



■ *ISS45 Version 7 – QDX Files*
■ *Technical Reference*
■ *Version 7130*
■
■

ISS45 V7 QDX Files Technical Reference

Date of Issue	Product Identification Number	Part Number	Brief Description
August 1995	45001/005	80316777	Version 7.1
October 1995	45001/005	80316203	Version 7.2
October 1996	45001/005	80328027	Version 7.3
June 1997	45001/005	80328462	Version 7.4
June 1998	45001/005	80602962	Version 7.5
February 1999	45001/005	80602986	Version 7.6
August 2000	45001/005	89000053	Version 7.7 (Unchanged)
May 2010	45001/005	89000566a	Version 7.1.3.0-140 (a: updated)

**© Copyright StoreNext Retail Technologies LLC 1995-2010
All rights reserved**

This publication is protected by federal copyright law. No part of this publication may be reproduced or transmitted into any human or computer language in any form or by any means, stored in a retrieval system, transmitted, redistributed, translated or disclosed to third parties, or de-compiled in any way including, but not limited to, photocopy, photograph, electronic, mechanical, magnetic or manual without the express written permission of StoreNext Retail Technologies LLC or its licensors, if any. This document, notwithstanding the above, may be distributed in electronic or printed form to personnel who are employed by either (1) a StoreNext customer of the subject system of this document or (2) personnel from an authorized StoreNext dealer of the subject system of this document. All copies, so distributed and/or so authorized, shall contain a full copy of this copyright notice.

StoreNext Retail Technologies LLC endeavors to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. StoreNext Retail Technologies LLC makes no representation or warranties with respect to the contents hereof, and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose or non-infringement. No commitments by StoreNext or its suppliers are made from this documentation which is provided for information only.

Development of StoreNext products and documentation is continuous: StoreNext Retail Technologies LLC reserves the right to revise this publication and to make changes from time to time in the contents hereof or in the products herein described or discussed without notice and without any obligation of StoreNext Retail Technologies LLC to notify any person or organization of such revision or changes. Information published in this document will likely become obsolete over time and it is recommended that users regularly check for updates and newer versions.

StoreNext Retail Technologies LLC has prepared this manual for use by users, authorized third parties and personnel of StoreNext Retail Technologies LLC as a guide to the proper installation, operation, customization and/or maintenance of StoreNext Retail Technologies LLC equipment and software. The drawings and specifications contained herein are the property of StoreNext Retail Technologies LLC and/or its licensors.

Third-party products, services, or company names referenced in this document may be trademarked or copyrighted by their respective owners, and are for identification purposes only.

Copyrights, trademarks and license agreements shall be governed and construed in accordance with the laws of the State of Texas and the Federal Arbitration Act, and shall benefit Retailix, its successors, and assigns.

Address comments and corrections to:

StoreNext Retail Technologies LLC
Software Program Director
6100 Tennyson Parkway
Suite 130
Plano, Texas 75024

Table of Contents

Table of Contents	i
Introduction	3
QDX Files	4
PLURELAT.QDX (0)	4
PLUPOS.QDX (1)	11
PLUITEM.QDX (2)	15
PLUINTER.QDX (3)	16
TICKINF.QDX (4)	17
PLUNAME.QDX (5)	19
PLUDEP.QDX (6)	20
POSVAL.QDX (7)	21
NOTFOUND.QDX (9)	26
CASHIER.QDX (10)	28
EMPL.QDX (11)	30
MEDIA.QDX (12)	38
RETN.QDX (13)	48
DISC.QDX (14)	52
TRANSACT.QDX (15)	54
EXPN.QDX (16)	55
ACCT.QDX (18)	56
CHCK.QDX (19)	57
TAX.QDX (20)	59
ECCA.QDX (21)	61

DAYDEP.QDX (22)	63
MONDEP.QDX (23)	64
SAVERCL.QDX (24)	66
CRNG.QDX (25)	68
GROUPNM.QDX (26)	74
SHORTGRP.QDX (27)	75
AUDIT.QDX (31)	77
COUPON.QDX (32)	80
POSLOGO.QDX (35)	91
SECURE.QDX (36)	92
SUBRELAT.QDX (37)	93
SUBDEP.QDX (38)	95
BUYAIDS.QDX (39)	96
DEPART.QDX (40)	101
PROMOTION.QDX (41)	105
BROM.QDX (42)	107
SAFE.QDX (43)	110
OFFERS.QDX (44)	114
MIXMATCH.QDX (45)	115
PARAM.QDX (46)	116
HEADFOOT.QDX (47)	149
WEEKTTL.QDX (48)	151
LOCATION.QDX (49)	153
CUSTOMER.QDX (50)	165
CUSTRLAT.QDX (51)	166
POSCONF.QDX (53)	169
KEY_FUNC.QDX (54)	212
KBD_DEF.QDX (55)	214

PRESET.QDX (56)	216
POSMMSG.QDX (73)	219
ALERTYPE.QDX (74)	234
LOYAL.QDX (75)	236
SAVERS.QDX (76)	239
T_CUST.QDX (77)	247
BAD_ACC2.QDX (79)	250
GEN_TBL.QDX (80)	254
PACKAGES.QDX (83)	259
HOURPROD.QDX (84)	261
EEXPEN.QDX (85)	269
BASKET.QDX (88)	271
LOGINFO.QDX (90)	273
VDR_CPN.QDX (93)	274
POSACCT.QDX (94)	276
ALERT.QDX (95)	285
GEN_DB.QDX (96)	287
NAMEIDX.QDX (98)	299
ALTHIST.QDX (100)	302
CROSS.QDX (101)	303
PLUEXT.QDX (105)	311
ASSISTNT.QDX (130)	313
PLUEXT.QDX (131)	315
MESSAGE.QDX (167)	317
HOTCARD.QDX (205)	320
WICITEMS.QDX (206)	321
WICCAT.QDX (207)	323

Introduction

This manual contains all the descriptions, specifications and record formats of the ISS45 system files.

- File numbers missing from this document are either not defined for use, or are reserved for internal use only.
- All structures are described in C.
- Next to each field is the field type and description. Here is a list of the field types:

Field Type	Description
ASCII	ASCII field
ASC_N	ASCII numeric
ASC_ND	ASCII numeric with a decimal point
BCD	Numeric field stored in BCD
BCD_D	BCD date, stored as YYMMDD or DDMMYY
BCD_T	Time in BCD
BIN	Binary numeric - MSB is on the left
BIT	Bit field
DATE_I	Date store in an integer field. bits:yyyyyyymmmddddd. Year 0 is 1980.
NU	Not used
NUM	Integral numeric
NUM_D	Numeric with decimal point
STRUCT	C structure
UNION	C union

QDX Files

PLURELAT.QDX (0)

File Type	Relative expansion
Record Size	512
Key Length	0

Record Format

	//OFS	Type	Description
typedef struct rlt_plu_rec_			
{			
unsigned char plu_number[7];	//0	BCD	PLU number
unsigned char dep[2];	//7	BCD	Department number
unsigned char name[40];	//9	ASCII	ASCII name
unsigned char name_pos[20];	//49	ASCII	POS name
unsigned char tax;	//69	BITS	
	//		Bit 0- TAX rate 1
	//		Bit 1- TAX rate 2
	//		Bit 2- TAX rate 3
	//		Bit 3- TAX rate 4
	//		Bit 4- TAX rate 5
	//		Bit 5- TAX rate 6
	//		Bit 6- TAX rate 7
	//		Bit 7- TAX rate 8
struct	//70	BITS	
{			
char pay_credit_foodstamp :1;	//		Payment by food stamps
char non_merch :1;	//		Non merchandize
char decimal_qty_req :1;	//		Decimal quantity
char negative_entry :1;	//		Negative entry

```

    char store_coupon      :1;      //      Store coupon
    char vendor_coupon     :1;      //      Vendor coupon
    char wic_item          :1;      //      WIC item
    char assume_qty        :1;      //      Assume Qty (WinPos)
} item_flags1;
struct                      //71  BITS
{
    char promotional      :1;      //      Item on promotion
    char not_for_sale     :1;      //      Sale prohibit
    char default_price    :1;      //      Not used
    char manual_price     :1;      //      Manual price
    char force_qty        :1;      //      Force quantity
    char weighed_item     :1;      //      Weighed item
    char inhibit_qty_rpt  :1;      //      Inhibit repeat
    char non_discount     :1;      //      Discount prohibit
} item_flags2;
struct                      //72
{
    char sku_entry        :1;      //      Force 15 digits SKU entry
    char opt_bonus_cpn    :1;      //      Save for later use [was disalow_bonus_cpn]
    char cost_plus        :1;      //      Cost plus item
    char price_verify     :1;      //      Price verify
    char price_override   :1;      //      Price override
    char supplier_prom    :1;      //      Supplier promotion
    char save_disc_flag   :1;      //      Save discount flag
    char item_on_sale     :1;      //      Item on sale
} item_flags3;
unsigned long price;        //73  AMT_D  Item price
unsigned short date_price;  //77  DATE_I  Price change date
long center_price;         //79  NUM_D  Center price
unsigned short date_ctr_price; //83  DATE_I  Center price change date
unsigned char cost_per_case_price[4]; //85  BCD_D  Cost per case
unsigned short date_cost_price; //89  DATE_I  Cost per case change date
unsigned char unit_per_case[2]; //91  BCD  Unit per case

```



```

unsigned char msu_c;                //93  NUM    multiply sell unit
unsigned char mm_c;                //94  NUM    Mix & match
unsigned char return_no;           //95  NUM    return type
unsigned char mm_flags;            //96  BITS
//                                bit 0: save for internal number
//                                bit 1: continuation record of internal
//                                bit 2: reserved
//                                bit 3: no internal
//                                bit 4: reserved
//                                bit 5: continuation record of m&m
//                                bit 6: return to user with OK_MORE
//                                bit 7: no mix match
unsigned char subdept[6];          //97  BCD    Subdepartment
unsigned char family_num[2];       //103 BCD    Family number
unsigned char user_defined[13];    //105 UNDEF  User defined fields
unsigned long weight;              //118 NUM_D   Weight
struct                            //122
{
    char opt_WIC_CVV                :1;    //      Payment by WIC CVV
    char opt_not_used                :7;    //      Save for later use
} item_flags4;
unsigned char nu2[2];              //123 NU     Not used
unsigned char disc_num;            //125 NUM    Linked discount
unsigned long second_price;        //126 NUM_D   Frequent shopper discount
unsigned char tare;                //130 NUM    Tare weight
unsigned char inter_num[7];        //131 BCD    Internal number
unsigned char label_count;         //138 NUM    Label count
struct                            //139 BITS
{
    unsigned char scale              :1;    //      send to scale
    unsigned char shelf_stock        :1;    //      maintain shelf stock
    unsigned char local_del          :1;    //      marked for local deletion
    unsigned char host_del           :1;    //      marked for host deletion
    unsigned char head_office_dep    :1;    //      report to head office

```

```

    unsigned char  weighed_scale      :1;  //          weighed at scale
    unsigned char  no_member_points   :1;  //          No member card points
    unsigned char  no_redemption       :1;  //          No redemption possible
} bo_flags1;

unsigned char freq_shop_type;           //140  NUM    Frequent shopper type
unsigned short second_family;           //141  NUM    Second family
long          save_price;               //143  NUM_D   Save price
unsigned char  pos_message;            //147  NUM    Linked message
unsigned char  modify_user;            //148  NUM    User number that modified the item
unsigned char  hierarchy_code[5];       //149  BCD    Hierarchy code
unsigned short shelf_life;              //154  NUM    Shelf life (in days)
unsigned short promo_no;                //156  NUM    Promotion number
unsigned char  bucket_no;               //158  NUM    Bucket number
unsigned char  nu[10];                  //159  NU     Not used
unsigned char  vendor [5];              //169  BCD    Vendor number
unsigned char  restrict_plu;            //174  NUM    Restrict PLU by age & time
unsigned char  tmpl_no;                 //175  NUM    Template number
unsigned char  limit_on_discount;       //176  NUM    Quantity limit for frequent
//                                     shopper discount in ticket

char nul[5];                           //177  NUM    Lead time in days

// Bonus Buy for V8

unsigned char  comp_type;               //182  NUM    Comparison type
long          comp_price;               //183  NUM_D   Comparison price
long          comp_qty;                 //187  NUM    Comparison quantity
unsigned char  valid_date[3];           //191  CHR    Valid date YYMMDD
unsigned char  label_format;            //194  BCD    Label format,
unsigned char  not_used[35];            //195          Not Used  TBA - moved to pluext.qdx
unsigned short freq_shop_item_group;    //230  NUM    Freq Shopper item group
unsigned short freq_shop_bonus_points;  //232  NUM    Freq Shopper bonus points
unsigned char  nu6[2];                 //234          Not used
unsigned short qty_rest_grp;            //236  NUM    Quantity restriction (DEA)
unsigned short excise_tax;              //238  NUM    Excise tax group
unsigned long  org_price;                //240  NUM    original price
unsigned char  org_qty;                 //244  NUM    original qty

```

unsigned long	mix_match2;	//245	NUM	Mix match code (v8)
unsigned char	info_nu[12];	//249	NU	Not used
// *** END OF INFORMATIVE PART ***				
// *** START OF SALES PART ***				
unsigned char	sale_nu[9];	//261	NU	Not used
long	additive_prod_cost;	//270	NUM	Additive online product cost
long	z_additive_prod_cost;	//274	NUM	Additive product cost since last z
long	day_vat_tax[MAX_TAX];	//278	NU	Daily Vat
// The 6 following field are for V7.3.4				
long	p_day_items_cost;	//310	NUM	Previous Day Cost
long	week_items_cost;	//314	NUM	Weekly Cost
long	p_week_items_cost;	//318	NUM	Previous Week Cost
long	month_items_cost;	//322	NUM	Monthly Cost
long	p_month_items_cost;	//326	NUM	Previous Month Cost
long	year_items_cost;	//330	NUM	Yearly Cost
long	inv_start;	//334	NUM	Opening stock
long	inv_saved;	//338	NUM	Saved stock
long	inv_in_out;	//342	NUM	Stock in/out
long	inv_counted;	//346	NUM	Counted stock
long	day_promo_revenue;	//350	NUM_D	Daily revenue when promoted
short	day_promo_qty;	//354	NUM	Daily quantity when promoted
unsigned short	day_date_sales;	//356	DATE_I	Date of last sale
long	day_qty_sales;	//358	NUM	Daily sales
long	day_revenue;	//362	NUM_D	Daily revenue
long	day_discount;	//366	NUM_D	Daily discount
long	day_promotion;	//370	NUM_D	Daily promotion revenue
long	day_qty_promotion;	//374	NUM	Daily promotion count
long	p_day_qty_sales;	//378	NUM	Previous day quantity sold
long	p_day_revenue;	//382	NUM_D	Previous day revenue
long	p_day_discount;	//386	NUM_D	Previous day discount
long	week_qty_sales;	//390	NUM	Weekly quantity sold
long	week_revenue;	//394	NUM_D	Weekly revenue
long	week_discount;	//398	NUM_D	Weekly discount

```

long p_week_qty_sales;           //402  NUM    Previous week quantity sold
long p_week_revenue;            //406  NUM_D   Previous week revenue
long p_week_discount;           //410  NUM_D   Previous week discount
long month_qty_sales;           //414  NUM    Monthly quantity sold
long month_revenue;             //418  NUM_D   Monthly revenue
long month_discount;            //422  NUM_D   Monthly discount
long p_month_qty_sales;         //426  NUM    Previous month quantity sold
long p_month_revenue;           //430  NUM_D   Previous month revenue
long p_month_discount;          //434  NUM_D   Previous month discount
long year_qty_sales;            //438  NUM    Yearly quantity sold
long year_revenue;              //442  NUM_D   Yearly revenue
long year_discount;             //446  NUM_D   Yearly discount
long z_qty_sales;               //450  NUM    Quantity sold since last Z
long z_revenue;                 //454  NUM_D   Revenue since last Z
long z_discount;                //458  NUM_D   Discount since last Z
long z_promotion;               //462  NUM_D   Promotion since last Z
long z_qty_promotion;           //466  NUM    Promotion quantity since last Z
long pccs_save_qty_sales;       //470  NUM    PCCS save quantity sold
long pccs_save_revenue;         //474  NUM_D   PCCS save revenue
short day_price_override;       //478  NUM    Daily price overrides
short week_price_override;      //480  NUM    Weekly price overrides
long day_reduce_qty_prom;       //482  NUM    Daily reduction quantity
long day_sale_item;             //486  NUM_D   Daily item on sale revenue
long day_qty_sale_item;         //490  NUM    Daily item on sale quantity
long period_sale_item;          //494  NUM_D   Period item on sale revenue
long period_qty_sale_item;      //498  NUM    Period item on sale quantity
long period_markdown_sale;      //502  NUM_D   Period item on sale markdown
struct maint_up_                //506  STRUCT  Update time stamp
{
    unsigned char hi_date;       //          High part of update date
    unsigned long pc_created :1; //          PC number generated
    unsigned long time          :23; //        Update time
    unsigned long low_date      :8; //          Low part of update date
} time_stamp;

```

```
    unsigned char q_dex_flags;           //511 BITS    QDEX reserved flag
} PLU_RELAT_REC;                        //512
```

PLUPOS.QDX (1)

File Type	Index
Record Size	160
Key Length	7

Record Format

typedef struct plu_pos_	//OFS	Type	Description
{			
char code[7];	//0	BCD	PLU number
char name[20];	//7	ASCII	Description
unsigned char mult_sell_unit;	//27	NUM	multiply sale unit
unsigned char mix_match;	//28	NUM	mix & match code
unsigned char discount_num;	//29	NUM	Linked discount
struct	//30	BITS	
{			
unsigned char rate_1 :1;	//		Bit 0- TAX rate 1
unsigned char rate_2 :1;	//		Bit 1- TAX rate 2
unsigned char rate_3 :1;	//		Bit 2- TAX rate 3
unsigned char rate_4 :1;	//		Bit 3- TAX rate 4
unsigned char rate_5 :1;	//		Bit 4- TAX rate 5
unsigned char rx_item :1;	//		Rx Item
unsigned char non_rx_item :1;	//		Non Rx Item
unsigned char rate_8 :1;	//		Bit 7- TAX rate 8
}tax;			
long price;	//31	NUM_D	Item price
short dept;	//35	NUM	Department number
struct	//37	BITS	
{			
unsigned char pay_fs :1;	//		Payment by food stamps
unsigned char non_merch :1;	//		Non merchandise

```
    unsigned char decimal_qty      :1;    //          Decimal quantity required
    unsigned char negative         :1;    //          Negative item
    unsigned char store_cpn        :1;    //          Store coupon
    unsigned char vendor_cpn       :1;    //          Vendor coupon
    unsigned char WIC_item         :1;    //          WIC item
    unsigned char assume_qty       :1;    //          Assume quantity
} flags1;

struct                                //38   BITS
{
    unsigned char promotion        :1;    //          Item on promotion
    unsigned char not_for_sale     :1;    //          Item not for sale
    unsigned char default_price    :1;    //          Not used
    unsigned char manual_price     :1;    //          Manual price
    unsigned char force_qty        :1;    //          Force quantity
    unsigned char weight_item      :1;    //          Weighed item
    unsigned char no_qty_and_repeat:1;    //          Repeat/Quantity not allowed
    unsigned char no_discount      :1;    //          Discount not allowed
} flags2;

struct                                //39   BITS
{
    char sku_entry                 :1;    //          Force 15 digits SKU entry
    char opt_bonus_cpn             :1;    //          Not used          [was disalow_bonus_cpn]
    char cost_plus                 :1;    //          Cost plus item
    char price_verify              :1;    //          Price verify
    char price_override            :1;    //          Price override
    char supplier_prom             :1;    //          Supplier promotion
    char save_disc_flag            :1;    //          Save discount flag
    char opt_item_on_sale          :1;    //          Item on sale
} flags3;

unsigned char return_type;            //40   NUM    Return type
long  second_price;                  //41   NUM_D   Frequent shopper discount
```

```

unsigned char tare_weight;           //45  NUM    Tare weight pointer
short      family_no;               //46  NUM    Family number
unsigned char second_name[20];      //48  ASCII  Item second name
unsigned char freq_shop_type;       //68  NUM    Frequent shopper type
unsigned short second_family;       //69  NUM    Second family
unsigned short promo_no;            //71  NUM    Promotion number
struct                                           //73
{
    char opt_WIC_CVV          :1;    //      Payment by WIC CVV
    char opt_not_used        :7;    //      Save for later use
} item_flags4;
unsigned char nul[1];               //74  NU     Not used
unsigned char bucket_no;           //75  NUM    Bucket number
unsigned char nu[8];               //76  NU     Not used
long center_price;                 //84  NUM_D  Head office price
unsigned char pos_message;         //88  NUM    Item message
unsigned char user_defined[13];    //89  UNDEF  User defined fields
unsigned short freq_shop_item_group; //102 NUM    Freq Shopper item group
unsigned short freq_shop_bonus_points; //104 NUM    Freq Shopper bonus points
unsigned short excise_tax;         //106 NUM    Excise tax ptr
unsigned char nu5[2];              //108 NU     Not used
unsigned long org_price;            //110 NUM    original price
unsigned char org_qty;              //114 NUM    original qty
unsigned long mix_match2;           //115 NUM    Long mix match code (v8)
unsigned char filler[23];           //119 NU     Not used
// Bonus Buy for V8
unsigned char comp_type;            //142 NUM    Comparison type
long      comp_price;               //143 NUM_D  Comparison price
long      comp_qty;                 //147 NUM    Comparison quantity
unsigned char valid_date[3];        //151 CHR    Valid date YYMMDD
unsigned short qty_rest_grp;        //154 NUM    Quantity restriction (DEA)
unsigned char limit_on_discount;    //156 NUM    Quantity limit for FS discount in ticket
unsigned char tmpl_no;              //157 NUM    Template number
unsigned char restrict_layout;      //158 NUM    Time restriction layout

```



```
    unsigned char qdx;                //159 BITS    QDEX reserved flag
}POSPLU_REC;                          //160
```

PLUITEM.QDX (2)

File Type	Index
Record Size	10
Key Length	7

Record Format

	//OFS	Type	Description			
typedef struct plu_index_ { unsigned char plu_number[7]; unsigned char filler[3]; } PLU_INDEX_REC; typedef struct { PLU_INDEX_REC idx; PLU_RELAT_REC rlt; }PLU_CMB; #define PLU_KEY_LEN (lenof(PLU_INDEX_REC, plu_number))						
	//0	BCD	PLU number			
	//7	NU	Not used			

PLUINTER.QDX (3)

File Type	Index
Record Size	17
Key Length	14

Record Format

struct plu_intern_idx_	//OFS	Type	Description
{			
unsigned char inter_nu[7];	//0	BCD	Internal number
unsigned char plu_number[7];	//7	BCD	PLU number
unsigned char filler[3];	//14	NU	Not used
} ;			
typedef struct			
{			
struct plu_intern_idx_ idx;			
PLU_RELAT_REC rlt;			
}PLU_INTERN;			

TICKINF.QDX (4)

File Type	Index
Record Size	128
Key Length	7

Record Format

```
typedef struct ticket_info_rec_  
{  
    struct ticket_info_key_  
    {  
        unsigned char    pos_no;           // POS number  
        unsigned char    ticket_no[3];     // ticket number (BCD)  
        unsigned char    trs_opcode;       // transaction record opcode  
        unsigned char    trs_subopcode[2]; // TRS subopcode (BCD)  
    }key;  
  
    union trans_    t_data; // TRS record  
    unsigned char    nu[56]; // not used  
    unsigned char    qdx;    // QDX flag  
  
} TICKET_INFO_REC;  
  
//struct for 24 hours SelfScanning report  
struct partial_trs_ticket_total_inf_  
{  
    short            items;  
    long             amount;  
    unsigned char    hour,  
                    scanpoint_item;  
};
```

```
//struct for additional data department report
struct additional_trs_data_depart_
{
    unsigned char item_number[7];
    long daily_qty;
    long daily_amount;
    long weekly_qty;
    long weekly_amount;
};
```

PLUNAME.QDX (5)

File Type	Index
Record Size	18
Key Length	15

Record Format

struct plu_name_idx_	//OFS	Type	Description
{			
unsigned char name[8];	//0	ASCII	First 8 characters from POS name
unsigned char plu_number[7];	//8	BCD	PLU number
unsigned char filler[3];	//15	NU	Not used
};			
typedef struct			
{			
struct plu_name_idx_ idx;			
PLU_RELAT_REC rlt;			
}PLU_NAME;			

PLUDEP.QDX (6)

File Type	Index
Record Size	12
Key Length	9

Record Format

struct plu_dep_idx_	//OFS	Type	Description
{			
unsigned char dep[2];	//0	BCD	Department number
unsigned char plu_number[7];	//2	BCD	PLU number
unsigned char filler[3];	//9	NU	Not used
}			
};			
typedef struct			
{			
struct plu_dep_idx_ idx;			
PLU_RELAT_REC rlt;			
}PLU_DEP;			

POSVAL.QDX (7)

File Type	Index
Record Size	1024
Key Length	3

Record Format

typedef struct posval_	//OFS	Type	Description
{			
char pos_num[2];	//0	BCD	POS number
char rec_no;	//2	NUM	Record number
	//		0 - Current period
	//		10- Previous period
struct	//3	BITS	
{			
char op_nu :5;	//		Not used
char disallow_log :1;	//		Don't allow sign on
char signed_on :1;	//		Is signed on
char signed_today :1;	//		Signed on today
} flags;			
short pos_no;	//4	NUM	Not used
char name[20];	//6	ASCII	Not used
unsigned short report_count;	//26	NUM	Report counter
struct	//28	BITS	Not used
{			
char priv8 :1;			


```

char  priv7      :1;

char  priv6      :1;

char  priv5      :1;

char  priv4      :1;

char  priv3      :1;

char  priv2      :1;

char  priv1      :1;

} sale_mode_privilage;

struct                                //29   BITS   Not used
{

    char  priv8      :1;

    char  priv7      :1;

    char  priv6      :1;

    char  priv5      :1;

    char  priv4      :1;

    char  priv3      :1;

    char  priv2      :1;

    char  priv1      :1;

} train_mode_privilage;

long transaction_Stamp;                //30   NUM    Number of last transaction processed.

long gross_sales_value;                //34   NUM_D   Gross sales value

short gross_sales_count;               //38   NUM    Gross sales count

long gross_return_value;               //40   NUM_D   Gross return value

short gross_return_count;              //44   NUM    Gross return count

long item_sold          ;              //46   NUM    Number of items sold

long item_returned      ;              //50   NUM    Number of items returned

```

long pos_disc_value ;	//54	NUM_D	Value of discounts given
short pos_disc_count ;	//58	NUM	Number of discounts given
long pos_prom_value ;	//60	NUM_D	Promotions value
short pos_prom_count ;	//64	NUM	Promotions count
long pos_neg_dep_value ;	//66	NUM_D	Negative dept sales
short pos_neg_dep_count ;	//70	NUM	Negative dept count
long pos_non_march_value;	//72	NUM_D	Non-merchandise sales
short pos_non_march_count;	//76	NUM	Non-merchandise count
long pos_europ_round_value;	//78	NUM_D	Tax value
short pos_europ_round_count;	//82	NUM	Tax count
long absolute_void_value ;	//84	NUM_D	Cancelled tickets value
short absolute_void_count ;	//88	NUM	Cancelled tickets count
long absolute_refund_value;	//90	NUM_D	Absolute value of refunds
short absolute_refund_count;	//94	NUM	Absolute count of refunds
long pos_price_overrides_val;	//96	NUM_D	Price overrides value
short pos_price_overrides_c ;	//100	NUM	Price overrides count
long absolute_cancel_value ;	//102	NUM_D	Absolute value of cancels
short absolute_cancel_count ;	//106	NUM	Absolute count of cancels
long absolute_subtract_value;	//110	NUM_D	Absolute value of subtracts
short absolute_subtract_count;	//112	NUM	Absolute count of subtracts
long cash_payout_value ;	//114	NUM_D	Cash payouts value
short cash_payout_count ;	//118	NUM	Cash payouts count
long cash_receipts_value ;	//120	NUM_D	Cash receipts value
short cash_receipts_count ;	//124	NUM	Cash receipts count
short number_of_no_sales ;	//126	NUM	No-sale count
short number_of_customers ;	//128	NUM	Customers count

```

unsigned short oprator_item_reg_time ; //130 NUM Item regs. time (seconds)
unsigned short idle_time ; //132 NUM Idle time (seconds)
unsigned short secure_time ; //134 NUM Secure time (seconds)
unsigned short tender_time ; //136 NUM Tender time (seconds)
unsigned short wait_time ; //138 NUM Wait time (seconds)
unsigned short inq_time ; //140 NUM Inquiry. time (seconds)
unsigned short other_time ; //142 NUM Other time (seconds)
unsigned char dclr_count; //144 NUM Period declaration counter
long sec_payout_value ; //145 NUM_D 2nd tender payouts value
short sec_payout_count ; //149 NUM 2nd tender payouts count
long sec_receipts_value ; //151 NUM_D 2nd tender receipts value
short sec_receipts_count ; //155 NUM 2nd tender receipts count
unsigned char not_used[16] ; //157 NU Not used

```

// The following structure repeats itself 25 times, once per media

```

struct pos_media_data_
{
    long media_value ; // NUM_D Value of media received
    short media_count ; // NUM Count of media received
    long lifts_value ; // NUM_D Pick-ups value
    short lifts_count ; // NUM Pick-ups count
    long opening_float ; // NUM_D Opening float value
    long add_float_value ; // NUM_D Additional float value
    short add_float_count ; // NUM Additional float count
    long declared_value ; // NUM_D Declared amount
    short declared_qty ; // NUM Declared quantity
}

```

```
long carry_over_value    ;           //    NUM_D    Carry over value

short carry_over_count   ;           //    NUM      Carry over count

} media_data[MAX_MEDIA];

unsigned char qdx_flag;      //1023 BITS    QDEX reserved flag

} POSVAL;

#define PS_KEY_LEN    (lenof(POSVAL, pos_num))
```

NOTFOUND.QDX (9)

File Type	Index
Record Size	12
Key Length	7

Record Format

```

typedef struct notfound_rec_           //OFS  Type  Description
{
    unsigned char  plu[7];             //0    BCD    PLU number
    unsigned char  keyd;               //7    NUM    Not found counter
    unsigned char  scan_count;//filler1; //8    NU    Not used
    struct
    {
        unsigned int   day   :5;      //          Day
        unsigned int   month :4;      //          Month
        unsigned int   year  :7;      //          Year
    } date;
    unsigned char  qdx_flag;           //          BITS    QDEX reserved flag
} NOTFOUND_REC;

typedef struct
{
    unsigned char desc[16];            // 0
    struct
    {
        unsigned char filler1:7;
        unsigned char valid:1;
    } flags;                          // 16
    unsigned char filler2[2];          // 17
    long          sales1 ;              // 19
    long          return1;              // 23

```

```
long      sales2 ;           // 27
long      return2;          // 31
unsigned char qdx_flag;      // 35
} CUSTC_REC;                 // 36
```

CASHIER.QDX (10)

File Type	Index
Record Size	2
Key Length	0

Record Format

typedef struct cash_rec_		//OFS	Type	Description
{				
char	key[2];	//0	BCD	Cashier number
char	security[2];	//2	NU	Not used
char	name[32];	//4	ASCII	Cashier name
char	address[32];	//36	ASCII	Address
char	tel[15];	//68	ASCII	Phone number
char	date[3];	//83	BCD_D	Start date
char	remark[32];	//86	ASCII	Remarks
char	report_counter[2];	//118	BCD	Report counter
CASH_MEDIA	dclr_current[MAX_MEDIA];	//120	STRUCT	Current period declarations
unsigned char	password[4];	//320	ASCII	Password
unsigned char	password_change_date[3];	//324	BCD_D	Password change date
unsigned char	type;	//327	NUM	Cashier type
unsigned char	priority;	//328	NUM	Priority
unsigned char	birth[3];	//329	BCD_D	Birth date
unsigned char	group;	//332	NUM	Reporting group number
struct		//333	STRUCT	Include in cashier reports
{				
char	active	:1;	//	BIT Only Active cashier can sign on
char	cbnk_signon	:1;	//	BIT Sign on allowed only in checkout banks
char	sign_on_today	:1;	//	BIT Did Sign on today? - set by trans
char	sign_on_yesterday	:1;	//	BIT Did sign on yesterday? - set by trans
char	nul	:4;		
}	flag;			

```

unsigned char board_messages[8];          //334 BITS   Cashier got board message
char          filler2[178];              //342 NU     Not used
CASH_MEDIA    dclr_prev[MAX_MEDIA];      //520 STRUCT Prev. period declarations
char          filler3[200];              //720 NU     Not used
char          cashier_flag;              //920 NU     Not used
char          prv_sale_mode;              //921 NU     Not used
char          prv_training_mode;          //922 NU     Not used
long          over_short_daily;           //923 NUM_D   Daily over/short value
long          over_short_prv_day;         //927 NUM_D   Prev. day over/short value
long          over_short_weekly;          //931 NUM_D   Weekly over/short value
long          over_short_prv_week;        //935 NUM_D   Prev. week over/short value
long          over_short_monthly;         //939 NUM_D   Monthly over/short value
long          over_short_prv_month;       //943 NUM_D   Prev. month over/short value
long          over_short_yearly;          //947 NUM_D   Yearly over/short value
long          pp_media[5];                //951 NUM_D   5 Pay Path tender totals
unsigned char cb_sign_on[3];              //971 BITS   Bit per each checkout bank Cashier may sign on
unsigned char declaring_pc_number;        //974         PC number
char          filler4[46];                //975 NU     Not used
unsigned char current_dclr_count;         //1021 NUM    Current period declarations counter
unsigned char prev_dclr_count;            //1022 NUM    Current period declarations counter
char          qdx_flag;                   //1023 BITS   QDEX reserved flag
}CASH_REC;

#define CS_KEY_LEN    (lenof(CASH_REC, key))

```


EMPL.QDX (11)

File Type	Index
Record Size	1024
Key Length	3

Record Format

// Applications using this structure :

// PCMQ2ASC

```
typedef struct empl_rec0_           //OFS  Type  Description
char cash_num[2];                  //0    BCD    POS number
char rec_no;                       //2    NUM    Record number
                                   //      0 - Current period
                                   //      10- Previous period

struct                             //3    BITS
char op_nu           :3;          //      Not used
char add_to_dclk_file :1;          //      Cashier was sign on - set at EOD by PCMEMPL
                                   //      Reset at EOD by PCMTextr
char message         :1;          //      Message send
char disallow_log     :1;          //      Don't allow sign on
char signed_on        :1;          //      Is signed on
char signed_today     :1;          //      Signed on today
} cashier_flag;

short pos_no;                      //4    NUM    Not used
char name[20];                    //6    ASCII  Not used
short password;                   //26   NUM    Not used
struct                             //28   BITS    Not used
char priv8           :1;
char priv7           :1;
char priv6           :1;
char priv5           :1;
```

```

char  priv4    :1;
char  priv3    :1;
char  priv2    :1;
char  priv1    :1;
} sale_mode_privilage;

struct          //29   BITS   Not used
char  priv8    :1;
char  priv7    :1;
char  priv6    :1;
char  priv5    :1;
char  priv4    :1;
char  priv3    :1;
char  priv2    :1;
char  priv1    :1;
} train_mode_privilage;

long transaction_Stamp;          //30   NUM   Number of last transaction processed
long gross_sales_value;         //34   NUM_D  Gross sales value
short gross_sales_count;        //38   NUM   Gross sales count
long gross_return_value;        //40   NUM_D  Gross return value
short gross_return_count;       //44   NUM   Gross return count
long item_sold      ;          //46   NUM   Number of items sold
long item_returned  ;          //50   NUM   Number of items returned
long pos_disc_value  ;          //54   NUM_D  Value of discounts given
short pos_disc_count ;         //58   NUM   Number of discounts given
long pos_prom_value  ;          //60   NUM_D  Promotions value
short pos_prom_count ;         //64   NUM   Promotions count
long pos_neg_dep_value ;       //66   NUM_D  Negative dept sales
short pos_neg_dep_count ;      //70   NUM   Negative dept count
long pos_non_march_value;      //72   NUM_D  Non-merchandise sales
short pos_non_march_count;     //76   NUM   Non-merchandise count
long pos_europ_round_value;    //78   NUM_D  Tax value
short pos_europ_round_count;   //82   NUM   Tax count
long absolute_void_value ;     //84   NUM_D  Cancelled tickets value

```

```

short absolute_void_count ;           //88  NUM    Cancelled tickets count
long absolute_refund_value;           //90  NUM_D   Absolute value of refunds
short absolute_refund_count;           //94  NUM    Absolute count of refunds
long pos_price_overrides_val;         //96  NUM_D   Price overrides value
short pos_price_overrides_c ;         //100 NUM    Price overrides count
long absolute_cancel_value ;          //102 NUM_D   Absolute value of cancels
short absolute_cancel_count ;          //106 NUM    Absolute count of cancels
long absolute_subtract_value;          //110 NUM_D   Absolute value of subtracts
short absolute_subtract_count;         //112 NUM    Absolute count of subtracts
long cash_payout_value ;              //114 NUM_D   Cash payouts value
short cash_payout_count ;             //118 NUM    Cash payouts count
long cash_receipts_value ;            //120 NUM_D   Cash receipts value
short cash_receipts_count ;           //124 NUM    Cash receipts count
short number_of_no_sales ;            //126 NUM    No-sale count
short number_of_customers ;           //128 NUM    Customers count
unsigned short oprator_item_reg_time ; //130 NUM    Item regs. time (seconds)
unsigned short idle_time ;            //132 NUM    Idle time (seconds)
unsigned short secure_time ;          //134 NUM    Secure time (seconds)
unsigned short tender_time ;          //136 NUM    Tender time (seconds)
unsigned short wait_time ;            //138 NUM    Wait time (seconds)
unsigned short inq_time ;             //140 NUM    Inquiry. time (seconds)
unsigned short other_time ;           //142 NUM    Other time (seconds)
unsigned char dclr_count;             //144 NUM    Period declaration counter
long sec_payout_value ;               //145 NUM_D   2nd tender payouts value
short sec_payout_count ;              //149 NUM    2nd tender payouts count
long sec_receipts_value ;             //151 NUM_D   2nd tender receipts value
short sec_receipts_count ;            //155 NUM    2nd tender receipts count
unsigned short signon_time ;          //157 NUM    Total sign on time(WinPOS)
unsigned char signon_date[3];         //159 BCD    Date when signed on
unsigned char not_used[11] ;          //162 NU     Not used

// The following structure repeats itself 25 times, once per media
struct media_data_
{

```

```

    long media_value      ;           //    NUM_D    Value of media received
    short media_count     ;           //    NUM      Count of media received
    long lifts_value      ;           //    NUM_D    Pick-ups value
    short lifts_count     ;           //    NUM      Pick-ups count
    long opening_float    ;           //    NUM_D    Opening float value
    long add_float_value  ;           //    NUM_D    Additional float value
    short add_float_count ;           //    NUM      Additional float count
    long declared_value   ;           //    NUM_D    Declared amount
    short declared_qty    ;           //    NUM      Declared quantity
    long carry_over_value ;           //    NUM_D    Carry over value
    short carry_over_count ;          //    NUM      Carry over count
} media_data[MAX_MEDIA];

unsigned char qdx_flag;           //1023 BITS    QDEX reserved flag
} EMPL_REC0;

// The following structure is used to keep cashier statistics data for
// performance reporting. Data is collected by PCMEMPL.
// The data is kept in 3 records:
// rec_no      11: Daily
// rec_no      12: Weekly
// rec_no      13: Monthly
//
#define EMPL_STT_DAILY      11
#define EMPL_STT_WEEKLY    12
#define EMPL_STT_MONTHLY   13

typedef struct empl_stt_           //OFS  Type    Description
    char cash_num[2];             //0     BCD     POS number
    char rec_no;                  //2     NUM     Record number
    struct                        //3     BITS
        char op_nu      :5;       //      Not used
        char disallow_log :1;     //      Don't allow sign on
        char signed_on   :1;     //      Is signed on
        char signed_today :1;    //      Signed on today
    } cashier_flag;

```

```

short pos_no;                //4    NUM    Not used
char  name[20];              //6    ASCII  Not used
short password;              //26   NUM    Not used
long  item_sold              //28   NUM    Number of items sold
long  absolute_void_value    //32   NUM_D  Cancelled tickets value
long  absolute_void_count    //36   NUM    Cancelled tickets count
long  pos_price_overrides_val; //40   NUM_D  Price overrides value
long  pos_price_overrides_c ; //44   NUM    Price overrides count
long  absolute_cancel_count ; //48   NUM    Absolute count of cancels
long  absolute_subtract_value; //52   NUM_D  Absolute value of subtracts
long  absolute_subtract_count; //56   NUM    Absolute count of subtracts
long  number_of_no_sales     //60   NUM    No-sale count
long  number_of_customers    //64   NUM    Customers count
unsigned long oprator_item_reg_time ; //68   NUM    Item regs. time (seconds)
unsigned long idle_time      //72   NUM    Idle time (seconds)
unsigned long secure_time    //76   NUM    Secure time (seconds)
unsigned long tender_time    //80   NUM    Tender time (seconds)
unsigned long wait_time      //84   NUM    Wait time (seconds)
unsigned long inq_time       //88   NUM    Inquiry. time (seconds)
unsigned long other_time     //92   NUM    Other time (seconds)
long  itm_scan;              //96   NUM    Items scanned
long  itm_key;               //100  NUM    Items keyed
long  itm_dept;              //104  NUM    items sold by dept
long  rings;                 //108  NUM    Number of item rings
long  amount_of_sales_training_mode; //112  NUM    Amount of sales in training mode
long  amount_of_non_taxable_sales ; //116  NUM    Amount of non taxable sales
long  amount_of_taxable_sales ; //120  NUM    Amount of taxable sales
long  amount_of_tax_refunded ; //124  NUM    Amount of tax refunded [option for V8]
long  amount_of_non_taxable_refunds; //128  NUM    Amount of non taxable refunds
unsigned long signon_time ; //132
unsigned char not_used[87] ; //134  NU     Not used
// The following structure repeats itself 25 times, once per media
// 25*32=800 bytes

```

```

struct media_stt_                                //223  STRUCT
{
    long media_value        ;                //    NUM_D    Value of media received
    short media_count       ;                //    NUM        Count of media received
    long lifts_value        ;                //    NUM_D      Pick-ups value
    short lifts_count       ;                //    NUM        Pick-ups count
    long opening_float      ;                //    NUM_D      Opening float value
    long add_float_value    ;                //    NUM_D      Additional float value
    short add_float_count   ;                //    NUM        Additional float count
    unsigned char  nu[10];
} media_data[MAX_MEDIA];
unsigned char qdx_flag;                        //1023 BITS    QDEX reserved flag
} EMPL_STT ;

#define EMPL_CASH_LEN      (lenof(EMPL_REC0, cash_num))
#define EMPL_KEY_LEN      3
#define EMPL_INFO_PRD_DAY  0
#define EMPL_INFO_PRD_PDAY 1
#define EMPL_INFO_PRD_WEEK 2
#define EMPL_INFO_PRD_PWEEK 3
#define EMPL_INFO_PRD_MONTH 4
#define EMPL_INFO_PRD_PMONTH 5
#define EMPL_INFO_PRD_YEAR 6

typedef struct empl_info_
{
    char cash_num[2];                // 0
    char rec_no;                    // 2  opcode 22
    int  pos_no;                    // 3
    char nu[10];                    // 5
    struct empl_info_prd_          // 15
    {
        // various counters
        int  cards_swiped_qty;
    }
}

```

```
int    cards_keyed_qty;
long   no_trans;
long   gross_sales_value;
long   item_sold;
int     number_of_customers;
long   promotion_value;
char   nu_qty[10];

// Special functions values & counters
long   void_last_value;
int     void_last_qty  ;
long   void_tender_value;
int     void_tender_qty ;
long   void_item_value;
int     void_item_qty   ;           //
long   roa_value;           //
int     roa_qty   ;           //    - 144 * 7 = 1008
long   refund_value;           //
int     refund_qty  ;           //
long   price_override_value;
int     price_override_qty  ;
long   cpn_out_tran_value;
int     cpn_out_tran_qty  ;
long   abort_trans_value;
int     abort_trans_qty  ;
long   stored_trans_value;
int     stored_trans_qty  ;
long   recall_trans_value;
int     recall_trans_qty  ;
long   open_depts_value;
int     open_depts_qty  ;
long   coupon_outside_value;
int     coupon_outside_qty  ;
```

```
int      oprator_item_reg_time ;
unsigned int idle_time      ;
unsigned int secure_time    ;
unsigned int tender_time    ;
unsigned int wait_time      ;
unsigned int inq_time       ;
unsigned int other_time     ;
char val_qty_nu[26];
} prd[7];
char qdx;                // 1023
} EMPL_INFO;
```


MEDIA.QDX (12)

File Type	Index
Record Size	512
Key Length	2

Record Format

typedef struct med_rec_	//OFS	Type	Description
{			
unsigned char key[2];	//0	BCD	Tender number
unsigned char name[16];	//2	ASCII	Tender name
unsigned char acc_prompt[16];	//18	ASCII	Account prompt
unsigned char eft_desc[16];	//34	ASCII	EFT description
unsigned char tender_type;	//50	NUM	Tender type
	//		0 - not used
	//		1 - Cash
	//		2 - Check
	//		3 - Credit card / EFT
	//		4 - Other/Charge posting
	//		5 - Generic credit card
	//		6 - Foreign currency
	//		equivalent
	//		7 - Food stamps
	//		8 - Vendor coupons
	//		9 - Store coupons
	//		10- In-house charge
	//		11- Debit card
	//		12- WIC
	//		13- Buy aid
	//		14- Staff discount type 1
	//		15- Staff discount type 2

```

//          16- Coupons
//          17- General tender
//          18- EBT food stamps
//          19- EBT cash
//          20- Accumulative
//          21- In-house purchase
//          22- Customer account payment
//          23- Credit Notes
//          24- black box I/F
//          25- WIC CVV
struct      //51  BITS
{
    unsigned char val_not_req :1;    //          Value not required
    unsigned char cancel_sub  :1;    //          SVR at Cancel/Subtract
    unsigned char inhibit    :1;    //          Cancel/Subtract inhibit
    unsigned char purchas    :1;    //          Tender purchase allowed
    unsigned char mag_card    :1;    //          MCR allowed
    unsigned char acc_req     :2;    //          Account number required
//          0-none
//          1-entered by cashier
//          2-entered by customer
    unsigned char overtendering:1;    //          Over-tender allowed
} flags1;
struct      //52  BITS
{
    unsigned char round_down  :1;    //          Not used
    unsigned char allow_refound:1;    //          Refund allowed
    unsigned char round       :1;    //          Rounding allowed
    unsigned char verify      :1;    //          Check bad accounts file
    unsigned char cdv_req     :1;    //          CDV check on account
    unsigned char sig_req     :1;    //          Check signature
    unsigned char bankable    :1;    //          Tender bankable
    unsigned char always_req  :1;    //          SVR keylock required

```

```

} flags2;

struct                                //53   BITS
{
    unsigned char floatable    :1;    //      Tender has float
    unsigned char open_loan    :1;    //      Automatic open loan
    unsigned char print_tmpl   :1;    //      Print template
    unsigned char eca          :1;    //      Use ECA
    unsigned char foreign      :1;    //      Foreign currency
    unsigned char autorisation :1;    //      Authorization on receipt
    unsigned char state_code   :1;    //      State code required
    unsigned char enter        :1;    //      Hit ENTER before slip printing
} flags3;

struct                                //54   BITS
{
    unsigned char open_drawer_options :2; //      Open drawer options
                                           //      0 - None
                                           //      1 - Open drawer at once
                                           //      2 - Open drawer if change
                                           //      3 - Open drawer on change
    unsigned char foreign_change_allow:1; //      Allow foreign currency change
    unsigned char roa_payout_allow    :1; //      ROA/Payout allowed
    unsigned char invalid_pay         :1; //      Invalid for payment
    unsigned char value_not_req_4_neg :1; //      Value not required for
                                           //      negative
    unsigned char credit_debit        :2; //      Credit/Debit options
                                           //      0 - credit or debit
                                           //      1 - credit
                                           //      2 - debit
} flags4;

unsigned char mr_req;                //55   NUM   Multi-part receipt
unsigned short endors_temp_no ;      //56   NUM   Endorsement template number
unsigned short template_no ;        //58   NUM   Template number

```

```

unsigned long currency_rate    ;           //60  NUM_D  Currency rate
unsigned char mcr;             //64  NUM    MCD number
unsigned short eca_timeout;    //65  NUM    ECA timeout
unsigned char other_tender;    //67  NU     Not used
long      max;                 //68  NUM_D  Maximum value
long      min;                 //72  NUM_D  Minimum value
long      over;                //76  NUM_D  Over tender value
long      floor;               //80  NUM_D  Floor limit value
long      amount_req_control_chck; //84  NUM_D  Amount required for CC
unsigned char max_chck_no;      //88  NUM    Maximum value CC
unsigned char min_chck_no;      //89  NUM    Minimum value CC
unsigned char over_tender;      //90  NUM    Over tender CC
unsigned char floor_chck_no;    //91  NUM    Floor limit CC
unsigned char daily_chck_no;    //92  NUM    Daily value CC
unsigned char repeat_chck_no;   //93  NUM    Repeat CC
unsigned char autho_chck_no;    //94  NUM    Authorization CC
unsigned char repcard_chck_no;  //95  NUM    Repeat card number CC
unsigned char filler_chck_no[3]; //96  NU     Not used
unsigned char retailer;         //99  NUM    Retailer information
long      loan_amount;         //100 NUM_D  Load amount
unsigned char loan_control_check; //104 NUM    CC for changing opening float
unsigned char Rounding_Value;    //105 NU     Not used
unsigned char mcr_timeout;       //106 NUM    MCR timeout
unsigned char mcr_misread;       //107 NUM    MCT misread
struct                                     //108 BITS
{
    unsigned char tender_cash_purch :2; // Automatic purchase of...
                                     // 0 - None
                                     // 1 - Tender
                                     // 2 - Cash
    unsigned char clear_at_eod      :1; // Clear totals at EOD
    unsigned char ecca              :1; // Use ECCA on this media
    unsigned char use_acct2         :1; // Use enhanced bad accounts

```

```

unsigned char  print_on_right      :1; //      Not used
unsigned char  EFTPOS              :1; //      EFTPOS
unsigned char  cash_in_drawer      :1; //      Cash in drawer check
} flags5;

unsigned char  tender_group;        //109  NUM    Tender group
                                     //      0 - Not defined
                                     //      1 - Cash
                                     //      2 - Check
                                     //      3 - EFTPOS
                                     //      4 - Manual Cr/Dr sale
                                     //      5 - Scanned coupons
                                     //      6 - Non scanned coupons
                                     //      7 - Staff discount
                                     //      8 - Stamps/Voucher
                                     //      9 - Credit voucher issued
                                     //     10- Paid outs
                                     //     11- ROA

unsigned char  till_clearance_type; //110  NUM    Clearance type
struct                                                //111  BITS
{
    unsigned char  expected_is_declared:1; //      Non declarable
    unsigned char  carry_over          :1; //      Carry over tender
    unsigned char  add_float_only      :1; //      Add float only
    unsigned char  use_crng            :1; //      Use card ranges
    unsigned char  receipt_on_denial   :1; //      Print receipt on denial
    unsigned char  no_banking          :1; //      Declare Banking disallowed
    unsigned char  gift_certificate    :1; //      Gift certificate tender
    unsigned char  bank_num_required   :1; //      Bank number is required
} flags6;

unsigned char  gift_vcr_len;         //112  NU     Not used
unsigned char  EFT_rep_group;        //113  NUM    EFT reporting group
unsigned char  ECCA_template;        //114  NUM    ECCA template
unsigned char  scan_as_account;      //115  NUM    Scan barcode as an account

```

```

//          0 - None
//          1 - Scan as account
//          2 - Without check digit
unsigned char mpr_number;          //116 NUM Multi part receipt number
unsigned char POS_rep_group;       //117 NUM POS reporting group
unsigned char fee_tab_no;          //118 NUM Fee table number
unsigned char fee_type;            //119 NUM 0-cashback 1-tender value
struct
{
    unsigned char select_sub_tender :1; //120 BITS Select sub tender PHIL
    unsigned char doc_manual_balnce :1; // Document manual balancing
    unsigned char chk_reader_in_use :1; // Check reader in use
    unsigned char duplicate_receipt :1; // V8: Tender maintenance
    // duplicate receipt
    unsigned char check_nmbr_require:1; // Check number required
    unsigned char use_checkdef_file :1; // Use Check definition file CHECKDEF
    unsigned char mu :2;
} flags7;
unsigned char doc_start_print_offset; //121 NUM Document start print offset PHIL
unsigned char doc_print_size;         //122 NUM Document print size PHIL
long preset_amount[10];               //123 NUM preset amount
long max_ticket_amount;               //163 NUM_D Maximum value in ticket
unsigned char max_ticket_count ;      //167 NUM Maximum times in ticket
unsigned char cc_max_ticket_amount;    //168 NUM CC for maximum value
unsigned char cc_max_ticket_count ;    //169 NUM CC for maximum count
long max_purchase;                    //170 NUM Maximum Purchase amount
unsigned char cc_max_purchase;         //174 NUM CC for maximum purchase
unsigned char cust_template_number;    //175 NUM Customer Template Number
unsigned char reserve0[17];            //176 NU Not used
//////////
// EPS parameters
//////////
unsigned long offline_tndr_limit;      //193 NUM Offline tender limit

```

```

unsigned long  cashback_amt_1;           //197  NUM    Pinpad cashback display amount #1
unsigned long  max_cashback_limit;       //201  NUM    Max cashback limit at pinpad
struct
{
    unsigned char  nu1                   :1; //205  BITS
    unsigned char  offline_approval      :1; //          Allow offline approval
    unsigned char  cashback_amt_pinpad:1; //          Capture cashback amt at pinpad
    unsigned char  manual_account_entry:1; //          Allow manual account entry
    unsigned char  mu                     :4;
} eps_flags;
unsigned long  cashback_amt_2;           //206  NUM    Pinpad cashback display amount #2
unsigned char  eps_nu1;                  //210
//////////
char account_number_length;              //211  NUM    Account number length
char cdv_formula;                        //212  NUM    CDV scheme for this buy-aid
char voucher_number_length;              //213  NUM    Voucher number length
unsigned char  check_formats;             //214  NUM    Special check formats
unsigned char  change_tender_key[2];      //215  BCD    "change" tensor
unsigned char  reserved[294];             //217  NUM    Not used
unsigned char  QuickDEX_flg;              //511  BITS   QDEX reserved flag
} MED_REC;
#define MED_KEY_LEN (lenof(MED_REC,key))

```

RETN.QDX (13)

File Type	Relative
Record Size	64
Key Length	0

Record Format

typedef struct	//OFS	Type	Description
{			
unsigned char name[16];	//0	ASCII	Description
struct	//16	BITS	
{			
unsigned char valid :1;	//		Valid return type
unsigned char return_to_stock :1;	//		Return item to stock
unsigned char type :3;	//		Type of return
	//		0 - transaction return
	//		1 - general item return
	//		2 - specific item return
	//		3 - link item
unsigned char cash_only :1;	//		Cash only given
unsigned char price_conf :1;	//		Confirm price
unsigned char multi_tender :1;	//		Allow multi tender
} flags;			
unsigned char item_code[7];	//17	BCD	Return item code
unsigned char item_no;	//24	NUM	Return control check
unsigned char print_description[20];	//25	ASCII	print description
unsigned short balancing_dept;	//45	NUM	balancing department
unsigned char slip_template[10];	//47	ASCII	slip template
unsigned char slip_copies;	//57	NUM	slip copies
unsigned char opt_bottle :1;	//58	BIT	Bottle deposit/refund
unsigned char opt_nu1 :1;	//58	BIT	Not used
unsigned char opt_nu2 :1;	//58	BIT	Not used
unsigned char opt_nu3 :1;	//58	BIT	Not used
unsigned char opt_nu4 :1;	//58	BIT	Not used
unsigned char opt_nu5 :1;	//58	BIT	Not used
unsigned char opt_nu6 :1;	//58	BIT	Not used
unsigned char opt_nu7 :1;	//58	BIT	Not used
unsigned char nu[5];	//59	NU	Not used
} RETN_REC;			


```
// Docking stations general parameters
typedef struct docking_parm_           //OFS  Type    Description
{
    struct                             //0
    {
        unsigned char exist           :1;
        unsigned char un              :7;
    }condition[256];
    unsigned char nu[768];             //256    NU      Not used
} DOCKING_PARM;
//***** for Quick docking station *****
typedef struct docking_station_
{
    int usage_counter ;                //0
    char un;                           //2
    struct                             //3
    {
        unsigned char exist           :1;
        unsigned char un              :7;
    }condition;
    struct                             //4
    {
        unsigned char manual          :1;
        unsigned char credit          :1;
        unsigned char staff_disc      :1;
        unsigned char un              :5;
    }clubcard_option;
    struct                             //5    ASCII current clubcard number in use
    {
        char number[20];
        char track1[110];
        char track2[50];
    }clubcard;
    struct                             //185   ASCII current credit card number in use
    {
        char number[20];
        char track1[110];
        char track2[50];
    }credit;

    struct                             //365   ASCII current credit card number in use
    {
```

```
char number[20];
char track1[110];
char track2[50];
}staff_disc;
unsigned char out_time[2];      //545  when Quick left cradle
unsigned char dey_use_time[2]; //547
unsigned char un1[300];        //549
struct                          //849
{
    unsigned char out_cradle      :1;
    unsigned char charged        :1;      //90% charged
    unsigned char filler         :6;
    unsigned char sale_status;      //850 value 1-255
    unsigned int  reserved       :16;      //851
    unsigned char last_broadcast_time[2]; //853 BCD HHMM
}quick_broadcast;
unsigned char lock_drvfile_reserved[10]; //855
char filler[159];                      //865
} DOCKING_STATION;
```

RETN.QDX (13)

File Type	Relative
Record Size	64
Key Length	0

Record Format

typedef struct	//OFS	Type	Description
{			
unsigned char name[16];	//0	ASCII	Description
struct	//16	BITS	
{			
unsigned char valid :1;	//		Valid return type
unsigned char return_to_stock :1;	//		Return item to stock
unsigned char type :3;	//		Type of return
	//		0 - transaction return
	//		1 - general item return
	//		2 - specific item return
	//		3 - link item
unsigned char cash_only :1;	//		Cash only given
unsigned char price_conf :1;	//		Confirm price
unsigned char multi_tender :1;	//		Allow multi tender
} flags;			
unsigned char item_code[7];	//17	BCD	Return item code
unsigned char item_no;	//24	NUM	Return control check
unsigned char print_description[20];	//25	ASCII	print description
unsigned short balancing_dept;	//45	NUM	balancing department
unsigned char slip_template[10];	//47	ASCII	slip template
unsigned char slip_copies;	//57	NUM	slip copies

```

unsigned char opt_bottle    :1;           //58   BIT    Bottle deposit/refund
unsigned char opt_nu1      :1;           //58   BIT    Not used
unsigned char opt_nu2      :1;           //58   BIT    Not used
unsigned char opt_nu3      :1;           //58   BIT    Not used
unsigned char opt_nu4      :1;           //58   BIT    Not used
unsigned char opt_nu5      :1;           //58   BIT    Not used
unsigned char opt_nu6      :1;           //58   BIT    Not used
unsigned char opt_nu7      :1;           //58   BIT    Not used
unsigned char nu[5];        //59   NU     Not used

```

```

} RETN_REC;

```

```

// Docking stations general parameters

```

```

typedef struct docking_parm_           //OFS   Type    Description
{
    struct                             //0
    {
        unsigned char exist           :1;
        unsigned char un              :7;
    }condition[256];
    unsigned char nu[768];             //256   NU     Not used
} DOCKING_PARM;

```

```

//***** for Quick docking station *****

```

```

typedef struct docking_station_
{
    int usage_counter ;                //0
    char un;                           //2

    struct                             //3
    {
        unsigned char exist           :1;
        unsigned char un              :7;
    }

```

```
}condition;
```

```
struct //4
```

```
{
    unsigned char manual    :1;
    unsigned char credit    :1;
    unsigned char staff_disc :1;
    unsigned char un        :5;
```

```
}clubcard_option;
```

```
struct //5 ASCII current clubcard number in use
```

```
{
    char number[20];
    char track1[110];
    char track2[50];
```

```
}clubcard;
```

```
struct //185 ASCII current credit card number in use
```

```
{
    char number[20];
    char track1[110];
    char track2[50];
```

```
}credit;
```

```
struct //365 ASCII current credit card number in use
```

```
{
    char number[20];
    char track1[110];
    char track2[50];
```

```
}staff_disc;
```

```
unsigned char out_time[2]; //545 when Quick left cradle
```

```
unsigned char dey_use_time[2]; //547
```

```
unsigned char un1[300]; //549
```

```
struct                                     //849
{
    unsigned char out_cradle             :1;
    unsigned char charged                 :1;    //90% charged
    unsigned char filler                  :6;

    unsigned char sale_status;            //850 value 1-255
    unsigned int  reserved                 :16;   //851
    unsigned char last_broadcast_time[2]; //853 BCD HHMM
}quick_broadcast;

unsigned char lock_drvfile_reserved[10]; //855

char  filler[159];                        //865

} DOCKING_STATION
```

DISC.QDX (14)

File Type	Relative
Record Size	64
Key Length	0

Record Format

```

typedef struct disc_rec_                                //OFS  Type  Description
{
    unsigned char desc[16];                             //0    ASCII  Description
    struct                                               //16   BITS
    {
        unsigned char upcharge      :1;                //      Upcharge discount
        unsigned char dollar_disc   :1;                //      0-percent 1-amount
        unsigned char additive      :1;                //      Additive discount
        unsigned char op_nu         :1;                //      Not used
        unsigned char default_disc  :1;                //      Default discount
        unsigned char fixed         :1;                //      Fixed discount
        unsigned char item_only     :1;                //      Item discount only
        unsigned char valid_disc    :1;                //      Valid discount
    } flag1;
    struct                                               //17   BITS
    {
        unsigned char auto_if_fixed :1;                //      Automatic if fixed
        unsigned char filler        :6;                //      Not used
        unsigned char override      :1;                //      Price override
    } flag2;
    // The following union depends on the dollar_disc field

    union disc_data_un_                                //18   UNION
    {

```

```
struct
{
    long rate;                //          Discount percentage
    long max_rate;            //          Maximum percentage
} by_prct;

struct
{
    long amount;              //          Discount value
    long max_amount;          //          Maximum value
} by_amount;
} disc_data;

unsigned char chck;           //26   NUM   Control check
unsigned short int dep_number; //27   NUM   Discount department
unsigned char filler4[35];    //29   NU    Not used
} DISC_REC;
```


TRANSACTION.QDX (15)

File Type	FIFO
Record Size	64
Key Length	0
Key Offset	0
QDX Flag Offset	63
Maximum Records	20000
Description	This file contains all POS terminal and back office generated transactions and events.

Note This file is described in its entirety in the Technical Reference TRANSACTION FILE Manual.

EXPN.QDX (16)

File Type	Relative
Record Size	64
Key Length	0

Record Format

typedef struct exp_rec_	//OFS	Type	Description
{			
unsigned char desc[16];	//0	ASCII	Description
struct	//16	BITS	
{			
unsigned char filler:6;	//		Not used
unsigned char receipt:1;	//		0-payout 1-receipt
unsigned char valid:1;	//		Valid expense code
} flags;			
unsigned char nul;	//17	NU	Not used
struct	//18	BITS	
{			
unsigned char nu	:8;	//	Not used
} flags2;			
unsigned char tender_use;	//19	NUM	0-cash only
	//		1-other tender
	//		2-by operator selection
unsigned char nu[44];	//20	NU	Not used
} EXP_REC;			

ACCT.QDX (18)

File Type	Index
Record Size	12
Key Length	10

Record Format

	//OFS	Type	Description
{			
unsigned char key[10];	//0	BCD	Account number
unsigned char chk;	//10	NUM	Control check number
unsigned char qdx_flag;			
} BAD_ACCT;			
#define BA_KEY_LEN (lenof(BAD_ACCT, key))			
#define BA_CHK_LEN (lenof(BAD_ACCT, chk))			

CHCK.QDX (19)

File Type	Relative
Record Size	64
Key Length	0

Record Format

typedef struct chck_rec_	//OFS	Type	Description
{			
unsigned char desc[16];	//0	ASCII	Description
struct	//16	BITS	
{			
unsigned char filler :2;	//		Not used
unsigned char ask_yes_no :1;	//		ask user yes/no
unsigned char mgr :1;	//		Manager keylock required
unsigned char inhibit :1;	//		Inhibit activity
unsigned char svr :1;	//		Supervisor keylock required
unsigned char warning :1;	//		Warning only
unsigned char valid :1;	//		Valid control check
} opt1;			
unsigned char priv;	//17	NUM_A	Privilege
unsigned char templat[2];	//18	NUM_A	Template number
struct	//20	BITS	
{			
unsigned char nu :6;	//		Not used
unsigned char dlyd_author :1;	//		Delayed Authorization
unsigned char qty_amount_on_eot :1;	//		Display quantity & amount
	//		at end of text.
} opt2;			
unsigned char self_checkout_behavior;	//21	NUM	How to behave on a Self Checkout:
	//		0 - Default
	//		1 - None/Ignore
	//		2 - Accept always
	//		3 - Accept w/delayed approval
	//		4 - Reject Always
	//		5 - Reject & Stop
unsigned char rpo_action;	//22	NUM	RPO Action:

```

                                //          0- None
                                //          1- Inform Only
                                //          2- Allow Approval
    unsigned char  nu[41];      //23    NU    Not used
} CHECK_REC;                  //64
```

TAX.QDX (20)

File Type	Relative
Record Size	256
Key Length	0

Record Format

typedef struct tax_rec_	//OFS	Type	Description
{			
unsigned char desc[12];	//0	ASCII	Description
struct	//12	BITS	
{			
char nu :6;	//		Not used
char exemptable :1;	//		Tax exemptable
char use_brakpoint :1;	//		Use break points
} flag1;			
long rate;	//13	NUM_D	Tax rate
unsigned char non_repeat_count;	//17	NUM	Number of non-repeat entries
unsigned char repeat_count ;	//18	NUM	Number of repeat entries
unsigned short min_tax;	//19	NUM_D	Minimum taxable amount
unsigned char tax_char;	//21	ASCII	Tax print character
unsigned char type;	//22	NUM	Tax type
	//		0 - Undefined
	//		48- Normal (USA style)
	//		49- Price includes (GST)
	//		50- Price excludes (GST)
	//		51- Tax calc. before other tax (PST)
	//		52- Tax calc. after other (PST)
unsigned short breaks[50];	//23	NUM_D	Tax brackets
unsigned short non_repeat_value;	//123	NUM_D	Non-repeat value
unsigned short repeat_value;	//125	NUM_D	Repeat value

```
    unsigned char nu[129];          // 127
} TAX_REC;                          // 256
```

ECCA.QDX (21)

File Type	Index
Record Size	256
Key Length	0

Record Format

typedef struct ecca_rec_	//OFS	Type	Description
{			
unsigned char account[10];	//0	BCD	Account number
short information;	//10	NUM	New account inf. options
	//		0 - None
	//		1 - Prompt Y/N at POS for
	//		info
	//		2 - Prompt for info
unsigned char name[20];	//12	ASCII	Name
unsigned char address[40];	//32	ASCII	Address
unsigned char phone[10];	//72	ASCII	Phone number
struct	//82	BITS	
{			
unsigned char check_file :1;	//		Check file
unsigned char bad_account :1;	//		Look up in bad accounts
unsigned char filler :6;	//		Not used
}general_option;			
unsigned char cc_for_bad_account;	//83	NUM	CC for bad account
// The following structure repeats itself for day, week & period			
struct ecca_period_	//84	STRUCT	
{			
unsigned char check_amount :1;	//	BIT	Check amount against limit
unsigned char check_count :1;	//	BIT	Check count against limit
unsigned char check_cash_back :1;	//	BIT	Check cash back against limit
unsigned char filler1 :5;			
long amount;	//	NUM_D	Total amount
long count;	//	NUM	Total count
long cash_back;	//	NUM_D	Total cash back
long limit_amount;	//	NUM_D	Limit amount
unsigned char limit_amount_cc;	//	NUM	CC for limit amount


```
long limit_count;           //    NUM    Limit count
unsigned char limit_count_cc; //    NUM    CC for limit count
long limit_cash_back;       //    NUM_D   Limit cash back
unsigned char limit_cash_back_cc; //    NUM    CC for limit cash back
unsigned char filler2[20];   //    NU     Not used
}ecca_option[3];
char period_start_date[3];   //228  BCD    Period start date
unsigned short period_length; //231  NUM    Period length (in days)
unsigned char bank[5];        //233  BCD    Bank number
unsigned char filler3[17];    //238  NU     Not used
unsigned char qdx_flag[1];    //255  BIT    QDEX reserved flags

} ECCA;
```

DAYDEP.QDX (22)

File Type	Index
Record Size	32
Key Length	5

Record Format

typedef struct	//OFS	Type	Description
{			
unsigned char date[3];	//0	BCD	Trading date
unsigned char dep[2];	//3	BCD	Department number
long sales;	//5	NUM_D	Daily sales VAT exclusive
long items;	//9	NUM	Items sold
short customers;	//13	NUM	Customer count
short return_count;	//15	NUM	Returned items
long sales_with_vat;	//17	NUM	Daily sales VAT inclusive
long return_sales;	//21	NUM_D	Value of returns
short item_scan_count;	//25	NUM	Items scanned
short item_key_count;	//27	NUM	Items keyed
short item_dept_count;	//29	NUM	Open departments sales count
unsigned char qdx;	//31	BITS	QDEX reserved flags
} DAYDEP_REC;			

MONDEP.QDX (23)

File Type Index

Record Size 71

Key Length 6

Record Format

```
// For Hyperama & V7.3, use bin6
typedef struct mondep_rec_          //OFS  Type  Description
{
    unsigned char  year;             //0    BCD    Year
    unsigned char  month;            //1    BCD    Month
    unsigned char  dep[2];           //2    BCD    Department number
    long           nu02 ;             //4    NU    Not used
    long           nu03 ;             //8    NU    Not used
    long           nu04;              //12   NU    Not used
    long           customers;         //16   NUM   Customers count
    long           items;             //20   NUM   Items count
    char           budget6[6];        //24   BIN   Monthly budget
    char           sales6[6];         //30   BIN   Monthly sales VAT exclusive
    char           sales_with_tax6[6]; //36   BIN   Monthly sales VAT inclusive
    unsigned char  nu01[21];          //42   NU    Not used
    unsigned char  qdx;               //63   BITS   QDEX reserved flags
} MONDEP_REC;                       //64

#define MONDEP_KEY_LEN (4)
typedef struct srv_dig_
{
    unsigned char  dig[50][5];
    unsigned char  nu[6];
}SRVDIG_REC;
struct logo_header_    // 32 bytes header
```

```
{  
    char signature[8]; // must be POSLOGO  
    short options;    // for future versions, now must be 0  
    short width;  
    short height;  
    char reserved[18];  
};
```

SAVERCL.QDX (24)

File Type	Index
Record Size	71
Key Length	6

Record Format

typedef struct saved_trans_header_	//OFS	Type	Description
{			
unsigned char pos_no[2];	//0	BCD	POS number
unsigned char ticket_no[2];	//2	BCD	Ticket number
unsigned char count[2];	//4	BCD	Record count
unsigned char transaction_recalled :1;	//6	BIT	Transaction recalled
unsigned char save_at_sale_mode :1;	//6	BIT	Saved at sale mode
unsigned char save_at_tender_mode :1;	//6	BIT	Saved at tender mode
unsigned char bad_transaction :1;	//6	BIT	Bad transaction
unsigned char save_at_training :1;	//6	BIT	Save At Training
unsigned char nu05 :1;	//6	BIT	Not used
unsigned char nu06 :1;	//6	BIT	Not used
unsigned char nu07 :1;	//6	BIT	Not used
unsigned char save_date[3];	//7	BCD	Save date
unsigned char save_time[3];	//10	BCD	Save time
unsigned short save_cashier_no;	//13	NUM	Cashier number
unsigned char save_free[10];	//15	NU	Not used
unsigned char recall_date[3];	//25	BCD	Recall date
unsigned char recall_time[3];	//28	BCD	Recall time
unsigned short recall_cashier_no;	//31	NUM	Cashier number
unsigned char recall_pos_no;	//33	NUM	Recalled at POS number
unsigned char recall_free[9];	//34	NU	Not used
long ticket_amount;	//43	NUM_D	Ticket amount
long media_total;	//47	NUM_D	Media total

```
unsigned short ticket_items;      //51  NUM_D  Ticket Items
unsigned short trans_count;      //53  NUM_D  Trans Count
unsigned char  phone[10];        //55  ASCII  Telephone Number
char filler[6] ;                //65  NU     Not used
}SR_RPT;
```

CRNG.QDX (25)

File Type	Relative
Record Size	1024
Key Length	0

Record Format

typedef struct crng_rec_	//OFS	Type	Description
{			
long iin_start;	//0	NUM	IIN start
long iin_end;	//4	NUM	IIN end
struct	//8	BITS	
{			
unsigned char credit_card :1;	//		1 - credit 0-debit
unsigned char cont_file :1;	//		Continue on validation fail
unsigned char autorisation :1;	//		Force authorization
unsigned char force_auth_key :1;	//		Force authorization for key entry
unsigned char tender_purchase :1;	//		Tender purchase allowed
unsigned char refunds :1;	//		Refund allowed
unsigned char sales :1;	//		Sales allowed
unsigned char valid :1;	//		Valid range
} flags1;			
struct	//9	BITS	
{			
unsigned char print_codes :1;	//		Not used
unsigned char sec_digit :1;	//		Tender predefined
unsigned char key_entry :1;	//		Key entry not allowed
unsigned char date_format :1;	//		Date format
	//		0 - YYMM
	//		1 - MMY
unsigned char start_req :1;	//		Start date required
unsigned char card_issue :1;	//		Card issued required

```

    unsigned char expiry      :1;      //          Expiry date required
    unsigned char if_keyed    :1;      //          Use the above 4 flags
} flags2;

unsigned char pan_len;        //10   NUM    PAN length
unsigned char min_len;        //11   NUM    Minimum length track 2
unsigned char type;           //12   NUM    Tender number
unsigned char club_card_scheme; //13   NUM    Clubcard/Member card scheme
unsigned char class_id ;      //14   NUM    Card range type
                                //          0 - Undefined
                                //          48 - Any class
                                //          49 - Check
                                //          50 - Specific CR/DB card
                                //          51 - Other
                                //          52 - Generic CR/DB card

unsigned long schem_min ;     //15   NUM_D   Scheme minimum value
unsigned long schem_max ;     //19   NUM_D   Scheme maximum value
unsigned char floor_option ;  //23   NUM    Floor options
                                //          0 - Tender amount
                                //          1 - Transaction amount

unsigned long schem_floor_limit ; //24   NUM_D   Scheme floor limit
unsigned long schem_daily_limit ; //28   NUM_D   Scheme daily limit
unsigned char daily_count ;    //32   NUM    Scheme daily count
long         fallback_schem_min; //33   NUM_D   Fallback scheme minimum
long         fallback_schem_max; //37   NUM_D   Fallback scheme maximum
unsigned char mpr_number;      //41   NUM    Multi part receipt number
unsigned char eca_timeout;     //42   NUM    Number of bad swipes before manual entry
unsigned char exp_date ;       //43   NUM    Expiry date start position
unsigned char pan_lay_1 ;      //44   NUM    PAN layout 1
unsigned char pan_lay_2;       //45   NUM    PAN layout 2
unsigned char pan_lay_3;       //46   NUM    PAN layout 3
unsigned char pan_lay_4;       //47   NUM    PAN layout 4
unsigned char pan_lay_5;       //48   NUM    PAN layout 5

struct                        //49   BITS

```


[illegible]

```

//          3=PHH Modulu 10 double and double
//          4=Diacard modulu 10 double & double
} flags5;

struct          //53  BITS
{
    unsigned char  purchase_type      :4; //          Purchase type
//          0=all products
//          1=TESCO domestic cards
//          2=TESCO fuel cards
//          3=PHH cards
//          4=override cards
//          5=dialcard
    unsigned char  grty_cheque        :1; //          Guarantee check
    unsigned char  fuel_test_card     :1; //          Fuel test card
    unsigned char  extra_auth_consec_use:1; //          Extra authorization on consecutive usage
    unsigned char  variable_PAN_lengh :1; //          Variable PAN length
} flags6;

unsigned char control_check;          //54  NUM          Control check
unsigned char service_code_loc;       //55  NUM          Service code location
unsigned char one_in_n;               //56  NUM          One in 'n' authorization
unsigned char mult_trans_check_no;    //57  NUM          Multiple transaction check number
unsigned char nu2[2];                 //58  NU           Not used
struct          //60  BITS
{
    unsigned char  luhn_check_digit   :1; //          Luhn check digit
    unsigned char  extra_check_digit  :1; //          Extra check digit
    unsigned char  receipt_on_denial  :1; //          Print receipt on denial
    unsigned char  fly_buy            :1; //          loyalty fly buy
    unsigned char  non_edc_cards      :1; //          non-EDC Cards
    unsigned char  nu                 :3; //          Not used
} flags7;

unsigned char loc_of_reg_check;        //61  NUM          Not used
unsigned char loc_of_milage ;          //62  NUM          Location of mileage & registration

```

unsigned char loc_of_check_digit;	//63	NUM	Location of check digit
unsigned char tender_desc[16];	//64	ASCII	Tender description
unsigned char validity_period;	//80	NUM	Location of validation period
unsigned long refund_schem_min ;	//81	NUM_D	Refund scheme minimum
unsigned long refund_schem_max ;	//85	NUM_D	Refund scheme maximum
unsigned long cash_back_schem_min ;	//89	NUM_D	Cash back scheme minimum
unsigned long cash_back_schem_max ;	//93	NUM_D	Cash back scheme maximum
unsigned char loc_of_start_date;	//97	NUM	Location of start date
unsigned char loc_of_items_range;	//98	NUM	Location of items range
unsigned char service_code_length;	//99	NUM	Service code length
unsigned char narrative_credit;	//100	NUM	Card text narration for credit
unsigned char narrative_debit ;	//101	NUM	Card text narration for debit
unsigned char srv_valid_digits;	//102	NUM	Service code valid digits
long staff_discount_precent;	//103	NUM_D	Staff discount percent
unsigned char merchant_number[15];	//107	ASCII	Merchant number
unsigned char sponcer_no;	//122	NUM	Sponsor number
unsigned char mask_character;	//123	ASCII	PAN masking character
unsigned char loc_schem;	//124	NUM	Location of scheme number in card number
unsigned char len_schem;	//125	NUM	Length of scheme number in card number
unsigned char pan_mask_start;	//126	NUM	PAN masking start pos.
unsigned char pan_mask_end;	//127	NUM	PAN masking end position

// For ISS45 V7.3 & TESCO, card range record length is 1024. For other

// versions, record length is 128

```

struct                                //128  BITS
{
    unsigned char  produce_cashback_rec :1; //          Produce CASHBACK record in APACS50 file
    unsigned char  ola_avail_sale_cash  :1; //          OLA available on sale / cashback
    unsigned char  ola_avail_refund      :1; //          OLA available on sale / refund
    unsigned char  voice_autho_avail     :1; //          Voice authorization available
    unsigned char  reversale_supported   :1; //          Reversals supported
    unsigned char  date_valida_key       :1; //          Date validation on key entry
    unsigned char

```

```

    card_not_avail_on_opt_sales      :1; //          Card not available on OPT sales.
    unsigned char
        balance_processing_enabled    :1; //          balance_processing_enabled
} flags8;
unsigned long offline_limit_sale;      //129  NUM_D    Offline limit sale
unsigned long offline_limit_cashback;  //133  NUM_D    Offline limit cashback
unsigned long offline_limit_refund;    //137  NUM_D    Offline limit refund
short        ola_min_responce_time;    //141  NUM      OLA minimum response time
short        ola_time_out;             //143  NUM      OLA time out
unsigned char max_pan_length;          //145
unsigned char branch_on_track2;        //146
unsigned char staff_disc_tnd_no;       //147          NUM Like: Tender number
unsigned char card_usage;              //148  NUM      0 Don't care
                                           //          1 swipe only
                                           //          2 Manual entry
unsigned char tnd2;                   //149  NUM      Linked tender #2
unsigned char tnd3;                   //150  NUM      Linked tender #3
unsigned char tnd4;                   //151  NUM      Linked tender #4
unsigned char tnd5;                   //152  NUM      Linked tender #5
unsigned char filler[826];            //153  NU       Not used
unsigned char kiosk_merchant_number[15]; //979  ASCII    Reserved
unsigned char opt_merchant_number[15]; //994  ASCII    Reserved
long        opt_daily_limit;          //1009 NUM_D    Reserved
long        opt_night_mode_limit;     //1013 NUM_D    Reserved
long        opt_ola_pre_autho_amount;  //1017 NUM_D    Reserved
unsigned char no;                     //1021 NU       Not used
unsigned char internal_use;           //1022 NUM      Internal use
unsigned char qdx_flag;               //1023 BITS     QDEX reserved flags
} CRNG_REC;

```

GROUPNM.QDX (26)

File Type	Relative
Record Size	32
Key Length	0

Record Format

typedef struct grp_rec_	//OFS	Type	Description
{			
unsigned char name[16];	//0	ASCII	Group name
unsigned char super;	//16	NUM	Group type
	//		0-group
	//		1-super group
unsigned char super_no;	//17	NUM	Linked super group number
struct	//18	BITS	
{			
unsigned char non_bank :1;	//		Non-bankable super group
unsigned char nu :7;	//		Not used
} opt1;			
unsigned char nu [13];	//	NU	Not used
} GRP_REC;			

SHORTGRP.QDX (27)

File Type	Index
Record Size	64
Key Length	25

Record Format

```

typedef struct plu_group_rec_           //OFS  Type    Description
{
    unsigned char num;                  //0     NUM     Group number
    unsigned char opcode;               //1     NUM     Opcode
                                         //      NUM     0 - header
                                         //      NUM     1 - data
    unsigned char name[16];             //2     ASCII   Group/item description
    unsigned char plu[7];               //18    BCD     PLU number
    long price;                        //25    NUM_D   Item price
    unsigned char delete_eod :1;        //29    BIT     delete in EOD
    unsigned char nu[33];               //30    NU      Not used
    char qdx_flag;                     //63    BITS    QDEX reserved flags
} PLU_GROUP;

#define PG_KEY_LEN    (lenof(PLU_GROUP, num)+lenof(PLU_GROUP,opcode) + \
                      lenof(PLU_GROUP,name)+lenof(PLU_GROUP,plu))

typedef struct pos_miscell_menu_        //OFS  Type    Description
{
    unsigned char opcode;               //0     NUM     opcode 0xa0
    unsigned char num;                 //1     NUM     menu number
    unsigned char order;               //2     NUM     lines order index
    unsigned char filler[22];          //3     NU      Not used
    unsigned char name[16];            //25    ASCII   menu description
    unsigned short func;               //41    NUM     Keyboard function
    unsigned char key;                 //43    NUM     Keylock
    unsigned char privilage;           //44    NUM     Privilege

```

```

    unsigned char nu[18];                //45    NU        Not used
    char qdx_flag;                       //63    BITS    QDEX reserved flags
} POS_MISCELL_MENU;

```

[illegible]

```
struct hist_cash_rec_
```

```
{
    unsigned char pos_num;

    unsigned int date;

    unsigned char time[3]; // ICL pos format

    unsigned short cashier;

    union
    {
        unsigned char buff[120];

        struct trs_lift_ t_lift;

    } trn;
};
```

```
#define MAX_POS_PARAM_RECS    200
#define POS_PARAM_QUICK_REC    20    // Quick parameters
#define POS_PARAM_USER_DEF    50    // 1 record for user defined parameters
#define POS_PARAM_SYSTEM_REC  150    // some system parameters
#define POS_PARAM_FUEL_REC    180    // start of fuel/pump info
#define MAX_POS_MENU_LEN    50
```

```
#define POS_PARAM_SYSTEM_OFS ((long)POS_PARAM_SYSTEM_REC*1024L)
```

```
#define MAX_COST_PLUS 20
```

```
#define MAX_CDV_SCHEME 16
```

AUDIT.QDX (31)

File Type	FIFO
Record Size	128
Key Length	0

Record Format

```

typedef struct audit_rec_          //OFS  Type  Description
{
    char opcode;                   //0    NUM    Opcode
                                   //      1 - Insert
                                   //      2 - Update
                                   //      3 - Delete
    unsigned char plu[7];          //1    BCD    PLU number

    // The following structure is repeated twice - for the old & new data
    struct audit_data_             //8     STRUCT
    {
        unsigned char dep[2];      //8 -54 BCD    Department number
        unsigned char name_pos[12]; //10-56 ASCII  POS name
        unsigned char tax;         //22-68 BITS   Tax flags
        unsigned char flags1;      //23-69 BITS   Item flags1 (see PLU relative structure)
        unsigned char flags2;      //24-70 BITS   Item flags2 (see PLU relative structure)
        long price;                //25-71 NUM_D  Item price
        unsigned char cost_per_case_price[4]; //29-75 BCD    Cost per case
        unsigned char unit_per_case[2]; //33-79 BCD    Units per case
        unsigned char tare;        //35-81 NUM    Tare weight pointer
        unsigned char type;        //36-82 NUM    Item type
                                   //      1 - MSU > 1
                                   //      2 - Weighed item
                                   //      4 - Store coupon
        unsigned char msu_c;       //37-83 NUM    MSU
    }

```

```

char          inter_num[7];          //38-84 BCD      Internal code
long second_price;                   //45-91 NUM      Frequent shopper discount value
char freq_shop_type;                 //49-95 NUM      Frequent shopper discount type
char          nu[4];                  //50-96

} new_inf, old;

struct                                           //100 BITS
{
    unsigned char head_office    :1;    //          Report to head office
    unsigned char host_change    :1;    //          Change made by host
    unsigned char nu             :6;    //          NU      Not used
} opt1;

struct                                           //101 BITS
{
    unsigned char app_gen        :1;    //          generated by application, not DRVFILE
    unsigned char ignore_next    :1;    //          Ignore next record
    unsigned char nu             :6;
} opt2;

unsigned char time[4];                    //102          HH:MM:SS:##
unsigned char modify_user;                //106 NUM      User number that modified the item
char          nu[12];                     //107 NU      Not used
// old area
long          o_sales;                    //119 NUM      Old quantity sold
long          o_revenue;                  //123 NUM_D    Old revenue
char          qdx_flag;                   //127 BITS     QDEX reserved flags
} AUDIT_REC;

typedef struct audit_ascii_rec_              //OFS  Type  Description
{
    unsigned char opcode;                  //0    ASC_N    Opcode
                                           //          1 - Insert
                                           //          2 - Update
                                           //          3 - Delete
    unsigned char plu[12];                  //1    ASC_N    PLU number
// The following structure is repeated twice - for the old & new data

```

```

struct audit_ascii_data_          //13
{
    unsigned char price[7];        //    ASC_ND  Item price
    unsigned char name[12];        //    ASCII  POS name
    unsigned char dep[3];          //    ASCII  Department number
    unsigned char type;            //    ASC_N   Item type
                                   //           1 - MSU > 1
                                   //           2 - Weighed item
                                   //           4 - Store coupon

    unsigned char msu[2];          //    ASC_N   MSU
    unsigned char filler1[3];      //    NU      Not used
    unsigned char tax[4];          //    ASCII  Tax flags
                                   //           'Y' - tax on
                                   //           'N' - tax off

    unsigned char tare[2];         //    NUM     Tare weight pointer
    unsigned char pay_credit_foodstamp; //    ASCII  Payment by food stamps
    unsigned char force_qty;       //    ASCII  Force quantity
    unsigned char cost_per_case[7]; //    ASC_ND  Cost per case price
    unsigned char unit_per_case[3]; //    ASC_N   Units per case
} new_inf, old;

unsigned char o_sales[7];          //105 ASC_N   Old quantity sold
unsigned char o_revenue[11];       //112 ASC_ND  Old revenue
unsigned char nu[5];               //123 NU      Not used
} AUDIT_ASCII;

#define WEEK_DEPTS      133
#define WEEK_TAXES      8
#define WEEK_MEDIAS     25

```

COUPON.QDX (32)

File Type	Relative
Record Size	1024
Key Length	0

Record Format

typedef struct coupon_p_rec_	//OFS	Type	Description
{			
struct	//0	BITS	
{			
char in_use :1;	//		Record in use
char item_deprtmnt :1;	//		Item/department
	//		0 - item
	//		1 - department
char found :1;	//		Item/department found/not
	//		found in basket
char print_barcode :1;	//		Print barcode
char filler :4;	//		Not used
} flag;			
unsigned char plu_dep_number[7];	//1	BCD	PLU/department number
unsigned char print_control_1 ;	//8	NUM	Print control for line 1
	//		0 - normal
	//		1 - bold
unsigned char text_line_1[40] ;	//9	ASCII	Text line 1
unsigned char print_control_2 ;	//49	NUM	Print control for line 2
unsigned char text_line_2[40] ;	//50	ASCII	Text line 2
unsigned char print_control_3 ;	//90	NUM	Print control for line 3
unsigned char text_line_3[40] ;	//91	ASCII	Text line 2
unsigned char print_control_4 ;	//131	NUM	Print control for line 4
unsigned char text_line_4[40] ;	//132	ASCII	Text line 2

```

unsigned char print_control_5 ;           //172  NUM    Print control for line 5
unsigned char text_line_5[40] ;           //173  ASCII  Text line 2
unsigned char barcode[7] ;               //213  BCD    Barcode number
unsigned char post_logo ;                 //220  NUM    POST logo
char nu[3] ;                             //221  NU     Not used coupon
long ticket_amount_to_print_coupon ;      //224  NUM_D  Ticket amount to print coupon
long minimum_purchase_amount_required;    //228          Minimum purchase amount to print coupon
unsigned char CC_for_minimum_purchase;    //232  NUM    Control check for minimum purchase
long amt_for_cpn_threshold;               //233  NUM    Show CC when amt to reach threshold is less then
unsigned char filler[786];                //237  NU     Not used
unsigned char qdx_flag;                   //1023  NU     Not used
} COUPON_P;

// QDX file 78
// TESCO coupons
// From V7.3, record length is 128
//-----
typedef struct t_coupon_                  //OFS  Type  Description
{
    unsigned char coupon_number[7];       // 0  BCD    coupon number
    unsigned char description[48];         // 7  ASCII  coupon description
    unsigned char threshold;               // 55  NUM    threshold
    unsigned char expiry_date[3];          // 56  BCD    expiry date
    unsigned char type;                    // 59  NUM    coupon type
    long coupon_value;                     // 60  NUM    coupon value
    long trans_total;                      // 64  NUM    transaction total
    struct flags_                          // 68
    {
        unsigned char blocked :1;         //      BIT    blocked coupon
        unsigned char filler :7;          //      BITS   Not used
    }flags;

    unsigned char limit_upto;              // 69  NUM    limit up to

```

```
    unsigned char  filler[45];           // 70  NU    Not used
    long           day_value;            //115  NUM    Daily value
    short          day_qty;              //119  NUM    Daily count
    long           week_value;           //121  NUM    Weekly value
    short          week_qty;             //125  NUM    Weekly count
    unsigned char  qdx_flag;             //127  BITS   QDEX reserved flags
}T_COUPON;
```

```
// Member promotion types
```

```
#define MPT_IMMEDIATE      1
#define MPT_EHNANCED      2
#define MPT_GROUP          3
#define MPT_ORDER          4
#define MPT_ORD_LEVEL     5
#define MPT_ORD_GROUP      6
#define MPT_FREE_ITEM      7
```

```
// Member promotion rewards
```

```
#define MPT_NOT_DEFINED    0
#define MPT_CASH_VALUE     1
#define MPT_PERCENT        2
#define MPT_POINTS        3
#define MPT_PER_ITEM_DISC  4
#define MPT_BOGO_LOW       5
#define MPT_BOGO_HIGH      6
#define MPT_STUB           7
#define V8_NEW_PRICE       8
#define MPT_TARGET_MSG     12
#define MPT_PERCENT_OFF_ORDER 13
#define MPT_NEW_PRICE_PER_QTY 17
#define MAX_REWARD_TYPE   99
```

```
//*****//
```

```
//          Additional Promotion (Savers)          //
//
// File Name   : SAVERS.QDX           File Number: 76   //
//
// File Type    : Relative              //
// Record Size  : 256                  //
// Key Length   : 0                    //
//
//*****//
```

```
typedef struct promotion_
```

```
{
    //-- GENERAL part -----

                                //OFS  Type  Description
unsigned char  promotion_number[2];  //0    BCD    Promotion number
unsigned char  promotion_type;       //2    NUM    Promotion type
                                //          1 Store/Immediate Promotion
                                //          2 Multisaver/Enhanced
                                //          3 Linksaver/Group promotion
                                //          4 Extended Multisaver/Order promotion
                                //          5 Extended Linksaver/order group promotion
                                //          6 Order/Group promotion
                                //          7 One free
unsigned char  start_day[3];         //3    BCD    Start day YYMMDD
unsigned char  end_day[3];           //6    BCD    End day   YYMMDD

struct prom_flags_                  //9    STRUCT
{
    unsigned char  active             :1; //    BIT    Active prom
    unsigned char  m_card_dependency :1; //    BIT    Member card dependency
    unsigned char  avail_all_schemes :1; //    BIT    Available for all schemes
    unsigned char  allow_when_offline :1; //    BIT    Allow this promo when Server is offline COREMA
    unsigned char  low_to_high        :1;
```

```

unsigned char delayed_promotion      :1;
unsigned char value_enter_by_cshr    :1;
unsigned char update_loyalty_server  :1; //COREMA

}flags;

unsigned char  description[20];        //10  ASCII  Description
unsigned char  department[2];          //30   BCD    Department
unsigned char  reward_type;            //32   NUM    Reward type
                                         //    1 - Cash value
                                         //    2 - Percentage off
                                         //    3 - Clubcard/Member card points
                                         //    4 - Per item discount
                                         //    5 - Low BOGO
                                         //    6 - High BOGO
                                         //    7 - STUB
                                         //    8 - New price
                                         //   12 - Target Message
                                         //   17 - New price per quantity
struct scheme_list_                   //33   NUM    Scheme numbers 1-10
{
    unsigned char scheme;

}scheme_list[10];

long          reward;                  //43   NUM_D   Reward
short         club_card_scheme_number; //47   NUM    Clubcard/Member card scheme number
long          limited_qty;             //49   NUM    Limited quantity
unsigned char free[5];                 //53   NU     Not used

union
{
    struct ext_banded_                 //58   STRUCT  (banded)

```

```

{
    unsigned char  bucket_type;    //58  NUM    Bucket type
                                    //      1 Standard / Units
                                    //      2 Loose / Weighed
                                    //      3 Department

    long  minimum_item_value;    //59  NUM_D  Minimum item value
    long  maximum_reward;        //63  NUM_D  Maximum reward
    struct ext_banded_band_      //67  STRUCT
    {
        long  quantity;          //67  NUM    Quantity
        long  value;              //71  NUM_D  Value
        long  reward;             //75  NUM_D  Reward
    }ext_banded_bands[5];
}ext_banded;

struct ext_not_banded_          //58  STRUCT
{
    unsigned char  bucket_type;    //58  NUM    Bucket/Group type
                                    //      1 Standard / Units
                                    //      2 Loose / Weighed
                                    //      3 Department

    long  minimum_item_value;    //59  NUM_D  Minimum item value
    long  maximum_reward;        //63  NUM_D  Maximum reward Purchase requirements
    long  treshold_quantity;      //67  NUM    Threshold quantity
    long  treshold_value;         //71  NUM_D  Threshold value
    long  treshold_reward;        //75  NUM_D  Threshold reward
    long  step_quantity;          //79  NUM    Step quantity
    long  step_value;             //83  NUM_D  Step value
    long  step_reward;            //87  NUM_D  Step reward
    unsigned char reserved[36];    //91

}ext_not_banded;

```



```

struct ext_linksave_           //58  STRUCT
{
    long maximum_reward;       //58  NUM_D  Maximum reward
    struct ext_linksave_bucket_ //62  STRUCT
    {
        unsigned char bucket_type; //62  NUM    Bucket/Group type
                                   //          1 Standard / Units
                                   //          2 Loose / Weighed
                                   //          3 Department
                                   //          Purchase requirements
        long minimum_item_value; //63  NUM_D  Minimum item value
        long treshold_quantity;  //67  NUM    Threshold quantity
        long treshold_value;     //71  NUM_D  Threshold value
    }ext_linksave_buckets[3];

    unsigned char reserved[26];  //101

}ext_linksave;

struct linksave_              //58  STRUCT
{
    struct linksave_bucket_    //58  STRUCT
    {
        unsigned char bucket_type; //58  NUM    Bucket/Group type
                                   //          1 Standard / Units
                                   //          2 Loose / Weighed
                                   //          3 By value
        long treshold_quantity;    //59  NUM    Threshold quantity
    }linksave_buckets[3];

    unsigned char reserved[54];    //73

}linksave;

```

```

struct multisaver_           //58  STRUCT
{
    unsigned char  bucket_type; //58  NUM      Bucket/Group type
                                //          1 Standard / Units
                                //          2 Loose / Weighed
                                //          3 By value

    long  treshold_quantity; //59  NUM      Threshold quantity
    long  step_quantity;    //63  NUM      Step quantity
    unsigned char reserved[60]; //67

}multisaver;

struct store_promotion_     //58  STRUCT
{
    //Store promotion (immediate)
    unsigned char  bucket_type; //58  NUM      Bucket/Group type
                                //          1 Standard / Units
                                //          2 Loose / Weighed

    long  treshold_quantity; //59  NUM      Threshold quantity
    long  step_quantity;    //63  NUM      Step quantity
    unsigned char reserved[60]; //67

}store_promotion;

struct free_item_
{
    struct free_bucket_     //58  STRUCT
    {
        unsigned char  bucket_type; //58  NUM      Bucket/Group type
                                    //          1 Standard / Units
                                    //          2 Loose / Weighed

        long  treshold_quantity; //59  NUM      Threshold quantity
    }free_buckets[3];
}

```

```
    unsigned char  plu[7];                //73   BCD       PLU number
```

```
    unsigned char reserved[47];           //80
```

```
    } free_item;
```

```
}purchase_request;
```

```
struct member_segments_                  //127
```

```
{
```

```
    unsigned short seg_1    :1;
```

```
    unsigned short seg_2    :1;
```

```
    unsigned short seg_3    :1;
```

```
    unsigned short seg_4    :1;
```

```
    unsigned short seg_5    :1;
```

```
    unsigned short seg_6    :1;
```

```
    unsigned short seg_7    :1;
```

```
    unsigned short seg_8    :1;
```

```
    unsigned short seg_9    :1;
```

```
    unsigned short seg_10   :1;
```

```
    unsigned short seg_11   :1;
```

```
    unsigned short seg_12   :1;
```

```
    unsigned short seg_13   :1;
```

```
    unsigned short seg_14   :1;
```

```
    unsigned short seg_15   :1;
```

```
    unsigned short seg_16   :1;
```

```
} segments;
```

```
unsigned char  start_time[2];             //129  BCD_T    Promotion start time
```

```
unsigned char  end_time[2];               //131  BCD_T    Promotion end time
```

```
struct prom_active_day_                   //133  BITS    Active days for promotion
```

```
{
```

```
    unsigned char  opt_Sunday    :1;
```

```
    unsigned char  opt_Monday    :1;
```

```

    unsigned char  opt_Tuesday      :1;
    unsigned char  opt_Wednesday   :1;
    unsigned char  opt_Thursday    :1;
    unsigned char  opt_Friday      :1;
    unsigned char  opt_Saturday     :1;
    unsigned char  opt_07           :1;
} act_day;

struct adv_prom_day_time_          //134  STRUCT  "Happy Hour" for each day of the week
    unsigned char  start_time[2];   //      BCD_T   Start time
    unsigned char  end_time[2];     //      BCD_T   End time
} day_time[7];

struct
{
    unsigned char  trig_grp_1  :1;   //162  BIT      Group 1 is the trigger
    unsigned char  trig_grp_2  :1;   //162  BIT      Group 2 is the trigger
    unsigned char  trig_grp_3  :1;   //162  BIT      Group 3 is the trigger
    unsigned char  trig_grp_nu4:1;   //162  BIT      not used
    unsigned char  trig_grp_nu5:1;   //162  BIT      not used
    unsigned char  nu           :3;   //162  BIT      not used
} flags2;

unsigned char  cc_for_min_purchase; //163          Control check for min purchase
long          min_purchase_amt;     //164          Min. purchase amount
unsigned char  inf_nu[23];          //168  NU      Not used

// *** END OF INFORMATIVE PART ***

// *** START OF SALES PART ***

long          day_qty;              //191  NUM      Unit Quantity sold today
long          day_weight;           //195          Weight qty sold today

```

long	day_sales;	//199	NUM	Sales today
long	day_markdown;	//203	NUM	Markdown/Points rewarded
long	prd_qty;	//207	NUM	Unit Quantity sold
long	prd_weight;	//211		Weight qty sold today
long	prd_sales;	//215	NUM	Sales
long	prd_markdown;	//219	NUM	Markdown/Points rewarded
unsigned char	sales_nu[32];	//223	NUM	Not used
unsigned char	qdx_flag;	//255	BITS	QDEX reserved flags

} TESCO_PROMO;

File Type	Relative
Record Size	1024
Key Length	0

```
// The POS logo is kept in two types of records.
```

```
// First record is a header.
```

```
// Records 1-32 hold the bitmap.
```

	//OFS	Type	Description
<pre>typedef union pos_logo_rec_</pre>			
{			
struct pos_logo_header_	//	STRUCT	Logo header
{			
unsigned char name[16];	//0	ASCII	Logo name
struct	//16	BITS	
{			
unsigned char used :1;	//		Logo in use
unsigned char nu :7;	//		Not used
}opt;			
short width;	//17	NUM	Image width
short height;	//19	NUM	Image height
unsigned char nu[1003];	//21	NU	Not used
} head;			
unsigned char bmp[1024];	//0	NUM	Image data
} LOG_REC;	// 1024		

```
#define RECS_PER_LOGO      32
```

SECURE.QDX (36)

File Type	Relative
Record Size	34
Key Length	0

Record Format

typedef struct secure_rec_	//OFS	Type	Description
{			
char number[2];	//0	ASC_N	User number
char fullname[24];	//2	ASCII	User name
char password[5];	//26	ASC_N	Password
char priority;	//31	NUM	Priority
char filler1;	//32	NU	Not used
char qdexflag;	//33	BIT	QDEX reserved flags
char cash_num[2];	//34	BCD	cashier number for privilege
char notused[220];	//36	not used	
} SEC_REC;			

SUBRELAT.QDX (37)

File Type	Relative
Record Size	128
Key Length	6

Record Format

typedef struct subdep_rel_rec_	//OFS	Type	Description
{			
unsigned char sub_key[6];	//0	BCD	Subdepartment number
unsigned char sub_name[16];	//6	ASCII	Description
unsigned char department_number[2];	//22	BCD	Department number
unsigned char group;	//24	NUM	Group number
unsigned char sub_character[1];	//25	NU	Not used
unsigned char sub_filler[6];	//26	NU	Not used
long tmp_day_sub_revenue_sale;	//32	NUM	Temporary Daily revenue sales
long day_sub_qty_sale;	//36	NUM	Daily quantity sales
long day_sub_revenue;	//40	NUM_D	Daily revenue
long prv_sub_qty_sale;	//44	NUM	Previous day quantity sold
long prv_sub_qty_revenue;	//48	NUM_D	Previous day revenue
long week_sub_qty_sale;	//52	NUM	Weekly quantity sales
long week_sub_revenue;	//56	NUM_D	Weekly revenue
long prv_sub_week_qty_sale;	//60	NUM	Previous week quantity sold
long prv_sub_week_revenue;	//64	NUM_D	Previous week revenue
long month_sub_qty_sale;	//68	NUM	Monthly quantity sales
long month_sub_revenue;	//72	NUM_D	Monthly revenue
long prv_sub_month_qty_sale;	//76	NUM	Previous month quantity sold
long prv_sub_month_revenue;	//80	NUM_D	Previous month revenue
long year_sub_qty_sale;	//84	NUM	Yearly quantity sales
long year_sub_revenue;	//88	NUM_D	Yearly revenue
long z_sub_qty_sale;	//92	NUM	Quantity sold since last Z
long z_sub_revenue;	//96	NUM_D	Revenue since last Z


```
unsigned char sub_filler1[27];          //100  NU      Not used
unsigned char qdx_fl[1];                //127  BITS    QDEX flags
} SUBDEP_RELAT;                         //128

typedef struct subdep_co_rec_
{
    SUBDEP_INDEX   idx;          // 0
    SUBDEP_RELAT   rlt;          // 9
} SUBDEP_REC;                  // 9+128 = 137
```

SUBDEP.QDX (38)

File Type	Index
Record Size	9
Key Length	6

Record Format

	//OFS	Type	Description
{			
char key[6];	//0	BCD	Subdepartment number
char filler[3];	//6	NU	Not used
} SUBDEP_INDEX;			
#define SUB_KEY_LEN (lenof(SUBDEP_INDEX, key))			

BUYAIDS.QDX (39)

File Type	Index
Record Size	512
Key Length	2

Record Format

typedef struct buy_aid_rec_	//OFS	Type	Description
{			
unsigned char key[2];	//0	BCD	Buy aid number
unsigned char society[16];	//2	ASCII	Description
unsigned char acc_prompt[16];	//18	ASCII	Account prompt
unsigned char eft_desc[16];	//34	ASCII	EFT description
unsigned char tender_type;	//	NU	Not used
struct	//51	BITS	
{			
unsigned char val_not_req :1;	//		Value not required
unsigned char cancel_sub :1;	//		SVR at Cancel/Subtract
unsigned char inhibit :1;	//		Cancel/Subtract inhibit
unsigned char purchas :1;	//		Tender purchase allowed
unsigned char mag_card :1;	//		MCR allowed
unsigned char acc_req :2;	//		Account number required
	//		0-none
	//		1-entered by cashier
	//		2-entered by customer
unsigned char overtendering:1;	//		Over-tender allowed
} flags1;			
struct	//52	BITS	
{			
unsigned char round_down :1;	//		Not used
unsigned char allow_refound:1;	//		Refund allowed
unsigned char round :1;	//		Rounding allowed

```

unsigned char mr_req;                //55  NU    Not used
unsigned      endors_temp_no  ;      //56  NUM    Endorsement template number
unsigned      template_no    ;      //58  NUM    Template number
unsigned long currency_rate  ;      //60  NUM_D   Currency rate
unsigned char mcr;                //64  NUM    MCD number
unsigned      eca_timeout;      //65  NUM    ECA timeout
unsigned char other_tender;      //67  NU    Not used
long          max;                //68  NUM_D   Maximum value
long          min;                //72  NUM_D   Minimum value
long          over;              //76  NUM_D   Over tender value
long          floor;             //80  NUM_D   Floor limit value
long          amount_req_control_chck; //84  NUM_D   Amount required for CC
unsigned char max_chck_no;        //88  NUM    Maximum value CC
unsigned char min_chck_no;        //89  NUM    Minimum value CC
unsigned char over_tender;        //90  NUM    Over tender CC
unsigned char floor_chck_no;      //91  NUM    Floor limit CC
unsigned char daily_chck_no;      //92  NUM    Daily value CC
unsigned char repeat_chck_no;     //93  NUM    Repeat CC
unsigned char autho_chck_no;      //94  NUM    Authorization CC
unsigned char repcard_chck_no;    //95  NUM    Repeat card number CC
unsigned char filler_chck_no[3];  //96  NU    Not used
unsigned char retailer;           //99  NUM    Retailer information
                                   //100 NUM_D   Load amount
long          loan_amount;        //104 NUM    CC for changing opening
unsigned char loan_control_check; //    float
unsigned char Rounding_Value;     //105 NU    Not used
unsigned char mcr_timeout;        //106 NUM    MCR timeout
unsigned char mcr_misread;        //107 NUM    MCT misread
struct                                                //108 BITS
{
    unsigned char tender_cash_purch:2; //    Automatic purchase of...
    unsigned char nul          :1; //    Not used
    unsigned char ecca          :1; //    Use ECCA on this media
    unsigned char nu2           :4; //    Not used

```

```

} flags5;

unsigned char  Reserved[102];           //109  NU      Not used
char account_number_length;           //211  NUM     Account number length
char cdv_formula;                     //212  NUM     CDV scheme for this buy-aid
char voucher_number_length;           //213  NUM     Voucher number length
unsigned char  reserved2[273];         //214  NU      Not used
long  banked_amount;                  //487  NUM_D   Amount banked
long  banked_qty;                    //491  NUM     Quantity banked

// The following structure repeats itself for current & previous shift
struct buyaid_rec_dclr_                //495  STRUCT
{
    long  amount;                     //      NUM_D   Total cashier declarations amount
    long  qty;                        //      NUM     Total cashier declarations quantity
}shift[2];

unsigned char  QuickDEX_flg;           //511  BITS    QDEX reserved flags

} BUYAID_REC;

#define  BUY_KEY_LEN  (lenof(BUYAID_REC,key))

typedef struct
{
    unsigned char  staff_num[3];      // 0
    unsigned char  name[26];          // 3
    unsigned char  inits[4];          // 29
    int  store_num;                   // 33
    unsigned char  group;              // 35
    long  salary;                     // 36
    long  dis_alow;                    // 40
    long  dis_received;                // 44
    long  tick_per;                    // 48      percent per ticket
    unsigned int  date;                // 52
    long  last_mon_disc;               // 54

```

```
    char        filler[6];          // 58
} STAFF_REC;                        // 64
```

```
typedef struct
{
    long percent_dis;               // 0
    long tick_per;                 // 4
    struct                          // 8
    {
        unsigned char valid        :1;
        unsigned char nu          :7;    // not used
    } flags;
    unsigned char filler[23];       // 9
} STAFF_GROUP_REC;                 // 32
```

DEPART.QDX (40)

File Type	Index
Record Size	1024
Key Length	2

Record Format

```
// department periods
#define DEP_PER_P_DAY      0
#define DEP_PER_WEEK      1
#define DEP_PER_P_WEEK    2
#define DEP_PER_MONTH     3
#define DEP_PER_P_MONTH   4
#define DEP_PER_YEAR      5
#define DEP_PER_Z         6

typedef struct dep_rec_           //OFS  Type  Description
{
    char key[2];                 //0    BCD    Department number
    char dep_name[16];           //2    ASCII Department description
    struct                       //18   BITS
    {
        unsigned char rate1:1;   //          Tax rate 1
        unsigned char rate2:1;   //          Tax rate 2
        unsigned char rate3:1;   //          Tax rate 3
        unsigned char rate4:1;   //          Tax rate 4
        unsigned char rate5:1;   //          Tax rate 5
        unsigned char rate6:1;   //          Tax rate 6
        unsigned char rate7:1;   //          Tax rate 7
        unsigned char rate8:1;   //          Tax rate 8
    } tax;
}
```



```
char filler;                                //19  NU    Not used

struct                                     //20  BITS
{
    unsigned char credit      :1;  //      Payment with food stamps
    unsigned char non_merch   :1;  //      Non merchandise
    unsigned char negative    :1;  //      Negative entry
    unsigned char decimal_point :1; //      Decimal quantity
    unsigned char weight      :1;  //      Allow weighing
    unsigned char discountable :1;  //      Allow discount
    unsigned char wic_item     :1;  //      Payment by WIC
    unsigned char valid       :1;  //      Department in use
} flags1;

struct                                     //21  BITS
{
    unsigned char cost_plus      :1; //      Cost plus department
    unsigned char report_ho      :1; //      Report PLU changes to head office
    unsigned char no_staff_disc   :1; //      Don't allow staff discounts
    unsigned char no_club_card_points :1; //      No clubcard points
    unsigned char key_entry_disallowed :1; //      No manual amount entry
    unsigned char counter_dept    :1; //      Counter department
    unsigned char non_scanned_dept :1; //      Non scanned department
    unsigned char trading_eligible :1; //      Eligible for trading stamps
} flags2;

struct                                     //22  BITS
{
    unsigned char rcpt_ttl       :1; //      Print total on receipt
    unsigned char store_coupon   :1; //      Store coupon dept.
    unsigned char dep_vendor_cpn_flag :1; //      V8: Department vendor coupon
    unsigned char dep_bonus_cpn_flag :1; //      V8: Department bonus coupon
    unsigned char valid_pfs      :1; //      TESCO_EE: PFS Department in use
    unsigned char pop_msg_plu    :1; //      Display POS message for linked items
```

```

unsigned char nu          :1; //          Not used (reserved for V8)
unsigned char opt_WIC_CVV :1; //          Payment by WIC CVV

} flags3;

long dep_max_value;          //23  NUM_D  Maximum value
long dep_min_value;          //27  NUM_D  Minimum value
unsigned char dep_max_check_no; //31  NUM    Control check for maximum value
unsigned char dep_min_check_no; //32  NUM    Control check for minimum value
char filler0[4]      ;      //33  NU     Not used
short group_no;        //37  NUM    Group number
unsigned char fuel_group; //39  NUM    Fuel group department
unsigned char disc_num;  //40  NUM    Discount type
unsigned short re_dep;   //41  NUM    RE department
unsigned short extended_promo_no; //43  NUM    Extended promotion number
unsigned char extended_bucket_no; //45  NUM    Extended bucket number
unsigned char inter_num[7]; //46  BCD    Internal number
unsigned char restrict_layout; //53  NUM    Time restriction layout

// V7.3.4 - Namibia
long coust_disc_type_1;      //54  NUM    customer discount type 1
long coust_disc_type_2;      //58  NUM    customer discount type 2
long coust_disc_type_3;      //62  NUM    customer discount type 3
long coust_disc_type_4;      //66  NUM    customer discount type 4
long coust_disc_type_5;      //70  NUM    customer discount type 5
char          info_nu[249];   //74  NU     Not used

// *** END OF INFORMATIVE PART ***

// *** START OF SALES PART ***

unsigned char sales_nu[96];    //323  NU     Not used
short date_last_sales;        //419  DATE_I  Date of last sale

```

```
// The following structure repeats itself for the following periods:
```

```
// Previous day, week, last week, month, last month, year, since last Z
```

```
// 86 bytes * 7 periods = 602
```

```
struct dep_period_data_
```

```
{
    unsigned char  net_sales6[6] ;           //  BIN    Net sales
    long           net_count      ;           //  NUM    Net sales count
    long           void_sales     ;           //  NUM_D  Canceled sales
    long           void_count     ;           //  NUM    Canceled items count
    long           return_sales   ;           //  NUM_D  Returned items value
    long           return_count   ;           //  NUM    Returned items count
    long           s_coupon_sales ;           //  NUM_D  Store coupon sales
    long           s_coupon_count ;           //  NUM    Store coupon count
    long           v_coupon_sales ;           //  NUM_D  Vendor coupon sales
    long           v_coupon_count ;           //  NUM    Vendor coupon count
    long           b_coupon_sales ;           //  NUM_D  Bonus coupon sales
    long           b_coupon_count ;           //  NUM    Bonus coupon count
    long           discount_sales ;           //  NUM_D  Discounts given value
    long           discount_count ;           //  NUM    Discounts given counter
    long           item_scanned   ;           //  NUM    Items scanned
    long           item_keyed     ;           //  NUM    Items keyed
    long           item_dept_count;           //  NUM    Items sold through open
    long           non_net_disc   ;           //  NUM_D  Non-netted discounts
    long           customers      ;           //  NUM    Number of customers
    long           non_net_fs_disc;           //  NUM    Mon-netted Freq. Shop. Disc.
    unsigned char  nu[4]         ;           //  NU     Not used
} period[7];
char qdx_flag;                               //  BITS   QDEX reserved flags
} DEP_REC;
```

```
#define DEPART_REC DEP_REC
```

```
#define DEP_KEY_LEN (lenof(DEP_REC, key))
```

PROMOTION.QDX (41)

File Type	Index
Record Size	128
Key Length	11

Record Format

typedef struct prom_rec_	//OFS	Type	Description
{			
unsigned char plu[7];	//0	BCD	PLU number
unsigned char invoke_date[3];	//7	BCD_D	Start date YYMMDD
unsigned char zero;	//10	NU	Not used
unsigned char revoke_date[3];	//11	BCD_D	End date DDMMYY
unsigned long promotion_price;	//14	NUM_D	Promotion/discount price/percent
unsigned char promo_name[16];	//18	ASCII	Description
unsigned char promo_id[4];	//34	ASCII	Promotion ID
unsigned char standard_price[2];	//38	BCD	Quantity at standard price
unsigned char discount_price[2];	//40	BCD	Quantity at discount price
struct promo_flag_	//42	BITS	
{			
unsigned char filler1 :1;	//		Not used
unsigned char filler2 :1;	//		Not used
unsigned char limited_qty :1;	//		Limited quantity
unsigned char linked_offer :1;	//		Linked offer
unsigned char price_percent :1;	//		Price is percentage
unsigned char promo_item :1;	//		Promotion
unsigned char offer_item :1;	//		Offer
unsigned char rdc_item :1;	//		Reduction
} promo_flag;			
unsigned char offer_no[2];	//43	HEX	Offer number (reversed int)
unsigned char action_flag;	//45	ASC_N	Value of '3' means ignore price compare

```
unsigned char  active;                //46  NUM    Active promotion
unsigned char  revoke_flag;           //47  NU     Not used
unsigned char  seq_pointer;           //48  NU     Not used
unsigned char  min_purchase_flag;     //49  NUM    Minimum purchase required
long          min_purchase_amount;    //50  NUM_D  Minimum purchase amount
unsigned char  min_purchase_ctrl  ;    //54  NUM    CC for minimum purchase

// Used in V8 and WinPos only
unsigned char  limited_qty_order;     //55  NUM    Limited quantity
unsigned char  start_time[2];         //56  BCD_T  Promotion start time
unsigned char  end_time[2];           //58  BCD_T  Promotion end time
unsigned char  opt_Sunday      :1;    //60  BIT    Active on ...
unsigned char  opt_Monday      :1;    //60  BIT    Active on ...
unsigned char  opt_Tuesday     :1;    //60  BIT    Active on ...
unsigned char  opt_Wednesday   :1;    //60  BIT    Active on ...
unsigned char  opt_Thursday    :1;    //60  BIT    Active on ...
unsigned char  opt_Friday      :1;    //60  BIT    Active on ...
unsigned char  opt_Saturday    :1;    //60  BIT    Active on ...
unsigned char  opt_07          :1;    //60  BIT    Active on ...
struct prom_day_times_
{
    //61
    unsigned char  start_time[2];      //    BCD_T  Start time on day..
    unsigned char  end_time[2];        //    BCD_T  End time on day..
}day_time[7];
unsigned char  promo_data[37];        //89  NU     Not used
unsigned char  prom_by;               //126 NUM    Promotion is by:
//                                     0 - PLU
//                                     1 - mix & match
//                                     2 - Department
unsigned char  Qdex_flag;             //127 BITS   QDEX reserved flags
} PROM_REC;
#define  PROM_KEY_LEN    (lenof(PROM_REC, plu)+lenof(PROM_REC, invoke_date)+1)
```

BROM.QDX (42)

File Type	Index
Record Size	128
Key Length	7

Record Format

typedef struct brom_rec_	//OFS	Type	Description
{			
unsigned char item_code[7];	//0	BCD	PLU number
unsigned char seq_number;	//7	NU	Not used
unsigned char prom_name[16];	//8	ASCII	Description
long prom_price ;	//24	NUM_D	Promotion price
unsigned short standard_price ;	//28	NUM	Quantity at standard price
unsigned short discount_price ;	//30	NUM	Quantity at discount price
struct	//32	BITS	
{			
char filler :2;	//		Not used
char limet_qty :1;	//		Limited quantity
char linked_offer :1;	//		Linked offer
char price_is_prcent :1;	//		Price is percent
char promotion :1;	//		Promotion
char offer :1;	//		Offer
char reduction :1;	//		Reduction
} prom_flag;			
unsigned char invoke[3];	//33	NU	Start date
	//		[0]=day, [1]=mon, [2]=year
unsigned char revoke[3];	//36	NU	End date
	//		[0]=day, [1]=mon, [2]=year
unsigned char prom_id[4];	//39	ASCII	ID number
short offer_num ;	//43	NUM	Offer number

```
unsigned char action_flag;           //45  NU    Not used
unsigned char filler   ;             //46  NU    Not used
unsigned char revoke_flag;          //47  NU    Not used
unsigned char seq_pointer;          //48  NU    Not used
unsigned char min_purchase_flag;     //49  NUM   Minimum purchase required
long          min_purchase_amount;   //50  NUM_D  Minimum purchase amount
unsigned char min_purchase_ctrl  ;    //54  NUM   CC for minimum purchase

union
{
    struct brom_sales_v7_
    {
        char          filler2[11];    //55  NU    Not used
        long          sales_value;     //66  NUM_D  Daily Sales value
        long          sales_count;     //70  NUM    Daily Sales count
        long          markdown_value;  //74  NUM_D  Daily Markdown value
        long          markdown_count;  //78  NUM    Daily Markdown count
        long          prd_sales_value;  //82  NUM_D  Period Sales value
        long          prd_sales_count;  //86  NUM    Period Sales count
        long          prd_markdown_value; //90  NUM_D  Period Markdown value
        long          prd_markdown_count; //94  NUM    Period Markdown count
        long          ttl_sales_value;  //98  NUM_D  Value of sales while on promotion (daily)
        short         ttl_sales_count;  //102 NUM    Quantity sold while on promotion (daily)
        long          prd_ttl_sales_value; //104 NUM_D  Value of sales while on promotion (period)
        short         prd_ttl_sales_count; //108 NUM    Quantity sold while on promotion (period)
        unsigned char prom_data[17];    //110 NU    Not used
    } sales;

    struct brom_data_v8_
    {
        unsigned char limited_qty_order; //55  NUM    Limited quantity

        unsigned char start_time[2];     //56  BCD_D  Promotion start time
        unsigned char end_time[2];       //58  BCD_D  Promotion end time
    }
}
```

```

unsigned char opt_Sunday      :1;  //60  BIT    Active on ...
unsigned char opt_Monday      :1;  //60  BIT    Active on ...
unsigned char opt_Tuesday     :1;  //60  BIT    Active on ...
unsigned char opt_Wednesday   :1;  //60  BIT    Active on ...
unsigned char opt_Thursday    :1;  //60  BIT    Active on ...
unsigned char opt_Friday      :1;  //60  BIT    Active on ...
unsigned char opt_Saturday    :1;  //60  BIT    Active on ...
unsigned char opt_07          :1;  //60  BIT    Active on ...
struct brom_day_times_       //61
    unsigned char start_time[2];  //    BCD_T    Start time on day..
    unsigned char end_time[2];    //    BCD_T    End time on day..
}day_time[7];
    unsigned char reserved[38];    //89  NU      Not used
} data;
} version;

char      qdx_flag;              //127  BITS    QDEX reserved flags
} BROM_REC;

```


SAFE.QDX (43)

File Type	Relative
Record Size	1001
Key Length	0

Record Format

```
typedef struct safe_rec_                                //OFS  Type    Description
{
    // The following structure repeats itself 100 times for 100 tenders

    struct safe_media_                                  //0    STRUCT
    {
        long  decl;                                     //    NUM_D    Safe declarations
        long  in_safe;                                  //    NUM_D    Expected value in safe
    } safe_data[100];

    // The following structure holds value for the following action:
    // safe_ttl[0]  - Add loan
    // safe_ttl[1]  - Pickups
    // safe_ttl[30] - Receipts
    // safe_ttl[31] - Deposits
    // safe_ttl[32] - ROA
    // safe_ttl[33] - Payouts
    // safe_ttl[34] - Media transfer

    struct safe_bank_                                  //800  STRUCT
    {
        long  sum;
    } safe_ttl[50];

    char      qdx_flag;                                // 1000
```

```
} SAFE_REC;
```

```
typedef struct safe_media_ SAFE_MEDIA ;
```

```
// Safe file - record 2
```

		//OFS	Type	Description
typedef struct safe_rec2_				
{				
long	os_day;	//0	NUM_D	Daily over/short value
long	os_pday;	//4	NUM_D	Previous day over/short value
long	os_week;	//8	NUM_D	Weekly over/short value
long	os_pweek;	//12	NUM_D	Previous week over/short value
long	os_month;	//16	NUM_D	Monthly over/short value
long	os_pmonth;	//20	NUM_D	Previous month over/short value
long	os_year;	//24	NUM_D	Yearly over/short value
long	os_last;	//28	NUM_D	Last Over/short value
char	filler[959];	//32	NU	Not used
long	sum_empl_os;	//991	NUM_D	Ttl cashiers ovr/short since last safe locking
long	sum_empl_sales;	//995	NUM_D	Total sales of cashiers on locking time
unsigned char	bank_count;	//999	NUM	Bankings counter
char	qdx_flag;	//1000		
}	SAFE_REC2;			

```
// Safe file - record 3
```

		//OFS	Type	Description
typedef struct safe_rec3_				
{				
// The following structure holds information for 25 tenders				
struct safe_med_dclr_		//0	STRUCT	
{				
long	cshr_dclr_amount;	//	NUM_D	Cashier declarations amount
long	cshr_dclr_qty ;	//	NUM	Cashier declarations quantity
long	banked_amount ;	//	NUM_D	Banked amount
long	banked_qty ;	//	NUM	Banked quantity
long	receipt_amount ;	//	NUM_D	Receipts value
long	receipt_qty ;	//	NUM	Receipts quantity

```
    long  pay_amount      ;                //    NUM_D    Payouts amount
    long  pay_qty         ;                //    NUM      Payouts quantity
} media[MAX_MEDIA];

// The following structure holds cashier declarations in current period

struct safe_med_dclr_current_           //800
{
    long  cur_cshr_dclr_amount;           //    NUM_D    Declared amount
    long  cur_cshr_dclr_qty;             //    NUM      Declared quantity
} cur_media[MAX_MEDIA];

    char  filler1;                       //    NU      Not used
} SAFE_REC_MED_DCLR;

// safe records 4,5,6,7,8
typedef struct safe_rec4_8_
{
    // The following structure holds information for 100 tenders
    // 64 * 100 = 6400 bytes
    struct safe_med_tr_                   //0    STRUCT
    {
        long  cshr_dclr_amount;           //    NUM_D    Cashier declarations amount
        long  cshr_dclr_qty  ;           //    NUM      Cashier declarations qty
        long  deposit_amount ;           //    NUM_D    Banked/deposit amount
        long  deposit_qty   ;           //    NUM      Banked/deposit quantity
        long  receipt_amount ;           //    NUM_D    Receipts value
        long  receipt_qty   ;           //    NUM      Receipts quantity
        long  pay_amount    ;           //    NUM_D    Payouts amount
        long  pay_qty       ;           //    NUM      Payouts quantity
        long  p_up_amount   ;           //    NUM_D    Picked up amount
        long  p_up_qty      ;           //    NUM      Picked up quantity
        long  roa_amount    ;           //    NUM_D    ROA amount
        long  roa_qty       ;           //    NUM      ROA quantity
    }
```

```

    long  float_amount  ;           //    NUM_D    Float amount
    long  float_qty     ;           //    NUM      Float qty
    long  safe_dclr     ;           //    NUM_D    safe declaration
    char  nu[4];         //    NU
} media[100];
} SAFE_REC_MED_TR

```

OFFERS.QDX (44)

File Type	Index
Record Size	10
Key Length	2

Record Format

	//OFS	Type	Description
typedef struct offer_rec_ { unsigned char num[2]; unsigned char plu[7]; unsigned char qdx_flag; } OFFER_REC;			
	//0	BCD	Offer number
	//2	BCD	PLU number
	//9	BITS	QDEX reserved flags

MIXMATCH.QDX (45)

File Type	Index
Record Size	9
Key Length	8

Record Format

	//OFS	Type	Description
typedef struct mm_rec_ { unsigned char code; unsigned char plu_number[7]; char filler[1]; } MM_REC;			
	//0	NUM	Mix & match code
	//1	BCD	PLU number
	//8	BITS	QDEX reserved flags

```
typedef struct  
{  
    short  nu;  
    short  qty;  
    long  decl;  
} CASH_MEDIA;
```

PARAM.QDX (46)

File Type	Relative
Record Size	1000
Key Length	0

Record Format

typedef struct syspar_rec_	//OFS	Type	Description
{			
unsigned char olasrv_pc;	//0	NUM	PC number to run OLA server
unsigned char bat_name[12];	//1	ASCII	Name of batch file to run every X minutes
unsigned char bat_min;	//13	NUM	Execute batch every X min.
unsigned char ccms_pc;	//14	NUM	CCMS SRV pc
unsigned short coff_end;	//15	DATE_I	Date of last safe locking
unsigned char subdep_eod_report1;	//17	NUM	# of user-define subdept rpt to be created @EOD
unsigned char subdep_eod_report2;	//18	NUM	# of user-define subdept rpt to be created @EOD
unsigned char subdep_eod_report3;	//19	NUM	# of user-define subdept rpt to be created @EOD
unsigned char subdep_eod_report4;	//20	NUM	# of user-define subdept rpt to be created @EOD
unsigned char subdep_eod_report5;	//21	NUM	# of user-define subdept rpt to be created @EOD
unsigned char ccms_pc2;	//22	NUM	2nd CCMS PC
unsigned short store_no;	//23	NUM	Store number
unsigned char item_check_digit;	//25	NUM	Item must have valid check digit
unsigned char pccs_pc;	//26	NUM	PC number to use for PCCS
struct	//27	BITS	General flags
{			
unsigned char			
do_not_allow_flag_weight:1;	//		Do not allow flagging on or
	//		off the weighted item flag
	//		in the case of having sales in
	//		the current period.
unsigned char eod_edit :1;	//		Allow parameters editing during EOD.
unsigned char auto_pickup :1;	//		Automatic pickup of O/S during EOD
}			

```

    unsigned char filler      :5;
}flags;
unsigned char date;          //28  ASCII  Date format
                             //      'A' - American
                             //      'E' - European
unsigned char dec_point;     //29  NUM    Decimal point
unsigned char ccms_type;     //30  NUM    0 - None
                             //      1 - Corema
                             //      2 - Fry's
unsigned char stamp_00_01;   //31  NUM    Stamp for write validation
unsigned short ej_days;      //32  NUM    Days to keep EJ
unsigned char sys_lang;      //34  NUM    0 - English
                             //      1 - Arabic
unsigned char prom_ext;      //35  NUM    Promotion extract
                             //      0 - None
                             //      1 - On
unsigned char pbrcopy_active; //36  NUM    File is currently being copied by PBRCOPY
unsigned char scr_exit_prog; //37      Screen saver exit from program:
                             //      0 - None
                             //      1 - to previous menu
                             //      2 - to Sign On screen
unsigned char user_ap1[30];  //38  UNDEF  Reserved for user appl.
unsigned char user_ap2[30];  //68  UNDEF  Reserved for user appl.
unsigned char mix_match_update; //98  NUM    Update all linked items if m&m code changed
unsigned char batch_fields;  //99  NUM    PLU batch type
                             //      78 - Basic
                             //      89 - Enhanced
                             //      100 - Extended
                             //      101 - Super (Size 400)
unsigned char kilograms;    //100  NUM    Pounds/Kilograms weight
unsigned char PCCS_Z_on_boot; //101  UNDEF  Reserved for PCCS use
unsigned char PCCS_enable;   //102  UNDEF  Reserved for PCCS use
unsigned char nu05;          //103  NU     Not used
unsigned char origin_batch;  //104  NUM    Use original batch number for error batch file

```


[illegible]

```

    unsigned char  fs_disc_pric:1;          //          Allow Frequent Shopper discount > price
    unsigned char  enh_po_roa  :1;          //          Enhanced Paid-out/ROA
} flags1;

struct                                //124  BITS  General flags
{
    unsigned char  dclr_bank   :1;          //          Declare banking
    unsigned char  safe_lock_p :1;          //          Prohibit safe lock if unlocked cashiers found
    unsigned char  safe_reps   :1;          //          Specify reports for safe  locking
    unsigned char  remove_group_in_eod :1; //          delete PLU Group in EOD
    unsigned char  declare_while_signon:1; //          Allow cashier/POST declarations while cashier
                                                //          signed on
    unsigned char  total_by_group :1;        //          Use groups numbers to total checks
    unsigned char  auto_dclr_fix  :1;        //          Allow manual fixing of automatic declarations
    unsigned char  auto_dclr_eod  :1;        //          Update cashier/pos decl. during EOD
} flags2;

unsigned char subdep_in_system;           //125  NUM  Subdepartment in system
unsigned char min_wait_pos_trans;         //126  NUM  Minutes to wait for POS to
                                                //          send transactions before EOD
unsigned short int cheque_group_1;        //127  NUM  Check totals group 1
unsigned short int cheque_group_2;        //129  NUM  Check totals group 2
unsigned short corema_port;               //131  NUM  Corema port number
unsigned char stamp_00_02;                //133  NUM  Stamp for write validation
unsigned char local_ccms_type;             //134  NUM  Local CCMS type
                                                //          0-None
                                                //          1-Reward check
unsigned char new_nu;                     //135  NU   Not Used
unsigned char price;                      //136  NU   Not Used
unsigned char update;                     //137  NU   Not Used
unsigned char chk_check3;                 //138  NU   Not Used
unsigned char sort_promo_add_loss;         //139  NUM  Sort promotion add/loss by department
unsigned char sort_sale_add_loss;         //140  NUM  Sort promotion add/loss by department
unsigned char corema_host[16];            //141  ASCII Corema host alias

```

```
unsigned char sys_ver[15];           //157  ASCII   System version
unsigned char chk_check4[1];         //172  NU      Not Used
unsigned char chk_line[2];           //173  NU      Not Used
unsigned char inq;                   //175  NU      Not Used
unsigned char ap1_file[20];          //176  NU      Not Used
unsigned char ap2_file[20];          //196  NU      Not Used
unsigned char ecr;                   //216  NU      Not Used
unsigned char sts;                   //217  NU      Not Used
unsigned char bo_ent;                //218  NU      Not Used
unsigned char maint;                 //219  NU      Not Used
unsigned char sys_shut;              //220  NU      Not Used
unsigned char update_safe;           //221  NUM     Update safe immediately
//                                  after B/O function at POS
unsigned char po_roa;                //222  NUM     Print payout/roa reports during EOD
unsigned char mgr_str;               //223  NU      Not Used
unsigned char fun_f8;                //224  NU      Not Used
unsigned char fun_f9;                //225  NU      Not Used
unsigned char exit_DOS;              //226  NU      Not Used
unsigned char pcm_EOD;               //227  NU      Not Used
unsigned char disc_sales;            //228  NUM     Set "discount prohibit" flag if item on sale
unsigned char disc_promotion;        //229  NUM     Set "discount prohibit" flag if item on promo
unsigned char disc_reduction ;       //230  NUM     Set "discount prohibit" flag
//                                  if item on reduction
unsigned char disc_offer;            //231  NUM     Set "discount prohibit" flag if item on offer
unsigned char save_plu;              //232  NUM     Save PLU batch
unsigned char save_promo;            //233  NUM     Save promotion flag
struct                               //234  BITS
{
    unsigned char opt_mm_in_sys      :1;// Money market in system
    unsigned char foreign_trans_rep_generate :1;// Foreign transaction report generate
    unsigned char foreign_trans_rep_print :1;// Foreign transaction report print
    unsigned char send_foreign_trans_alert :1;// Send foreign transaction alert
    unsigned char opt_mm_nu          :4;// Money market reserved flags
}foreign_trans;
```

```

unsigned char bo_sec1[19];           //235  NU      Not used
struct                               //254  BITS
{
    unsigned char safe_tot_af_dclr    :1;//      Update safe totals immediately
                                   //      after cashier/pos declaration
    unsigned char un                  :7;//      Not used
}safe_flag;
struct cash_lock_                     //255  BITS
{
    unsigned char lock_all_cashiers    :1;//      Perform cashier locking on all cashiers
    unsigned char print_rpt_all_cashr  :1;//      Print reports for all cashiers,
                                   //      active and non-active
    unsigned char specify_report       :1;//      Specify rpt to print for each cashier at lock
    unsigned char cashier_and_declr    :1;//      Cashier Report, Cashier Declaration Report
    unsigned char operator_trial       :1;//      Operator Trial Balance Report
    unsigned char summary_trial        :1;//      Summary Trial Balance Report
    unsigned char restrict_lock_to_file:1;//      Restrict cashier/post locking to file.
    unsigned char un                  :1;//      Not used
} cash_lock;
unsigned char eft_driver[60];         //256  ASCII   EFT device driver command
struct eod_report_to_print_           //316
{
    struct                             //316  BITS
    {
        unsigned char control          :1;// Control report
        unsigned char post             :1;// Post Report (Grand total) report
        unsigned char media            :1;// Media report
        unsigned char hourly_productivity :1;// Hourly Productivity report
        unsigned char fifteen_minutes_productivity :1;// Fifteen Minutes Productivity report
        unsigned char department       :1;// Department report
        unsigned char expanded_department :1;// Expanded Department report
        unsigned char weekly_expanded_department :1;// Weekly Expanded Department report
    }eod_report1;

```

```
struct //317 BITS
{
    unsigned char monthly_expanded_department :1;// Monthly Expanded Department report
    unsigned char yearly_expanded_department :1;// Yearly Expanded Department report
    unsigned char department_by_cashier :1;// Department by Cashier report
    unsigned char department_by_terminal :1;// Department by Terminal report
    unsigned char department_by_hour :1;// Department by Hour report
    unsigned char department_by_customer :1;// Department by Customer report
    unsigned char department_information :1;// Department Information report
    unsigned char traders_sales :1;// Traders Sales report
}eod_report2;

struct //318 BITS
{
    unsigned char department_exc_inc :1;// Department Exclusive/Inclusive
    unsigned char group_sales :1;// Group Sales report
    unsigned char promotion_add_loss :1;// Promotion Add/Loss report
    unsigned char sale_items_add_loss :1;// Sale Items Add/Loss report
    unsigned char sort_promotion_add_loss :1;// Sort Promotion Add/Loss report by Dept
    unsigned char sort_sale_item_add_loss :1;// Sort Sale Item Add/Loss report by Dept
    unsigned char payout_roa :1;// Payout/ROA reports
    unsigned char cashier_performance :1;// Cashier Performance report
}eod_report3;

struct //319 BITS
{
    unsigned char Cashier_Productivity_by_Hour :1;// Cashier Productivity by Hour report
    unsigned char Training :1;// Training report
    unsigned char Daily_Operator_Performance :1;// Daily Operator Performance report
    unsigned char Weekly_Operator_Performance :1;// Weekly Operator Performance report
    unsigned char Monthly_Operator_Performance :1;// Monthly Operator Performance report
    unsigned char Branch_Performance :1;// Branch Performance report
    unsigned char Daily_Refunds :1;// Daily Refunds report
    unsigned char Period_Refunds :1;// Period Refunds report
}eod_report4;

struct //320 BITS
```

```

{
    unsigned char department_flash_sales          :1;// Department Flash Sales & Sales Mix
    unsigned char net_department_sales_flash      :1;// Net Department Sales Flash
    unsigned char statement_of_trade_receivables  :1;// Statement Of Trade Receivables
    unsigned char weekly_business                 :1;// Weekly Business
    unsigned char cashier_history                  :1;// Cashier History
    unsigned char pos_sales                        :1;// POS Sales
    unsigned char saved_transactions               :1;// Saved Transactions
    unsigned char location                         :1;// Location
}eod_report5;

struct                                     //321  BITS
{
    unsigned char item_member_promotions_sales    :1;// Item Member Promotions Sales report
    unsigned char queue_length_weekly_summary    :1;// Queue Length Weekly Summary
    unsigned char itemized_check_report_by_cashier :1;// Itemized Check report by Cashier
    unsigned char itemized_check_report_by_post   :1;// Itemized Check report by POST
    unsigned char vendor_coupon_totals            :1;// Vendor Coupon totals report
    unsigned char eft_authorized_transactions     :1;// EFT Authorized Transactions report.
    unsigned char branch_expenses_summary_during_eod :1;// Branch Expenses Summary rpt during EOD
    unsigned char branch_expenses_summary_during_eow :1;// Branch Expenses Summary rpt during EOW
}eod_report6;

struct                                     //322  BITS
{
    unsigned char detailed_daily_expenses_during_eod :1;// Detailed Daily Expenses rpt during EOD
    unsigned char detailed_weekly_expenses_during_eod :1;// Detailed Weekly Expenses rpt during EOD
    unsigned char daily_markdown                    :1;// Daily Markdown report
    unsigned char weekly_markdown                    :1;// Weekly Markdown report
    unsigned char operator_summary_trial_balance     :1;// Operator Summary Trial Balance report
    unsigned char operator_trial_balance             :1;// Operator Trial Balance report
    unsigned char voided_transactions               :1;// Voided Transactions report
    unsigned char credit_card_transactions_schedule  :1;// Credit Card Transactions Schedule
}eod_report7;

struct                                     //323  BITS
{

```

```
    unsigned char EFTPOS_Daily_Reconciliation      :1;// EFTPOS Daily Reconciliation report
    unsigned char Subtract_and_Cancel              :1;// Subtract and Cancel Report
    unsigned char Accountability_for_current_period :1;// Cashier Accountability Rpt for curr per
    unsigned char Accountability_for_previous_period :1;// Cashier Accountability Rpt for prev per
    unsigned char Ranking_report                   :1;// Ranking Reports
    unsigned char Subdepartment_Report              :1;// Subdepartment Report
    unsigned char trading_stamps_distribution       :1;//Trading stamps distribution report
    unsigned char media_in_posts                    :1;//Media in POSTs signed-on at close report
}eod_report8;

struct                                     //324 BITS
{
    unsigned char Audit_report                  :1;//Audit report - PLU changes
    unsigned char Posts_eod_error               :1;//POSTs error in EOD rep
    unsigned char buyaid_day                    :1;//Buy Aids daily report
    unsigned char buyaid_week                   :1;//Buy Aids weekly report
    unsigned char Member_Prom_Daily              :1;//Member Promotion Daily sales
    unsigned char basket                        :1;// Basket analysis
    unsigned char dwb                           :1;// Department weekly budget
    unsigned char dsa                           :1;// Department sales analysis
}eod_report9;

struct                                     //325 BITS
{
    unsigned char comp_ean                      :1;// Compare EANs
    unsigned char daily_lift_float              :1;// Daily Cashier/POS Transactions
    unsigned char weekly_lift_float             :1;// Weekly Cashier/POS Transactions
    unsigned char eft_daily_trans               :1;// EFT daily transactions
    unsigned char
        prepaid_card_report                    :1;// Prepaid card report
    unsigned char
        reward_check_report                    :1;// Daily reward check report
    unsigned char plu_not_found                 :1;// PLU not found report
    unsigned char ss_24h                       :1;// print SelfScan 24h/hourly report
}eod_report10;
```

```
}eod_report_to_print;
```

```
struct                                //326  BITS    General flags
{
    unsigned char  chk_valid_subdep   :1;      //Check valid subdepartment on plu maintenance
    unsigned char  seperatly_media   :1;      //Print media deposit separately
    unsigned char  display_desc       :1;      //Display PLU batch's description
    unsigned char  show_batch_desc    :1;      //Show PLU batch description in batch list
    unsigned char  suppress_ssl_prompt:1;      //Suppress store/safe locking prompt
    unsigned char  branch_per_tax_ex  :1;      //Branch performance report is tax excl.
    unsigned char  no_eod_if_save_tran:1;      //Don't allow EOD if non-recalled saved
                                                // transactions left
    unsigned char  assistant_in_sys   :1;      //Assistant in system

}flags3;

unsigned char stamp_00_03;             //327  NUM     Stamp for write validation

//                                     EFT PARAMETERS

unsigned char  ip_address_connect[20]; //328  ASCII   IP address to connect
unsigned short port_num_to_use;        //348  NUM     Port number to use
unsigned long  timeout_ms;             //350  NUM     Timeout ms

unsigned char mm_trs_file[30];         //354  ASCII   Money market TRS file
unsigned short free_space;             //384  NUM     KB to initiate "out of disk space" warning
unsigned char garbage_tender;         //386  NUM     Tender # (0-25) to accum. tenders above 25

struct                                //387  BITS
{
    unsigned char  pass_or_cash       :1;//      0-Use security maintenance as users list
                                                //      1-Use cashier maintenance as users list
    unsigned char  pass_change_0      :1;//      Prompt for password change if password 0
    unsigned char  alerts_in_sys      :1;//      Alerts in system
    unsigned char  sunday_trading_store:1;//      Sunday trading store
}
```



```
unsigned char  exchnge_while_signon:1;//      Allow Tender Exch. while CSH/POST signed on
unsigned char  enhance_dep_sale_rep:1;//      Prt Enhanced Exclusive/Inclusive Dep Sales Rpt
unsigned char  enh_cshr_loan_pickup:1;//      Show media in POS on cashier Pickup/Loan trans
unsigned char  default_next_working_day :1;//  Default next working day
} op;

unsigned char time_pass_change;                //388  NUM      Days delay before prompting for password chg
struct                                              //389  BITS
{
    unsigned char  delayed                :1;  //      Allow delayed POS balancing
    unsigned char  pos_period_eow         :1;  //      Transfer POS period on EOW
    unsigned char  close_safe_locking     :1;  //      Default "Yes" or "No" in Safe/Store
                                              //      Locking ("Close office period")
    unsigned char  del_member_prom_at_end :1;  //      Delete non-active member promotion at EOD
    unsigned char  del_promotion_at_end   :1;  //      Delete non-active promotion at EOD
    unsigned char  eftt7_void_rep_generate:1;  //      EFT T7 Voided Tender Report
    unsigned char  eftt7_void_rep_print   :1;  //      EFT T7 Voided Tender Report Print
    unsigned char  warn_chk_prev_period   :1;  //      Check "Previous period" selection warning
} pacct_opt;

unsigned char days_skip_pos_period;            //390  NUM      Days to skip POS acct. transfer
unsigned short mm_dflt_cshr;                   //391  NUM      Money market default cashier
unsigned short mm_dflt_pos;                    //393  NUM      Money market default pos
unsigned char  mm_poll_delay;                  //395  NUM      How often to look for new TRS
unsigned char  bo_sec3[10];                   //396  NU       Not used
unsigned char  pc_list[10];                   //406  NUM      PC's in system
unsigned char  save_plu_number[7];            //416  NU       Not used
unsigned char  store_name[28];                //423  ASCII    Store name
unsigned char  minute_boot;                   //451  NU       Not used
unsigned char  cshr_rpt_in_use;               //452  NU       Not used
unsigned char  safe_rpt_in_use;               //453  NU       Not used
unsigned char  factura_in_sys;                //454  NUM      Factura in system
unsigned char  multiply_by_10;                //455  NU       Not used
```

```

unsigned char promo_bat_file[25];          //456  ASCII  Promotion batch files name
unsigned char inter_num_in_sys;            //481  NUM    Internal number in system
unsigned char weely_rpt_tax_ex;            //482  NUM    Weekly business report is tax exclusive
unsigned char copyright [33];              //483  ASCII  Copyright
unsigned char staff_disc_eom ;             //516  NUM    Day in month to clear staff discount
unsigned char ean_check_digit;             //517  NUM    PLU must have a valid CD
unsigned char inter_check_digit;           //518  NUM    Internal number must have a valid CD
unsigned char inter_update_group;          //519  NUM    Update all items linked by internal number
long          os_percent_lock;             //520  NUM_D   Maximum over/short percent to allow locking
unsigned char cross_in_system;             //524  NUM    Cross file in system
struct                                                //525  BITS
{
    unsigned char  test_corrupt_trans:1; //          Test transaction corruption
                                   //          (0xc0)
    unsigned char  boot_corrupt_trans:1; //          Boot PC if corrupted
    unsigned char  ascii_tran_file :1; //          Create ASCII transaction
                                   //          file during EOD
    unsigned char  weekly_bud_tax_ex :1; //          Weekly budget is tax excl.
    unsigned char  clear_save_recall :1; //          clear save/recall file at EOD
    unsigned char  multiplying_qty :1; //          multiplying the unit sales by 1000 when
                                   //          updating sub-department
    unsigned char  enhance_media_rep :1; //          Enhance Media Report
    unsigned char  subdep_ascii :1; //          Create subdepartment ASCII file at EOD
} opt2;

unsigned char VAT_number[16];              //526  ASCII  VAT number
unsigned long debt_arrears_prct;           //542  NUM_D   Debtors arrears percent
unsigned long debt_arrears_amount;         //546  NUM_D   Debtors arrears amount

// Integrity
unsigned char prompt_frequency_tickets;    //550  NUM    PLU FAST bkgd load prompt frequency (tickets)
unsigned short prompt_frequency_seconds;   //551  NUM    PLU FAST bkgd load prompt frequency (seconds)
unsigned char load_threshold_percentage;    //553  NUM    FAST/SLOW foreground PLU load threshold %
unsigned short switch_delay_seconds;       //554  NUM    PLU foreground/background switch delay

```

[illegible]

```

unsigned char  transfer_inactiv_pos :1; //      Transfer Balances for Inactive POS//TBA bug 391
unsigned char  nu                    :7; //      Not Used

} pacct_opt2;

unsigned char nu2[1];                //564  NU      Not used
unsigned char prevent_cashier_decl;  //565
unsigned char batch_2;               //566  NUM      Use opcode 2 as insert/update
unsigned char mix_match_group[3];    //567  ASC_N    Maximum m&m group number
unsigned char mm_eod_flag[30];       //570  ASCII    Money market EOD flag
unsigned char mm_eod_wait;           //600  NUM      Minutes to wait for money market EOD flag
unsigned char crd_com_file[23];      //601  NU      Not used
unsigned char crd_com_len[3];        //624  NU      Not used
unsigned char weeks_local_plu_del;   //627  NUM      Weeks before local deletion
unsigned char weeks_host_plu_del ;   //628  NUM      Weeks before host deletion
unsigned char inter_cdv_ptr;         //629  NUM      Internal number CDV scheme
struct                                               //630  BITS
{
    unsigned char  eod_workstation :1; //      Allow EOD from LFS
    unsigned char  cshr_period_eow :1; //      transfer cashier periods on EOW only
    unsigned char  plu_price_0      :1; //      Allow zero price PLU
    unsigned char  skip_store_cd    :1; //      Skip store check digit test
    unsigned char  disalow_mult_son :1; //      prohib user to signon at more than one server
    unsigned char  plu_extention    :1; //      Maintain plu extension file
    unsigned char  build_pos_acct   :1; //      build pos accountability file
    unsigned char  tran_qdx_2_ascii :1; //      translate qdx file to Ascii
}opt1;

unsigned char office_cash_copies;    //631  NUM      Copies of "Office Cash Balance" at store lock
unsigned char dsa_copies;            //632  NUM      Copies of DSA at store locking
unsigned char week_receipts_copies;  //633  NUM      Copies of Weekly Receipts rpt at store locking
unsigned char week_payout_copies;    //634  NUM      Copies of Weekly payouts rpt at store locking
unsigned char week_expense_copies;   //635  NUM      Copies of Weekly Expenses rpt at store locking
unsigned char device_pos[15];        //636  NUM      External device

```

unsigned char safe_copies;	//651	NUM	Copies of 'Safe Report' at store locking
unsigned char cshr_acct_copies;	//652	NUM	Copies of 'Cashier acct' at store locking
unsigned char bcr_copies;	//653	NUM	Copies of BCR at store locking
unsigned char rec5_fil[1];	//654	NU	Not used
unsigned char bat_com1[12];	//655	NU	Not used
unsigned char bnu [3];	//667	NU	Not used
unsigned char out_d[12];	//670	NU	Not used
unsigned char out_len[3];	//682	ASC_N	PLU report file record size
unsigned char exp_d[11];	//685	NU	Not used
unsigned char stamp_00_04;	//696	NUM	Stamp for write validation
unsigned char bat_echo;	//697	NUM	Echo PLU batch data
unsigned char batch_1;	//698	NUM	Use PLU batch opcode 1 as insert/update
unsigned char bat_file[25];	//699	NUM	Use PLU batch opcode 2 as insert/update
unsigned char sec_file[25];	//724	NU	Not used
unsigned char bat_key_size[2];	//749	NU	Not used
unsigned char eod_hour[2];	//751	ASC_N	Hour to kick auto EOD
unsigned char eod_minute[2];	//753	ASC_N	Minute to kick auto EOD
unsigned char bat_com2[13];	//755	NU	Not used
unsigned char aud_com;	//768	NUM	PLU maintenance audit
unsigned char aud_d[12];	//769	NU	Not used
unsigned char aud_len[3];	//781	ASC_N	Audit file record length
unsigned char pro_d[12];	//784	NU	Not used
unsigned char pro_len[3];	//796	ASC_N	Promotion batch file record length
unsigned char non_d[24];	//799	NU	Not used
unsigned char aud_mrg;	//823	NU	Not used
unsigned char aud_file[25];	//824	ASCII	Audit file name
unsigned char pnu [3];	//849	NU	Not used
unsigned char crd_out_file[25];	//852	ASCII	ASCII credit file name
unsigned char department_rp;	//877	NUM	Department report in EOD
unsigned char cntrl_rp;	//878	NUM	Control report in EOD
unsigned char media_rp;	//879	NUM	Media report in EOD
unsigned char hourly_rp;	//880	NUM	Hourly prod. report in EOD
unsigned char edr_rp;	//881	NUM	EDR report in EOD
unsigned char edr_weekly_rp;	//882	NUM	Weekly EDR report in EOD

```

unsigned char edr_monthly_rp;           //883  NUM    Monthly EDR report in EOD
unsigned char edr_yearly_rp;           //884  NUM    Yearly EDR report in EOD
unsigned char post_rp;                 //885  NUM    POST report in EOD
unsigned char prom_add_loss_rp;        //886  NUM    Promotion add/loss report in EOD
unsigned char dep_cshr_rp;             //887  NUM    Dept by cashier report in EOD
unsigned char dep_terminal_rp;         //888  NUM    Dept by terminal report in EOD
unsigned char dep_hour_rp;             //889  NUM    Dept HP report in EOD
unsigned char dep_customer_rp;         //890  NUM    Dept by customer report in EOD
unsigned char check_rp;                //891  NUM    Itemized check report in EOD
unsigned char dep_info_rp;             //892  NUM    Department information report in EOD
unsigned char cashier_info_rp;         //893  NUM    Cashier information report in EOD
unsigned char delayed_balance;         //894  NUM    Delayed cashier balancing
unsigned char sale_add_loss_rp;        //895  NUM    Sale add/loss report in EOD
unsigned char otb_rep;                 //896  NUM    Operator trial balance report
unsigned char max_tills_upgrade;       //897  NUM    Max No. of tills to upgrade simultaneously
struct                                //898  BITS
{

    unsigned char  rsession_active  :1;  //          Rsession active  yes/no
    unsigned char  rsession_nu      :7;

}rsession_flag;

unsigned short rsession_comm_time_out; //899  NUM    Rsession communication time out
unsigned char opt_non_sales_mode;      //901
//RPO
unsigned char action_key_func;         //902  NUM    Action for kbd func:
                                     //          0 - None
                                     //          1 - Info Only
                                     //          2 - Allow Approval
unsigned char show_last_alert_time;    //903  NUM    0-24, default 2 (hours)
struct                                //904
{
    unsigned char rma_cashier_notify :1;  //904    BIT RMA cashier notification 1=Y

```

```
    unsigned char rpo_nu          :7;    //904    not used
} rpo;

unsigned char cashier_notify_icon_timeout;//905 NUM    1-15 minutes

//CDS blackout times
unsigned char cds_blkout_mon_thur_start[2];//906    BCD TIME CDS blackout time mon-thursday start
unsigned char cds_blkout_mon_thur_end[2]; //908    BCD TIME CDS blackout time mon-thursday end
unsigned char cds_blkout_fri_start[2];    //910    BCD TIME CDS blackout time friday start
unsigned char cds_blkout_fri_end[2];      //912    BCD TIME CDS blackout time friday end
unsigned char cds_blkout_sat_start[2];    //914    BCD TIME CDS blackout time saturday start
unsigned char cds_blkout_sat_end[2];      //916    BCD TIME CDS blackout time saturday end
unsigned char cds_blkout_sun_start[2];    //918    BCD TIME CDS blackout time sunday start
unsigned char cds_blkout_sun_end[2];      //920    BCD TIME CDS blackout time sunday end

struct
{
    unsigned char cds_blackout_valid_sun :1; //922 BIT    CDS blackout time valid on sunday
    unsigned char cds_blackout_valid_mon :1; //922 BIT    CDS blackout time valid on monday
    unsigned char cds_blackout_valid_tue :1; //922 BIT    CDS blackout time valid on tuesday
    unsigned char cds_blackout_valid_wed :1; //922 BIT    CDS blackout time valid on wed
    unsigned char cds_blackout_valid_thu :1; //922 BIT    CDS blackout time valid on thursday
    unsigned char cds_blackout_valid_fri :1; //922 BIT    CDS blackout time valid on fri
    unsigned char cds_blackout_valid_sat :1; //922 BIT    CDS blackout time valid on sat
    unsigned char nu_1                   :1; //922 NU     Not used

} cds_blackout_valid_days;

//
unsigned char poll_2[37];          //923    NU     Not used
unsigned char stamp_00_05;         //960    NUM     Stamp for write validation
unsigned char rec8_fil[63];        //961    NU     Not used
unsigned char out_file[30];        //1024 ASCII    PLU report file name
unsigned char pro_file[30];        //1054 ASCII    Promotion report file name
unsigned char dep_prd_rep;         //1084 NUM     Dept period report in EOD
```

```

unsigned char dep_vat_rep;           //1085 NUM    Dept VAT report in EOD
unsigned char dep_grp_rep;           //1086 NUM    Dept groups report in EOD
unsigned char cs_hp_rep;              //1087 NUM    Cashier HP report in EOD

struct                               //1088 BITS
{
    unsigned char  eod_exp_sum      :1;    // Summary expenses report in EOD
    unsigned char  eow_exp_sum      :1;    // Summary expenses report in EOW
    unsigned char  eod_detail_exp   :1;    // Detailed expenses report in EOD
    unsigned char  eow_detail_exp   :1;    // Detailed expenses report in EOW
    unsigned char  train_peform     :1;    // Training performance report in EOD
    unsigned char  oper_perform     :1;    // Operator performance report in EOD
    unsigned char  save_recall      :1;    // Save/recalled report in EOD
    unsigned char  vdr_cpn          :1;    // Vendor coupons report in EOD
} eod_reps1;

struct                               //1089 BITS
{
    unsigned char  dsa              :1;    // DSA report in EOD
    unsigned char  flash_mix        :1;    // Flash sales & sales mix report in EOD
    unsigned char  refund           :1;    // Refunds report in EOD
    unsigned char  buy_aids         :1;    // Buy-aids report in EOD
    unsigned char  compare_ean      :1;    // Compare Multiple EAN's report in EOD
    unsigned char  cshr_hist        :1;    // Cashier history report in EOD
    unsigned char  dept_sales_flash :1;    // Dept sales flash report in EOD
    unsigned char  month_basket     :1;    // Monthly Basket Analysis report in EOD
} eod_reps2;

struct                               //1090 BITS
{
    unsigned char  week_refund      :1;    // Weekly refunds report in EOD
    unsigned char  week_buyaid      :1;    // Weekly Buy-aids report in EOD
    unsigned char  branch_perform   :1;    // Branch performance report in EOD
    unsigned char  location         :1;    // Location report in EOD
}

```



```
unsigned char  categ_totals      :1;  //      Category totals report in EOD
unsigned char  audit_account     :1;  //      Terminal accountability in EOD
unsigned char  terminal_audit    :1;  //      Terminal audit report in EOD
unsigned char  dept_budget       :1;  //      Dept weekly budget report in EOD
} eod_reps3;

struct                                                //1091 BITS
{
    unsigned char  class_daily      :1;  //      Classification sales report in EOD
    unsigned char  cash_office_sales :1;  //      Cash office sales report in EOD
    unsigned char  dept_sales       :1;  //      Dept sales report in EOD
    unsigned char  acct_rep         :1;  //
    unsigned char  make_staff_d_rep :1;  //      Staff disc weekly report in EOD
    unsigned char  make_cb_daily_rep :1;  //      Checkout bank daily report in EOD
    unsigned char  make_q_len_rep   :1;  //      Queue length report in EOD
    unsigned char  make_cb_eod_rep  :1;  //      Checkout bank report in EOD
} eod_reps4;

struct                                                //1092 BITS
{
    unsigned char  fifteen_min_prod_rep :1;//      15 minutes productivity report in EOD
    unsigned char  performance_summary  :1;//      Performance summary report in EOD
    unsigned char  operator_audit       :1;//      Operator audit report in EOD
    unsigned char  operator_performance :1;//      Operator performance report in EOD
    unsigned char  item_promotions_sales:1;//      Item promotion sales report in EOD
    unsigned char  statement_of_trade   :1;//      statement of trade receivables
    unsigned char  cshr_prod_week       :1;//      Weekly cashier productivity
    unsigned char  cshr_prod_mon        :1;//      Monthly cashier productivity
} eod_reps5;

struct                                                //1093 BITS
{
    unsigned char  item_chk_by_pos      :1;//      Itemized check by POS report
    unsigned char  eft_by_pos           :1;//      EFT by pos report
```

```

unsigned char trd_dtmp_rep      :1;//      trading stamps distribution
unsigned char day_pir_over_rep  :1;//      daily markdown report
unsigned char week_pir_over_rep :1;//      weekly markdown report
unsigned char wbr               :1;//      Weekly business report
unsigned char summary_trial     :1;//      Summary trial balance
unsigned char void_rep          :1;//      voided transactions report
} eod_reps6;

struct                          //1094 BITS
{
    unsigned char Gst_dep_rep    :1;//      At Dept Rprt if this bit is on report on a GST
                                   //      Exclusive basis.
    unsigned char Gst_net_dep_sales_flash :1;//      Same as above for Net Department Sales Flash
    unsigned char Gst_trader_sales_rep  :1;//      Same as above for Trader Sales Report
    unsigned char ccard_trs_sched      :1;//      Credit cards TRS schedule
    unsigned char eftpos_recon         :1;//      EFTPOS Daily Reconciliation
    unsigned char void_cancel_rep      :1;//      canceled subtract transactions report
    unsigned char acc_curr_period      :1;//      Cashier accountability report - current per
    unsigned char acc_prev_period      :1;//      Cashier accountability report - previous per
} eod_reps7;

unsigned long apacs50_seq_no;      //1095 NUM      APACS 50 sequential number
unsigned char eod_week;           //1099 NUM      Day in week to run EOW
unsigned char eod_jan[2];         //1100 ASC_N      Day in month for EOM: JAN
unsigned char eod_feb[2];         //1102 ASC_N      Day in month for EOM: FEB
unsigned char eod_mar[2];         //1104 ASC_N      Day in month for EOM: MAR
unsigned char eod_apr[2];         //1106 ASC_N      Day in month for EOM: APR
unsigned char eod_may[2];         //1108 ASC_N      Day in month for EOM: MAY
unsigned char eod_jun[2];         //1110 ASC_N      Day in month for EOM: JUN
unsigned char eod_jul[2];         //1112 ASC_N      Day in month for EOM: JUL
unsigned char eod_aug[2];         //1114 ASC_N      Day in month for EOM: AUG
unsigned char eod_sep[2];         //1116 ASC_N      Day in month for EOM: SEP
unsigned char eod_oct[2];         //1118 ASC_N      Day in month for EOM: OCT

```

```
unsigned char eod_nov[2];           //1120 ASC_N   Day in month for EOM: NOV
unsigned char eod_dec[2];           //1122 ASC_N   Day in month for EOM: DEC
unsigned char e_o_y_month[2];       //1124 ASC_N   End of year month
unsigned char e_o_y_day[2];         //1126 ASC_N   End of year day in month
unsigned char stamp_01_01;          //1128 NUM     Stamp for write validation
unsigned char day_weekly_movement;  //1129 NUM     Day in week to perform weekly item movement
                                   //              (0 = regular EOW)

unsigned char rec9_fil[20];         //1130 NU      Not used
unsigned char pfs1_pos;             //1150 NUM     POS number for PFS1
unsigned char pfs2_pos;             //1151 NUM     POS number for PFS2
unsigned char poll_list[32];        //1152 NU      Not used
unsigned char cust_audit;           //1184 NU      Not used
unsigned char cust_aud_name[30];    //1185 NU      Not used
unsigned char cust_aud_size[3];     //1215 NU      Not used
unsigned char cust_host_name[30];   //1218 NU      Not used
unsigned char cust_host_size[3];    //1248 NU      Not used
unsigned char cust_host_code1;      //1251 NU      Not used
unsigned char cust_host_code2;      //1252 NU      Not used
unsigned char factura_number[9];    //1253 NU      Not used
unsigned char post_logo[30];        //1262 NU      Not used
unsigned char ascii_trans_name[25]; //1292 ASCII   ASCII transaction file name
struct                               //1317 BITS
{
    unsigned char  build              :1;//      Build ASCII transaction file
    unsigned char  acc_rep_act_cshr    :1;//      Print accountability rpt for active cashiers
    unsigned char  eft_nomad_rep_in_system:1;//      EFT Reconciliation Report[Nomad] in system
    unsigned char  eft_nomad_rep_to_print :1;//      Print EFT Reconciliation Report[Nomad]
    unsigned char  dep_VAT_calc_from_trans_file:1;//      Depart VAT is calculated from transaction file
    unsigned char  cash_office_balance_generate :1;//      Cash Office Balance Report generate
    unsigned char  cash_office_balance_rep_print:1;//      Cash Office Balance Report print
    unsigned char  eft_nomad_report_execute  :1;//      Execute EFT Nomad Report

} tr_opt1;

struct                               //1318 BITS
```

```

{
    unsigned char  include_avg_item_sales  :1;//      Include field "Average Item Sales" in EOD
                                                    //      Expanded Department Reports

    unsigned char  expand_depart_rpt_tax_ex:1;//      Expanded Dept Reports Tax Exclusive Sales

    unsigned char  pos_perform_rpt_generate:1;//      Pos Performance Report Generate

    unsigned char  pos_perform_rpt_print   :1;//      Pos Performance Report Print

    unsigned char  warn_safe_locked_twice  :1;//      Show warning message:Lock Safe already
                                                    //      actioned today

    unsigned char  nu      :3;                //      Not used
} tr_opt2;

struct                                     //1319 BITS
{
    unsigned char  nu      :8;                //      Not used
} tr_opt3;

struct                                     //1320 BITS
{
    unsigned char  nu      :8;                //      Not used
} tr_opt4;

struct                                     //1321 BITS
{
    unsigned char  nu      :8;                //      Not used
} tr_opt5;

unsigned char pc_lpt[10];                  //1322 NUM      LPT used by PC

unsigned char days_skip_period;             //1332 BITS      Day in week to skip period transfer

unsigned char printer_type[10];             //1333 NUM      printer type

unsigned char ranking_report1;              //1343 NUM      Ranking report to do at EOD

unsigned char ranking_report2;              //1344 NUM      Ranking report to do at EOD

unsigned char ranking_report3;              //1345 NUM      Ranking report to do at EOD

unsigned char ranking_report4;              //1346 NUM      Ranking report to do at EOD

unsigned char ranking_report5;              //1347 NUM      Ranking report to do at EOD

unsigned char stamp_01_02;                  //1348 NUM      Stamp for write validation

unsigned char act_nomad_report_time[2];     //1349 NU      Not used

unsigned char PC_1_p_batch_in_us[4];       //1351 ASC_N      Promotion batch in use on PC 1

```

unsigned char PC_2_p_batch_in_us[4];	//1355 ASC_N	Promotion batch in use on PC 2
unsigned char PC_3_p_batch_in_us[4];	//1359 ASC_N	Promotion batch in use on PC 3
unsigned char PC_4_p_batch_in_us[4];	//1363 ASC_N	Promotion batch in use on PC 4
unsigned char PC_5_p_batch_in_us[4];	//1367 ASC_N	Promotion batch in use on PC 5
unsigned char PC_1_cst_batch_use;	//1371 NU	Not used
unsigned char PC_2_cst_batch_use;	//1372 NU	Not used
unsigned char PC_3_cst_batch_use;	//1373 NU	Not used
unsigned char PC_4_cst_batch_use;	//1374 NU	Not used
unsigned char PC_5_cst_batch_use;	//1375 NU	Not used
unsigned char PC_1_cshr_trn_use;	//1376 NUM	Cashier transaction in use on PC 1
unsigned char PC_2_cshr_trn_use;	//1377 NUM	Cashier transaction in use on PC 2
unsigned char PC_3_cshr_trn_use;	//1378 NUM	Cashier transaction in use on PC 3
unsigned char PC_4_cshr_trn_use;	//1379 NUM	Cashier transaction in use on PC 4
unsigned char PC_5_cshr_trn_use;	//1380 NUM	Cashier transaction in use on PC 5
unsigned char PC_1_safe_trn_use;	//1381 NUM	Safe transaction in use on PC 1
unsigned char PC_2_safe_trn_use;	//1382 NUM	Safe transaction in use on PC 2
unsigned char PC_3_safe_trn_use;	//1383 NUM	Safe transaction in use on PC 3
unsigned char PC_4_safe_trn_use;	//1384 NUM	Safe transaction in use on PC 4
unsigned char PC_5_safe_trn_use;	//1385 NUM	Safe transaction in use on PC 5
unsigned char PC_1_batch_in_use[4];	//1386 NUM	PLU batch in use on PC 1
unsigned char PC_2_batch_in_use[4];	//1390 NUM	PLU batch in use on PC 2
unsigned char PC_3_batch_in_use[4];	//1394 NUM	PLU batch in use on PC 3
unsigned char PC_4_batch_in_use[4];	//1398 NUM	PLU batch in use on PC 4
unsigned char PC_5_batch_in_use[4];	//1402 NUM	PLU batch in use on PC 5
unsigned char pcm_c;	//1406 NU	Not used
unsigned char stamp_01_03;	//1407 NUM	Stamp for write validation
char company_name[30];	//1408 ASCII	Company name
char store_tel_num[10];	//1438 ASCII	Phone number
unsigned char filler3[8];	//1448 NU	Not used
unsigned char second_eom[12];	//1456 NUM	Second day in month to run EOD
char store_adr[30];	//1468 ASCII	Store address
char store_fax_num[10];	//1498 ASCII	Store fax number
char store_tax_num[15];	//1508 ASCII	Store tax number
char store_retail[15];	//1523 ASCII	Store retail number

```

unsigned long  invoice;                //1538 NUM    Invoice number
unsigned long  stamp_num;              //1542 NUM    Stamp number
struct                                                //1546 BITS
{
    unsigned char med_in_pos           :1;//    BIT    Media in POSTs signed-on at EOD Rpt
    unsigned char do_audit_rep         :1;//    BIT    do Audit report
    unsigned char posts_error_in_eod   :1;//    BIT    POSTs error report in auto EOD
    unsigned char member_prom_daily    :1;//    BIT    Member Promotion Daily sales
    unsigned char eft_daily_reps       :1;//          EFT daily transactions
    unsigned char plu_not_found        :1;//          PLU not found report
    unsigned char add_data_dep_generate:1;//    BITS    Generate additional data department report
    unsigned char add_data_dep_print   :1;//    BITS    Print additional data department report
} eod_reps8;

struct                                                //1547 BITS    (Nirit)
{
    unsigned char daily_liftflot_generate :1;//          Generate daily lift-float report
    unsigned char weekly_liftflot_generate :1;//          Generate weekly lift-float report
    unsigned char prepaid_card_generate    :1;//          Generate prepaid card report
    unsigned char reward_check_generate    :1;//          Generate reward check report
    unsigned char ss_24h_generate          :1;//          Generate SelfScan 24h/hourly report
    unsigned char post_checkout_bank_report:1;//          Generate POST Checkout Bank report
    unsigned char print_post_checkbank_rep :1;//          Print POST Checkout bank report
    unsigned char sort_promo_add_loss_by_ID:1;//          Sort Promotion Add Loss Report by ID
} eod_reps9;

struct                                                //1548
{
    unsigned char lfs_01                 :1;
    unsigned char lfs_02                 :1;
    unsigned char lfs_03                 :1;
    unsigned char lfs_04                 :1;
    unsigned char lfs_05                 :1;
    unsigned char lfs_06                 :1;

```

```
unsigned char lfs_07          :1;
unsigned char lfs_08          :1;
} lfs_boot;

unsigned char stamp_01_04;      //1549 NUM      Stamp for write validation
struct                          //1550
{
    unsigned char enter_key_as_upc_key :1;
    unsigned char print_saving_and_total:1;
    unsigned char nu                   :6;
} v8_param1;

struct                          //1551
{
    unsigned char summary_report_header :1;
    unsigned char summary_report_type1  :1;
    unsigned char summary_report_type2  :1;
    unsigned char summary_report_type3  :1;
    unsigned char summary_report_type4  :1;
    unsigned char summary_report_type5  :1;
    unsigned char summary_rep_for_total :1;
    unsigned char summary_report_footer :1;
} v8_bonus_buy;

unsigned char total_item_on_receipt; //1552 V8: total item on receipt
unsigned char sign_wic_sale;         //1553 V8:
unsigned char sign_dea_sale;         //1554 V8:
struct                                //1555 BIT
{

    unsigned char checkout_bank_number_1 :1;//      Checkout bank number 1
    unsigned char checkout_bank_number_2 :1;//      Checkout bank number 2
    unsigned char checkout_bank_number_3 :1;//      Checkout bank number 3
    unsigned char checkout_bank_number_4 :1;//      Checkout bank number 4
```

```

unsigned char checkout_bank_number_5 :1;//          Checkout bank number 5
unsigned char checkout_bank_number_6 :1;//          Checkout bank number 6
unsigned char checkout_bank_number_7 :1;//          Checkout bank number 7
unsigned char checkout_bank_number_8 :1;//          Checkout bank number 8

} chk_bank_for_rep1;
struct
{
unsigned char checkout_bank_number_9 :1;//1556 BIT    Checkout bank number 9
unsigned char checkout_bank_number_10 :1;//          Checkout bank number 10
unsigned char nu2                      :6;//          BIT    Not used
} chk_bank_for_rep2;

unsigned char filler4[243];              //1557 NU    Not used
unsigned char
    slip_desc[10][MAX_SLIP_LEN];        //1800 ASCII  Bank names
} SYSPAR_REC;

typedef struct
{
    unsigned char filler1[1099];
    unsigned char eod_week;              // 1099
    unsigned char eod_jan[2];            // 1100
    unsigned char eod_feb[2];            // 1102
    unsigned char eod_mar[2];            // 1104
    unsigned char eod_apr[2];            // 1106
    unsigned char eod_may[2];            // 1108
    unsigned char eod_jun[2];            // 1110
    unsigned char eod_jul[2];            // 1112
    unsigned char eod_aug[2];            // 1114
    unsigned char eod_sep[2];            // 1116
    unsigned char eod_oct[2];            // 1118
    unsigned char eod_nov[2];            // 1120

```



```
unsigned char eod_dec[2];      // 1122
unsigned char e_o_y_month[2]; // 1124
unsigned char e_o_y_day[2];   // 1126
unsigned char filler2[872];   // 1128
} PARAM_REC0_1;
```

```
typedef struct syspar_rec2_      //OFS  Type  Description
{
    // Bonus coupon options
    struct par_coupon_range_      //0      STRUCT
    {
        short low;                //      NUM_D  "From" value
        short high;               //      NUM_D  From amount
        short tender;             //      NUM_D  "To" value
        short mult;               //      NUM_D  Multiply
        short min;                //      NUM_D  Minimum order
    } range[8];

    // Bonus coupon days in use. Entry in the
    // following array for each day of the
    // week.
    struct par_coupon_day_is_use_ //80
    {
        char range1_in_use :1;    //      Range 1 in use
        char range2_in_use :1;    //      Range 2 in use
        char range3_in_use :1;    //      Range 3 in use
        char range4_in_use :1;    //      Range 4 in use
        char nu              :4;   //      Not used
    } day[7];

    struct coupon_net_opt_        //87
    {
        char bonus_cpn_on_disc_plu :1; //      Bonus coupon on discountable items only
        char vendor_dept_net        :1; //      Vendor coupons are netted
    }
}
```

```

char store_dept_net      :1;      //      Store coupons are netted
char bonus_dept_net      :1;      //      Bonus coupons are netted
char allow_cpn_in_no_sale_dep:1;  //      Allow coupons in depts with no sales.
char no_cpn_kbd_sound     :1;      //      No keyboard sound for cpn w/o dept or price
char allow_store_vendor_co :1;      //      Allow Store/Vendor coupon only sale
char EAN99_as_UPC5       :1;      //      Validate EAN99 as UPC5
} opt1;

unsigned char cc_4_cpn_less_0 ;      //88  NUM      CC to allow coupons to bring subtotal below 0
long max_cpn_face_value;      //89  NUM_D      Maximum coupon face value
unsigned char tnd_4_vendr_c ;      //93  NUM      Tender to vendor coupon
unsigned char tnd_4_store_c ;      //94  NUM      Tender to store coupon
unsigned char tnd_4_bonus_c ;      //95  NUM      Tender to bonus coupon
unsigned char level_valid ;      //96  NUM      Level of coupon validation.

// Tax forgive parameters.
// 1 bytes per coupon type:
// 1 byte is vendor
// 2 byte is store
// 3 byte is bonus
struct par_cpn_tax_forgive_      //97  BITS
{
    char tax1      :1;      //      Tax rate 1
    char tax2      :1;      //      Tax rate 2
    char tax3      :1;      //      Tax rate 3
    char tax4      :1;      //      Tax rate 4
    char tax5      :1;      //      Tax rate 5
    char tax6      :1;      //      Tax rate 6
    char tax7      :1;      //      Tax rate 7
    char tax8      :1;      //      Tax rate 8
} tax_forgive[3];

unsigned char tnd_4_upc5_c ;      //100  NUM      Tender for UPC 5

```

```
// Denominations table
struct denom_tbl_                                //101
{
    long value;                                  // NUM_D Value

    struct                                        // BITS
    {
        unsigned char coin :1; // coin
        unsigned char left_till :1; // Left in till
        unsigned char coinage :1; // coinage
        unsigned char nu :5; // Not used
    }opt;

    char nu[3]; // NU Not used
} denom[MAX_DENOM];

long coupon_ceiling[8]; //301 NUM_D Coupon ceiling
long add_amount[8]; //333 NUM_D

unsigned char tender_no_99_coupon; //365 NUM tender number for 99 coupon
unsigned char stamp_02_01; //366 NUM Stamp for write validation
unsigned char upc5_cpn_markdown[2]; //367 BCD Markdown department for UPC5
// coupons with 992 bypass code

unsigned char filler1[30]; //369 NU Not used
unsigned char cc_redm_out_of_sale; //399 NUM CC allow redemption of 99 coupons out of sale
unsigned char cc_more_cpn_items; //400 NUM CC for more coupons than items.
unsigned char cc_allow_cpn_first; //401 NUM CC to allow coupon as first item in the ticket
struct coupon_depart_sales_ //402 BIT
{
    char more_store_coupon :1; // Allow more store coupons than # of items sold.
    char store_coupon_value :1; // Allow store coupon value > than value sold.
    char more_vendor_coupon :1; // Allow more vendor coupons than # of items sold
    char vendor_coupon_value :1; // Allow Vendor coupon value > than value sold
    char nu :4; //
```

```

} opt2;

unsigned char  cc_more_store_coup  ;    //403  NUM    CC store coupons
unsigned char  cc_store_coup_value ;    //404  NUM    CC store coupon value
unsigned char  cc_more_vendor_coup ;    //405  NUM    CC vendor coupons
unsigned char  cc_vendor_coup_value;    //406  NUM    CC Vendor coupon value
unsigned short max_store_cpn   ;        //407  NUM    max store coupons
long  max_store_value         ;        //409  NUM    store coupon value amount
unsigned short max_vendor_cpn  ;        //413  NUM    max Vendor coupons
long  max_vendor_value        ;        //415  NUM    Vendor coupon value amount
unsigned char  stamp_02_02;          //419  NUM    Stamp for write validation
unsigned char  filler2[580];          //420  NU     Not used
                                         //1000

} PARAM_REC2;
#define PARAM_COUPONS_REC  2

typedef struct syspar_rec3_          //OFS  Type  Description
{
    char tare_weight_table_des[30][16]; //0    ASCII  tare weight description
    char tare_weight_percent_des[30][16]; //480  ASCII  tare weight percent description
    unsigned char  stamp_03_01;          //      NUM    Stamp for write validation
    unsigned char  filler[39];
}PARAM_REC3;
#define PARAM_TARE_WEIGHT_REC  3

#define MAX_QDX_CHKSUM  30
typedef struct syspar_rec4_          //OFS  Type  Description
{
    // qdex files checksum table
    struct qdx_chksum_parm_          //      STRUCT
    {
        unsigned char  file_no;        //      NUM    QDX file number
        unsigned short ofs1;           //      NUM    Offset 1
        unsigned short len1;           //      NUM    Bytes from offset 1
    }

```

```
    unsigned short ofs2;           //    NUM    Offset 2
    unsigned short len2;           //    NUM    Bytes from offset 2
    unsigned short ofs3;           //    NUM    Offset 3
    unsigned short len3;           //    NUM    Bytes from offset 3
    long          max_rec;         //    NUM
    unsigned char  nu[7];          //    NU     Not used
} file[MAX_QDX_CHKSUM];
    unsigned char filler1[280];    //    NU     Not used
} PARAM_REC4;
```

```
#define POLL_RECORD 5
typedef struct syspar_rec5_
{
    // poll list table
    struct pos_poll_rec_          //    STRUCT
    {
        unsigned char  primary;   //    NUM    Primary PC
        unsigned char  secondary; //    NUM    Secondary PC
        unsigned char  in_pc;     //    NUM    In PC
        char          filler[1];  //    NU     Not used
    } pos[250];
} POLL_REC;
```

```
typedef struct
{
    struct pos_poll_
    {
        char  primary;
        char  secondary;
        char  in_pc;
        char  filler[1];
    } pos[250];
} PARAM_REC5;
```

```
{
    unsigned char    grand_total[6];
    unsigned int     z_counter;
    unsigned int     last_date;
    unsigned char    total_sales[6];
    unsigned char    free[6];
}
```

[illegible]

```

//          3 - Monthly
//          4 - Previous month
//          5 - Yearly
unsigned char nu02[380];      //620  NU      Not used
} PARAM_REC8
```

HEADFOOT.QDX (47)

File Type	Relative
Record Size	1024
Key Length	0

Record Format

// This structure defines a single header/footer line

```
struct HeaderFooterLine_           //
{
    unsigned char data[62];         //   ASCII   Line text
    unsigned char type;             //   NUM     0-no-print
                                    //           1-normal
                                    //           2-enlarge (bold)
    unsigned char print;            //   NUM     0-not printed
                                    //           1-print
};
```

```
struct HeaderFooter_               //OFS  Type   Description
{
    unsigned char RecordType;       //0     NUM     0-Header 1-Footer
    unsigned char name[30];         //1     ASCII   H/F description
    unsigned char filler[352];      //31    NU      Not used
    struct HeaderFooterLine_ hfData[10]; //383  STRUCT
                                    //
    unsigned char qdx;              //1023 BITS    QDEX reserved flags
};
```

// qdx file #52


```
struct cust_post_  
{  
    unsigned char cust_key[6];        // HEX: 6 bytes  
    unsigned char last_name[15];      // ASCII  
    unsigned char first_name[10];     // ASCII  
    unsigned char mother_name[15];    // ASCII  
    unsigned char address1[30];       // ASCII  
    unsigned char address2[30];       // ASCII  
    unsigned char flag;  
    unsigned char referance[15];      // ASCII  
    unsigned char filler[5];          // ASCII  
    unsigned char qdex_flag;  
};
```

```
struct tare_weight_rec_  
{  
    char name[16];  
    short weight;
```

WEEKTTTL.QDX (48)

File Type	Relative
Record Size	1000
Key Length	0

Record Format

typedef struct	//OFS	Type	Description
{			
char date[3];	//0	BCD	Date
struct week_dep_	//3	STRUCT	
{			
unsigned short num;	//	NU	Department number
long sales;	//	NUM_D	Daily sales
} dep[WEEK_DEPTS];			
char nu00[2];	//801	NU	Not used
struct week_tax_	//803	STRUCT	
{			
long taxable;	//	NUM_D	Taxable amount
long tax;	//	NUM_D	Tax amount
} tax[WEEK_TAXES];			
long voids;	//867	NUM_D	Voids value
long returns;	//871	NUM_D	Returns value
long discounts;	//875	NUM_D	Discounts value
long v_coupon;	//879	NUM_D	Vendor coupons value
long s_coupon;	//883	NUM_D	Store coupons value
long b_coupon;	//887	NUM_D	Bonus coupons value
long media[WEEK_MEDIAS];	//891	NUM_D	Media value
long customer;	//991	NUM	Customer count
long items;	//995	NUM	Items count

```
    char qdx_flag;                                //999 BITS    QDEX reserved flags
} WEEKLY_REC;                                    //1000

// record 0-49: groups
// record 50- : super groups
#define GROUP_SUPER_START    50
#define MAX_GROUP            100
```

LOCATION.QDX (49)

File Type	Relative
Record Size	1024
Key Length	0

Record Format

typedef struct location_rec_	//OFS	Type	Description
{			
unsigned char logo_no;	//0	NUM	Header / Footer
unsigned char kbd_no;	//1	NUM	keyboard number
unsigned char till_profile;	//2	NUM	Till profile:
	//		0 - not used
	//		1 - Super store
	//		2 - Garage
	//		3 - Coffee
	//		4 - Pharmacy
	//		5 - Quick
unsigned char scale;	//3	NUM	Scale type
	//		0-None
	//		1-9000
	//		2-Shekel Nix2
	//		3-Shekel Nix3
	//		4-NCR scanner scale
	//		5-Magellan
	//		6-Orion scanner scale
unsigned char cust_display;	//4	NUM	Customer display
unsigned char printer;	//5	NUM	Printer type
unsigned char authorization;	//6	NUM	Authorization
	//		0 - None
	//		1 - SMT

```

//          2 - NTN
//          3 - EFT
//          4 - MTXEPS
//          5 - Nomad
//          6 - Quest
unsigned char grand_total[5] ; //7  BIN  POS grand total
unsigned short z_number ; //12  NUM  Z number counter
unsigned short last_date ; //14  DATE_I  Last date of grand total update
unsigned char total_sales[5] ; //16  BIN  POS total sales
unsigned char pos_type; //21  NUM  POS type (Select, TeamPOS 5000)
unsigned char monitor; //22  NUM  Monitor type
struct //23  STRUCT
{
    unsigned char drawer :1; // BIT  Drawer
    unsigned char drawer_c_to_printer:1; // BIT  Drawer connected to printer
    unsigned char dual_drawer_in_use :1; // BIT  Dual drawer in use
    unsigned char nul :5; //

}flags;

unsigned char debug; //24  NUM  Debug mode
unsigned char graphic_logo; //25  NUM  Graphic Logo
unsigned char auth_port; //26  NUM  Authorization port
unsigned short auth_baud; //27  NUM  Authorization baud rate
unsigned char apt; //29  NUM  APT Y/N
unsigned char apt_port; //30  NUM  APT port
unsigned short apt_baud; //31  NUM  APT baud rate
struct //33  STRUCT
{
    unsigned char bagger_id :1; // bagger ID
    unsigned char scanner_handshake :1; // 0-none 1-dtr
    unsigned char text_to_trs :1; // create text transactions
    unsigned char battery_backup :1; // battery backup
    unsigned char ext_trs_file :1; // test external transaction file
}
```

```

    unsigned char keyboard_disp_2x20 :1; //          for future used
    unsigned char weight_on_CRT      :1; //          for future used
    unsigned char chk_reader_in_use  :1; //          Check reader in use
} flags1;

unsigned char scanner_baud  ;          //34  NUM    scanner baud rate
unsigned char scanner_bits  ;          //35  NUM    scanner bits
unsigned char scanner_parity ;          //36  NUM    scanner parity
long          min_draw_amount;          //37  NUM_D   Top Up message TESCO
long          max_draw_amount;          //41  NUM_D   Assistance Message TESCO
short         scale_baud    ;          //45  NUM    Scanner baud rate
short         scale_port    ;          //47  NUM    Scanner port
unsigned char scale_irq     ;          //49  NUM    Scale IRQ number
unsigned char delay_auto_eod ;          //50  NUM    Delay between message & EOD start
unsigned char till_type     ;          //51  NUM    Till type
                                     //          1 - Main
                                     //          2 - Petrol Kiosk
                                     //          3 - Opt
unsigned long EFT_location  ;          //52  NUM    EFT location
unsigned char checkout_bank ;          //56  NUM    for productivity & sales reports
unsigned short pace_of_q_display;          //57  NUM    Queue length prompt delay
unsigned char q_length_alert ;          //59  NUM    Queue length to send alert
unsigned char touch_type;          //60  NUM    Touch screen type
unsigned char CC_min_amount;          //61  NUM    Control check for minimum cash in drawer amt
unsigned char CC_max_amount;          //62  NUM    Control check for maximum cash in drawer amt
unsigned char net_sales[5];          //63  BIN    POS net sales
long          over_short_daily;          //68  NUM_D   Daily over/short value
long          over_short_prv_day;          //72  NUM_D   Prev. day over/short value
long          over_short_weekly;          //76  NUM_D   Weekly over/short value
long          over_short_prv_week;          //80  NUM_D   Prev. week over/short value
long          over_short_monthly;          //84  NUM_D   Monthly over/short value
long          over_short_prv_month;          //88  NUM_D   Prev. month over/short value
long          over_short_yearly;          //92  NUM_D   Yearly over/short value
struct          //96  STRUCT

```

```

{
    char allow_eft_reconciliation    :1;    //          Allow EFT reconciliation
    char un                          :7;    //          Not used

} flags2;

unsigned char printer_port;          //97    NUM      Printer port
unsigned char port_for_print_command; //98    NUM      Port for print commands
unsigned char pos_beep_from;         //99    NUM      Generate 'beep' sound from..

unsigned char wp_kbd;                //100   NUM      WinPOS Keyboard
//                                //          0 Standard
//                                //          1 ICL POST KBD
//                                //          2 ICL Table KBD
//                                //          3 SNIKEY
//                                //          4 Dynakey

unsigned char wp_sound;              //101   NUM      WinPOS Generate sound from
//                                //          0 POST Keyboard
//                                //          1 PC Speaker
//                                //          2 Both

unsigned char wp_printer;            //102   NUM      WinPOS printer
//                                //          0 None
//                                //          1 EPSON_TM930
//                                //          2 FUJITSU
//                                //          3 EPSON TMT80
//                                //          4 AXIOHM 7156
//                                //          5 EPSON TM950
//                                //          6 EPSON TM200
//                                //          7 AXIOHM 4101
//                                //          8 AXIOHM A758
//                                //          9 IBM 4610
//                                //          10 EPSON H6000

unsigned char wp_prn_port;           //103   NUM      WinPOS printer port
unsigned char wp_prn_baud;           //104   NUM      WinPOS baud rate
//                                //          0 None

```

```

//          2 2400
//          3 4800
//          4 9600
//          5 19200
//          6 38400
struct      //   STRUCT  WinPOS Flags
{
    unsigned char wp_MICR      :1;    //105  BIT    WinPOS check reader
    unsigned char disp_const_wgt :1;    //105  BIT    Display constant weight
    unsigned char battery      :1;    //105  BIT    Battery backup
    unsigned char oprt_2X20    :1;    //105  BIT    Operator display 2X20
    unsigned char use_partial_cut :1;    //105  BIT    Printer with partial cut
    unsigned char check_flip    :1;    //105  BIT    Check flip feature
    unsigned char opos_printer  :1;    //105  BIT    Use OPOS for printer
    unsigned char nu :1;        //105  BIT    POS is a self Checkout
} wp_flags1;
unsigned char wp_scn1port;        //106  NUM    WinPOS scanner port (0-10)
unsigned char wp_scn1baud;        //107  NUM    WinPOS scanner baud rate
//                                2400,4800,9600,19200,38400
unsigned char wp_scn1data_bits;   //108  NUM    WinPOS scanner data bits
unsigned char wp_scn1parity;      //109  NUM    WinPOS scanner parity
//                                0 None
//                                1 Odd
//                                2 Even
unsigned char wp_scn2port;        //110  NUM    WinPOS scanner port (0-10)
unsigned char wp_scn2baud;        //111  NUM    WinPOS scanner baud rate
//                                2400,4800,9600,19200,38400
unsigned char wp_scn2data_bits;   //112  NUM    WinPOS scanner data bits
unsigned char wp_scn2parity;      //113  NUM    WinPOS scanner parity
//                                0 None
//                                1 Odd
//                                2 Even
unsigned char wp_scl_type;        //114  NUM    WinPOS scale type
//                                0 None

```



```

//          1 ICL 9000 or 9500
//          2 Shekel NIX2 Protocol
//          3 Shekel NIX3 Protocol
//          4 NCR 7870 Scanner/Scale
//          5 'W' Protocol scales
unsigned char wp_scl_port;    //115  NUM  WinPOS Scale port
unsigned char wp_scl_baud;    //116  NUM  WinPOS Scale baud
//          0 None
//          2 2400
//          3 4800
//          4 9600
//          5 19200
//          6 38400
unsigned char wp_scl_data_bits; //117  NUM  WinPOS data bits
unsigned char wp_scl_parity;    //118  NUM  WinPOS scale parity
//          0 None
//          1 Odd
//          2 Even
unsigned char wp_cust_type;    //119  NUM  WinPOS customer disp. type
//          0 None
//          1 ICL standard
//          2 SNI standard
unsigned char wp_cust_port;    //120  NUM  WinPOS customer disp. port
unsigned char wp_cust_baud;    //121  NUM  WinPOS
//          0 None
//          2 2400
//          3 4800
//          4 9600
//          5 19200
//          6 38400
unsigned char wp_drawer_type;  //122  NUM  WinPOS drawer type
//          0 None
//          1 Basic
//          2 Connected to printer
```

```

unsigned char host_ip_address[6];          //123  BCD
unsigned char host_port_number[4];        //129  BCD
unsigned char eft_app_prog_version[4];    //133  BCD
unsigned char eft_app_param_version[4];   //137  BCD
unsigned char pinpad_unit_type[8];        //141  ASCII

unsigned char sacnner_irq;                 //149      Scanner IRQ number
unsigned char cust_display_irq;            //150      Customer Display IRQ number

//unsigned char authorization_irq;         //151      Authorization IRQ number. (may not be used)
//                                         //         Added WinPOS parameters instead

struct WP_flags3__                        //151
{
    unsigned char pos_cds_active           :1; //active POS for Code distribution
    unsigned char menu_shift_keys         :1;
    unsigned char screen_scaling          :1;
    unsigned char use_touch_screen        :1;
    unsigned char touch_oposite           :1;
    unsigned char old_mci_video            :1;
    unsigned char nu                       :2;
} flag3;

struct WP_opos_                          //152  Struct for OPOS flags
{
    unsigned char OPOS_scanner_1           :1;
    unsigned char OPOS_scanner_scale_1     :1; //if the scanner is in a scanner scale set this one
    unsigned char OPOS_scanner_2           :1;
    unsigned char OPOS_scanner_scale_2     :1; //if the scanner is in a scanner scale set this one
    unsigned char OPOS_scale               :1;
    unsigned char OPOS_drawer              :1;
    unsigned char OPOS_keyboard            :1;
    unsigned char OPOS_customer_display    :1;

```

```
unsigned char reserved[19];
    }WP_opos;

unsigned char pos_hw_type;           //172  NU    POS type
                                     //          0 DOS
                                     //          1 WinPOS
                                     //          2 Linux
                                     //          3 Money Market

unsigned char colors;               //173        Number of colors
                                     //          0 - 256
                                     //          1 - More than 256

unsigned char wp_display_type;       //174        WinPOS display type
                                     //          0 - Full screen
                                     //          1 - Wide menu panel
                                     //          2 - Narrow menu panel
                                     //          3 - Promotional Screen

unsigned char dual_vga;              //175        Dual VGA
                                     //          0 - No
                                     //          1 - Yes
                                     //          2 - Simulation

unsigned char receipt_disp_mode;     //176        Receipt display mode
                                     //          0 - Full
                                     //          1 - Partial

unsigned char disp_mode;             //177        Display mode
                                     //          0 - 640X480
                                     //          1 - 800X600

unsigned char touch_rows;            //178
unsigned char touch_cols;            //179
unsigned char dynakey_type;          //180        Dynakey type
                                     //          0 - Old
                                     //          1 - New

unsigned char opos_PrinterName[20];  //181
unsigned char opos_LineDisplayName[20]; //201
unsigned char opos_DrawerName[20];   //221
```

```

unsigned char opos_ScannerName[20];      //241
unsigned char opos_ScaleName[20];       //261
unsigned char opos_MicrName[20];        //281
unsigned char opos_MsrName[20];         //301

// WinPOS - Self Checkout
unsigned short self_checkout_cashier;    //321      Self checkout cashier number
unsigned short self_checkout_POS_num;    //323      Self checkout pos number
unsigned char self_checkout_type;        //325      // self checkout type
                                                // 0 = None
                                                // 1 = NCR SCOT
                                                // 2 = PSI

unsigned char opt_e_ticket_pos           :1;      //326
unsigned char opt_nu                     :7;
unsigned short e_ticket_virtual_pos_number; //327
unsigned short e_ticket_cashier;          //329

unsigned char storeline_eps_pinpad_type;  //331      0 = None
                                                //          1 = Concord
                                                //          2 = enTouch1000
                                                //          3 = enCrypt2100

unsigned char kbd_port;                   //332      NUM port number 0-10
unsigned char filler3[668];               //333

struct                                   //1001 BITS
{
    unsigned short cash_office_dep_sales :1; //      CODS report for this POS in
                                                //      EOD
    unsigned short acct_rep_pos          :1; //      Accountability report for
                                                //      this POS in EOD
    unsigned short sign_on_today         :1;
    unsigned short sign_on_yesterday     :1;

```

```
    unsigned short nu          :12; //          reserved
} eod_rep;

    unsigned char ok_pump[20]   ;          //1003 NUM      Pumps linked to POS
    unsigned char qdx_flag     ;          //1023 BITS      QDX flag
}LOCATION;

typedef struct
{
    unsigned char  code;          // 0
    unsigned char  type;          // 1
    unsigned char  plu[13];       // 2
    unsigned char  start_date[6]; // 15
    unsigned char  end_date [6];  // 21
    unsigned char  price[8];      // 27
    unsigned char  desc[16];      // 35
    unsigned char  id[4];         // 51
    unsigned char  offer[4];      // 55
    unsigned char  qty_std[4];     // 59
    unsigned char  qty[4];        // 63
    unsigned char  price_is_percent; // 67
    unsigned char  limited_qty    ; // 68
    unsigned char  flags[6];      // 69
    unsigned char  min_purch_req;  // 75
    unsigned char  min_pur_amount[8]; // 76
    unsigned char  c_check[2];    // 84
} PROM_BATCH;

typedef struct
{
    //86 compatible part:
    unsigned char  code;          // 0
    unsigned char  type;          // 1
    unsigned char  old_plu13[13]; // 2
```

```
unsigned char  start_date[6];      // 15
unsigned char  end_date  [6];      // 21
unsigned char  price[8];           // 27
unsigned char  desc[16];           // 35
unsigned char  id[4];             // 51
unsigned char  offer[4];           // 55
unsigned char  qty_std[4];         // 59
unsigned char  qty[4];            // 63
unsigned char  price_is_percent; // 67
unsigned char  limited_qty      ; // 68
unsigned char  flags[6];          // 69
unsigned char  min_purch_req;     // 75
unsigned char  min_pur_amount[8]; // 76
unsigned char  c_check[2];        // 84

//new part
unsigned char  plu[20];           // 86
unsigned char  not_used[94];      // 106

}  PROM_BATCH_200;                //200

typedef struct
{
    char code;           // 0 '9'
    char type;           // 1
    char date[6];        // 2
    char time[4];        // 8
    char name[12];       //12
    char fiiler[62];     //24
} PROM_BAT_FOOTER;      //86

typedef struct
{
    char code;           // 0 '9'
```

```
char type;           // 1
char date[6];        // 2
char time[4];        // 8
char name[12];        //12
char filler[176];    //24
} PROM_BAT_FOOTER_200; //200
```

```
union bat_rec_un_
{
    PLU_BAT_UN      plu;
    PROM_BATCH      prom;
    PROM_BAT_FOOTER prom_foot;
    char buff[200];
}
```

CUSTOMER.QDX (50)

File Type	Index
Record Size	9
Key Length	6

Record Format

struct debtor_idx_	//OFS	Type	Description (Versions below 7.3.3)
struct cust_idx_	//OFS	Type	Description (Version 7.3.3. and Up)
{			
unsigned char number[6];	//0	BCD	Account number
unsigned char nu[3];	//6	NU	Not used
};			

CUSTRLAT.QDX (51)

File Type	Relative expansion
Record Size	512
Key Length	0

Record Format

typedef struct debtors_rec_	//OFS	Type	Description
{			
unsigned char number[6];	//0	BCD	Account number
char nu01[6];	//6	NU	Not used
long credit_limit;	//12	NUM_D	Credit limit 1
long credit_limit2;	//16	NUM_D	Credit limit 2
long previous_limit;	//20	NUM_D	Previous limit
long over_limit_percent;	//24	NU	Not used
unsigned char account_type;	//28	NUM	Account type
unsigned char discount_type;	//29	NUM	Discount type
unsigned int store_number;	//30	NUM	Store number
unsigned char language;	//32	NUM	Language
unsigned char notes1[80];	//33	ASCII	Notes 1
unsigned char notes2[80];	//113	ASCII	Notes 2
unsigned char surname[20];	//193	ASCII	Surname
unsigned char initials[4];	//213	ASCII	Initials
unsigned char title[4];	//217	ASCII	Title
unsigned char address1[30];	//221	ASCII	Address 1
unsigned char address2[30];	//251	ASCII	Address 2
unsigned char address3[30];	//281	ASCII	Address 3
unsigned int postalCode;	//311	NUM	Post code
unsigned char telephone_home[16];	//313	ASCII	Phone number at home
unsigned char telephone_work[16];	//329	ASCII	Phone number at work
unsigned char date_opened[3];	//345	BCD_D	Date opened

```

unsigned char nu02[6];           //348  NU      Not used
unsigned char last_lim_change_date[3]; //354  BCD_D   Date of last limit change
long current_installment;        //357  NUM_D   Current installment
long balance30d;                 //361  NUM_D   30 days arrears
long balance60d;                 //365  NUM_D   60 days arrears
long balance90d;                 //369  NUM_D   90 days arrears
long balance120d;                //373  NUM_D   120 days arrears
char nu03[8];                    //377  NU      Not used
long value_current_due;          //385  NUM_D   Current due
unsigned char id_number[13];      //389  ASCII   ID number
long opening_balance;            //402  NUM_D   Opening balance
unsigned char statistics1[6];     //406  ASCII   Statistics part 1
unsigned char statistics2[6];     //412  ASCII   Statistics part 2
unsigned char statistics3[6];     //418  ASCII   Statistics part 3
unsigned char nu04[57];          //424  NU      Not used

// Update by Bt
unsigned char record_status;      //481  NUM     Record status
unsigned char account_status;     //482  NUM     Account status
long account_balance;            //483  NUM_D   Account balance
long value_last_purchase;        //487  NUM_D   Last purchase value
long value_last_payment;         //491  NUM_D   Last payment value
unsigned char last_purchase_date[3]; //495  DATE    Last purchase date
unsigned char last_payment_date[3]; //498  DATE    Last payment date
unsigned char filler[10];         //501  NU      Not used

unsigned char qdx_flag[1];        //511  BITS    QDEX reserved flags

} DEBTORS;

typedef struct
{

```

```
    struct debtor_idx_ idx;  
    DEBTORS           rlt;  
} DEBT_REC;
```

```
typedef struct  
{  
    unsigned char name[9];  
    unsigned char number[6];  
    unsigned char nu[3];  
} DNAME_REC;
```

```
typedef struct  
{  
    DNAME_REC      key;  
    DEBTORS        rlt;  
} DEBTNAME_REC;
```

POSCONF.QDX (53)

File Type	Relative
Record Size	1024
Key Length	0

Record Format

typedef struct POS_PARAM_	//OFS	Type	Description
{			
long cash_in_drawer_limit;	//0	NUM_D	Cash in drawer limit
	//		
unsigned char curr_symbol_at_tend :1; //4		BIT	Currency symbol before each tender & Bal Due.
unsigned char curr_symbol_at_price :1; //4		BIT	Current symbol at each price on receipt
unsigned char open_drawer_in_training:1; //4		BIT	Open drawer in training
unsigned char allow_foreign_change :1; //4		BIT	Allow foreign currency change
unsigned char dept_repeat_in_use :1; //4		BIT	Dept. repeat key in use
unsigned char void_chk_sig_allowed :1; //4		BIT	Void transaction at "Check signature allowed
unsigned char spvr_req_for_void :1; //4		BIT	SVR keylock to void non sale functions
unsigned char force_dept_price :1; //4		BIT	Force dept/price entry of unknown items
struct	//5	STRUCT	
{			
unsigned char hooks_yn :1; //		BIT	User hooks in system
unsigned char debug :1; //		BIT	Debug mode while in hook
unsigned char nu :6; //		NU	Reserved
} hooks_prm;			
// The following array include a bit per			
// hook. Total of 72 possible hooks.			
unsigned char user_hooks[8];	//6	BITS	0-Initialisation
	//		1-Before No-sale
	//		2-At end of transaction

```
//      3-Before sign on,sign off, secure, wait
//      4-After total key
//      5-Before PLU read
//      6-After PLU read
//      7-After PLU sale completion
//      8-Before dept. read
//      9-Afer dept. read
//     10-After dept. sale completion
//     11-Before discount read
//     12-After discount read
//     13-Afrer discount completion
//     14-Not used
//     15-Before return read
//     16-After return read
//     17-Not used
//     18-Before tender read
//     19-After tender read
//     20-After tender completion
//     21-Not used
//     22-Not used
//     23-Not used
//     24-After frequent shopper entry
//     25-Before "Check signature" prompt
//     26-User hook functions
//     27-Not used
//     28-Before BROM read
//     29-After BROM read
//     30-After BROM completion
//     31-Before maintenance exec.
//     32-Ager maintenance exec.
//11    40-After total key and order promo calculation
unsigned char stamp 00 01; //14  NUM  Stamp for write validation
unsigned char max_digits_for_dep_value; //15  NUM  Max digits for dep. entry
unsigned char max_digits_for_med_value; //16  NUM  Max digits for media entry
```

```

unsigned char MSU_Option;           //17  NUM  MSU rounding options:
                                     //      0- 34 34 32
                                     //      1- 34 33 33

unsigned char charge_payment_tender; //18  NUM  Charge payment tender
unsigned char rev_charge_payment_tender; //19  NUM  Reversal charge payment tender
unsigned char audit_detail;         //20  NUM  Level of detail on audit:
                                     //      0-Minimum
                                     //      1-Medium
                                     //      2-Maximum

unsigned long default_min_purch_promo; //21  NUM_D  Default min. purchase for promotions
unsigned char empl_no_at_bo_func;     //25  NUM  Empl No. & password required
                                     //      at B/O function:
                                     //      0 - Never required
                                     //      1 - Number only, always
                                     //      2 - Number only, signed off
                                     //      3 - Number & password, always
                                     //      4 - Number & password, s.off

unsigned char bo_confirm_prompt      :1; //26  BIT  Prompt to confirm selection of B/O function
unsigned char disp_expected_figs     :1; //26  BIT  Not used
unsigned char print_bal_due_at_sub_to:1; //26  BIT  Display Bal. Due at subtotal
unsigned char password_req_at_signon :1; //26  BIT  Password required at sign on
unsigned char change_password_at_soff:1; //26  BIT  Password required at s. off
unsigned char ATOMIC_on_system       :1; //26  BIT  Not used
unsigned char duplicate_receipt_loan :1; //26  BIT  Duplicate receipt on load / pickup
unsigned char display_fs_total       :1; //26  BIT  Display F/S total after F/S tender
unsigned char cash_lift_options;     //27  NUM  Cash lift options:
                                     //      0-Quantity/Amount
                                     //      1-Cash by Denom.
                                     //      2-Free format

unsigned char disp_extended_subtotaln; //28  NUM  Type of extended bal due:
                                     //      0 - None
                                     //      1 - Food stamps

unsigned short clubcard_weeks_reentry; //29  NUM  clubcard weeks to reentry
unsigned char clubcard_reentry_active :1; //31  BIT  clubcard reentry active

```

unsigned char	mname_cusdisp	:1; //	BIT	Member Name on cust. display
unsigned char	mname_cshdisp	:1; //	BIT	Member name on cshr. display
unsigned char	mname_receipt	:1; //	BIT	Print Member Name on receipt
unsigned char	tax_follow_pos	:1; //	BIT	Tax report after POS report
unsigned char	separate_receipt	:1; //	BIT	Prt separate rcpts for each tender on pickup
unsigned char	cashier_file_pos	:1; //	BIT	Cashier file resides on POS
unsigned char	allow_pswrd_at_signon	:1; //	BIT	Change password allowed at sign-on.
unsigned short				
	max_cncelsub_items_in_tick;	//32	NUM	Max cancel/subtract items per ticket
unsigned char	sign_off_report_option;	//34	NUM	Report at sign off:
		//		0-No report
		//		1-X-read on receipt
		//		2-Name & No. on receipt
		//		3-X-read on audit
		//		4-Name & No. on audit
unsigned char	auto_secure_minutes;	//35	NUM	Minutes before auto secure
unsigned char	auto_signoff_secure_opt;	//36	NUM	Automatic secure mode:
		//		0-Secure
		//		1-Sign off
unsigned char	open_drawer_signon	:1; //37	BIT	Open drawer at sign on
unsigned char	open_drawer_signoff	:1; //37	BIT	Open drawer at sign off
unsigned char	open_drawer_eod	:1; //37	BIT	Open drawer at EOD
unsigned char	print_sleep_plu	:1; //37	BIT	Print slip if PLU not found
unsigned char	limit_fs_elig_bal_due	:1; //37	BIT	Limit F/S eligible for Bal. due.
unsigned char	allow_select_exempt	:1; //37	BIT	Allow cashier to select
		//		specific tax for tax exempt
unsigned char	allow_select_reversal	:1; //37	BIT	Allow cashier to select
		//		specific tax for tax rev.
unsigned char	chk_digit_if_no_scan	:1; //37	BIT	No CDV check on item scan/
		//		ENTER key
long	max_val_item_cncelsub;	//38	NUM	Max item value cancel/ subtract
long	max_cncelsub_val_per_tick;	//42	NUM	Max cancel/subtract value per ticket

```

unsigned char foodstamp_tend_no;          //46  NUM    F/S tender number
unsigned char taxes_to_accumulate;        //47  NUM    Number of rates to accumulate amounts
unsigned char taxes_to_reverse;           //48  BITS   Taxes to be set ON for tax reversal.
unsigned char cheque_payment_tender;      //49  NU     Not used

struct                                    //50  STRUCT Supervisor privileges
{
    unsigned char  priv_1      :1;
    unsigned char  priv_2      :1;
    unsigned char  priv_3      :1;
    unsigned char  priv_4      :1;
    unsigned char  priv_5      :1;
    unsigned char  priv_6      :1;
    unsigned char  priv_7      :1;
    unsigned char  priv_8      :1;
} super_priv;

unsigned char WIC_display_options;         //51  NUM    WIC display options:
//                                         //         0-No extra prompt
//                                         //         1-Prompt for amount
//                                         //         2-Prompt for expiry date
//                                         //         3-Prompt for amount & expiry date
unsigned char WIC_print_option;            //52  NUM    WIC print options:
//                                         //         0-No extra print
//                                         //         1-Print amount
//                                         //         2-Print expiry date
//                                         //         3-Print amount & expiry date
unsigned char CC_exceed_wic_amt;           //53  NUM    CC to exceed WIC check amt.
unsigned char CC_exceed_wic_date;          //54  NUM    CC to exceed WIC expiry date
unsigned char CC_bypass_wic_amt;           //55  NUM    CC to bypass WIC check amt.
unsigned char CC_bypass_wic_date;          //56  NUM    CC to exceed WIC expiry date
struct                                    //57  STRUCT
{

```



```
unsigned char multi_lang      :1; // BIT Multi language in system
unsigned char reset_pos_ctr   :1; // BIT Reset POS counter at EOD
unsigned char eod_chksum      :1; // BIT Perform checksum during EOD
unsigned char read_plu_pc     :1; // BIT Always read PLU from PC
unsigned char open_tender_purc :1; // BIT Open drawer on tender purch.
unsigned char print_fs_sub_if_ten:1; // BIT Print F/S total after F/S
unsigned char cost_plus_in_sys :1; // BIT Cost plus in system
unsigned char plu_upc_enter_repea:1; // BIT PLU/UPC/ENTER repeat key in use
} flags1;

unsigned char FS_forgives_tax :1; //58 BIT Any F/S eligible items paid
// with F/S is tax forgiven
unsigned char FS_forgives_all_tax :1; //58 BIT Any F/S tender forgive tax
// on all F/S elig. items
unsigned char allow_non_whole_FS_tend:1; //58 BIT Allow non whole-Dollar F/S
unsigned char freq_shop_in_use :1; //58 BIT Frequent shopper feature in use
unsigned char use_FS_num_for_check_tn:1; //58 BIT Use Freq. Shop. No. for
// check tender account number
unsigned char total_individual_tax :1; //58 BIT Total/Individual tax on receipt
unsigned char cpn_printing_in_use :1; //58 BIT Coupon printing in use
unsigned char bypass_control_check :1; //58 BIT Bypass CC if conditions met
struct //59 STRUCT
{
    unsigned char cheque_guaranty_fun:1; // BIT Check guarantee function.
    unsigned char cpn_rep_at_eot :1; // BIT Coupons report at End of Ticket
    unsigned char sub_cshr_declare :1; // BIT Subt. cashier declarations from X reports.
    unsigned char no_pos_rep_eod :1; // BIT No POS report at EOD
    unsigned char cust_file_inc_cd :1; // BIT Customer file include check digit.
    unsigned char ticket_trans :1; // BIT Not used
    unsigned char disal_qty_with_scan:1; // BIT Disallow quantity with scanning
    unsigned char ENTER_key_is_total :1; // BIT Use ENTER key as total
} flags3;

struct //60 STRUCT
```

```

{
    unsigned char wic_exp_issue_date :1; // BIT WIC issue date
    unsigned char no_upc5_in_sys :1; // BIT No UPC5 in system
    unsigned char pass_rq_at_signoff :1; // BIT Password required at sign off
    unsigned char dont_print_tax_if_0:1; // BIT Don't print tax if zero
    unsigned char coupon_in_system_99:1; // BIT 99 coupon in system
    unsigned char not_allow_tndr_mode:1; // BIT Do not allow sales in tender mode
    unsigned char fly_cd :1; // BIT Suppress Fly Buys CD
    unsigned char key_in_mcard :1; // BIT Key in member card without check digit
} flags4;

unsigned char freq_shop_range_from[7]; //61 BCD Frequent shopper range from
unsigned char freq_shop_range_to[7]; //68 BCD Frequent shopper range to
unsigned char freq_shop_merch_options; //75 NUM Freq. Shop. merchandising:
// 0-None
// 1-Character
// 2-Text line
// 3-Text line & savings
unsigned char bonus_buy_print; //76 NUM Bonus buy printing
// 0 - None
// 1 - Summary
// 2 - Line
// 3 - Summary & line
unsigned char bad_account_fail_cc; //77 NUM
unsigned char offline_signon_signoff_cc; //78 NUM CC for offline sign on/off
unsigned char drawer_open_timeout; //79 NUM Delay for open drawer alarm

unsigned char key_in_fly_buys_number :1;//80 BIT Key-in Fly Buys number without check digit
unsigned char payment_amount_prompt :1;// BIT Payment amount prompt defaults to balance due
unsigned char print_unique_invoice_num:1;// BIT Prt unique invoice # on charge posting tickets
unsigned char exc_non_merch_items_cpn_print:1;//BIT Exclude non-merch. items from coupon printing
unsigned char mcard_qualifies_for_freq_shop:1;//BIT Member card qualifies for Frequent Shopper
unsigned char ATOMIC_fail_option :3;// NU Not used
unsigned char CC_offline_bad_account; //81 NUM CC for offline bad acct.

```

unsigned char CC_offline_svr_request;	//82	NUM	CC for offline SVR request
unsigned short receipt_print_mode;	//83	NUM	Not used
unsigned long staff_break_amount;	//85	NUM_D	Staff break amount
unsigned long freq_shop_min_purch_amt;	//89	NUM_D	Freq. Shop. minimum purchase amount
unsigned char CC_freq_shop_min_purch;	//93	NUM	CC for not meeting Freq.
	//		Shop. minimum purchase.
unsigned long min_purch_amt_for_disc;	//94	NUM_D	Minimum purchase for trans. discount.
unsigned char CC_for_disc_min_purch;	//98	NUM	CC for not meeting minimum
	//		purchase for trans. discount
unsigned char ACCT_file_option;	//99	NUM	Not used
unsigned char CC_for_not_found;	//100	NUM	Not used
unsigned char no_digits_chg_pst_acct;	//101	NUM	Digits for charge posting account number.
unsigned long check_tend_daily_amt;	//102	NUM_D	Check tender daily limits on account, amount
unsigned char check_tend_daily_cnt;	//106	NUM	Check tender daily limits on account, count
unsigned char CC_cpns_subtotal_less_0;	//107	NUM	CC to allow coupons to bring Subtotal below 0.
unsigned long max_cpn_face_value;	//108	NUM_D	Maximum coupon face value
unsigned char tend_no_vendor_coupons;	//112	NUM	Tender number for vendor c.
unsigned char tend_no_store_coupons;	//113	NUM	Tender number for store c.
unsigned char tend_no_bonus_coupons;	//114	NUM	Tender number for bonus c.
unsigned char max_cpns_to_print;	//115	NUM	Maximum number of coupons to print per trans.
unsigned long min_purch_amt_cpn_print;	//116	NUM_D	Minimum purchase to print coupons.
unsigned char CC_min_purch_cpn_print;	//120	NUM	CC for minimum purchase
unsigned char CC_audit_low;	//121	NUM	CC for paper low: audit
unsigned char CC_key_weight;	//122	NUM	CC for keying weight
unsigned char CC_can_sub_limit;	//123	NUM	CC for exceeding cancel / subtract limit
unsigned long can_sub_limit_amt;	//124		CC for exceeding cancel / subtract lmt amount
unsigned char CC_trans_void;	//128	NUM	CC for transaction void
unsigned char CC_sub_next;	//129	NUM	CC for subtract next
unsigned char CC_item_cancel;	//130	NUM	CC for item cancel
unsigned char CC_can_sub_post_action;	//131	NUM	CC for subtract after completion
unsigned char CC_cancel_tmpl_print;	//132	NUM	CC for pressing item cancel during print
unsigned char CC_return_tran_post_act;	//133	NUM	CC for return transaction, after completion.
unsigned char CC_no_sale_in_trans;	//134	NUM	CC for No-sale within tran.
unsigned char CC_no_sale_out_trans;	//135	NUM	CC for No-sale out of tran.

unsigned char CC_credit_card_elig;	//136	NUM	CC for CC tender exceeding eligible amount
unsigned char CC_save_tran_tend;	//137	NUM	CC for saving transaction with partial tender.
unsigned char CC_price_override;	//138	NUM	CC for price override
unsigned char in_house_style;	//139	NUM	In house style 0-Charge posting 1- balancing
// Variable price/weight options			
struct variable_price_	//140	STRUCT	
{			
unsigned char option :1;	//	BIT	Option in use
unsigned char PCDV_reqd :1;	//	BIT	PDCV required
unsigned char price_weight_opt :1;	//	BIT	Price/Weight
unsigned char nu3 :1;	//	BIT	Not used
unsigned char nu4 :1;	//	BIT	Not used
unsigned char nu5 :1;	//	BIT	Not used
unsigned char nu6 :1;	//	BIT	Not used
unsigned char nu7 :1;	//	BIT	Not used
unsigned char digits_in_price;	//	NUM	Digits in price
unsigned char decimal_digits;	//	NUM	Decimal places
}			
VP02,	//140	STRUCT	Starting digits 02
VP04,	//143	STRUCT	Starting digits 04
VP20,	//146	STRUCT	Starting digits 20
VP21,	//149	STRUCT	Starting digits 21
VP22,	//152	STRUCT	Starting digits 22
VP23,	//155	STRUCT	Starting digits 23
VP24,	//158	STRUCT	Starting digits 24
VP25,	//161	STRUCT	Starting digits 25
VP26,	//164	STRUCT	Starting digits 26
VP27,	//167	STRUCT	Starting digits 27
VP28,	//170	STRUCT	Starting digits 28
VP29,	//173	STRUCT	Starting digits 29
VP99;	//176	STRUCT	Starting digits 99

```
unsigned short  cost_plus_tax_surch_dept;//179  NUM      Cost plus surcharge dept
unsigned short
    non_cost_plus_tax_surch_dept;//181  NUM      Cost plus non-taxable surcharge department.
long          staff_disc_min_purchase;  //183  NUM_D    Staff discount minimum purchase.
unsigned char  staff_disc_min_purchase_c; //187  NUM      CC for not meeting staff disc. min. purchase.
unsigned char  CC_insufficient_funds;    //188  NUM      CC for insufficient funds.
unsigned char  CC_restrict_redemption;   //189  NUM      CC restrict redemption of customer
//                                          of customer accounts out sale
unsigned char  cc_max_cnlsb_itme_val;    //190  NUM      CC for Max item value cancel/subtract
unsigned char  cc_max_cnlsb_val_per_tick;//191  NUM      CC for Max cancel/subtract value per ticket
unsigned char  cc_max_cnlsb_items;       //192  NUM      CC for Max cancel/subtract items per ticket

unsigned char  cc_not_in_i_h_ch_file;    //193  NUM      CC for offline in house charge no file
unsigned char  cc_offline_ecca;          //194  NUM      CC for offline ECCA
unsigned char  cc_offline_debtors;       //195  NUM      CC for offline debtor
unsigned char  cc_bypass_freq_shop;      //196  NUM      CC to bypass frequent shop.
unsigned char  cc_receipt_low;           //197  NUM      CC for paper low: receipt.
unsigned char  CC_chq_printing           ; //198  NUM      CC for bypass check print
unsigned char  CC_chq_endorsment         ; //199  NUM      CC for bypass check endorsement.
unsigned char  CC_tender_template_print; //200  NUM      CC for bypass tender template printing.
unsigned char  restrict_bday_in_use :1;  //201  BIT      Restricted sale/birthday feature in use.
unsigned char  accept_bday_prompt   :1;  //201  BIT      Prompt to accept birthday
unsigned char  assist_no_for_payment:1;  //201  BIT      Assistant number required for account payments
unsigned char  forgive_tax_coupon_fs:1;  //201  BIT      Forgive tax on vendor coupon if F/S tendered.
unsigned char  password_req_bar_code:1;  //201  BIT      Password required with cashier bar-code
unsigned char  prt_cash_name_on_rcit:1;  //201  BIT      print cashier name on the receipt
unsigned char  no_check_digit         :1; //201  BIT      No check digit for customers accounts
unsigned char  mcard_during_trs       :1; //201  BIT      Allow member card during transaction
//                                          without hitting button or menu
unsigned char  CC_to_bypass_bday;        //202  NUM      CC to bypass birthday prompt
unsigned char  bday_val_print_option;    //203  NUM      Birthday validation print:
//                                          0-None
//                                          1-Receipt
//                                          2-Audit
```

```

//                                     //          3-Both

// The following structure defines
// restricted Sale/Birthday department
// sets
struct set_dep_                        //204  STRUCT
{
    unsigned short  dept_no;           //      NUM      Department number
    unsigned char   restrict_layout;   //      NUM      Layout number
    unsigned char   CC_restrict;       //      NUM      CC restricted sale
    unsigned char   age_req;           //      NUM      Age required
    unsigned char   CC_age_req;        //      NUM      CC birthday val.
}set_dep[10];

// The following structure defines
// valid sale times.
struct LAY_                            //264  STRUCT
{
    unsigned short  Mon_Thu_Start1_Time; //      TIME_I   Monday-Thursday start time 1
    unsigned short  Mon_Thu_Stop1_Time;  //      TIME_I   Monday-Thursday stop time 1
    unsigned short  Mon_Thu_Start2_Time;  //      TIME_I   Monday-Thursday start time 2
    unsigned short  Mon_Thu_Stop2_Time;   //      TIME_I   Monday-Thursday stop time 2
    unsigned short  Fri_Start1_Time;      //      TIME_I   Friday start time 1
    unsigned short  Fri_Stop1_Time;       //      TIME_I   Friday stop time 1
    unsigned short  Fri_Start2_Time;      //      TIME_I   Friday start time 2
    unsigned short  Fri_Stop2_Time;       //      TIME_I   Friday stop time 2
    unsigned short  Sat_Start1_Time;      //      TIME_I   Saturday start time 1
    unsigned short  Sat_Stop1_Time;       //      TIME_I   Saturday stop time 1
    unsigned short  Sat_Start2_Time;      //      TIME_I   Saturday start time 2
    unsigned short  Sat_Stop2_Time;       //      TIME_I   Saturday stop time 2
    unsigned short  Sun_Start1_Time;      //      TIME_I   Sunday start time 1
    unsigned short  Sun_Stop1_Time;       //      TIME_I   Sunday stop time 1
    unsigned short  Sun_Start2_Time;      //      TIME_I   Sunday start time 2
    unsigned short  Sun_Stop2_Time;       //      TIME_I   Sunday stop time 2
}lay[3];
```

```
struct                                //360  STRUCT  Reorganize Receipt options
{
    unsigned char buffer_printing      :1; //    BIT    Buffer printing
    unsigned char combine_sales        :1; //    BIT    Combine same items
    unsigned char sort_receipt         :1; //    BIT    Sort receipt
    unsigned char combine_cancel_subt  :1; //    BIT    Combine cancel/subtracts
    unsigned char devide_receipt_group:1; //    BIT    Not used
    unsigned char print_saving_today   :1; //    BIT    Print savings today
    unsigned char print_saving_allways:1; //    BIT    Print Savings even if buffer printing is off
    unsigned char combine_weight_items:1; //    BIT    Combine weight items

} org_recpt_opt1;

unsigned char stamp_00_02;             //361  NUM    Stamp for write validation
unsigned char org_recpt_main_sort;      //362  NUM    Main sort:
//                                     //          0-None
//                                     //          1-Department name
//                                     //          2-Department name + total
//                                     //          3-Item name
//                                     //          4-Price ascending
//                                     //          5-Price descending
//                                     //          6-Department number
unsigned char org_recpt_second_sort;    //363  NUM    Secondary sort:
//                                     //          0-None
//                                     //          1-Alpha
//                                     //          2-Price ascending
//                                     //          3-Price descending

// The following structure defines
// groups to be printed at end of ticket
// Group 0 - Items
// Group 1 - Departments
// Group 2 - Discounts
// Group 3 - Promotions
```

```
// Group 4 - Reductions
// Group 5 - Offers
// Group 6 - Coupons
// Group 7 - Refunds
// Group 8 - Age ID
// Group 9 - Freq. Shopper
// Group 10 - Cancel/Subtracts
```

```
struct org_recpt_group_ //364 STRUCT
```

```
{
    unsigned char priority      ; // NUM Priority
    unsigned char print_data    :1; // BIT Print Y/N
    unsigned char opt_nu        :7; // NU Not used
} org_recpt_group[20];
```

```
unsigned char prom_effect_date[3]; //404 BCD_D Promotion effective date
unsigned char multisaver_desc[20]; //407 ASCII Multisaver description
unsigned char ttl_multisaver_desc[20]; //427 ASCII Total multisaver description
unsigned char ttl_saving_desc[20]; //447 ASCII Total saving description
unsigned short multisaver_start_no; //467 NUM Multisaver start number
unsigned short link_save_start_no; //469 NUM Linksave start number
unsigned short ext_prom_start_no; //471 NUM Extended promotion start no.
unsigned short ext_link_save_start_no; //473 NUM Extended linksave start no.
unsigned short end_prom_no; //475 NUM End of promotion number
unsigned char consider_limit; //476 NUM
unsigned char host_batch_no; //478 NUM Next host batch number
unsigned char rounding_tender; //479 NUM Rounding tender
unsigned short wic_validity_period; //480 NUM WIC validation period
long cash_in_drawer_limit_to_raise_CC; //482 NUM_D Cash in drawer limit to raise CC
long qty_cc_qty; //486 NUM Quantity list
unsigned char CC_qty; //490 NUM CC for quantity limit
```

```
// The following structure defines
// valid sale days.
```



```
struct lay_days_set_ //491
{
    unsigned char sun:1; // BIT sun
    unsigned char mon:1; // BIT mon
    unsigned char tue:1; // BIT tue
    unsigned char wed:1; // BIT wed
    unsigned char thu:1; // BIT thu
    unsigned char fri:1; // BIT fri
    unsigned char sat:1; // BIT sat
    unsigned char nu :1;

}lay_days_set[3];

unsigned short member_points; //494 NUM member card points
long member_value; //496 NUM member card value
unsigned char opt_not_net_prom :1; //500 BIT Member promotion discounts are not netted
unsigned char opt_no_tax_rep :1; // No tax report
unsigned char plu_not_on_inquiry :1; // Print "PLU not found on inquiry" slip
unsigned char freq_shop_use_msu :1; // Freq Shop Discount to apply
// per item price (MSU)
unsigned char enhance_frq_shop_feature:1;// Enhanced frequent shopper feature in system
unsigned char use_customer_disc_scheme:1;// Use customers discount scheme
unsigned char no_per_item_cc_on_ngativ:1;// Suppress "per item CCs" for
// negative transaction on ticket total
unsigned char allow_non_whole_FS_EBT :1; // BIT Allow non whole dollar EBT food stamps
unsigned char CC_exceed_credit; //501 NUM CC to allow customer exceed credit limit
unsigned char loyalty_member_text [12]; //502 Loyalty member text prompt
unsigned char CC_drawer_insuff_funds; //514 NUM CC for insufficient funds in the drawer
unsigned short
    inclusive_tax_exempt_department; //515 inclusive tax exempt department
unsigned char costumer_id_during_trans:1;//517 prompt for customer id on save transactions
unsigned char bad_account_in_post :1;// bad account file is resident on post.
unsigned char freq_shop_begin_ticket :1;// Do not prompt for freq shopper @ beg of ticket
unsigned char apply_freq_shop_disc :1;// Apply freq Shopper discount also with any
```

```

// manual price entries
unsigned char auto_acpt_cmbine_card :1;// Automatically accept combine card as tender
unsigned char print_chnge_on_receipt :1;// Print change at the end of Receipt
unsigned char premium_freq_shopper :1;// Premium (Enhanced+) Frequent Shopper
unsigned char print_freq_shop_disc_total :1;// BIT Print Freq Shop disc.total only (no details)
unsigned char CC_bypass_cshr_not_fnd; //518 NUM CC to bypass 'cashier not found'
// during offline sign-on.
unsigned char stamp_00_03; //519 NUM Stamp for write validation
//unsigned char add_data_for_depart[2]; //520 BCD Add. data capture required for department
unsigned char report_member_prom_as_tender:1; //520 used by V8
unsigned char member_name_from_corema :1; // Show member name received from COREMA
unsigned char use_freq_shopper_dummy :1; // 520 Use Frequent Shopper dummy card
unsigned char freq_shop_dummy_automatic:1; // Insert the dummy card auto. into all tickets
unsigned char member_prom_affect_dept :1; // TBA #2080 Member promotion affects department promos
unsigned char print_all_offer_progress :1; // Print offer progress on receipt even if
// promotion was not triggered y/n
unsigned char forgive_fs_tax_by_highest_rate :1; //
unsigned char nu520 :1;

unsigned char nu; //521 not used
unsigned char CC_15_sku; //522 NUM Override mandatory entry of 15 digits SKU
unsigned char split_reward :1; //523 BIT Split reward amount between a promotion's item
unsigned char eft_opt_01 :1; //523 BIT NU
unsigned char eft_opt_02 :1; //523 BIT NU
unsigned char eft_opt_03 :1; //523 BIT NU
unsigned char eft_opt_04 :1; //523 BIT NU
unsigned char eft_opt_05 :1; //523 BIT NU
unsigned char eft_opt_06 :1; //523 BIT NU
unsigned char eft_opt_07 :1; //523 BIT NU
unsigned char eft_window_start[2]; //524 BCD Window start time
unsigned char eft_window_end[2]; //526 BCD Window end time
unsigned char cc_max_cnlsb_itme_lev2; //528 NUM CC for Max item value cancel /subtract Level-2
unsigned char cc_max_cnlsb_itme_lev3; //529 NUM CC for Max item value cancel /subtract Level-3
unsigned char cc_max_cnlsb_val_per_lev2;//530 NUM CC for Max cancel/subtract

```

```

// value per ticket Level-2
unsigned char cc_max_cnlsb_val_per_lev3;//531 NUM CC for Max cancel/subtract
// value per ticket Level-3
unsigned char cc_max_cnlsb_items_lev2; //532 NUM CC for Max cancel/subtract
// items per ticket Level-2
unsigned char cc_max_cnlsb_items_lev3; //533 NUM CC for Max cancel/subtract
// items per ticket Level-3
long max_val_item_cncelsub_lev2; //534 Max item value cancel/ subtract Level-2
long max_val_item_cncelsub_lev3; //538 Max item value cancel/ subtract Level-3
long max_cncelsub_val_per_tick_lev2; //542 Max cancel/subtract value per ticket Level-2
long max_cncelsub_val_per_tick_lev3; //546 Max cancel/subtract value per ticket Level-3
unsigned short
    max_cncelsub_items_in_tick_lev2;//550 Max cancel/subtract items per ticket - Level-2
unsigned short
    max_cncelsub_items_in_tick_lev3;//552 Max cancel/subtract items per ticket - Level-3
unsigned char CC_bypass_cash_not_found; //554 NUM CC to bypass "cashier not found" at sign
// on at OFFLINE
unsigned char cc_exceed_qtylimit_scan; //555 NUM CC for exceeding qty limit at scanning
long scanning_qty_limit; //556 NUM CC for Scanning qty limit
unsigned char cc_exc_amnt_neg_trans; //560 NUM CC for exceeding max. amount in neg. trans.
long max_amnt_neg_trans; //561 NUM CC for Max. amount for negative transactions
long foodstmp_in_drawer_limit; //565 NUM_D Food Stamps in drawer limit
unsigned char cc_offline_save_recall; //569 NUM CC for Save/Recall ticket in off-line mode
unsigned char cc_max_return_itme_lev1; //570 NUM CC for Max item value return Level-1
unsigned char cc_max_return_itme_lev2; //571 NUM CC for Max item value return Level-2
unsigned char cc_max_return_itme_lev3; //572 NUM CC for Max item value return Level-3
unsigned char cc_max_return_val_per_lev1;//573 NUM CC for Max return value per ticket Level-1
unsigned char cc_max_return_val_per_lev2;//574 NUM CC for Max return value per ticket Level-2
unsigned char cc_max_return_val_per_lev3;//575 NUM CC for Max return value per ticket Level-3
unsigned char cc_max_return_items_lev1; //576 NUM CC for Max return items per ticket Level-1
unsigned char cc_max_return_items_lev2; //577 NUM CC for Max return items per ticket Level-2
unsigned char cc_max_return_items_lev3; //578 NUM CC for Max return items per ticket Level-3
long max_val_item_return_lev1; //579 Max item value return Level-1
long max_val_item_return_lev2; //583 Max item value return Level-2
```

```

long      max_val_item_return_lev3;      //587      Max item value return Level-3
long      max_return_val_per_tick_lev1;  //591      Max return value per ticket Level-1
long      max_return_val_per_tick_lev2;  //595      Max return value per ticket Level-2
long      max_return_val_per_tick_lev3;  //599      Max return value per ticket Level-3
unsigned short
        max_return_items_in_tick_lev1;  //603      Max return items per ticket Level-1
unsigned short
        max_return_items_in_tick_lev2;  //605      Max return items per ticket Level-2
unsigned short
        max_return_items_in_tick_lev3;  //607      Max return items per ticket Level-3
unsigned char cc_post_voids_lev1;        //609      NUM      CC for Post voids
long      val_post_voids_lev1;          //610      Value for Post voids
struct                                         //614      BITS
{
    unsigned char  transaction_voids  :1; //      BIT      Display "reason table" on Transaction Voids
    unsigned char  post_voids         :1; //      BIT      Display "reason table" on POST Voids
    unsigned char  transaction_returns:1; //      BIT      Display "reason table" on Transaction Returns
    unsigned char  cancel              :1; //      BIT      Display "reason table" on Cancel
    unsigned char  subtracts          :1; //      BIT      Display "reason table" on Subtracts
    unsigned char  merchandise_returns:1; //      BIT      Display "reason table" on Merchandise Returns
    unsigned char  nu                 :2; //      BIT      not used
} reason_table_;

unsigned short active_cashier;             //615      NUM      use active cashier list
unsigned long min_change_amount;           //617      NU      Minimum change amount to print credit notes
unsigned char CC_for_min_change_amount;    //621      NU      CC if change amount is less then minimum
//                                         change amount to print credit notes
long      total_stub_amt_level1;          //622      NUM      min amount for total stub printing (level 1)
unsigned char prom_num_stub_level1[2];    //626      BCD      promotion number for level 1
long      total_stub_amt_level2;          //628      NUM      min amount for total stub printing (level 2)
unsigned char prom_num_stub_level2[2];    //632      BCD      promotion number for level 2
long      total_stub_amt_level3;          //634      NUM      min amount for total stub printing (level 2)
unsigned char prom_num_stub_level3[2];    //638      BCD      promotion number for level 3
unsigned char cc_post_voids_lev2;         //640      NUM      CC for Post voids

```

```

long          val_post_voids_lev2;          //641      Value for Post voids
unsigned char cc_post_voids_lev3;           //645      NUM      CC for Post voids
long          val_post_voids_lev3;          //646      Value for Post voids
unsigned char opt_log_related :1;           //650      BIT      Log related account number
unsigned char opt_print_related :1;         //650      BIT      print related account on receipt
unsigned char mcard_on_cash_disp :1;        //650      BIT      Display member card desc on cashier display
unsigned char mcard_on_cust_disp :1;        //650      BIT      Display member card desc on
unsigned char upc99_vndr_cpn_in_sys :1;     //          UPC99 Vendor Coupon in system
unsigned char print_price_in_group :1;      //650      BIT      Don't print price in PLU Group List
unsigned char freq_shop_receipt_tmpl:1;    //650      BIT      F.S. receipt template same as member promotion
unsigned char check_ID_num_in_bad_acct:1;   //650      BIT      Check for second ID number in bad account
unsigned char related_acct_tmpl[20];        //651      ASCII    Related account template
unsigned char cc_valid_scanned_upc21;       //671          CC for validate scanned UPC21
unsigned char euro_rate[8];                 //672          Euro rate

struct                                     //680      STRUCT
{
    unsigned char manual_weight :1; //      BIT      Allow manual weight
    unsigned char cashier_age_verify :1; //          Cashier age verification on restricted sales
                                     //          in use
    unsigned char allow_delay_verify :1; //          Allow delayed Age verification in system
    unsigned char opt_omit_cnfrm_manual_sec :1; //BIT      Omit confirmation on manual secure mode
    unsigned char verify_weight_before_sale_execution :1; //BIT
    unsigned char do_not_display_expiration_date_on_rcpt :1; //BIT
    unsigned char GetSecIdFromPinpad :1; //BIT      Prompt for member secondary ID at the pin pad
    unsigned char nul :1;
} flags5;

unsigned char print_ttl_itm_on_receipt;      //681          How to print Total items count on receipt
                                     //          (for WinPOS. DOS has it as a default).
                                     //          0 = Default
                                     //          1 = Balance
                                     //          2 = Separate
unsigned char nu2[1];                        //682          Reserved - Used In Postype.h

```

```

unsigned char receipt_line_spacing;    //683      Receipt line spacing
                                         //        0 = Normal
                                         //        1 = Reduced
                                         //        2 = Compact

unsigned char cc_save_trs_log_off;      //684 NUM    CC for log off with saved TRS
unsigned char max_item_in_exp_ticket;   //685      Maximum item in express ticket
unsigned char stamp_00_04;             //686 NUM    Stamp for write validation
unsigned char mcard_name_in_sys[12];    //687 ASCII  Member Card Name in system
unsigned char sec_id_num_length;        //699 NUM    Length of secondary ID number
unsigned char gold_card_tndr_num;       //700      tender # for gold card (BLACK BOX)
unsigned char silver_card_tndr_num;     //701      tender # for silver card (BLACK BOX)
unsigned char coupon_card_print_msg[37]; //702      Coupon card printout message
long          max_neg_ttl_in_order;     //739      Albertson's 30+
unsigned char cc_max_neg_in_order;      //743      Albertson's 30+
long          max_neg_percent_of_order; //744      Albertson's 30+
unsigned char cc_max_neg_percent_order; //748      Albertson's 30+
unsigned char name_of_self_scan_syst[20]; //749 ASCII  Name of Self-Scanning system
short        self_scan_nt_pos_num;      //769 NUM    Self Scanning: NT POS number( 1-255)
struct                                              //771
{
    unsigned char self_scan_debug_mode:1; //          Self Scanning: Debug mode(Y/N)
    unsigned char nu:7;
} self_scan_flags;

unsigned char cng_mnum_text_on_ticket[12]; //772 ASCII  Change Member 'CARD NO' text printed on
                                         //          ticket
unsigned char cng_mname_text_on_ticket[12]; //784 ASCII  Change 'MEMBER NAME' text printed on ticket
unsigned char CC_request_approval_T7_off;   //796 NUM    CC for Request for Approval when T-7 is
                                         //          Offline
unsigned char CC_request_void_T7_off;       //797 NUM    CC for Request for Void when T-7 is Off-line
unsigned char CC_request_approval_bank_off; //798 NUM    CC for Request for Approval when bank is
                                         //          Offline
unsigned char CC_request_void_bank_off;     //799 NUM    CC for Request for Void when bank is Off-line
unsigned char CC_void_without_invoice_num;  //800 NUM    CC for Voiding without invoice number

```

```
unsigned char CC_void_non_EDC_cards;          //801 NUM    CC for Voiding non-EDC cards
unsigned char discounts_description[20];       //802 ASCII   Discounts description for "Your saving today"
unsigned char promotions_description[20];      //822 ASCII   Promos description for "Your saving today"
unsigned char reductions_description[20];      //842 ASCII   Reductions desc. for "Your saving today"
unsigned char offers_description[20];          //862 ASCII   Offers description for "Your saving today"
unsigned char template_on_receipt_end[10];     //882        Name of additional template printed at end
                                                //          of receipt.

// The following structure is an additional
// restricted Sale/Birthday.
// CC for Cashier restriction
struct set_dep_1                               //892  STRUCT
{
    unsigned char CC_restrict_cashier;         //    NUM    CC restricted sale for Cashier
    unsigned char cashier_age;                 //    NUM    Cashier required age
}set_dep1[10];

unsigned char freq_shop_dummy_card[7];        //912  BCD    Frequent Shopper dummy card number
unsigned char WIC_date;                       //919  NUM    WIC date:
                                                //          0- issue date
                                                //          1- expiry date
                                                //          2- both
unsigned char cc_exc_item_refund_trans;        //920  NUM    CC for exceeding max. amount
                                                //          in item refund transactions
long          max_amnt_item_refund;            //921  NUM    max. amount in item refund transactions
unsigned char cc_exc_order_refund_trans;       //925  NUM    CC for exceeding max. amount
                                                //          in order refund transactions
long          max_amnt_order_refund;           //926  NUM    max. amount in order refund transactions
unsigned char tare_weight_on_rcpt;             //930        Print tare weight on receipt
                                                //          0 - No: No Tare Weight is printed.
                                                //          1 - Print all tares.
                                                //          2 - Print manual tares: Only the manually
                                                //          entered tares are printed.
unsigned short print_tpl_every_X_tickets;     //931  NUM    Print slip after every X tickets (0 - 9999)
```

```

unsigned char cust_slip_tpl_every_X_tickets[10]; //933 ASCII Customer Slip template name
unsigned char MLime_tender_no; //943 NUM Mobile Lime - tender number
unsigned char MLime_discount_tender_no; //944 NUM Mobile Lime - tender number
unsigned char MLime_timeout; //945 NUM Mobile Lime - transaction timeout 0-90
unsigned char MLime_prefix_for_term_id[7]; //946 BCD Mobile Lime - prefix of Terminal ID number
unsigned char MLime_discount_no; //953 NUM Mobile Lime - discount number
//TBA - sig tenders
unsigned char signature_tender1; //954 NUM Signature Tenders 1
unsigned char signature_tender2; //955 NUM Signature Tenders 2
unsigned char signature_tender3; //956 NUM Signature Tenders 3

//ECC options
unsigned char manual_check_ecc_tender_no; //957
unsigned char ecc_tender_no1; //958
unsigned char ecc_tender_no2; //959
unsigned char ecc_tender_no3; //960
unsigned char electronic_check_message_template[10]; //961

unsigned char ForceSupervisorIDbarcode :1; //971
unsigned char MaskSupervisorIDandPassword :1; //971
unsigned char PinPadCustDisp :1; //971
unsigned char DispSigCustDisp :1; //971
unsigned char PromptFreqShopUscan :1; //971
unsigned char notu4 :3; //971

//Biometric Opt
struct
{
unsigned char PBTInSystem :1; //972 BIT
unsigned char DisplayLoyaltyMmbrNameFromPBT :1; //972 BIT
unsigned char GetDOBFromPBT :1; //972 BIT
unsigned char GetLoyaltyFromPBT :1; //972 BIT
unsigned char DisplayEFTIconOnCashierDisplay :1; //972 BIT
unsigned char customized_EFT_messaged_on_cashier_display :1; //972 BIT

```



```
unsigned char nu_972                                     :2; //972 BIT not used

} biometric_flags;

unsigned char ACH_tender_no;                             //973 NUM
struct
{
unsigned char MLime_PromptLoyaltyAtStartTicket          :1; //974 BIT
unsigned char MLime_nu7                                  :7; //974

} Mlime_bits;
long min_amt_trans_in_customer_count;                   //975 Bug

//EBT WIC params
unsigned char bEBTWICPrintAvailableBenefitsOnly        :1; //979 EBT WIC params
unsigned char nu_EBTWIC_bits                             :7; //979 EBT WIC params
unsigned char uchEBTWICTenderNumber;                    //980 EBT WIC params

//MTX
unsigned char MTXReceiptPrinting                         :1; //981 TBA - MTX receipt printing
unsigned char MTXSuppressStoreCopy                       :2; //981 Suppress Store Copy.
unsigned char nu_b7                                       :6; //981 nu
unsigned char CC_AllowEBTWICMixedBasket; //982

//<WIC CVV Options> // - WIC CVV [EL1G]
unsigned char WIC_CVV_display_options; //983 NUM WIC CVV display options:
// 0-No extra prompt
// 1-Prompt for amount
// 2-Prompt for expiry date
// 3-Prompt for amount & expiry
// date
unsigned char WIC_CVV_print_option; //984 NUM WIC CVV print options
// 0-No extra print
// 1-Print amount
```

```

//          2-Print expiry date
//          3-Print amount & expiry date
unsigned char CC_exceed_WIC_CVV_amt; //985 NUM CC to exceed WIC CVV check amt.
unsigned char CC_exceed_WIC_CVV_date; //986 NUM CC to exceed WIC CVV expiry date
unsigned char CC_bypass_WIC_CVV_amt; //987 NUM CC to bypass WIC CVV check amt.
unsigned char CC_bypass_WIC_CVV_date; //988 NUM CC to exceed WIC CVV expiry date
unsigned char WIC_CVV_date; //989 NUM WIC CVV date:
//          0- issue date
//          1- expiry date
//          2- both

unsigned short WIC_CVV_validity_period; //990 NUM WIC CVV validation period
//<WIC CVV Options> - WIC CVV [EL1G]

//<WIC CVV Options> - WIC CVV [EL1G]
struct
{
    unsigned char AllowWicCvvInaMixBasketTrans :1; // "Allow WIC CVV in a Mixed Basket Transaction"
// (Default 0)
    unsigned char AllowSplitTenderInWicCvvTrans:1; // "Allow Split Tendering in a WIC CVV transaction"
// (Default 0)
    unsigned char DisplayWicCvvEligibleTotal :1; // "Display WIC CVV Eligible Total on Cashier Display"
// (Default 0)
    unsigned char unused :5; //Not used
}WIC_CVV_flags; //992 BITS
short MaximumNumOfWicCvvVouchers; //993 NUM "Maximum Number of WIC Vouchers" (Default 0)
//</WIC CVV Options> // WIC CVV [EL1G]
unsigned char nul9[3]; //995 Not used
unsigned char stamp_00_05; //998 NUM Stamp for write validation
unsigned char total_message :1; //999 BIT "Frequent shopper?" appear at "Total" time
unsigned char display_qty :1; //999 BIT Display quantity of similar items sold
// simultaneously.
unsigned char display_numeric :1; //999 BIT Display numeric data on customers display

```

```

//          prior to entry key.
unsigned char dont_consolidate :1; //999 BIT Don't consolidate weighted items in buffer
//          printing.
unsigned char display_frequent :1; //999 BIT Display frequent shopper's price on inquiry.
unsigned char display_doc_prm :1; //999 BIT Display document printing message on E.J.
unsigned char round_unround_disp :1; //999 BIT Display rounded and UN-rounded
//          balance due on screen.
unsigned char no_print_on_barcode:1; //999 BIT Do not print weight and price
//          per Kilo on 02 bar-code
unsigned char redamp_end_date[3]; //1000 BCD_D Redemption end date
unsigned char pend_redamp_stat_date[3]; //1003 BCD_D Redemption start date
unsigned char redamp_tender[2]; //1006 BCD Redemption tender
unsigned char redamp_option; //1008 NUM Redemption option:
//          0-None
//          1-Use all redemption amount
//          2-Use partial redemption am.
unsigned char qf_option; //1009 NUM Qualified spend option:
//          0-Print QF for Clubcard /
//          member card customers only
//          1-Print QF on all receipts
//          2-As option 1 + Clubcard /
//          member card points message
//          3-As option 2 + Clubcard /
//          member card points message
//          4-Print only total points
struct //1010 BITS
{
    unsigned char mcard_on_sys :1; // BIT Member card on system
    unsigned char use_card_scheme :1; // BIT Use card scheme in printing member savings
    unsigned char suppress_print :1; // BIT Suppress printing points on receipt
    unsigned char bal_due_aftr_tndr:1; // BIT Display Bal. Due after each tender
    unsigned char prn_coupon_as_neg:1; // BIT Print Coupon as negative
    unsigned char prn_balance_due :1; // BIT Print Balance Due each time the media change
    unsigned char use_fsn_as_mcn :1; // BIT Use freq shopper no. as member card number

```

```

    unsigned char  cst_disp_at_tm    :1;  //    BIT    Enable customer display at training mode
} flags2;
unsigned char mcr_timeout;           //1011 NUM
unsigned char times_to_swipe;        //1012 NUM    Times to swipe card before manual entry
unsigned char force_zeros;           //1013 NUM    Force zeros on coupon number
unsigned char opt_use_pos_reason :1;  //1014 BIT    Use action reasons feature
unsigned char disallow_qty_key   :1;  //1014 BIT    Disallow the QTY key to be
                                   //              used twice in one transaction
unsigned char wic_tndr_wout_valid:1;  //1014 BIT    Allow WIC tender without validation
unsigned char coupon_card_in_use :1;  //1014 BIT    Coupon Card in use
unsigned char valid_scanned_upc21 :1;  //1014 BIT    Check validity of scanned non
                                   //              weighed price Embedded items
unsigned char freq_shop_disc_not_net :1; //1014 BIT    Freq. Shop. discount are not netted
unsigned char disp_cust_acct_blk_msg :1; //1014 BIT    Display Customer Account Black List Message
unsigned char bEBTWICinSystem      :1; //1014 BIT    EBT WIC in system
unsigned char CC_cash_in_drawer_limit; //1015 NUM    CC for cash in drawer limit
unsigned char
    delay_drawer_open_alarm_after_sign_off; //1016 NUM    Delay before open drawer alarm after sign off
unsigned char sales_prn_sound;           //1017 NUM    Sales printer sound:
                                   //              0-None
                                   //              1-Buffer printing
                                   //              2-Always
short          adjustment_fact_1;        //1018 NUM    Productivity report factor 1
short          adjustment_fact_2;        //1020 NUM    Productivity report factor 2
unsigned char  second_media_po_roa;      //1022 NUM    Second tender to Payout/ROA
struct y_eod_bits_                       //1023 STRUCT only in BO
{
    unsigned char yesterday_was_eow :1;  //    bit    yesterday was end of week
    unsigned char yesterday_was_eom :1;  //    bit    yesterday was end of month
    unsigned char yesterday_was_eoy :1;  //    bit    yesterday was end of year
    unsigned char                :5;  //
}y_eod_bits;

                                   //1024

```

```
//*****
// RECORD 1
//*****

// Start of Quick area
struct pos_param_Quick_
{
    unsigned char Q_running_total      :1; //0    BIT    Display running total
    unsigned char Q_pre_auth_req       :1; //0    BIT    Pre-authorization required
    unsigned char Q_clubcard_auth      :1; //0    BIT    Clubcard authorization req.
    unsigned char Q_secure              :1; //0    BIT    Security check
    unsigned char Q_coupon_accept      :1; //0    BIT    Coupon processing
    unsigned char Q_sig_override_crng:1; //0    BIT    Signature required to override card range
    unsigned char Q_opt01_nu           :2; //0    BIT    Not used
    unsigned char Q_key_barcode         :1; //1    BIT    Key entry of item barcode
    unsigned char Q_key_credit          :1; //1    BIT    Key entry of credit card
    unsigned char Q_key_debit           :1; //1    BIT    Key entry of debit card
    unsigned char Q_key_club            :1; //1    BIT    Key entry of clubcard
    unsigned char Q_key_coupon_barcode:1; //1    BIT    Key entry of coupon barcode
    unsigned char Q_key_dept            :1; //1    BIT    Key entry of department
    unsigned char Q_opt02_nu           :2; //1    BIT    Not used
    unsigned char Q_help_timeout ;      //2    NUM    Help screen timeout
    unsigned char Q_supervisor_timeout; //3    NUM    SVR screen timeout
    unsigned short Q_trolleys;          //4    NUM    Number of trolley devices
    unsigned short Q_abandon_timeout;   //6    NUM    Abandon trolley timeout
    unsigned char Q_card_failure_alert; //8    NUM    Number of card read failures
                                   //      prior to raising an alert
    unsigned short Q_not_recalled;      //9    NUM    Minutes before raising a "Not recalled" alert
    long          Q_pre_auth_amount;    //11   NUM_D   Pre-authorized amount
    unsigned char Q_nu01[71];           //15   NU      Not used
    unsigned char stamp_01_01;          //86   NUM     Stamp for write validation
} Quick;

// End of Quick area
```

```

unsigned char round_table[10];           //87  NUM    Round-to parameters
struct
{
    unsigned char round_at_total      :1; //97  BIT    Rounding on total
    unsigned char round_sign_reverse :1; //97  BIT    Reverse signs on Rounding tender ticket
    unsigned char nu                   :6; //97  NU
}rounding_param;

//FSA
struct
{
    unsigned char OptEnableFsaItemSubstantiation :1; //98 FSA
    unsigned char nu98                           :7; //98 not used
} fsa_flags;
unsigned char RxPrintChar;                //99  FSA
unsigned char NonRxPrintChar;             //100 FSA
unsigned char FsaTotalsTemplate[10];      //101 FSA
unsigned char FsaDisclaimerTemplate[10];  //111 FSA
unsigned char filler14[44];               //121 NU    Not used
unsigned char stamp_01_02;                //165 NUM    Stamp for write validation
unsigned char sign_currency2;              //166 ASCII  2nd Currency character
unsigned char sign_currency;              //167 ASCII  Currency sign
unsigned char sign_minus;                  //168 ASCII  Minus sign
unsigned char sign_decimal;                //169 ASCII  Decimal sign
unsigned char sign_format;                 //170 ASCII  Format sign
unsigned char sign_non_taxable;            //171 ASCII  Non-taxable sign
unsigned char sign_multi_tax;              //172 ASCII  Multi tax sign
unsigned char sign_food_stamp;             //173 ASCII  Food stamp sign
unsigned char sign_non_food_stamp;         //174 ASCII  Non food stamp sign
unsigned char cost_plus_sign[2];           //175 ASCII  Cost plus sign
unsigned char non_cost_plu_sign[2];        //177 ASCII  Non cost plus sign
unsigned char freq_shopper_sign[2];        //179 ASCII  Frequent shopper sign
unsigned char non_freq_shopper_sign[2];    //181 ASCII  Non frequent shopper sign
unsigned char sign_reserved[4];            //183 NU    Reserved

```

```
unsigned short tare_weight_table[30];    //187  NUM    Tare weight parameters
unsigned short tare_weight_percent[30];  //247  NUM    Tare weight percents

struct coupon_values_code_table_        //307  STRUCT  Coupon values code table
{
    unsigned char  type;                  //    NUM    Type
    unsigned char  buy_qty;               //    NUM    Buy quantity
    unsigned short get_value;             //    NUM_D   Get value
    char           filler[2];             //    NU     Not used
} coupon_value[100];

char print_on_check_dollar[10];          //907  ASCII  print currency on check dollar
char print_on_check_cents[10];           //917  ASCII  print currency on check cents
struct
{
    unsigned char in_system              :1; //927
    unsigned char calc_bef_vender_cpn   :1; //927
    unsigned char calc_bef_bonus_cpn    :1; //927
    unsigned char calc_bef_store_cpn    :1; //927
    unsigned char no_mssg_cshr_screen   :1; //          Display message on operator screen
    unsigned char points_with_decimal   :1; //          Display/Print points with decimal point
    unsigned char un11                  :2; //927
    long          value_of_one;          //928
    char          template_id[10];       //932
    unsigned char calc_method;           //942
    unsigned char filler[19];            //943

}trading_stamps;

long  express_items;                     //962  NUM    Number of items in express orders
long  idle_limit;                        //966  NUM    Multiply item time by this
                                           //          parameter to limit idle time

unsigned char opt_card_prog   :1;        //970  BIT    Community partner in system
unsigned char opt_nul         :7;        //970  BIT    NU
unsigned char card_prog_UPC;   //971  NUM    Card program UPC barcode
```

```

unsigned char card_prog_loc;           //972  NUM    location of prog. in card
unsigned char stamp_01_03;           //973  NUM    Stamp for write validation

//CCTV
Struct                               //974
{
unsigned char UseTcpDirect           :1;    //
unsigned char nu_975                 :7;    //

} CCTV_bits;

unsigned char TcpDirectHostName[19];  //975
unsigned short TcpDirectPort;         //994

unsigned char bLaneHawkInSystem :1;    //996
unsigned char notused             :7;    //996
unsigned short ushLaneHawkPort;      //997
unsigned short ushLaneHawkHBIntervalInSec; //999
unsigned short ushLaneHawkTimeoutInSec; //1001
unsigned char uchLaneHawkHostsAlias[16]; //1003
unsigned char filler18[4];           //1019  NU    Not used
char qdx_flag1;                     //1023  NU    Not used

//*****
// RECORD 2
//*****

unsigned char retailer_info[40][25]; //0    ASCII  Retailer information
unsigned char stamp_02_01;           //1000  NUM    Stamp for write validation
unsigned char filler17[22];          //1001  NU    Not used
char qdx_flag2;

//*****
// RECORDS 3,4,5

```



```
//*****
struct fuel_cards_item_groups_          //0    STRUCT   Fuel card item groups
{
    unsigned char  opt1;                  //    NUM     Option 1
    unsigned char  opt2;                  //    NUM     Option 2
    unsigned short dep[20];               //    NUM     Departments list
    char           filler[6];             //    NU      Not used
} fuel_entries[32];

struct fuel_type_item_                  //1536 STRUCT   Fuel type items
{
    struct fuel_dep_plu_
    {
        unsigned short dept;
        unsigned char  plu[7];
    } data[5];
    unsigned char  desc[10];
    char           filler;
} fuel_sale_type[16];
// 16 entries used, last 16 will probably
// will be used
struct fuel_tank_                       //2432
{
    unsigned char grade;
    unsigned char valid[8];
    unsigned char nu[7];
} fuel_tank[32];

unsigned short user_sign_on[10];         //2944 NUM     user signed on at each PC
unsigned char stamp_05_01;               //2964 NUM     Stamp for write validation
unsigned char nu02[106];                 //2965 NU      Not used
unsigned char qdx_flag5;                 //3071 NU      Not used

//*****
```

```
// RECORD 6
//*****
struct pos_menu_          //0    STRUCT
{
    unsigned char nu;      //    NU    Not used
    unsigned short func;   //    NUM    Keyboard function
    unsigned char key;     //    NUM    Keylock
    unsigned char privilege; //    NUM    Privilege
}
idle_menu[50],            //0    STRUCT  Idle mode menu
no_sale_menu[50];         //250  STRUCT  No sale menu
struct cost_plus_table_   //500  STRUCT  Cost plus table
{
    long    amount1;       //    NUM_D   Amount
    long    percent;       //    NUM_D   Percentage
} cost_plus[20];

struct xyear_weeks_       //580  STRUCT  Weeks in months (for DSA)
{
    char    start_date[3]; //    BCD_D   Year start date
    char    end_date[3];   //    BCD_D   Year end date
    char    mon_weeks[12]; //    NUM     weeks in month
    char    nu[6];         //    NU     Not used
} year_weeks[15];         //
unsigned char stamp_06_01; //    NUM     Stamp for write validation
char filler16[3];         //    NU     Not used

//*****
// RECORD 7
//*****

struct                    //0    STRUCT  Customer display messages:
                        //
                        //    Lane closed
{
```

```
    unsigned char len;                //    NUM    Message length
    unsigned char text[49];           //    ASCII  Message text
} cust_disp_pos_close[4];           //

struct                               //200  STRUCT  Customer display messages:
{                                   //
    unsigned char len;
    unsigned char text[49];           //    NUM    Message length
} cust_disp_next_cust[4];           //    ASCII  Message text
//

struct cdv_scheme_                   //400  STRUCT  CDV Schemes
{
    unsigned char desc[8];            //    ASCII  Description
    unsigned char nu1;                //    NU     Not used
    unsigned char weights[11];        //    NUM    Weights
    unsigned char mod;                //    NUM    Mod (divided by)
    unsigned char rem;                //    NUM    Reminder
    unsigned char nu2[3];             //    NU     Not used
} cdv_scheme[16];

struct pos_menu_ supervisor[50];     //800  STRUCT  Supervisor menu
struct pos_menu_ temp_stat_2[50];    //1050 STRUCT  Reserved
struct pos_menu_ temp_stat_3[50];    //1300 STRUCT  Reserved

// 5 Check fee tables.
struct check_fee_rec_                //1350 STRUCT  Check Fee tables
{
    long          from_amount;        //    NUM_D   From amount
    long          to_amount;          //    NUM_D   To amount

    union
    {
        unsigned char plu_no[7];     //    BCD     PLU Number
```

```

struct
{
    unsigned long percent;
    unsigned short dep_num;
    unsigned char nu;          // NU      Not used
}p;
}u;
unsigned char type;          // 0-plu 1-percent+dep

} check_fee[10][5];

unsigned char int_use[127];   //2150 NU      Reserved
unsigned char stamp_09_01;    // NUM      Stamp for write validation

struct sub_tender_           //2278
{
    unsigned char line[20];

}sub_tender[20];             // ASCII Gift Certificate

// The following structure defines
// valid date lay.
struct date_link_to_lay_     //2678 STRUCT
{
    Unsigned char date[3];    // BCD_D

    unsigned short Start_Time1; // TIME_I start time 1
    unsigned short Stop_Time1 ; // TIME_I stop time 1
    unsigned short Start_Time2; // TIME_I start time 2
    unsigned short Stop_Time2 ; // TIME_I stop time 2

    struct link_to_lay_

```

[illegible]

```

// options : never=0 , members=1 , always=2
long minimum_amount_for_printing; //3 NUM Minimum amount for printing signature receipt
unsigned char prompt_for_signature :1; //7 BIT When printing signature receipt , prompt the
// cashier that signature is required
unsigned char corema_non_loyalty :1; // for sending Corema only message
unsigned char opt_barcoded_tender :1; // COINSTAR in use
unsigned char opt_not_net_freq_shop :1; // Non-Netted promotions/freq shop
unsigned char gift_card_in_system :1; // Gift Card in system
unsigned char allow_offline_gift_card :1; // offline gift card activation is allowed
unsigned char GiftCardActivationAfterTender :1; // Gift Card Activation After Tender
unsigned char allow_auto_tender_of_available_balance :1; // allow auto tender of available balance for
// Gift Card

unsigned short fuel_roa_dept_code; //8 NUM PosPump: Balance Pre-Pay
unsigned char member_card_cd_formula; //10 NUM 0=UPC, 1=account
unsigned char enable_double_coupons; //11 NUM Enable double coupons Options: 0=all,
// 1=members
// 2=non members
// 3=disable all
unsigned char enable_triple_coupons; //12 NUM Enable triple coupons Options: 0=all
// 1=members
// 2=non members
// 3=disable all

unsigned short max_triples_per_ticket; //13 NUM Maximum triples per ticket
unsigned char opt_UPC5_desc_item :1; //15 BIT Opt UPC5 desc from item
unsigned char opt_baldue_enl_vert_f :1; // Opt balance due enlarged vertical fonts
unsigned char opt_sav_omit_cpn_ttl :1; // Opt savings omit cpn total
unsigned char opt_sav_detail_list :1; // Opt savings detail list
unsigned char opt_sav_omit_dtl_ttl :1; // Opt savings omit detail total
unsigned char opt_sav_omit_sumry_ttl :1; // Opt savings omit summary total
unsigned char opt_sav_present_as_ng :1; // Opt savings present as negative
unsigned char opt_savings_marke_msg :1; // Opt savings marketing message
unsigned char opt_omit_tax_report_ttl :1; //16 BIT Opt omit tax report total
unsigned char opt_omit_saving_rpt :1; // Opt omit saving report
unsigned char opt_not_used_rec14a :1; // Opt nu

```

unsigned char opt_lmt_gft_crd_EAN128 :1; //	Opt Limit Gift Card Entry for EAN-128
unsigned char opt_not_used8888 :4; //	Opt nu
unsigned char tax_report_hdr_text[40]; //17	Tax report header text
unsigned char sav_hdr_text[30]; //57	Savings header text
unsigned long sav_marketing_percent; //87	Savings marketing percent
unsigned char sav_marketing_tmpt1[10]; //91	Calc above percent
unsigned char sav_marketing_tmpt2[10]; //101	Calc below percent
unsigned char mmbr_card_template[10]; //111	Member card template
unsigned char mmbr_prom_desc[30]; //121	Member prom desc
unsigned char vendor_coupon_desc[30]; //151	Vendor coupon desc
unsigned char store_coupon_desc[30]; //181	Store coupon desc
unsigned char bonus_coupon_desc[30]; //211	Bonus coupon desc
unsigned char triple_coupon_desc[30]; //241	Triple coupon desc
unsigned char RR_group_all_coupons :1; //271 BIT	RR group all coupons
unsigned char RR_tax_rpt_bef_bal_due:1; //	RR tax report before bal due
unsigned char RR_net_price_on_system:1; //	RR net price on system
unsigned char RR_group_complex_prom :1; //	RR group complex promotions
unsigned char RR_linked_item_follow_the_triger_item :1; //	
unsigned char opt_print_tax_line_with_signs_off :1; //	
unsigned char DoNotMaskEFTDataInTLog:1; //	VISA_REQ (disabled and forced to zero)
unsigned char capture_ebt_gen :1; //	Capture EBT GEN number for manual account entry
unsigned char RR_special_format_for_neg; //272 NUM	RR special format for negatives 0-none
//	1-Bold
//	2-Italic
//	3-Bold &Italic
//	4-Color
unsigned char RR_spl_frmt_for_sav; //273	RR special format for savings
unsigned char RR_spl_frmt_for_net_price; //274	RR special format for net price
unsigned char total_sales_text[30]; //275	total sales text
unsigned char total_coupon_text[30]; //305	total coupon text
unsigned char addnal_savings_text[30]; //335	additional savings text
unsigned char addnal_sav_ttl_text[30]; //365	additional savings total text
unsigned char net_price_sale_text[30]; //395	net price sale text
unsigned char net_price_free_text[40]; //425	net price free text

```

unsigned char RR_spcl_fmt_addnal_sav; //465      RR special format for additional savings
unsigned char RR_spcl_fmt_coupons; //466      RR special format for coupons
unsigned char CC_prn_vucher_on_duple_rcp; //467      CC print voucher on duplicate receipt

//code distribution
unsigned char no_ticket_retries; //468 NUM      No. of tickets between retries of upgrade CDS
unsigned short max_sec_retries; //469 NUM      Max seconds between retries of upgrade CDS
unsigned short max_sec_wait_user; //471 NUM      Max seconds to wait user confirmation DS
unsigned short max_min_wait_2nd_pc; //473      Max min to wait for 2nd pc to sync
unsigned char num_months_for_backup; //475      number of months to keep backup
unsigned char nu_4[4]; //476
unsigned short min_before_pos_monitor; //480 NUM      minutes before displaying POSs monitoring
unsigned char package_prefix_letter; //482 ASCII letter to prefix package number
unsigned char reserved_cds[5]; //483
unsigned char tesco_filler[125]; //488 NUM      Tesco EE specific parameters.
// In Front Office SourceSafe

//Define By POS
unsigned char bcode_tender_no; //613 NUM      COINSTAR
unsigned char bcode_tender_prefix; //614 NUM      COINSTAR
unsigned char bcode_tender_value_loc; //615 NUM      COINSTAR
unsigned char bcode_tender_value_len; //616 NUM      COINSTAR
unsigned char bcode_tender_additional_prefix; //617 NUM      Additional Barcode Tenders
unsigned char not_use4[44]; //618      Not Used

// Gift Card
unsigned char gift_crd_range_start[7]; //662 BCD      PLU number to start the range
unsigned char gift_crd_range_end[7]; //669 BCD      PLU number to end the range
long gift_crd_min_value; //676      for a variable price
long gift_crd_max_value; //680      for a variable price
unsigned char gift_crd_recharge_item[7]; //684      PLU number is use for recharge function
unsigned short gift_card_activation_tndr; //691      Gift card activation tender number
unsigned short start_gift_card_range_from_dept; // 693
unsigned short end_gift_card_range_to_dept; // 695
unsigned short free_disk_space; //697      pos occupied

```



```
unsigned char max_gift_cards_in_single_order;    //699
unsigned char CC_max_gift_cards_in_single_order; //700
unsigned long max_gift_cards_amount;             //701 in single order
unsigned char CC_max_gift_cards_amount;         //705 in single order
unsigned char CC_price_override_gift_card_dep_range; //706 CC for Price Override on Gift Card
                                                    // Department range
unsigned char int_use4[91];                     //707 NU Not Used

//ScanMaster EPS parameters
struct
{
    unsigned char PinPadIdleLockingMode :1; //798 Apply Idle Locking mode for Pin Pad
    unsigned char bReloadDllAfterEOD :1; //798 ServerEPS reload after EOD or Cold Boot
    unsigned char nu :6; //798 NU
}eps_bits;

unsigned char pos_eft_timeout; //799 POS timeout for ISS45 Integrated Payments
unsigned char cc_overridable_decline; //800 CC for Overrideable Decline
unsigned char eps_provider; //801 StoreLine EPS payment provider
                                // 0 - None
                                // 1 - PLD
                                // 2 - MTX
                                // 3 - EPS
                                // 4 - KD
unsigned char auth_file_directory[40]; //802 Auth file directory on the server
unsigned char private_card_tender; //842 Private Credit Card Tender

//Q-Buster flags //843
unsigned char is_qbuster_in_sys :1; // BIT
unsigned char use_member_card_as_qbcard :1; // BIT
unsigned char print_saving_total_when_on_line_printing :1;
unsigned char gift_card_pre_activated :1; // Synch V7 with V8 EFT dll
unsigned char nu_flag4 :4;
```

```

unsigned char max_no_wic_items_print_on_check; //844      WIC_ENDORSE
unsigned char offline_credit_tender;           //845      Synch V7 with V8 EFT dll
unsigned char offline_ebt_tender;              //846      Synch V7 with V8 EFT dll
unsigned char private_card_tender_no;          //847      Synch V7 with V8 EFT dll
unsigned char not_us[2];                       //848

//ReMa
unsigned short retailer_id;                     //850      NUM      ReMa retailer ID 0 - 9999
unsigned char store_id[24];                    //852      ASCII   ReMA Store ID

//ReMa - Central Enhanced Bad Accounts
unsigned char rema_bad_accounts_in_sys :1; //876      BIT
unsigned char nu7                        :7; //          nu

unsigned char rema_ba_pos_timeout;            //877      NUM      0-90 POS timeout for ReMA
unsigned char rema_server_pc;                 //878      NUM      MFS used for ReMa server ( 0-2 )
unsigned char int_use5[1169];                 //879      NU       Not Used

} POS_PARAM ;

// User defined parameters. Record #50
struct pos_param_user_def_
{
    unsigned char nu[1024];
};

// General system parameters (for back office
// use only, not POS!)
struct pos_param_system_
{
    struct alert_general_ //0      STRUCT Alerts parameters
    {
        unsigned char on_mfs1 :1; //0      BIT      Display alerts on MFS 1
        unsigned char on_mfs2 :1; //0      BIT      Display alerts on MFS 2
    }

```

```
unsigned char  opt_nu1      :6;    //0    NU    Not used
unsigned char  on_lfs3     :1;    //1    BIT    Display alerts on LFS  3
unsigned char  on_lfs4     :1;    //1    BIT    Display alerts on LFS  4
unsigned char  on_lfs5     :1;    //1    BIT    Display alerts on LFS  5
unsigned char  on_lfs6     :1;    //1    BIT    Display alerts on LFS  6
unsigned char  on_lfs7     :1;    //1    BIT    Display alerts on LFS  7
unsigned char  on_lfs8     :1;    //1    BIT    Display alerts on LFS  8
unsigned char  on_lfs9     :1;    //1    BIT    Display alerts on LFS  9
unsigned char  on_lfs10    :1;    //1    BIT    Display alerts on LFS 10
unsigned char  opt_nu2     :8;    //2    NU    Not used
unsigned short delay;      //3    NU    Not used
unsigned char  nu[27];     //5    NU    Not used
} alt_gen;

unsigned short working_date;    //32    DATE_I    Trading date
unsigned short last_eow;       //34    DATE_I    Last EOW date
char          nu03[18];        //36    NU    Not used
unsigned char  print_reports_auto; //54    BIT    Print report automatically at EOD
unsigned char  auto_eod_pc;     //55    NUM    PC to run automatic EOD

// parameters for each day of the week
// for automatic EOD
struct auto_eod_day_          //56    STRUCT
{
    unsigned char  auto_day    :1;    //    BIT    Automatic EOD on this day
    unsigned char  opt_nu      :7;    //    NU    Not used
    unsigned char  time[2];    //    BCD    Time to run EOD
    unsigned char  nu[29];     //    NU    Not used
} auto_day[7];

// Next working date for exceptional
// dates
struct next_wd_              //280    STRUCT
{
    unsigned short date;      //    DATE_I    Trading date
```

```
    unsigned short next;                //      DATE_I  Next working date
} nwd[40];

unsigned short v8_pc;                   //440    BITS    Bit for each PC with V8
short          days_count;              //442    NUM     internal use
unsigned char  pc_1_name[20];           //444    ASCII   PC 1 computer name (NT)
unsigned char  pc_2_name[20];           //464    ASCII   PC 2 computer name (NT)
short          dual_counter;            //484    NUM     internal use
unsigned char  nu02[538];               //486    NU      Not used
};

typedef struct pos_param_fuel_  // starting from record 180
{
    struct pump_mode_
    {
        char text[30];    // 30 * 15 = 450
    } pump_mode[15];

    struct trans_stat_
    {
        char text[8];     // 8 * 15 = 120
    } trans_stat[15];

    struct general_stat_
    {
        char text[28];    // 28
    } general_stat0;

    struct general_stat_ general_stat[255]; //28 * 255 = 7140

    struct pump_warning_status_
    {
        char text[19];    // 19 * 15 = 285
    } pump_warning_status[15];

    struct grade_name_
```

```
{
    char text[15];    // 15 * 15 = 225
} grade_name[15];

struct pump_err_
{
    char text[30];    // 30 * 15 = 450
} pump_err[15];

struct pump_w_device_
{
    char text[15];    // 15
} pump_w_device0;

struct pump_w_device_ pump_w_device[14]; // 15 * 14 = 210

unsigned char nu[ 26];

struct status_info_
{
    unsigned char reg1[2];
    unsigned char reg2[2];

    //unsigned char reg_stat;
    unsigned char swap_needed;

    unsigned char war1[2];
    unsigned char war2[2];

    unsigned char show_warning;    // 45 * 20 = 900

    unsigned char att[2];
    unsigned char bgd[2];
```

```
unsigned char beep_needed;

unsigned char str[30];
} status_info[20];

struct fuel_master_param_
{
    char          dirmaster[20];
    long          card_total_limit;
    unsigned int   minutes_dealy_receipt;

    unsigned char console[8];
    unsigned char max_pump_controlled_by_operator;

    struct
    {
        unsigned char  priv8      :1;
        unsigned char  priv7      :1;
        unsigned char  priv6      :1;
        unsigned char  priv5      :1;
        unsigned char  priv4      :1;
        unsigned char  priv3      :1;
        unsigned char  priv2      :1;
        unsigned char  priv1      :1;
    } force_release_privilage;

    unsigned char nu[28];

} fuel_master_param;

unsigned int   force_release_privilege;// not use

} POS_PARAM_FUEL;
```

KEY_FUNC.QDX (54)

File Type	Index
Record Size	32
Key Length	2

Record Format

```

typedef struct kbd_fun_rec_           //OFS  Type    Description
{
    char  key[2];                     //0      BCD      Function number
    char  name[20];                   //2      ASCII   Function description
    unsigned char preset_code ;       //22     NUM      Preset number
    char  filler[8];                  //23     NU       Not used
    char  qdx_flag;                   //31     BITS     QDEX reserved flags
} KBD_FUNC;

typedef struct
{
    char  key[2];      // 0          // for memory index list
    char  keys[8];     // 2
} KBD_FUNC;          // 10

typedef struct
{
    char  key[2];      // 0          // for memory index list
    char  name[20];    // 2
    char  *keys;       // 22
} KBD_FUNC1;          // 26

struct kbd_def_           //      STRUCT
{
    int func1 ;           //      NUM      Function in sale mode

```

```
int func2 ;           //    NUM    Function in tender mode
char priv  ;          //    NUM    Privilege required
char key_lock ;       //    NUM    Key lock required
unsigned char alpha;   //    ASCII  Alphanumeric state
char filler[1] ;      //    NU     Not used;
```


KBD_DEF.QDX (55)

File Type	Relative
Record Size	1024
Key Length	0

Record Format

```
// The KBD_DEF file can hold definitions for 10 different keyboards. Each
// keyboard is made of 5 record:
// 1 header record, 1024 bytes long
// 4 data records, 1024 bytes long each, 4096 total
```

```
typedef struct                                //OFS  Type   Description
{
    struct kbd_header_
    {
        char kbd_name[25];                    //0     ASCII   Keyboard name
        char nu    [999] ;                    //25    NU      Not used
    } header;

    struct kbd_def_ kbd_def[512];

} KEYBOARD ;
```

```
typedef struct kbd_def_ KBD_DEF;
```

```
typedef struct
{
    struct kbd_def_ key[512];
    unsigned int items;
    char key_nam[512][25] ;
    int key_code[512] ;
```

```
}KBD_DATA;
```

```
typedef struct kbd_rec_          //OFS  Type  Description
{
    union kbd_data_
    {
        struct keyboard_header_  //0    STRUCT  Keyboard header
        {
            char name[25];        //0    ASCII   Keyboard name
            char filler[999];     //25    NU      Not used
        } header;

        struct kbd_key_          //0    STRUCT  Key definition
        {
            //
            unsigned short func;  //      NUM     Function number
            unsigned short tender; //      NUM     Tender/Dep...
            char    privilage;    //      NUM     Privilege
            char    key;         //      NUM     Keylock
            unsigned char alpha;  //      ASCII   Alpha character
            char    filler;      //      NU      Not used
        } keys[128];
    } data;
} KBD_REC;
```

PRESET.QDX (56)

File Type	Relative
Record Size	16
Key Length	0

Record Format

typedef struct key_preset_	//OFS	Type	Description
{			
char opcode ;	//0	NUM	Preset opcode:
	//		0-Not defined
	//		1-PLU
	//		2-Department
	//		3-Tender
	//		4-Discount
	//		5-Return type
	//		6-Expense code
union pre_data_	//1	UNION	Preset data
{			
struct pre_plu_	//	STRUCT	Preset data for PLU
{			
char plu_num[7];	//	BCD	PLU number
} plu;			
struct pre_dep_	//	STRUCT	Preset data for department
{			
int dep_num ;	//	NUM	Department number
unsigned long dep_cash;	//	NUM_D	Value
char filler ;	//	NU	Not used
} dep ;			

```

struct pre_tndr_          //    STRUCT   Preset data for tender
{
    int          tndr_num  ;      //    NUM      Tender number
    unsigned long tndr_cash ;      //    NUM_D    Value
    char filler          ;      //    NU       Not used
} tndr  ;

struct pre_disc_          //    STRUCT   Preset data for discount
{
    int disc_num  ;      //    NUM      Discount number
    char filler[5] ;      //    NU       Not used
} disc  ;

struct pre_retn_          //    STRUCT   Preset data for return type
{
    int retn_num   ;      //    NUM      Return type
    char filler[5] ;      //    NU       Not used
} retn  ;

struct pre_expn_          //    STRUCT   Preset data for expense code
{
    int expn_num   ;      //    Expense code
    char filler[5] ;      //    NU       Not used
} expn  ;
} data ;

char filler[8] ;          //8    NU       Not used
}PRESET ;

typedef struct pre_plu_    PRE_PLU;
typedef struct pre_dep_    PRE_DEP;
typedef struct pre_tndr_    PRE_TNDR;
typedef struct pre_disc_    PRE_DISC;
typedef struct pre_retn_    PRE_RETN;

```

```
typedef struct pre_expn_    PRE_EXPN;
typedef union  pre_data_    PRE_DATA;
```

```
typedef struct
{
    char func_name[16] ;
    int  func_opcode_numeric ;
    char func_opcode_string[16] ;
    char func_preset_num[14] ;
    char func_preset_name  [16] ;

}FUNC_DATA ;
```

```
typedef struct
{
    FUNC_DATA func1 ;
    FUNC_DATA func2 ;
    FUNC_DATA func3 ;
} FUNC_INF ;
```

POSMSG.QDX (73)

File Type	Relative
Record Size	256
Key Length	0

Record Format

// This file holds definitions of various tables. It is divided as follows:

//

// From Record	To Record	Subject
// -----	-----	-----
// 1	255	POS item messages
// 256	265	Checkout banks
// 266	275	Cashier groups
// 276	530	Action reasons
// 531	629	OLA messages
// 630	650	reserved for OLA messages
// 651	660	Clubcard/Member card promotion messages
// 661	700	reserved for promotion messages
// 701	1000	Clubcard/Member card customer messages
// 1001	1101	Clubcard/Member card attachment
// 1102	1131	Item sales report definitions
// 1132	1199	reserved for item sales report
// 1200	1249	Gift certificates
// 1250	1299	reserved for gift certificates
// 1300	1362	Immediate/Interrupt/1 day sign on messages
// 1365	1450	Multiple days Sign on messages
// 1460	1558	Markdown reasons
// 1580	1584	Trading Stamps Option
// 1600	1639	Printer control
// 1640	1689	Financial calendar planning
// 1690	1739	Control Check reason table

```
// 1750          1759          Member Program Points
//
// All structures are combined into union pos_msg_un_ .
//

#define FIRST_POS_ITEM_MESSAGE      1
#define LAST_POS_ITEM_MESSAGE      255

#define FIRST_CHECK_OUT_BANK        256
#define LAST_CHECK_OUT_BANK        265

#define FIRST_CASHIER_GROUP         266
#define LAST_CASHIER_GROUP         275

#define FIRST_ACTION_REASONS        276
#define LAST_ACTION_REASONS        530

#define FIRST_OLA                   531
#define LAST_OLA                   629

#define FIRST_CLBCRD_PROM_MSG       651
#define LAST_CLBCRD_PROM_MSG       660

#define FIRST_CLBCRD_CUST_MSG       701
#define LAST_CLBCRD_CUST_MSG       1000

#define FIRST_CLUBCARD_ATTACHMEN    1001
#define LAST_CLUBCARD_ATTACHMEN    1101

#define FIRST_ITEM_SALE_REPORT      1102
#define LAST_ITEM_SALE_REPORT      1131

#define FIRST_GIFT_CIRTIFICATE      1150
#define LAST_GIFT_CIRTIFICATE      1200
```

typedef struct pos_message_rec_	//OFS	Type	Description
{			
unsigned char nul[3];	//0	NU	Not used
unsigned char message[100];	//3	ASCII	Message text
unsigned char un2;	//103	NU	Not used
unsigned char type;	//104	NUM	Message type
	//		0-Not used
	//		1-Display when item sold
	//		2-Display when total pressed


```

//          3-Sale not allowed
//          4-Display when double scanned
unsigned char min_age;          //105  NUM  Message minimum age
unsigned char nu[150];          //106  NU   Not used
} POS_MSG_REC;                  //256

// Checkout bank record
typedef struct checkout_bank_
{
    unsigned char nul[3];        //0    NU   Not used
    unsigned char name[30];      //3    ASCII  Checkout bank name include on report:
    unsigned char include_on_q_len_rep :1; //33  BIT   Queue length weekly summary
    unsigned char include_on_15_min_rep :1; //33  BIT   15 minutes
    unsigned char nu2            :6; //33  NU   Not used
    unsigned char max_percent_tills; //34  NUM   CDS: Max % of tills in a checkout bank to
//                               upgrade simultaneously

    unsigned char filler[221];    //35  NU   Not used
} POS_CHK_BANK;                  // 128/256

// Cashier group record
typedef struct cashier_group_
{
    unsigned char nul[3];        //0    NU   Not used
    unsigned char name[30];      //3    ASCII  Group name

    struct                      //33  STRUCT
    {
        //          Include on report:
        unsigned char performance_summary :1;// BIT   Performance summary
        unsigned char operator_performance :1;// BIT   Operator performance
        unsigned char operator_audit      :1;// BIT   Operator audit
        unsigned char nu                  :5;// NU   Not used
    } flags;
}
```

[illegible]


```

//          1-Print text & points
//          2-Print text only
//          3-Print only if points > 0

unsigned char filler[225];          //31  NU    Not used

} CLBCRD_PROM_MSG;                  //128/256

typedef struct clbcd_reentry_reason_
{
    unsigned char text[30];          // 0  ASCII  Reentry reason text
    unsigned char type;              // 30  NUM    Reentry reason type
    unsigned char filler[225];       // 31  NU    Not used

} CLBCRD_REENTRY_REASON;           // 256

// Notice: The next 2 structures belong to the same record/program
typedef struct clbcd_cust_msg1_
{
    char          type;              //0    NUM    Message type:
                                     //          0-Not active
                                     //          1-Operator display only
                                     //          2-Receipt only
                                     //          3-Operator display & receipt
                                     //          4-Global message

    unsigned char operator_text[30][3]; //1  ASCII  Message text
    char          date[3];            //91  BCD_D  Effective date
    char          filler[162];        //94  NU    Not used
} CLBCRD_CUST_MSG1;                //256

// The following structure repeats itself twice, to allow 10 lines of receipt
// message
typedef struct clbcd_cust_msg2_
{
    struct clbcd_reciept_cust_msg      //0    STRUCT

```

```
{ //
    char text[40]; // ASCII Line text
    char type; // NUM Line type
                // 0-Not used
                // 1-Standard
                // 2-Enlarge
} receipt_msg[5]; //
                //
    char filler[51]; // NU Not used
} CLBCRD_CUST_MSG2; //256
```

```
typedef struct markdown_reasons_
```

```
{
    unsigned char description[16]; //0
    unsigned char nu01[4]; //16
    unsigned char disc_type; //20
    long value_present; //21
    unsigned char cc_num; //25

    char active :1; //26
    char nu:7; //26

    char filler[229]; //27 NU Not used

} MARKDOWN_REASONS;
```

```
typedef struct trd_stmp_option_
```

```
{
    unsigned char filler0[30]; //0
    struct
    {
        unsigned char active:1; //30
        unsigned char mon :1; //30
        unsigned char tue :1; //30
    }
```

```

    unsigned char wed    :1;           //30
    unsigned char thu    :1;           //30
    unsigned char fri    :1;           //30
    unsigned char sat    :1;           //30
    unsigned char sun    :1;           //30

    }flags;

    unsigned char time_form[2];         //31
    unsigned char time_to[2];           //33
    long          multiple;             //35
    long          min_purchase ;        //39
    char          filler1[213];         //43  NU      Not used

} TRD_STMP_OPTION;

typedef struct printer_control1_
{
    unsigned char  description[20];     //0
    char          active :1;           //20
    char          nu:7;                 //20
    char          controls[60];         //21
    unsigned short lines;               //81
    unsigned char printer_type;         //83
    char          line_type;            //84
    char          filler[171];          //85  NU      Not used

} PRINTER_CONTROL;

typedef struct fin_year_weeks_
{
    unsigned char  start_date[3];       //0   BCD_D   First day of financial year
    unsigned char  end_date[3];         //3   BCD_D   End date of financial year
    unsigned char  mon_weeks[12];       //6   NUM     Weeks per financial month
    unsigned char  first_month;         //18  NUM     First month in year

```

```
    unsigned char    filler[237];           //19   NU       Not used
} FIN_CLNDR;                               //256

#define FIN_CLNR_1ST_YEAR      1997
#define FIN_CLNR_LST_YEAR      (FIN_CLNR_1ST_YEAR+LAST_FIN_CLNDR-FIRST_FIN_CLNDR)

// Item sales report
typedef struct item_sale_report_
{
    char             report_name[16];       //0     ASCII   Report name
    char             period;                //16    NUM     Report period
                                           //       0-Daily
                                           //       1-Weekly
    unsigned char     sort_field;           //17    NUM     Sort criteria
                                           //       0-PLU
                                           //       1-Sales quantity
                                           //       2-Sales value
    char             department[2];         //18    BCD     Department number
    char             from_plu[7];           //20    BCD     Build report from PLU
    char             to_plu[7];             //27    BCD     Build report to PLU
    char             plu_number[15][7];     //34    BCD     Build report on 15 or less PLUs
    short            last_date;             //139   NUM     Last change date
    char filler[115];                       //141   NU      Not used
} ITEM_SALE_REPORT;                       // 256

// Control check reason table
typedef struct cc_reason_table_
{
    char             nul[16];               //0     NU       Not used
    struct            //16    NUM     Control Check type
    {
        unsigned transaction_voids :1;     //      BITS    Transaction voids
    }
}
```

```

    unsigned post_voids      :1;      //    BITS    POST voids
    unsigned transaction_returns :1;    //    BITS    Transaction Returns
    unsigned cancels         :1;      //    BITS    Cancels
    unsigned substracts      :1;      //    BITS    Substracts
    unsigned merchandise_return :1;    //    BITS    Merchandise return
    unsigned nu              :10;     //    BITS    not used
}cc_type;

char      filler1[7];          //18    NU      Not used
char      cc_num;              //25    NUM      Control Check number
unsigned char  actice :1;      //26    BIT      Activate reason
unsigned char  nu      :7;     //    BITS    not used
char description[25];          //27    ASCII   Description
char filler2[204];             //52    NU      Not used
}CC_REASON_TABLE ; //256

//FRYS Member Points Program
typedef struct member_program_points_
{
    unsigned char  program_name[16];    //0      ASCII
    unsigned char  nu[24];              //16
    unsigned long  min_points_for_redemp; //40
    unsigned long  min_purchase_for_redemp; //44
    unsigned char  redemption_rule;     //48    NUM      Redemption Rule type:
                                     //
                                     //      0 - % off ticket total
                                     //      1 - $ amount off ticket total
                                     //      2 - % off specific PLU sold
                                     //      3 - $ amount off specific PLU sold
                                     //      4 - free specific item
                                     //      5 - % off specific department
                                     //      6 - $ amount off specific department
                                     //      7 - coupon printed
                                     //      8 - fixed bar-code printed

    unsigned long  redemption_value;    //49
    unsigned long  formula_points;      //53

```



```
unsigned long  formula_amount;          //57
unsigned char  markdown_dept[2];        //61  BCD
unsigned char  nu_flags1;                //63  BITS   Future flags
unsigned char  nu_flags2;                //64  BITS   Future flags
unsigned char  filler1[35];              //65
unsigned char  plu_number[7];            //100 BCD    Attached PLU number
unsigned char  dep_num[2];               //107 BCD    attached Dep number
unsigned char  filler2[147];             //109

}MEMBER_POINTS;  //256

union pos_msg_un_
{
    struct pos_message_rec_      pms;
    struct checkout_bank_        cbk;
    struct cashier_group_        cgr;
    struct action_reasons_       ars;
    struct ola_                  oms;
    struct attachment_rec_       atc;
    struct gift_certificate_type_ gcr;
    struct clbcrd_prom_msg_       cpm;
    struct clbcrd_cust_msg1_       ccl;
    struct clbcrd_cust_msg2_       cc2;
    struct item_sale_report_       isl;
    struct cc_reason_table_        ccr;
};

//#####
//\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ END   P O S - M E S S A G E S  /////
//#####
//#####

// Multiple Messages
```

[illegible]

```
{
    unsigned short cash_num;

    }list_cash[20];                //          20*2 = 40

    char          filler[15];      //241

} MULTIPLE_MESSAGES;             //256

// Immediate Messages
typedef struct immediate_messages_
{
    unsigned char    filler1[8];
    unsigned char    select_cash;    //8          NUM          0-cashier , 1-Tills

    struct immidiate_list_text_      //9          ASCII    5 x 36 = 180
    {
        char line[36];

    }list_text[5];

    struct
    {
        //189
        unsigned char    active          :1;
        unsigned char    upon_receive    :1;    //          BIT          Upon receive
        unsigned char    null            :6;

    } flag1;

    unsigned char    msg_operation;    //190          NUM          Message operation
                                //          0-Hit C 1-Yes/No message

    unsigned char    filler2[5];

    unsigned char    select_mail;      //196          NUM          Select mail
```

```
                                //          0-All 1-from to 2-select
unsigned short    cash_from;    //197    NUM    Cash From
unsigned short    cash_to;      //199    NUM    Cash To

struct                                //201    NUM    select cash 1-20
{
    unsigned short cash_num;

    }list_cash[20];                  //          20*2 = 40

char              filler3[15];      //241

} IMMEDIATE_MESSAGES;              //256

typedef union
{
    MULTIPLE_MESSAGES    multiple_message;
    IMMEDIATE_MESSAGES    immediate_message;
} MESSAGE_BOARD
```

ALERTYPE.QDX (74)

File Type	Index
Record Size	128
Key Length	3

Record Format

typedef struct alert_prm_rec_	//OFS	Type	Description
{			
unsigned char num[3];	//0	BCD	Alert type number
unsigned char msg[80];	//3	ASCII	Alert message
 struct	 //83	 STRUCT	
{			
unsigned char pop_message :1;	//	BIT	Pop-up on the middle of screen
unsigned char pop_alert :1;	//	BIT	Pop-up on the bottom of screen
unsigned char ignore_alert :1;	//	BIT	Ignore this alert
unsigned char pass_req :1;	//	BIT	Password required
unsigned char send_to_rpo :1;	//	BIT	Send To RPO //TBA
unsigned char send_rpo_notif2pos :1;	//	BIT	Send RPO notification to pos //TBA
unsigned char nu :2;	//	NU	Not used
} opt1;			
 unsigned short delay_next;	 //84		 Delay between the next alert
 struct	 //86		
{			
unsigned char pager :1;	//	BIT	Send alerts to pager
unsigned char e_mail :1;	//	BIT	Send alerts to e-mail address
unsigned char nu :6;	//		
} devices;			

```
unsigned char e_mail_address    [30];    //87      E-Mail address to send alerts to
unsigned char nu[10];           //117    NU      Not used
char          qdx;              //127    BITS    Reserved for qdx
} ALERT_PRM_REC;                //128
```

LOYAL.QDX (75)

File Type	Relative
Record Size	256
Key Length	0

Record Format

struct club_card_scheme_	//OFS	Type	Description
{			
unsigned char effective_date[3];	//0	BCD	Date YYMMDD
unsigned char description[20];	//3	ASCII	Description
struct	//23	STRUCT	
{			
unsigned char Active :1;	//	BIT	Clubcard/Member card active
unsigned char use_frequent :1;	//	BIT	use also frequent shopper
unsigned char check_bad_acc :1;	//	BIT	check in bad account
unsigned char check_bad_enh :1;	//	BIT	check in enhanced bad acc.
unsigned char disc_member_in_file:1;	//		Restrict discount only to member in file
unsigned char filler :3;	//	NU	Not used
} flags;			
long treshold_value;	//24	NUM_D	Threshold amount
long threshold_points;	//28	NUM	Threshold points
long step_value;	//32	NUM_D	Every addition amount
long step_points;	//36	NUM	Points for Every addition amount
long bonus_treshold_points;	//40	NUM	Bonus threshold points
long bonus_step_points;	//44	NUM	Bonus step points
struct	//48	STRUCT	
{			
	//		Bonus active in:

```

unsigned char  monday   :1;          //    BIT    Monday
unsigned char  tuesday  :1;          //    BIT    Tuesday
unsigned char  wednesday:1;          //    BIT    Wednesday
unsigned char  thursday :1;          //    BIT    Thursday
unsigned char  friday   :1;          //    BIT    Friday
unsigned char  saturday :1;          //    BIT    Saturday
unsigned char  sunday   :1;          //    BIT    Sunday
unsigned char  filler   :1;          //    NU     Not used
} bonus_app_days;

unsigned char  bonus_start_time[2];  //49  BCD     Bonus start time HHMM
unsigned char  bonus_end_time[2];    //51  BCD     Bonus end time HHMM
unsigned char  filler1[27];          //53  NU     Not used

//----- Only for ADDITIONAL SCHEMS -----

unsigned char  trigger_type;          //80  NUM     Trigger type:
//                                     1-Gets points only from Current scheme
//                                     2- Gets points from Current + Basic scheme
unsigned char  reciept_print;         //81  NUM     Receipt print:
//                                     1- Print TOTAL include ADDITIONAL points
//                                     2- Don't print ADDITIONAL points
//                                     3- Print TOTAL include
//                                     ADDITIONAL + Additional separately
//                                     4- Points for each scheme printed separately
unsigned char  value_to_use;          //82  NUM     Value to use:
//                                     1- Give points for QS only
//                                     2- Points only for amount paid in this card
//                                     (not > QS )
//                                     3- Points for all amount including CHANGE.
unsigned char  reciept_description[20]; //83  ASCII   Receipt description
unsigned char  print_card_number;     //103 NUM     Print card number (0-No/1-Yes)
unsigned char  filler2[23];           //104 NU     Not used
unsigned char  disc_num;              //127 NUM     linked discount number

```



```
unsigned char  nu[127];           //128  NU      Not used
unsigned char  qdx_flag;          //255  BCD      Reserved for qdx
};
```

```
// Written by PCMEOD with problems detected during auto EOD
```

```
// Applications using this structure :
```

```
//      PCMEOD
```

```
struct auto_eod_log_rec_
```

```
{
    int  opcode;
    char info[1000];
    char nu[22]
```

SAVERS.QDX (76)

File Type	Relative
Record Size	256
Key Length	0

Record Format

```
typedef struct promotion_
{
    //-- GENERAL part -----
                                //OFS  Type   Description
    unsigned char  promotion_number[2];    //0    BCD    Promotion number
    unsigned char  promotion_type;         //2    NUM    Promotion type
                                //
                                //      1 Store/Immediate Promotion
                                //      2 Multisaver/Enhanced
                                //      3 Linksaver/Group promotion
                                //      4 Extended Multisaver/Order promotion
                                //      5 Extended Linksaver/order group promotion
                                //      6 Order/Group promotion
                                //      7 One free
    unsigned char  start_day[3];           //3    BCD    Start day YYMMDD
    unsigned char  end_day[3];             //6    BCD    End day   YYMMDD

    struct prom_flags_                //9    STRUCT
    {
        unsigned char  active            :1; //    BIT    Active prom
        unsigned char  m_card_dependency :1; //    BIT    Member card dependency
        unsigned char  avail_all_schemes :1; //    BIT    Available for all schemes
        unsigned char  allow_when_offline :1; //    BIT    Allow this promo when Server
                                                //
                                                //      is offline - COREMA.
        unsigned char  low_to_high       :1;
        unsigned char  delayed_promotion :1;
    }
};
```

```
unsigned char value_enter_by_cshr      :1;
unsigned char update_loyalty_server    :1; //COREMA

}flags;

unsigned char  description[20];          //10   ASCII   Description
unsigned char  department[2];           //30   BCD     Department
unsigned char  reward_type;              //32   NUM     Reward type
                                           //      1 - Cash value
                                           //      2 - Percentage off
                                           //      3 - Clubcard/Member card points
                                           //      4 - Per item discount
                                           //      5 - Low BOGO
                                           //      6 - High BOGO
                                           //      7 - STUB
                                           //      8 - New price
                                           //     12 - Target Message
                                           //     17 - New price per quantity
struct scheme_list_                     //33   NUM     Scheme numbers 1-10
{
    unsigned char scheme;

}scheme_list[10];

long          reward;                   //43   NUM_D   Reward
short         club_card_scheme_number; //47   NUM     Clubcard/Member card scheme number
long          limited_qty;              //49   NUM     Limited quantity
unsigned char free[5];                  //53   NU      Not used

union
{
    struct ext_banded_                  //58   STRUCT   (banded)
    {
                                           //
        unsigned char bucket_type;      //58   NUM     Bucket type
```

```

//          1 Standard / Units
//          2 Loose / Weighed
//          3 Department

long  minimum_item_value; //59  NUM_D  Minimum item value
long  maximum_reward;    //63  NUM_D  Maximum reward
struct ext_banded_band_  //67  STRUCT
{
//          Purchase requirements
    long  quantity;      //67  NUM    Quantity
    long  value;         //71  NUM_D  Value
    long  reward;        //75  NUM_D  Reward
}ext_banded_bands[5];
}ext_banded;

struct ext_not_banded_  //58  STRUCT
{
//
    unsigned char  bucket_type; //58  NUM    Bucket/Group type
//          1 Standard / Units
//          2 Loose / Weighed
//          3 Department

    long  minimum_item_value; //59  NUM_D  Minimum item value
    long  maximum_reward;    //63  NUM_D  Maximum reward purchase requirements
    long  threshold_quantity; //67  NUM    Threshold quantity
    long  threshold_value;   //71  NUM_D  Threshold value
    long  threshold_reward;  //75  NUM_D  Threshold reward
    long  step_quantity;     //79  NUM    Step quantity
    long  step_value;        //83  NUM_D  Step value
    long  step_reward;       //87  NUM_D  Step reward
    unsigned char reserved[36]; //91

}ext_not_banded;

struct ext_linksave_    //58  STRUCT
{
//
    long  maximum_reward;   //58  NUM_D  Maximum reward

```

```
struct ext_linksave_bucket_    //62   STRUCT
{
    unsigned char  bucket_type;    //62   NUM      Bucket/Group type
                                   //      1 Standard / Units
                                   //      2 Loose / Weighed
                                   //      3 Department
                                   //      Purchase requirements

    long  minimum_item_value;    //63   NUM_D   Minimum item value
    long  treshold_quantity;    //67   NUM      Threshold quantity
    long  treshold_value;    //71   NUM_D   Threshold value
}ext_linksave_buckets[3];

unsigned char reserved[26];    //101

}ext_linksave;

struct linksave_              //58   STRUCT
{
    //
    struct linksave_bucket_    //58   STRUCT
    {
        unsigned char  bucket_type;    //58   NUM      Bucket/Group type
                                   //      1 Standard / Units
                                   //      2 Loose / Weighed
                                   //      3 By value

        long  treshold_quantity;    //59   NUM      Threshold quantity
    }linksave_buckets[3];

    unsigned char reserved[54];    //73

}linksave;

struct multisaver_            //58   STRUCT
{
    //
    unsigned char  bucket_type;    //58   NUM      Bucket/Group type
```

```

//          1 Standard / Units
//          2 Loose / Weighed
//          3 By value
long  treshold_quantity; //59  NUM  Threshold quantity
long  step_quantity;    //63  NUM  Step quantity

unsigned char reserved[60]; //67

}multisaver;

struct store_promotion_ //58  STRUCT
{
//Store promotion (immediate)
    unsigned char  bucket_type; //58  NUM  Bucket/Group type
//          1 Standard / Units
//          2 Loose / Weighed
    long  treshold_quantity; //59  NUM  Threshold quantity
    long  step_quantity;    //63  NUM  Step quantity
    unsigned char reserved[60]; //67

}store_promotion;

struct free_item_
{
    struct free_bucket_ //58  STRUCT
    {
        unsigned char  bucket_type; //58  NUM  Bucket/Group type
//          1 Standard / Units
//          2 Loose / Weighed
        long  treshold_quantity; //59  NUM  Threshold quantity
    }free_buckets[3];
    unsigned char  plu[7]; //73  BCD  PLU number

    unsigned char reserved[47]; //80

```

```
} free_item;

}purchase_request;

struct member_segments_                                //127
{
    unsigned short seg_1      :1;
    unsigned short seg_2      :1;
    unsigned short seg_3      :1;
    unsigned short seg_4      :1;
    unsigned short seg_5      :1;
    unsigned short seg_6      :1;
    unsigned short seg_7      :1;
    unsigned short seg_8      :1;
    unsigned short seg_9      :1;
    unsigned short seg_10     :1;
    unsigned short seg_11     :1;
    unsigned short seg_12     :1;
    unsigned short seg_13     :1;
    unsigned short seg_14     :1;
    unsigned short seg_15     :1;
    unsigned short seg_16     :1;
} segments;

unsigned char  start_time[2];                          //129 BCD_T   Promotion start time
unsigned char  end_time[2];                            //131 BCD_T   Promotion end time
struct prom_active_day_                                //133 BITS    Active days for promotion
{
    unsigned char opt_Sunday   :1;
    unsigned char opt_Monday   :1;
    unsigned char opt_Tuesday  :1;
    unsigned char opt_Wednesday :1;
    unsigned char opt_Thursday :1;
    unsigned char opt_Friday   :1;
```

```

    unsigned char  opt_Saturday   :1;
    unsigned char  opt_07         :1;
}  act_day;

struct adv_prom_day_time_          //134  STRUCT  "Happy Hour" for each day of the week
    unsigned char  start_time[2];  //      BCD_T   Start time
    unsigned char  end_time[2];    //      BCD_T   End time
} day_time[7];

struct
{
    unsigned char  trig_grp_1  :1;    //162  BIT      Group 1 is the trigger
    unsigned char  trig_grp_2  :1;    //162  BIT      Group 2 is the trigger
    unsigned char  trig_grp_3  :1;    //162  BIT      Group 3 is the trigger
    unsigned char  trig_grp_nu4:1;    //162  BIT      not used
    unsigned char  trig_grp_nu5:1;    //162  BIT      not used
    unsigned char  nu          :3;    //162  BIT      not used
} flags2;

unsigned char  cc_for_min_purchase;  //163          Control check for min purchase
long          min_purchase_amt;      //164          Min. purchase amount
unsigned char  inf_nu[23];           //168  NU      Not used

// *** END OF INFORMATIVE PART ***

// *** START OF SALES PART ***

long          day_qty;               //191  NUM      Unit Quantity sold today
long          day_weight;            //195          Weight qty sold today
long          day_sales;             //199  NUM      Sales today
long          day_markdown;          //203  NUM      Markdown/Points rewarded
long          prd_qty;               //207  NUM      Unit Quantity sold
long          prd_weight;            //211          Weight qty sold today
long          prd_sales;             //215  NUM      Sales

```



```
long          prd_markdown;          //219  NUM    Markdown/Points rewarded
unsigned char sales_nu[32];          //223  NUM    Not used
unsigned char qdx_flag;              //255  BITS    QDEX reserved flags
} TESCO_PROMO
```

T_CUST.QDX (77)

File Type	Index
Record Size	128
Key Length	10

Record Format

typedef struct t_customer_	//OFS	Type	Description
{			
unsigned char cust_no[10];	//0	BCD	Customer number
long points;	//10	NUM	Points
long redemption_value;	//14	NUM	Redemption value
long pending_redemption_value;	//18	NUM	Pending redemption value
unsigned char update_date[3];	//22	BCD	Update date
unsigned char panel_flag;	//25	NUM	Panel_flag
unsigned char status;	//26	NUM	Customer status
unsigned char filler[6];	//27	NU	Not used
unsigned char second_id[10];	//33	ASCII	Secondary ID
unsigned char name[20];	//43	ASCII	Customer name
long promotional_points_1;	//63	NUM	Points type 1
long promotional_points_2;	//67	NUM	Points type 2
long promotional_points_3;	//71	NUM	Points type 3
long promotional_points_4;	//75	NUM	Points type 4
long promotional_points_5;	//79	NUM	Points type 5
unsigned char message_no;	//83	NUM	Customer message
unsigned short card_scheme_num;	//84	NUM	Card Scheme number
long saving;	//86	NUM	Saving
unsigned char nu[4];	//90		
struct members_segments_	//94		
{			
unsigned short seg_1 :1;			

```
unsigned short seg_2    :1;
unsigned short seg_3    :1;
unsigned short seg_4    :1;
unsigned short seg_5    :1;
unsigned short seg_6    :1;
unsigned short seg_7    :1;
unsigned short seg_8    :1;
unsigned short seg_9    :1;
unsigned short seg_10   :1;
unsigned short seg_11   :1;
unsigned short seg_12   :1;
unsigned short seg_13   :1;
unsigned short seg_14   :1;
unsigned short seg_15   :1;
unsigned short seg_16   :1;
} segments;

unsigned char  ticket_info[30];          //96   ASCII   Ticket information
unsigned char  nul;                      //126  NU     Not used

unsigned char  qdx_flag;                  //127  NUM     Reserved for qdx
}T_CUSTOMER;

typedef struct q_customer_
{
    unsigned char  cust_no[10];           //0     BCD
    long  points;                          //10 Display only
    long  redemption_value;               //14 Display only
    long  pending_redemption_value;       //18 Display only
    unsigned char  update_date[3];        //22 Display only
    unsigned char  panel_flag;            //25 Display only
    // 1.Homestore customer , Non panel
    // 2.Homestore customer
    // 3.Panel customer
}
```

```
// This field should be added to TRANSACTION record.
unsigned char  status;                //26 Display only
// 0- Redemption available
// 1- Redemption locked

// This flag is used so 2 POS cant update the same record at the same file. This flag should be
cleared after transaction ended OR during EOD.

unsigned char  cust_title;           //27
struct member_flags_
{
    unsigned char staff_member       :1;
    unsigned char home_shop         :1;
    unsigned char filler              :6;
}flag1;                             //28

unsigned char  filler[4];            //29
unsigned char  lock[10];             //33
unsigned char  name[20];             //43
unsigned char  qdx_flag;             //127
}Q_CUSTOMER;                        //128
```

BAD_ACC2.QDX (79)

File Type	Index
Record Size	128
Key Length	12

Record Format

```

typedef struct                                     //OFS  Type  Description
{
    unsigned char  num[11];                        //0    BCD    Bad account number
    unsigned char  seq_num;                        //11   NUM    Sequential number
    unsigned char  account_name[34];               //12   ASCII  Bad account name
    unsigned char  initials[10];                  //46   ASCII  Bad account initials
    unsigned char  tender_type[2];                //56   BCD    Tender type
    unsigned short date_tendered;                  //58   NUM    Date tendered
    unsigned short credit_check_type;              //60   NUM    Credit_check_type
    unsigned short orig_store;                     //62   NUM    Original store
    long           amount_tendered;                //64   NUM_D  Amount tendered
    long           amount_owing;                   //68   NUM_D  Amount_owing
    unsigned char  bank[5];                        //72   BCD    Bank number
    char           filler[50];                     //77   NU     Not used
    unsigned char  qdx_flag;                       //127  BITS   QDEX reserved flags
} BAD_ACCT2;

// structure of PO/ROA seq file
//qdx file #99
typedef struct poroa_rec_                          //OFS  Type  Description
{
    unsigned char  tr_type;                        //0    NUM    Record type
                                                    //      1 - payment
                                                    //      2 - receipt
                                                    //      3 - buy aid
    unsigned char  nu0[2];                         //1    NU     Not used

```

```

unsigned char exp_desc[20];           //3   ASCII   Expense code description
unsigned      local_branch;           //23   NUM     This branch number
unsigned char date[3];                //25   BCD_D   Date
unsigned char exp_reference[10];      //28   ASCII   Reference number
unsigned int  branch;                 //38   NUM     Branch number
unsigned long exp_code;               //40   NUM     Expense code
unsigned int  tender_type;            //44   NUM     Tender number
long          amount;                 //46   NUM_D   Amount
unsigned char template_no;            //50   NUM     Template used
//                                     //         0 - basic
//                                     //         1 - purchase
//                                     //         2 - debtors
//                                     //         3 - Christmas stamps
//                                     //         4 - bad debts
struct                                               //51   STRUCT
{
    unsigned char hist_rec    :1;    //      BIT     History record
    unsigned char nu          :7;    //              Reserved
} flags1;
unsigned char nu[48];                 //52   NU       Not used
union template_data_               //100  UNION    Different data according
{                                     //              to template used
    char templ_data[156];           //
//
    struct tmpl_purchase_          //      STRUCT  Purchase (1)
    {
        long          vat_amount;    //      NUM_D   VAT amount
        unsigned char vat_invoice[13]; //    ASCII   Invoice number
        unsigned long grv_number;    //      NUM     GRV number
        unsigned int  re_dept;       //      NUM     Head office department num.
        unsigned char desc[28];      //    ASCII   Expense code description
        char          nu[105];       //
    } purchase;

```

```
struct tmpl_debtor_          //    STRUCT  Debtor (2)
{
    long          vat_amount;    //    NUM_D   VAT number
    unsigned char  vat_invoice[13]; //    ASCII  Invoice number
    char          nu[139];      //    NU     Not used
} debtor;

struct tmpl_christ_         //    STRUCT  Christmas Stamps (3)
{
    unsigned long  agent;        //    NUM     Agent number
    char          nu[152];      //    NU     Not used
} Christ;

struct tmpl_bad_debt_       //    STRUCT  Bad debt recovered (4)
{
    unsigned char  acct_num[11]; //    BCD     Account number
    unsigned char  acct_name[34]; //    ASCII  Customer name
    unsigned char  initials[10]; //    ASCII  Initials
    unsigned       acct_tender;   //    NUM     Tender number causing error
    unsigned       date_tendered; //    DATE_I  Date tendered
    char          nu0;           //    NU     Not used
    long          amount_tendered; //    NUM_D   Amount tendered
    long          amount_owing;   //    NUM_D   Amount owing
    unsigned       orig_store;    //    NUM     Original store number
    char          nu[86];
} bad_debt;

} tmpl;
} POROA_REC;

#define MAX_BUY_HIST 5
//qdx file #99
typedef struct
{
    unsigned char  tr_type;        // 1 - payment    0
```

```

// 2 - receipt
// 3 - buy aid

struct buyaidhist_           // 1
{
    unsigned int buyaid_no;
    long    amount;
    long    voucher_no;
    unsigned char account[20];
    unsigned int  ticket_no;
    unsigned int  cashier;           // 50 * 5 = 250
    unsigned      pos_no;
    unsigned char date[3];
    unsigned char nu[11];
} buy[MAX_BUY_HIST];

unsigned char nu[4];           // 251
unsigned char qdx;             // 255
} BUYAIDHIST_REC;             // 256
```


GEN_TBL.QDX (80)

File Type	Index
Record Size	256
Key Length	24

Record Format

```
// Different record opcodes

#define GTBLE_DEA      (unsigned char)1 // Drug Enforcement Agency
#define GTBLE_CHK_BANK (unsigned char)2 // Checkout banks (V8 & WinPOS)
#define GTBLE_CUST_SRV (unsigned char)3 // Customer survey
#define GTBLE_BONUS_BUY (unsigned char)4 // Bonus buy types
#define GTBLE_CBNK_LOAN (unsigned char)5 // Opening loan per checkout bank
#define GTBLE_EXCISE    (unsigned char)6 // PLU Excise tax groups

#define GTBLE_EBTWIC_STATE_CODES (unsigned char)37 //

typedef struct gtbl_EBTWIC_state_codes_
{
    unsigned char opcode;           // 0    NUM    Record opcode. Always 37
    unsigned char KeyFiller[22];    // 1    NU     Not used
    unsigned char StateFIPS;        //23    BCD    State code numeric 0-99
    unsigned char StateName[20];    //24
    unsigned char StateCode[2];     //44
    unsigned char BIN[32];          //46
    short PANLength;               //78
    short TrainingDigit;           //80
    short PANMaskingPrintStart;     //82
    short PANMaskingPrintEnd;       //84
    unsigned char filler[169];      //86    NU     Not used
    unsigned char qdx;              //255   BITS   QuickDex reserved flags
} GTBL_EBTWIC_STATE_CODES;
```

typedef struct gtbl_gnrl_	//OFS	Type	Description
{			
// Record key			
unsigned char opcode;	//0	NUM	Record opcode
unsigned char key_filler[23];	//1	NU	Not used
unsigned char data[231];	//24	NU	Not used
unsigned char qdx;	//255	BITS	QuickDex reserved flags
} GTBL_GNRL;			

typedef struct gtbl_dea_grp_	//OFS	Type	Description
{			
unsigned char opcode;	//0	NUM	Record opcode. Always 1
unsigned char key_filler[21];	//1	NU	Not used
unsigned char grp_no[2];	//22	BCD	Group number
unsigned char name[16];	//24	ASCII	Group name
unsigned long qty_level1;	//40	NUM	Warning level
unsigned long qty_level2;	//44	NUM	Lock level
unsigned char nu[207];	//48	NU	Not used
unsigned char qdx;	//255	BITS	QuickDex reserved flags
}GTBL_DEA_GRP;			

typedef struct gtbl_chk_bank_	//OFS	Type	Description
{			
unsigned char opcode;	//0	NUM	Record opcode. Always 2
unsigned char key_filler[22];	//1	NU	Not used
unsigned char grp_no;	//2	BCD	Group number
unsigned char name[30];	//24	ASCII	Group name
unsigned char nu[201];	//54	NU	Not used
char qdx;	//255	BITS	QuickDex reserved flags
} GTBL_CHK_BANK;			

typedef struct gtbl_cust_srv_	//OFS	Type	Description
{			

```
unsigned char  opcode;                //0    NUM    Record opcode. Always 3
unsigned char  key_filler[20];        //1    NU     Not used
unsigned char  qst_no[3];             //21   BCD    Question number

unsigned char  start_date[3];         //24   BCD_D  Start date
unsigned char  end_date[3];           //27   BCD_D  End date
unsigned char  filler1[6];            //30   NU     Not used

unsigned char  opt_Sunday   :1;       //36   BIT    Don't ask on Sunday
unsigned char  opt_Monday   :1;       //36   BIT    Don't ask on Monday
unsigned char  opt_Tuesday  :1;       //36   BIT    Don't ask on Tuesday
unsigned char  opt_Wednesday:1;       //36   BIT    Don't ask on Wednesday
unsigned char  opt_Thursday :1;       //36   BIT    Don't ask on Thursday
unsigned char  opt_Friday   :1;       //36   BIT    Don't ask on Friday
unsigned char  opt_Saturday :1;       //36   BIT    Don't ask on Saturday
unsigned char  opt_07       :1;       //36   BIT    Not used
unsigned char  opt_end_ticket:1;       //37   BIT    0:Start ticket 1:end ticket
unsigned char  opt_active   :1;
unsigned char  opt_10       :1;
unsigned char  opt_11       :1;
unsigned char  opt_12       :1;
unsigned char  opt_13       :1;
unsigned char  opt_14       :1;
unsigned char  opt_15       :1;

struct day_times_                //38
{
    unsigned char  start_time[2];
    unsigned char  end_time[2];
    }day_time[7];

unsigned char  filler[14];          //66   NU     Not used
unsigned char  answer_type;         //80   NUM    Answer input type:
//                                0 - Y/N
```

```

//                                1 - Numeric

unsigned char  qst_line[36][4];    //81  ASCII  Question text
unsigned char  nu[30];            //225
unsigned char  qdx;                //255  BITS   QuickDex reserved flags
} GTBL_CUST_SRV;

typedef struct gtbl_bonus_buy_    //OFS  Type   Description
{
    unsigned char  opcode;         //0    NUM    Record opcode. Always 4
    unsigned char  key_filler[16]; //1    NU     Not used
    unsigned char  plu_no[7];      //17   BCD    Group number
    unsigned char  name[20];       //24   ASCII  Group name
    unsigned char  comp_type;      //44   NUM    Comparison type
    long          comp_price;      //45   NUM_D  Comparison price
    long          comp_qty;        //49   NUM    Comparison quantity
    unsigned char  valid_date[3];  //53   CHR    Valid date YYMMDD
    unsigned char  nu[199];        //56   NU     Not used
    unsigned char  qdx;            //255  BITS   QuickDex reserved flags
} GTBL_BONUS_BUY;

typedef struct gtbl_cbnk_loan_    //OFS  Type   Description
{
    unsigned char  opcode;         //0    NUM    Record opcode. Always 5
    unsigned char  key_filler[19]; //1    NU     Not used
    unsigned char  bank[2];        //20   BCD    Checkout bank number
    unsigned char  med_no[2];      //22   BCD    Tender number
    long          loan_amt;        //24   NUM_D  Opening loan amount
    unsigned char  nu[227];        //28   NU     Not used
    unsigned char  qdx;            //255  BITS   QuickDex reserved flags
} GTBL_CBNK_LOAN;

typedef struct gtbl_excise_tax_
{
    unsigned char  opcode;         //0    NUM    Record opcode. Always 6

```

```
unsigned char  key_filler[21];          //1    NU    Not used
unsigned char  excise_tax[2];           //22   BCD    Group number
unsigned char  desc[20];                //24   ASCII  Description
unsigned short dept;                   //44
unsigned long  value;                   //46
unsigned char  nu[205];                 //50   NU    Not used
char          qdx;                      //255  BITS   QuickDex reserved flags
} GTBL_EXCISE_TAX;
```

```
typedef struct gtbl_card_program_
{
    unsigned char  opcode;               //0     NUM    Record opcode. Always 7
    unsigned char  key_filler[21];       //1     NU    Not used
    unsigned char  program[2];           //22    BCD    program no.
    unsigned short type;                 //24    NU    Reserved
    unsigned char  desc[20];             //26    ASCII  Program name
    unsigned char  print_template[20];   //46    ASCII  Template name
    unsigned char  void_template[20];    //66    ASCII  Template name
    unsigned char  nu[169];              //86    NU    Not used
    char          qdx;
} GTBL_CARD_PROGRAM;
```

```
typedef union gtbl_rec_
{
    GTBL_GNRL          gnrl;
    GTBL_DEA_GRP       dea_grp;
    GTBL_CHK_BANK      chk_bank;
    GTBL_CUST_SRV       cust_srv;
    GTBL_BONUS_BUY     bonus_buy;
    GTBL_CBNK_LOAN     cbnk_loan;
    GTBL_EXCISE_TAX    excise;
    GTBL_CARD_PROGRAM  card_prog;
    unsigned char      data[256];
} GTBL_REC;
```

PACKAGES.QDX (83)

File Type	Index
Record Size	1024
Key Length	5

Record Format

```
typedef struct packages_rec_  
{  
    unsigned char complete;           //0  
    unsigned char num[4];             //1  
  
    unsigned char flag;               //5  
    unsigned char sensitivity;        //6  
  
    short registration_date;          //7  
    short start_date;                //9  
  
    unsigned char registrator;        //11  
    unsigned char initiator;          //12  
  
    unsigned char computer_state[256]; //13  
  
    unsigned char valid_base_1[4];    //269  
    unsigned char valid_base_2[4];    //273  
    unsigned char valid_base_3[4];    //277  
    unsigned char valid_base_4[4];    //281  
    unsigned char valid_base_5[4];    //285  
  
    unsigned char external_dependency :1; //289  
    unsigned char reregistration_err :1;  //289
```

```
unsigned char check_valid_bases      :1;    //289
unsigned char un                      :5;    //289

unsigned short script_line;           //290
unsigned char ex_dependency_states;   //292

unsigned char mfs1_start;             //293
unsigned char mfs2_start;             //294

char ctl_package_name[20];           //295
unsigned short version_array[4];      //315

unsigned short enh_pkg_error[256];     //323

unsigned char filler[188];            //835
unsigned char qdx;                    //1023 QDEX reserved flag
}PACKAGES_REC
```

HOURLPROD.QDX (84)

File Type	Relative
Record Size	1024
Key Length	0

Record Format

/*

The Hourly Productivity file is a flat file broken into several areas:

Ofs.	Data Type	Record Type	Size in Bytes
-----	-----	-----	-----
0	Store 15 minutes productivity	struct hp_rec_	1024 data length X 96 quarters = 98304 bytes
98304	POS 15 minutes productivity of last 7 days	struct hp_daily_	12 rec. length X 96 quarters X 7 days X 250 POSTs = 2016000 bytes
2114304	POS 15 minutes productivity, daily & weekly	struct hp_pos_	64 rec. length X 2 (day & week)X 96 quarters X 250 POSTs = 3072000 bytes

5186304

*/

```
#define HP_PRD_LAST_DAY 0
```



```
#define HP_PRD_WEEK      1
#define HP_PRD_LAST_WEEK 2
#define HP_PRD_MONTH     3
#define HP_PRD_LAST_MONTH 4
#define HP_PRD_YEAR      5
#define HP_PRD_LAST_YEAR 6

// The following structure keeps the store hourly productivity information in
// 15 minutes intervals. 24 X 4 = 96 records.
//
typedef struct hp_rec_          //OFS  Type    Description
{
    // The following structure repeats itself
    // 7 times, for the following periods:
    // 0 - Previous day
    // 1 - Weekly
    // 2 - Previous week
    // 3 - Monthly
    // 4 - Previous month
    // 5 - Yearly
    // 6 - Previous year
    struct hp_prd_
    {
        unsigned char  amount6[6];          //0    BIN    Sales value
        unsigned long  customers;            //6    NUM    Customers
        unsigned long  items;                //10   NUM    Items
        unsigned char  pos_active[32];       //14   BITS   Bit per active POS
                                                //
        // times handling variables          //
        unsigned long  item_reg_time;        //46   NUM    Item regs. time (seconds)
        unsigned long  idle_time;            //50   NUM    Idle time (seconds)
        unsigned long  secure_time;          //54   NUM    Secure time (seconds)
        unsigned long  tender_time;          //58   NUM    Tender time (seconds)
        unsigned long  wait_time;            //62   NUM    Wait time (seconds)
```

```

    unsigned long   inq_time;           //66   NUM    Inquiry. time (seconds)
    unsigned long   other_time;         //70   NUM    Other time (seconds)
    unsigned long   item_scan_count;    //74   NUM    Items scanned count
    unsigned char   nu[50];             //78
} prd[7];                             //
//
    unsigned char   nu[128];            //896   NU     Not used
} HP_REC;

```

// The following structure holds information per POS per 15 minutes per period:

```

// 0 - Previous day
// 1 - Weekly
// 2 - Previous week
// 3 - Monthly
// 4 - Previous month
// 5 - Yearly
// 6 - Previous year
//

```

```

struct hp_daily_           //OFS   Type    Description
{
    struct hp_daily_pos_   //0     STRUCT
    {
        // The following structure repeats
        // itself 96 (24 hours * 4 quarters)
        // X 7 (periods) times
        //
        struct hp_daily_15_min_ //      STRUCT
        {
            unsigned int   avg_q_len;           //      NUM    Avg. queue length
            unsigned long   sign_on_time;       //      NUM    Sign on time (seconds)
            int             q_len;              //      NUM    Queue length
            unsigned char   q_len_count;        //      NUM    Queue length count

            unsigned char   nu[3];              //      NU     Not used
        }
    }
}

```

```
// Total size - 12 bytes
} prd[96][7]; //Total size - 12*96*7 = 8064
}hp_daily_pos_no;

} ; // 250 pos *8064 = 2MB

typedef struct hp_daily_15_min_ HP_DAILY_15_MIN;
typedef struct hp_daily_ HP_DAILY;

// The following structure holds information per POS per 15 minutes
// per day & week.
//
struct hp_pos_ //OFS Type Description
{
    struct hp_pos_period //0 STRUCT 15 minutes data
    {
        struct hp_pos_15_min_ //0 STRUCT Daily & weekly data
        {
            unsigned char amount6[6]; //0 BIN Sales value
            unsigned customers; //6 NUM Customers
            unsigned card_key; //8 NUM Card number keyed count
            int items; //10 NUM Items count
            unsigned card_swipe; //12 NUM Card swipes count
            int q_len; //14 NUM Queue length total
            unsigned char q_len_count; //16 NUM Queue length count
            unsigned int avg_q_len; //17 NUM Avg. queue length
            unsigned char opt_active :1; //19 BIT POS active
            unsigned char opt_nu :7; //19 NU Not used

            // times handling variables
            unsigned long item_reg_time; //20 NUM Item regs. time (seconds)
            unsigned long idle_time; //24 NUM Idle time (seconds)
            unsigned long secure_time; //28 NUM Secure time (seconds)
            unsigned long tender_time; //32 NUM Tender time (seconds)
```

```

    unsigned long wait_time;           //36   NUM   Wait time (seconds)
    unsigned long inq_time;            //40   NUM   Inquiry. time (seconds)
    unsigned long other_time;          //44   NUM   Other time (seconds)
    unsigned long item_scan_count;     //48   NUM   Items scanned count
    unsigned      open_dept;           //52   NUM   Open department sales count
    unsigned char nu[10];              //54   NU    Not used
    //Total size - 64                  //
} day, week;                          //
} period[96];                          //

// Total size - 64 bytes record length
//           96 1/4 hour
//           2
//           250 POST = 3MB
//TOTAL FILE SIZE = 3MB
};

typedef struct hp_pos_15_min_ HP_POS_15_MIN;
typedef struct hp_pos_ HP_POS;

// This structure keeps tender counts:
// per pos, 15 min, tender type (1-24)
// entry 0 for tender type above 24
typedef struct hp_pos_15_min_tenders
{
    short type_count[25];
    // Total size - 25 bytes record length
    //           2
    //           250 POST = 3MB
    //TOTAL FILE SIZE = 12500 K
} HP_POS_15_MIN_TND;

//*****

```

```
#define MAX_LINES_IN_STOCK_PAGE 80
```

```
typedef struct
```

```
{
```

```
    unsigned char page_number[2]; // BCD // 0
```

```
    unsigned char stock_count_type; // 2 // 1 - plu  
                                     // 2 - vendor  
                                     // 3 - dept  
                                     // 4 - all
```

```
    unsigned char status; // 3 // 1 - during counting  
                          // 2 - end counting
```

```
    unsigned long trans_no; //4 // process on transactions until this trans number
```

```
    union def_range_ // 8
```

```
{
```

```
    struct def_plu_range_
```

```
{
```

```
        unsigned char plu1[7];
```

```
        unsigned char plu2[7];
```

```
    } plu_range;
```

```
    struct def_vendor_range_
```

```
{
```

```
        unsigned char vendor1[5];
```

```
        unsigned char filler1[2];
```

```
        unsigned char vendor2[5];
```

```
        unsigned char filler2[2];
```

```
    } vendor_range;
```

```
    struct def_dep_range_
```

```
{
```

```
        unsigned char dep1[2];
```

```
        unsigned char filler1[5];
```

```
        unsigned char dep2[2];
```

```
        unsigned char filler2[5];
    } dep_range;

} range;

unsigned char filler1[1001]; // 22
unsigned char qdx_flag;      // 1023
} STOCK_TYPE_DEF;

typedef struct
{
    unsigned char page_number[2]; // BCD // 0
    long          total_quantity; // 2
    unsigned char pos_no;          // 6
    unsigned char counter_name[20]; // 7
    unsigned int  cashier;          // 27
    struct        // 29
    {
        unsigned char opt_pc_generated :1;
        unsigned char nu                :7;
    } flag1;
    unsigned char filler1[113]; // 30

    struct lines_in_stock_page_ //143
    {
        unsigned char plu[7];
        long          qty;
    } line[MAX_LINES_IN_STOCK_PAGE];
    unsigned char qdx_flag; // 1023
} STOCK_PAGE;

// Common areas - location & fields
struct posware_common_
{
```

```
unsigned char  nu01[401];          // 0
unsigned char  prog_name[80];      // 401
unsigned char  nu02[20];          // 481
unsigned char  user_priority;      // 501
unsigned       user_no;            // 502
unsigned char  user_pass[5];       // 504
unsigned char  nu06[72];          // 509
unsigned char  shop_name[31];      // 581
unsigned char  nu03[8];            // 612
unsigned char  branch_name[40];    // 620
unsigned char  branch_no[10];      // 660
unsigned char  nw_date[6];         // 670
unsigned char  nu04[4];            // 676
unsigned char  eod_date[6];        // 680
unsigned char  pc_no[4];           // 686
unsigned char  nu05[334];          // 690
};                                  // 1024
```

EEXPEN.QDX (85)

File Type	Index
Record Size	128
Key Length	3

Record Format

typedef struct eexp_rec_	//OFS	Type	Description
{			
unsigned char code[3];	//0	BCD	Expense code number
unsigned char po_desc[20];	//3	ASCII	Paidout description
unsigned char roa_desc[20];	//23	ASCII	Receipt description
unsigned char template_no;	//43	NUM	Template number
	//		0 - Basic
	//		1 - P/C Purchase
	//		2 - Basic + VAT
	//		3 - Xmas Stamps
	//		4 - Bad Debts received
unsigned char type;	//44	NUM	Expense type
	//		0 - Paidout & Receipt
	//		1 - Paidout
	//		2 - Receipt
char nu[82];	//45	NU	Not used
unsigned char qdx_flag;	//127	BITS	QDEX reserved flags
} EEXP_REC;			
#define MAX_EXP_TEMPLATE	5		
#define TNDR_TYPE_NOT_USED	0		
#define TNDR_TYPE_CASH	1		
#define TNDR_TYPE_CHECK	2		
#define TNDR_TYPE_CREDIT_EFT	3		
#define TNDR_TYPE_OTHER_CHRAGE	4		


```
#define TNRD_TYPE_GENERIC_CARD      5
#define TNRD_TYPE_FOREIGN_CUR      6
#define TNRD_TYPE_FOOD_STAMPS      7
#define TNRD_TYPE_VENDOR_COUPONS    8
#define TNRD_TYPE_STORE_COUPONS     9
#define TNRD_TYPE_IN_HOUSE         10
#define TNRD_TYPE_DEBIT             11
#define TNRD_TYPE_WIC               12
#define TNRD_TYPE_BUY_AID           13
#define TNRD_TYPE_STAFF_D           14
#define TNRD_TYPE_CHECKERS_STAFF_D 15
#define TNRD_TYPE_STAMP_VOUCHER     16
#define TNRD_TYPE_GENERAL_GROUP     17
#define TNRD_TYPE_EBT_FS            18
#define TNRD_TYPE_EBT_CASH          19
#define TNRD_TYPE_ACCUMULATIVE      20
#define TNRD_TYPE_IN_HOUSE_PUR      21
#define TNRD_TYPE_CUST_PAYMENT      22
#define TNRD_TYPE_WIC_CVV           25
```

BASKET.QDX (88)

File Type	Relative
Record Size	64
Key Length	4

Record Format

// This file holds definitions of various tables. It is divided as follows:

```
//
// From offset          Length          To Offset          Subject
// -----
//      0              64*100=6400      6400              Basket
//                      6400              3840              10240      Reserved
//                      10240 (10k)      10*301K=3082240    3092480      10 graphic images
//
```

```
struct basket_          //OFS  Type  Description
{
    long limit;          //0    NUM_D  Limit to amount
    long count_today;    //4    NUM    Today
    long count_p_day;    //8    NUM    Previous day
    long count_week;     //12   NUM    This Week
    long count_p_week;   //16   NUM    Previous week
    long count_month;    //20   NUM    This month
    long count_p_month;  //24   NUM    Previous month
    long count_year;     //28   NUM    This year
    long count_p_year;   //32   NUM    Previous year
    int num_of_limits;    //36   NUM    Number of records in the file
    unsigned char filler[25]; //38   NU     Not used
    unsigned char qdx_flag; //63   BITS   QDEX reserved flags
};
```

```
typedef struct img_data_
```

	//OFS	Type	Description
{			
Char name[16];	//0		ASCII Image name
unsigned short delay;	//16	NUM	Delay in seconds
struct	//18	BITS	
{			
unsigned char used :1;	//		Image in use
unsigned char nu7 :7;	//		Not used
}opt;			
unsigned short no_recs;	//19	NUM	no recs
unsigned char nul[203];	//21	NU	Not used
struct pos_img_header_	//224	STRUCT	Image header
{			
unsigned char stamp[7];	//0	ASCII	"POS IMG"
unsigned char version;	//7	NUM	Version number
unsigned char sub_ver;	//8	NUM	Sub version
unsigned char nu[12];	//9	NU	Not-used
unsigned long colors;	//21	NUM	Colors in image
unsigned width;	//25	NUM	Image width
unsigned height;	//27	NUM	Image height
unsigned char palette[768];	//29	NUM	256X3 palette values
unsigned char nu2[3];	//797	NU	Not used
}head;			
// unsigned char img[640*480];	//	NUM	300KB of image data
	//1024		
}IMG_DATA			

LOGINFO.QDX (90)

File Type	FIFO
Record Size	128
Key Length	0

Record Format

typedef struct	//OFS	Type	Description
{			
unsigned char user_number;	//0	BCD	User number
unsigned char program[30];	//1	ASCII	Program name
unsigned char description[70];	//31	ASCII	Menu description
unsigned char date[6];	//101	BCD	Entry date
unsigned char time[4];	//107	BCD	Entry time
unsigned long user_num_cash;	//111	NUM	User number (Tesco)
unsigned char filler[12];	//115	NU	Not used
unsigned char qdx;	//127	BITS	QDEX reserved flags
} EURMENU_LOG;			


```
#define VDR_CPN_LAST_Z 0
#define VDR_CPN_DAY 1
#define VDR_CPN_WEEK 2
#define VDR_CPN_MONTH 3
#define VDR_CPN_YEAR 4
```

VDR_CPN.QDX (93)

File Type	Index
Record Size	64
Key Length	6

Record Format

```
typedef struct vdr_cpn_rec_           //OFS  Type  Description
{
    unsigned char  manf_num[6];       // 0   BCD    Manufacturer number
    struct vdr_cpn_period_            //6    STRUCT
    {
        int  qty;                     //6    NUM    Quantity
        long value;                   //8    NUM    Value
    } period[5];
    char          nu[27];              //36   NU     Not used
    char          qdx;                 //63   BITS   QDEX reserved flags
} VDR_CPN;                           //64

#define MAX_ITEM_SALES_REPS 100
#define MAX_ITEM_SALES_COND 50
typedef struct item_sales_rpt_
{
    unsigned char  name[30];           // 0
    unsigned char  rep_by   ;          // 30
    unsigned char  nul[97];            // 31

    struct item_sales_rpt_cond_        // 128
    {
        unsigned char  plu1[7],        //
        plu2[7];                    // 16 * 50 = 800
```

```
    unsigned char  nu[2];          //
} cond[MAX_ITEM_SALES_COND];      //

    unsigned char  nu2[96];        // 928
} ITEM_SALE_REP;                   // 1024
```

POSACCT.QDX (94)

File Type	Index
Record Size	512
Key Length	5

Record Format

```
// This file is made of different kinds of records, which differ from each  
// other by their opcode.
```

```
// Record type macros
```

```
#define PACCT_OPC_INF1    1  
#define PACCT_OPC_INF2    2  
#define PACCT_OPC_DEPT    3  
#define PACCT_OPC_MED     4  
#define PACCT_OPC_TILL    10
```

```
// Periods
```

```
#define PACCT_PRD_DAY     0  
#define PACCT_PRD_WEEK    1  
#define PACCT_PRD_MONTH   2  
#define PACCT_PRD_YEAR    3
```

```
#define PACCT_KEY_LEN     5
```

```
// General structure of a POS accountability record layout
```

```
struct pacct_gen_          //OFS  Type    Description  
{  
    struct pacct_key_      //0      STRUCT  
    {  
        unsigned char  pos[2];    //0      BCD      POS number
```

```

unsigned char  opcode;           //2    NUM    Opcode:
                                   //      1-Informative data 1
                                   //      2-Informative data 2
                                   //      3-Department data
                                   //      4-Media data
                                   //      5-Non scanned department
                                   //
unsigned char  data_key[2];      //3    BCD    Data key
} key;

char nu[507];                   //5    NU     Not used
};

// POS accountability record type 1 - General information
struct pacct_info_ttl_1_        //OFS  Type  Description
{
    unsigned char  pos[2];        //0    BCD    POS number
    unsigned char  opcode;        //2    NUM    1
    unsigned char  data_key[2];    //3    BCD    Should be 1
    unsigned char  grand_total[6]; //5    BIN    POS grand total
    unsigned char  grand_net_total[6]; //11   BIN    POS total net sales
    unsigned short z_counter;      //17   NUM    Z counter

    // The following structure repeats itself
    // 4 times, for daily, weekly, monthly &
    // yearly periods
    struct pacct_info_ttl_1_period_ //19   STRUCT
    {
        long      gross_sales_value; //19   NUM_D   Gross sales value
        long      gross_sales_count; //23   NUM     Gross sales count
        long      gross_return_value; //27   NUM_D   Gross return value
        long      gross_return_count; //31   NUM     Gross return count
        long      num_items;          //35   NUM     Items sold
        long      nu001                ; //39   NU     Not used
    }

```



```
long      disc_value;           //43  NUM_D  Discounts value
long      disc_count;          //47  NUM    Discounts count
long      prom_value;          //51  NUM_D  Promotions value
long      prom_count;          //55  NUM    Promotions count
long      neg_dep_value;        //59  NUM_D  Negative department value
long      neg_dep_count;        //63  NUM    Negative department count
long      non_merch_value;      //67  NUM_D  Non merchandise department value
long      non_merch_count;      //71  NUM    Non merchandise department count
long      price_override_value; //75  NUM_D  Price overrides value
long      price_override_count; //79  NUM    Price overrides count
long      net_dep_total;        //83  NUM_D  Net department sales
long      total_sales_med;      //87  NUM_D  Total media taken
long      cancel_value;         //91  NUM_D  Cancels value
long      cancel_count;         //95  NUM    Cancels count
long      sub_value;            //99  NUM_D  Subtracts value
long      sub_count;            //103 NUM    Subtracts count
long      void_value;           //107 NU    Not used
long      void_count;           //111 NU    Not used
long      abs_refund_value;      //115 NUM_D  Absolute value of refunds
long      abs_refund_count;      //119 NUM    Absolute count of refunds
short     return_tickets;       //123 NUM    Returned tickets
unsigned char filler[17];       //125 NU    Not used

} prd[4];

unsigned char qdx_flag;          //511 BITS  QDEX reserved flags
};                               //512

// POS accountability record type 2 - General information
struct pacct_info_ttl_2_        //OFS  Type  Description
{
    unsigned char pos[2];        //0    BCD    POS number
    unsigned char opcode;        //2    NUM    1
    unsigned char data_key[2];   //3    BCD    Should be 1
```

```

// The following structure repeats itself
// 4 times, for daily, weekly, monthly &
// yearly periods
struct pacct_info_ttl_2_period_      //5    STRUCT
{
    char        nu[8];                //5    NU    Not used
    long        payout_value;         //13    NUM_D  Payouts value
    long        payout_count;         //17    NUM    Payouts count
    long        reciepts_value;       //21    NUM_D  Receipts value
    long        reciepts_count;       //25    NUM    Receipts count
    long        no_sale_count;        //29    NUM    No sale count
    long        num_of_customers;     //33    NUM    Customers count
    long        item_reg_time;        //37    NUM    Item registration time (seconds)
    long        idle_time;            //41    NUM    Idle time (seconds)
    long        secure_time;          //45    NUM    Secure time (seconds)
    long        tender_time;          //49    NUM    Tender time (seconds)
    long        wait_time;            //53    NUM    Wait time (seconds)
    long        inq_time;             //57    NUM    Inquiry time (seconds)
    long        other_time;           //61    NUM    Other time (seconds)
    long        lift_value;           //65    NUM_D  Pick ups value
    long        lift_count;           //69    NUM    Pick ups count
    long        open_float_value;     //73    NUM_D  Opening float value
    long        add_float_value;      //77    NUM_D  Additional float value
    long        add_float_count;      //81    NUM    Additional float count
    long        tax_amount;           //85    NUM_D  Tax amount
    long        tax_count;            //89    NUM    Tax count
    long        taxable_amount;       //93    NUM_D  Taxable amount
    long        tax_exempt_amount;    //97    NUM_D  Tax exempt amount
    long        tax_exempt_count;     //101   NUM    Tax exempt count
    long        taxable_exempt_amount; //105   NUM_D  Taxable exempt amount
    short       cards_swiped_qty;     //109   NUM    Card swipes count
    short       cards_keyed_qty;      //111   NUM    Card number keyed count
    long        open_depts_value;     //113   NUM_D  Open department sale value

```

```
    unsigned char  filler[14];                //117  NU      Not used
} prd[4];

unsigned char  nu[2];                        //509  NU      Not used
unsigned char  qdx_flag;                     //511  BIT      QDEX reserved flags
};

// POS accountability record type 3 - Department totals
struct pacct_dept_ttl_                      //OFS  Type    Description
{
    unsigned char  pos[2];                   //0      BCD      POS number
    unsigned char  opcode;                   //2      NUM      3
    unsigned char  dept_no[2];               //3      BCD      Department number

    // The following structure repeats itself
    // 4 times, for daily, weekly, monthly &
    // yearly periods
    struct pacct_info_dept_ttl_period_      //5      STRUCT
    {
        long      net_sales;                 //5      NUM_D     Net sales
        long      net_count;                 //9      NUM      Sales count
        long      void_sales;                //13     NUM_D     Cancels & subtracts value
        long      void_count;                //17     NUM      Cancels & subtracts count
        long      return_sales;              //21     NUM_D     Returns value
        long      return_count;              //25     NUM      Returns count
        long      key_amount ;               //29     NUM_D     Keyed amounts
        long      nu002 ;                    //33     NU      Not used
        long      store_coupon_sales;        //37     NUM_D     Store coupons sales
        long      total_coupon_count;        //41     NUM      Store coupons count
        long      discount_sales;            //45     NUM_D     Discounts value
        long      discount_count;            //49     NUM      Discounts count
        long      item_scan_count;           //53     NUM      Items scanned count
        long      item_key_count;            //57     NUM      Items keyed count
        long      item_dept_count;           //61     NUM      Open department count
    }
}
```

```

    long        vendor_coupon_sales;        //65   NUM_D   Vendor coupon sales
    long        bonus_coupon_sales;         //69   NUM_D   Bonus coupon sales
    long        item_key_count_sales_refund;//73   NUM
    unsigned char filler[46];                //77   NU      Not used
} prd[4];

unsigned char  nu[34];                      //477  NU      Not used
unsigned char  qdx_flag;                    //511  BITS    QDEX reserved flags
};

// POS accountability record type 4 - Media totals
struct pacct_tender_ttl_                    //OFS  Type    Description
{
    unsigned char  pos[2];                  //0     BCD     POS number
    unsigned char  opcode;                  //2     NUM     3
    unsigned char  tender_no[2];            //3     BCD     Tender number

    // The following structure repeats itself
    // 4 times, for daily, weekly, monthly &
    // yearly periods
    struct pacct_info_tender_ttl_period_    //5     STRUCT
    {
        long        media_value;            //5     NUM_D   Media value
        long        media_count;            //9     NUM     Media count
        long        lift_value;             //13    NUM_D   Pick ups value
        long        lift_count;             //17    NUM     Pick ups count
        long        open_float_value;       //21    NUM_D   Opening float value
        long        add_float_value;        //25    NUM_D   Additional float value
        long        add_float_count;        //29    NUM     Additional float count
        long        payout_value;           //33    NUM_D   Payouts value
        long        payout_count;           //37    NUM     Payouts count
        long        receipt_value;          //41    NUM_D   Receipts value
        long        receipt_count;          //45    NUM     Receipts count
        long        media_count_increase;   //49    NUM     Media usage count
    }
};

```

```
    unsigned char  filler[48];                //53   NU      Not used
} prd[4];

unsigned char  nu[122];                      //389   NU      Not used
unsigned char  qdx_flag;                     //511   BITS    QDEX reserved flags
};

// POS accountability record type 10 - Non scanned departments
struct pacct_non_scanned_dept_
{
    unsigned char  pos[2];                   //0      BCD      Pos number
    unsigned char  opcode;                   //2      NUM      Opcode (10)
    unsigned char  data_key[2];              //3      BCD      Department number

    // The following structure repeats itself
    // 4 times, for daily, weekly, monthly &
    // yearly periods
    struct till_total_                      //5      STRUCT
    {
        struct dept_total_                  //5      STRUCT
        {
            //
            long      today;                 //      NUM_D    Today takings
            long      prev_day;              //      NUM_D    Prev. day takings
            long      prev_monday;           //      NUM_D    Prev. Monday takings
            long      prev_tuesday;          //      NUM_D    Prev. Tuesday takings
            long      prev_wednesday;        //      NUM_D    Prev. Wednesday takings
            long      prev_thursday;         //      NUM_D    Prev. Thursday takings
            long      prev_friday;           //      NUM_D    Prev. Friday takings
            long      prev_suterday;         //      NUM_D    Prev. Saturday takings
            long      prev_sunday;           //      NUM_D    Prev. Sunday takings

            struct                                          //      STRUCT
            {
                //
                unsigned char  sunday      :1; //      BIT      Data entered for Sunday
            }
        }
    }
}
```

```

        unsigned char    monday    :1; //    BIT    Data entered for Monday
        unsigned char    tuesday   :1; //    BIT    Data entered for Tuesday
        unsigned char    wednesday :1; //    BIT    Data entered for Wednesday
        unsigned char    thursday  :1; //    BIT    Data entered for Thursday
        unsigned char    friday    :1; //    BIT    Data entered for Friday
        unsigned char    saturday  :1; //    BIT    Data entered for Saturday
        unsigned char    null      :1; //    NU     Not used
    } day;

        unsigned char filler[63];    //    NU     Not used

    } dept_total;
} pos_[4];

unsigned char nu[106];                //    NU     Not used
unsigned char qdx_flag;                //511 BITS    QDEX reserved flags
};

//qdx file #94
typedef union
{
    struct pacct_gen_            gen;
    struct pacct_info_ttl_1_    inf1;    // opcode 1
    struct pacct_info_ttl_2_    inf2;    // opcode 2
    struct pacct_dept_ttl_      dept;    // opcode 3
    struct pacct_tender_ttl_     tender; // opcode 4
    struct pacct_non_scanned_dept_ till; // opcode 10
} POSACC_REC;

struct general_depweek_
{
    struct general_key_
    {

```

```
unsigned char  opcode;

unsigned char  dept_no[2];    // in BCD

unsigned char  data_key[2];

} key;


struct dep_hp_
{
    long qty;
    char filler [12];
} week_hour[24];


unsigned char  nu[634];
unsigned char  qdx_flag;

};                                     // 1024


// Qdx file #96, for general purpose
typedef union
{
    struct general_depweek_    gen_depweek; // Opcode 30

} GEN_REC
```

ALERT.QDX (95)

File Type	Index
Record Size	128
Key Length	4

Record Format

```

struct alert_key_                                //0    STRUCT
{
    unsigned char  confirmed;                    //0    NUM    0-Not confirmed yet
                                                //      NUM    1-Confirmed
    unsigned char  alt_no[2];                    //1    BCD    Alert type number
    unsigned char  pos_no;                       //3    NUM    POS number
};

#ifdef POSTRANS_H
typedef struct alert_rec_                        //OFS  Type    Description
{
    struct alert_key_ key;
    union trans_   trn;                          //4    STRUCT  The transaction record

    // confirmation fields
    unsigned char  conf_date[3];                 //71   BCD     Confirmed date
    unsigned char  conf_time[3];                 //68   BCD     Confirmed time
    unsigned short user_no;                      //74   NUM     User number
    unsigned char  conf_nu[8];                   //76   NU      Not used
    unsigned short delay_next;                   //84   NUM     Delay between alerts (sec)

    struct
    {
        unsigned char  pop_message   :1;        //      BIT     Popup message box
        unsigned char  pop_alert     :1;        //      BIT     Popup that indicate there is an alert
    }

```



```
unsigned char ignore_alert :1;    //    BIT    Ignore this alert
unsigned char pass_req      :1;    //    BIT    Password required
unsigned char send_rpo      :1;    //    BIT    Send alert to RPO
unsigned char send_rpo_notif2pos :1; //    BIT    Send RPO notification to pos    //TBA
unsigned char nu            :2;    //    NU     Not used
} opt1;

unsigned char alert_date[3];      //87    BCD    Alert trs date
unsigned char alert_time[3];      //90    BCD    Alert trs time
char          nu[34];             //93    NU     Not used
char          qdx;                //127   BITS   QDEX reserved flags
} ALERT_REC;                      //128
#endif
```

GEN_DB.QDX (96)

File Type	Index
Record Size	1024
Key Length	5

Record Format

```
// Different record opcodes

#define DAY_HP_OPCODE      (unsigned char)1
#define DEP_WEEK_OPCODE    (unsigned char)2
#define GDB_OPEN_CASH      (unsigned char)3
#define GDB_ROA             (unsigned char)4
#define GDB_MISC_TOTALS    (unsigned char)5
#define GDB_MED_DEPOSIT    (unsigned char)6
#define GDB_POUT            (unsigned char)7
#define GDB_SAFE_DCLR      (unsigned char)8
#define GDB_NON_MERC_DISC  (unsigned char)9
#define GDB_EFTPOS         (unsigned char)10
#define GDB_RECEIPT        (unsigned char)11
#define GDB_MEDIA          (unsigned char)12
#define GDB_TAX             (unsigned char)13
#define GDB_DEPOSIT        (unsigned char)14
#define GDB_RECEIPTS       (unsigned char)15
#define GDB_LIFT_AND_FLOAT (unsigned char)16


// on line records. Cleared at EOD

#define GDB_OLINE_ROA      (unsigned char)104
#define GDB_OLINE_DEPOSIT  (unsigned char)106
#define GDB_OLINE_POUT     (unsigned char)107
#define GDB_OLINE_RECEIPT  (unsigned char)111


typedef struct gdb_general_
```

```
{
    struct gdb_key_
    {
        unsigned char  opcode;           // 0  BCD      Opcode
        unsigned char  num[4];           // 1  BCD      Number
    } key;

    unsigned char      data[1018];       // 5          Data
    unsigned char      qdx_flag;         //1023 BITS   QuickDex reserved flags
} GDB_GNRL;

#define DEP_WEEK_MAX_WEEKS 53
typedef struct dep_week_ //12 //OFS Type Description
{
    unsigned char  opcode;           // 0  BCD      Opcode 2
    unsigned char  filler[2];        // 1  NU       Not used
    unsigned char  key[2];           // 3  BCD      Department number
    //
    struct dep_week_inf_             // 5  STRUCT
    {
        long  budget;               //      NUM     Plan department sale
        long  sales;                //      NUM     Department sale
        long  sales_tax_ex;          //      NUM     Department sale tax exempt
        unsigned char  nu[7];        //      NU      Not used
    } week[DEP_WEEK_MAX_WEEKS];     // 19 * 53 = 1007

    char  nu[11];                   //1012 NU      Not used
    char  qdx;                      //1023 BITS    QDEX reserved flags
} DEP_WEEK;

typedef struct gdb_day_hp_15_
{
    struct gdb_day_hp_15_key_       //0    STRUCT   Record key
    {
```

```

    unsigned char  opcode;           //    NUM    Opcode - always 1
    unsigned char  date[3];          //    BCD    Date
    unsigned char  num;               //    NUM    quarter in day - 0-96
}key;

short            custs;              //5     NUM    Customers
long             sales;              //7     NUM    Sales
long             items;              //11    NUM    Items
unsigned char    pos_active[32];     //15    BITS   Active POSTs
unsigned char    nu[976];            //47    NU     Not used
unsigned char    qdx;                //1023   BITS   QuickDex reserved flags
} GDB_DAY_HP_15;                    //1024

typedef struct gdb_open_cash_
{
    struct gdb_open_cash_key_        //0     STRUCT  Record key
    {
        unsigned char  opcode;       //    NUM    Opcode - always 3
        unsigned char  subopcode;     //    NUM    1 - tenders 0-49
                                           //          2 - tenders 50-99
        unsigned char  nu  [3];       //    NU     Not used
    }key;

    struct gdb_open_cash_med_        //5     STRUCT
    {
        //
        long            amount;       //    NUM_D   Opening amount
        unsigned char    nu01[2];     //    NU     Not used
        short           qty;          //    NUM     Quantity (not used)
        unsigned char    nu[12];      //    NU     Not used
    } med[50];                       //

    unsigned char    nu01[18];        //1005 NU     Not used
    unsigned char    qdx;              //1023 BITS   QuickDex reserved flags
} GDB_OPEN_REC ;                    //1024

```

```
typedef struct gdb_roa_  
{  
    struct gdb_roa_key_                //0    STRUCT    Record key  
    {  
        unsigned char  opcode;        //    NUM      Opcode - always 4  
        unsigned char  subopcode;     //    NUM      subopcode - 0  
        unsigned char  date[3];       //    BCD_D    Date  
    }key;  
  
    struct gdb_roa_type_              //5    STRUCT  
    {  
        //  
        long          amount;         //    NUM_D    ROA amount  
        short         qty;            //    NUM      Quantity (not used)  
        unsigned char  nu01[4];       //    NU       Not used  
    } med[100];                      //  
  
    unsigned char  nu01[18];          //1005 NU      Not used  
    unsigned char  qdx;               //1023 BITS    QuickDex reserved flags  
} GDB_ROA_REC ;                      //1024  
  
typedef struct gdb_safe_dclr_  
{  
    struct gdb_safe_dclr_key_        //0    STRUCT    Record key  
    {  
        unsigned char  opcode;        //    NUM      Opcode - always 4  
        unsigned char  subopcode;     //    NUM      subopcode - 0  
        unsigned char  date[3];       //    BCD_D    Date  
    }key;  
  
    struct gdb_safe_med_              //5    STRUCT  
    {  
        //  
        long          amount;         //    NUM_D    ROA amount
```

```

    short          qty;                //    NUM    Quantity (not used)
    unsigned char  nu01[4];            //    NU     Not used
} med[100];                            //

unsigned char  nu01[18];                //1005 NU     Not used
unsigned char  qdx;                    //1023 BITS   QuickDex reserved flags
} GDB_SAFE_DCLR_REC;                  //1024

// Daily miscellaneous totals
typedef struct gdb_misc_ttl_
{
    struct gdb_misc_ttl_key_            //0    STRUCT  Record key
    {
        unsigned char  opcode;          //    NUM    Opcode - always 5
        unsigned char  subopcode;        //    NUM    1 - this week
                                                //          2 - previous week
        unsigned char  date[3];          //    BCD_D   Date
    }key;

    long          charge_payments;        //5    NUM_D   Account payments
    long          discounts;              //9    NUM_D   Discounts
    long          receipts;               //13   num_D   Receipts
    unsigned char  nu01[1006];            //17   NU     Not used
    unsigned char  qdx;                   //1023 BITS   QuickDex reserved flags
} GDB_MISC_TTL ;                      //1024

// Daily deposits
typedef struct gdb_med_deposit_
{
    struct gdb_med_depost_key_           //0    STRUCT  Record key
    {
        unsigned char  opcode;          //    NUM    Opcode - always 1
        unsigned char  nu ;              //    NU     Not used
        unsigned char  date[3];          //    BCD    Date
    }

```

```
}key;

struct gbd_med_deposit_data_          //5    STRUCT
{
    long          amount;              //    NUM_D    Deposit amount
    unsigned char  nu01[2];            //    NU       reserved
    unsigned char  nu[4];              //    NU       Not used
} med[100];

    unsigned char  nu[18];              //1005    NU       Not used
    unsigned char  qdx;                //1023    BITS    QuickDex reserved flags
} GDB_MED_DEPOSIT_REC;

// Deposits
typedef struct gdb_deposit_
{
    struct gdb_depost_key_             //0    STRUCT    Record key
    {
        unsigned char  opcode;         //0    NUM       Opcode - always 1
        unsigned short  date;          //1    NUM       Date in swapped format
        unsigned char  media_num;      //3    NUM       Media number
        unsigned char  seq_num;        //4    NUM       Sequential number
    }key;

    long          amount;              //5    NUM_D    Deposit amount
    unsigned char  nu[1014];           //9    NU       Not used
    unsigned char  qdx;                //1023    BITS    QuickDex reserved flags
} GDB_DEPOSIT_REC;

typedef struct gdb_pout_
{
    struct gdb_pout_key_               //0    STRUCT    Record key
    {
        unsigned char  opcode;         //    NUM       Opcode - always 4
```

```

    unsigned char  subopcode;          //    NUM    subopcode - 0
    unsigned char  date[3];            //    BCD_D   Date
}key;

struct gdb_pout_type_                  //5    STRUCT
{
    //
    long          amount;              //    NUM_D   ROA amount
    short         qty;                 //    NUM     Quantity (not used)
    unsigned char  nu01[4];            //    NU      Not used
} med[100];                           //

unsigned char  nu01[18];               //1005 NU     Not used
unsigned char  qdx;                    //1023 BITS   QuickDex reserved flags
} GDB_POUT_REC ;                      //1024

typedef struct gdb_nmerch_disc_
{
    struct gdb_nmerch_disc_key_        //0    STRUCT  Record key
    {
        unsigned char  opcode;         //    NUM     Opcode - always 9
        unsigned char  nu      ;       //    NU      Not used
        unsigned char  date[3];        //    BCD_D   Date
    }key;

    struct gdb_nmerch_                  //5    STRUCT
    {
        //
        short         dep_no;          //    NUM     Department number
        long          amount;          //    NUM_D   Discount amount
        unsigned char  nu01[14];       //    NU      Not used
    } dep[50];                         //

    unsigned char  nu01[18];           //1005 NU     Not used
    unsigned char  qdx;                //1023 BITS   QuickDex reserved flags
} GDB_NMERCH_DISC_REC;                //1024

```



```
typedef struct gdb_poseft_  
{  
    struct gdb_pos_eft_key_  
    {  
        unsigned char  opcode;           // 0 BCD      Opcode - always 10  
        unsigned char  group;           // 1 NUM      POS number/10  
                                           //          0 - POS   1 - 9  
                                           //          1 - POS  10 - 19  
                                           //          2 - POS  20 - 29  
        unsigned char  nu[3];           // 1 NU      Not used  
    } key;  
  
    struct gdb_pos_eft_data_              // 5 STRUCT  
    {  
        unsigned char  opt_yt_eft_bef_cut :1; // BIT      EFT used yesterday before Cutover  
        unsigned char  opt_yt_eft_aft_cut :1; // BIT      EFT used yesterday after Cutover  
        unsigned char  opt_td_eft_bef_cut :1; // BIT      EFT used today before Cutover  
        unsigned char  opt_td_eft_aft_cut :1; // BIT      EFT used today after Cutover  
        unsigned char  opt_td_cutover      :1; // BIT      Cutover done today  
        unsigned char  opt_td_inquiry      :1; // BIT      Inquiry done today  
        unsigned char  opt_nu              :2; // BIT      Not used  
  
        struct geb_eft_prd_                // STRUCT Yesterday/Today values  
        {  
            long        eftpos_bef_cut_amt; // NUM      EFTPOS before Cutover  
            short       eftpos_bef_cut_cnt; // NUM      EFTPOS before Cutover  
            long        eftpos_aft_cut_amt; // NUM      EFTPOS after  
            short       eftpos_aft_cut_cnt; // NUM      EFTPOS after  
            long        subtotal_amount;    // NUM      subtotal  
            short       subtotal_count ;    // NUM      subtotal Cutover  
            unsigned char  nu[22];          // NU      Not used  
        } yt, td;  
    }  
};
```

```

    unsigned char  nu[19];                //    NU    Not used
} pos[10];

unsigned char      nu01[18];              //1005 NU    Not used
unsigned char      qdx_flag;              //1023 BITS   QuickDex reserved flags
} GDB_POS_EFT;

// Ver 7.3.4
typedef struct gdb_media_ttl_
{
    struct gdb_media_ttl_key_            //    STRUCT   Record key
    {
        unsigned char  opcode;           //0    NUM    Opcode - always 12
        unsigned char  tender_key[2];     //1    BCD    Tender number
        unsigned char  nu01[2];           //3    NU    Not used
    }key;

    // save field for the following period:
    // previous day[0],weekly[1],previous week[2],monthly[3],
    // previous month[4],yearly[5],last Z[6];
    struct media_field_                  //5
    {
        long          value;
        long          count;
        unsigned char  nu02[20];          //    NU    Not used
    }media_field[7];

    unsigned char  nu03[822];              //201 NU    Not used
    unsigned char  qdx;                    //1023 BITS   QuickDex reserved flags
} GDB_MEDIA_TTL;                          //1024

// Ver 7.3.4
typedef struct gdb_include_tax_ttl_
{

```

```
struct gdb_include_tax_ttl_key_          //    STRUCT  Record key
{
    unsigned char  opcode;                //0    NUM    Opcode - always 13
    unsigned char  dept[2];               //1    BCD    Department number
    unsigned char  nu01[2];               //3    NU     Not used
}key;

// save field for the following period:
// today[0],previous day[1],weekly[2],previous week[3],monthly[4],
// previous month[5],yearly[6]
struct include_tax_field_                //5
{
    long          total_tax[MAX_TAX];
    unsigned char  nu02[24];              //    NU     Not used
}include_tax[7];

unsigned char  nu03[626];                 //201  NU     Not used
unsigned char  qdx;                       //1023 BITS  QuickDex reserved flags
} GDB_INCLUDE_TAX_TTL;  //1024

// Ver 7.4.0
struct lift_float_
{
    unsigned char  date[3];               //0
    unsigned char  time[3];               //3
    unsigned char  post;                  //6
    short         lift_count;             //7
    long          lift_amount;            //9
    short         float_count;            //13
    long          float_amount;           //15
    unsigned char  tender;                //19
    unsigned long  supervisor_no;         //20
    unsigned short pos_cashier;           //22
    struct                                     //26
```

```

{
    char pc_pos :1;           //1 - PC, 0 - POS
    char nu      :7;
    }flags;
    char filler[7];           //27
}; //34

```

```
// Ver 7.4.0
```

```
typedef struct gdb_lift_float_ttl_
```

```

{
    struct gdb_lift_float_ttl_key_           //    STRUCT    Record key
    {
        unsigned char  opcode;               //0    NUM      Opcode - always 16
        unsigned short cashier_pos;          //1    NUM      Pos or Cashier number
        unsigned short seq;                  //3    NU       Not used
    }key;

    struct lift_float_ lift_float_data[25]; //5

    unsigned char  nu[168];                  //855  NU       Not used
    unsigned char  qdx;                      //1023 BITS    QuickDex reserved flags
} GDB_LIFT_FLOAT; //1024

```

```
typedef union
```

```

{
    GDB_GNRL          general;
    DEP_WEEK          d_week;
    GDB_DAY_HP_15     day_hp;
    GDB_OPEN_REC      op_cash;
    GDB_ROA_REC        roa;
    GDB_MISC_TTL       misc_ttl;
    GDB_MED_DEPOSIT_REC med_deposit;

```

```
GDB_POUT_REC      pout;
GDB_SAFE_DCLR_REC  safe_lock;
GDB_NMERCH_DISC_REC nmerch;
GDB_POS_EFT        eftpos;
GDB_MEDIA_TTL      media;
GDB_INCLUDE_TAX_TTL tax_ttl;
GDB_DEPOSIT_REC    deposit;
GDB_LIFT_FLOAT      lift_float;
unsigned char      data[1024];
} GDB_REC;
```


// 4 - delete

```
unsigned char number[6];           // 0 6
char          nu01[6];
long credit_limit;                 // 12 4
long credit_limit2;                // 16 4
long previous_limit;               // 20 4
long over_limit_percent;           // 24 4
unsigned char account_type;         // 28 1
unsigned char discount_type;        // 29 1
unsigned int  store_number;          // 30 2
unsigned char language;             // 32 1
unsigned char notes1[80];           // 33 80
unsigned char notes2[80];           // 113 80
unsigned char surname[20];          // 193 20
unsigned char initials[4];          // 213 4
unsigned char title[4];             // 217 4
unsigned char address1[30];         // 221 30
unsigned char address2[30];         // 251 30
unsigned char address3[30];         // 281 30
unsigned int  postalCode;           // 311 2
unsigned char telephone_home[16];   // 313 16
unsigned char telephone_work[16];   // 329 16
unsigned char date_opened[3];       // 345 3
unsigned char nu02[6];              // 348 3
unsigned char last_lim_change_date[3]; // 354 3
long current_installment;           // 357 4
long balance30d;                    // 361 4
long balance60d;                    // 365 4
long balance90d;                    // 369 4
long balance120d;                   // 373 4
char nu03[8];                       // 377 4
long value_current_due;             // 385 4
unsigned char id_number[13];        // 389 13
long opening_balance;               // 402 4
```

```
unsigned char statistics1[6];           // 406 6
unsigned char statistics2[6];           // 412 6
unsigned char statistics3[6];           // 418 6
unsigned char nu04[57];                 // 424 57

// Update by Bt
unsigned char record_status;
unsigned char account_status;
long account_balance;
long value_last_purchase;
long value_last_payment;
unsigned char last_purchase_date[3];
unsigned char last_payment_date[3];
unsigned char filler[9];                // 424 67

// end of debtor record - start of informative
unsigned int user_no;
unsigned char time[3];
unsigned char filler2[283];
unsigned char qdx;
} DEBT_CHANGE;                          // 800 bytes
```


ALTHIST.QDX (100)

File Type	FIFO
Record Size	128
Key Length	0

Record Format

```
#ifdef POSTRANS_H
typedef struct alert_hist_rec_           //OFS  Type   Description
{
    struct alert_key_ key;               //0    STRUCT  The alert key

    union trans_   trn;                 //4    STRUCT  The transaction record

    // confirmation fields
    unsigned char  conf_date[3];        //71   BCD     Confirmed date
    unsigned char  conf_time[3];        //68   BCD     Confirmed time
    unsigned short user_no;             //74   NUM     User number
    unsigned char  conf_nu[8];          //76   NU      Not used
    unsigned short delay_next;          //84   NUM     Delay between alerts (sec)
    char           nu[41];              //86   NU      Not used
    char           qdx;                 //127  BITS    QDEX reserved flags
} ALT_HIST_REC;                        //128
#endif
```

CROSS.QDX (101)

File Type	Index
Record Size	32
Key Length	15

Record Format

```
//KEY-1
#define PLU_PROM          0x20
#define DEPT_PROM         0x21
#define PLU_COUPON        0x22
#define MIXMATCH_PROM     0x23  // V8
#define CROSS_SEGMENT_PROM 0x28  // V8

//KEY-2
#define PROM_PLU          0x30
#define PROM_DEPT         0x31
#define COUPON_PLU        0x32
#define PROM_MIXMATCH     0x33  // V8
#define CROSS_PROM_SEGMENT 0x38  // V8
#define CROSS_CBANK_QUESTION 0x40  // V8
#define CROSS_CASHIER_CBANK 0x42  // V8

struct dept_              // STRUCT
{
    unsigned char  filler[5];    //1  NU    Not used
    unsigned char  dept[2];      //6  BCD   Department key
};

struct prom_              // STRUCT
{
    unsigned char  filler[5];    //1  NU    Not used
    unsigned char  prom[2];      //6  BCD   Promotion number
```



```

/-- GENERAL part -----
unsigned char  opcode;                //0   BCD   Opcode links
                                     //    1- Prom    -> PLU
                                     //    3- Prom    -> Department
                                     //    10- Coupon  -> PLU

                                     //    20-PLU      -> Prom
                                     //    21-Department -> Prom
                                     //    22-PLU      -> Coupon
                                     //
                                     //    30-Prom     -> PLU
                                     //    31-Prom     -> Department
                                     //    32-Coupon    -> PLU
                                     //    40-Checkout bank->customer survey

union
{
    unsigned char  plu[7];            //1   BCD   PLU key

    unsigned char  coupon[7];         //1   BCD   Coupon key

    struct dept_   dept;              //    STRUCT
    struct prom_   prom;              //    STRUCT
    struct cross_cust_q_   cust_q;
    struct cross_cbank_   cbank;
    struct cross_cashier_   cshr;
    struct cross_mixmatch_   mix;
    struct segment_key   segment_key;
}key1;                                //was 'prom_or_coupon'

union
{
    unsigned char  plu[7];            //8   BCD   PLU key
    unsigned char  coupon[7];         //8   BCD   Coupon key

```

```
struct dept_    dept;                //8      Department key
struct prom_    prom;                //8      Promotion number
struct cross_cust_q_    cust_q;
struct cross_cbank_    cbank;
struct cross_mixmatch_    mix;
    struct segment_key    segment_key;
}key2;

                                //was 'plu_or_dept'
struct cross_flags_                //15      STRUCT
{
    unsigned char delete_pending    :1;    //      BIT      Delete in EOD
    unsigned char free_item_plu      :1;    //              This is the Free Item Code PLU
    unsigned char filler              :6;    //      NU      Not used
}flags;

unsigned char    bucket_num;         //16      NUM      Bucket number
struct type_prom_                //17      STRUCT
{
    unsigned char    local_multisaver :1;    //      BIT      Immediate Promotion
    unsigned char    multisaver        :1;    //      BIT      Enhanced Promotion
    unsigned char    linksave          :1;    //      BIT      Group Promotions
    unsigned char    ext_prom          :1;    //      BIT      Order Promotions
    unsigned char    ext_prom_banded   :1;    //      BIT      Order/Levels Promotions
    unsigned char    ext_linksave      :1;    //      BIT      Order/Group Promotions
    unsigned char    free_item         :1;    //      BIT      Free item promotion
    unsigned char    v8_level          :1;    //      BIT      v8 level promotion
}type_prom;

unsigned char    fiiler[13];         //18      NU      Not used
unsigned char    qdx;                //31      BITS     QDEX reserved flags

} CROSS_REC;
```

```
//*****
// DESCRIPTION: Staff Discount
//
//*****
typedef struct staff_disc_rec_
{
    //KEY
    unsigned char  staff_number[10];  //0
    unsigned char  date[3];           //10
    unsigned char  trs_seq;            //13 transaction sequential number
    unsigned long  store_number;       //14
    unsigned char  time[3];           //18
    unsigned int   cashier;            //21
    long           ticket_amount;      //23
    long           discount_amount;    //27
    unsigned char  pos;                //31
    unsigned       ticket_no;          //32

    struct
    {
        unsigned char  opt_swiped   :1;
        unsigned char  opt_nu       :7;
    }option;                        //34

    unsigned char  name[20];           //35
    unsigned char  nu[72];             //55
    unsigned char  qdx;                //127

}STAFF_DISC;                        //128

typedef struct company_member_idx_
{
    unsigned char number[6];           //0 BCD
```

```
    unsigned char nu[3];                //6

}COMPANY_MEMBER_NUM_IDX;              //9

typedef struct company_member_
{
    //KEY
    unsigned char  number[6];           //0
    unsigned char  name[30];            //6
    unsigned char  addres[30];          //36
    unsigned int   zip_code;             //66
    char           telephone[10];       //68
    char           fax_number[10];      //78
    char           tax_number[15];      //88
    char           retail_number[15];   //103
    unsigned char  nu[138];             //118

}COMPANY_MEMBER;                      //256

typedef struct
{
    COMPANY_MEMBER_NUM_IDX    idx;
    COMPANY_MEMBER            rlt;

} COMPANY_MEMBER_NUM_REC;

typedef struct
{
    unsigned char name[9];
    unsigned char number[6];
    unsigned char nu[3];

}COMPANY_MEMBER_NAME_IDX;
```

```
typedef struct
{

    COMPANY_MEMBER_NAME_IDX    key;
    COMPANY_MEMBER              rlt;

} COMPANY_MEMBER_NAME_REC;

typedef struct delivery_notes_header
{
    //KEY
    unsigned char dn_num_by_ven[3];          //0
    unsigned char dn_type;                   //3
    unsigned char vendor_num[3];             //4
    unsigned char counter;                   //7
    unsigned char date[3];                   //8
    unsigned long stamp;                     //11
    unsigned char confirmed;                 //15
    unsigned long total_qty;                 //16
    unsigned long items;                     //20
    unsigned char nu[1000];                  //24

}DELIVERY_HEADER_REC;                      //1024

typedef struct delivery_notes_
{
    //KEY
    unsigned char dn_num_by_ven[3];          //0
    unsigned char dn_type;                   //3
    unsigned char vendor_num[3];             //4
    unsigned char counter;                   //7

    struct plu_sale_                      //8
    {
```



```
unsigned char   plu_num[7];           //0
unsigned long   qty;                  //7
unsigned long   not_used;             //11
unsigned long   actual_qty;           //15   32*31 = 992
unsigned long   bouns;                //19

struct flag1_           //23   STRUCT BIT
{
    unsigned char weighed_item :1;    //   weighed_item
    unsigned char nu           :7;    //
}flag1;

unsigned char   nu[8];                //24

}plu_sale[31];                        //32
//           992+8= 1000

unsigned char nu[24];                 //1000

}DELIVERY_NOTES_REC;                 //1024

typedef struct vendor_store_
{
    unsigned char opcode;              // 0 // 0x01-vendor 0x02-sotre
    unsigned char vnd_str_num[3];      // 1
    char          vnd_str_name[30];    // 4
    char          addres[30];          // 34
    char          phone[20];           // 64
    char          fax_number[20];      // 84
    char          filler[152];         //104
}VND_STR_DATA;                        //256
```

PLUEXT.QDX (105)

File Type	Index
Record Size	256
Key Length	9

Record Format

```
typedef struct plu_ext_
{
    //key
    unsigned char  opcode;                //0 future use opcode
    unsigned char  plu_number[7];        //1 BCD  PLU number
    unsigned char  seq;                  //8 future use sequence

    //Label Details from plu_relat

    unsigned char  lbl_vendor_id[8];      //9  ALPHA  Vendor ID
    unsigned char  lbl_vendor_num[20];    //17 ALPHA  Vendor item number
    unsigned char  lbl_item_size[10];     //37 ALPHA  Item size
    unsigned char  lbl_comparative_uom;    //47 NUM    Comparative UOM

    // 01 Ounce      09 Meter          17 Case
    // 02 Pound      10 Inch           18 Count
    // 03 Liter       11 Foot           19 Each
    // 04 Mililiter   12 Yard           20 Fluid Oz
    // 05 Box         13 Gallon         21 Sq Yard
    // 06 Package     14 Quart          22 Sq Feet
    // 07 Gram        15 Pint           23 Units
    // 08 Kilogram    16 Carton

    Long           lbl_compare_qty;       //48 Num Compare Qty  ( format 7.3 )
    unsigned short lbl_compare_unit;      //52 Num Compare Unit
}
```

```
//new fields
unsigned char   lbl_package_uom;           //54  Num values are like lbl_comparative_uom
unsigned long   lbl_package_size;          //55  Num ( format 7.2 )
unsigned long   lbl_unit_factor;          //59  Num
unsigned char   not_used[193];             //63  Not Used

} PLU_EXT;
```

ASSISTNT.QDX (130)

File Type	Index
Record Size	1024
Key Length	3

Record Format

```
#define ASST_PRD_DAY      0
#define ASST_PRD_WEEK    1
#define ASST_PRD_MONTH   2
#define ASST_PRD_YEAR    3
#define ASST_PRD_LAST_DAY 4
#define ASST_PRD_LAST_WEEK 5
#define ASST_PRD_LAST_MONTH 6

typedef struct asst_rec_
{
    unsigned char number[3];           // 0
    unsigned char name[32];           // 3
    struct                               // 35
    {
        unsigned char status          :1;
        unsigned char filler          :7;
    } flags;

    long          comission;           // 36
    unsigned char address[32];         // 40
    unsigned char telephone[15];       // 72
    unsigned char remarks[32];         // 87
    unsigned char nu_informative[200]; // 119

    // End of informative part. Sales information starts here
```

```
// period figures
struct asst_period_                                // 319
{
    int    customers;

    long   sales;
    int    items;

    long   returns_value;        // 64 * 7 = 448
    int    returns_count;

    long   discounts_value;
    int    discounts_count;

    long   tax_value;
    int    tax_count;

    char   nu[38];
} prd [7];

unsigned char nu_sales[256];                        // 767

unsigned char qdx_flag[1];                          // 1023
} ASSISTNT;                                          // 1024
```

PLUEXT.QDX (131)

File Type	Index
Record Size	256
Key Length	7

Record Format

```
typedef struct plu_extention_rec_  
{  
    unsigned char  plu_number[7]; // 0  
  
    // Sales information - To be updated during EOD  
    //-----  
    long  day_return;           // 7  
    short day_return_qty;       // 11  
    long  day_acct_revenue;     // 13  
    short day_acct_qty;         // 17  
    long  day_acct_return;      // 19  
    short day_acct_return_qty;  // 23  
    long  week_return;          // 25  
    short week_return_qty;      // 29  
    long  week_acct_revenue;    // 31  
    short week_acct_qty;        // 35  
    long  week_acct_return;     // 37  
    short week_acct_return_qty; // 41  
    long  month_return;         // 43  
    short month_return_qty;     // 47  
    long  month_acct_revenue;   // 49  
    short month_acct_qty;       // 53  
    long  month_acct_return;    // 55  
    short month_acct_return_qty; // 59  
    long  p_year_revenue;       // 61
```

```
long  day_acct_discount;      // 65
char  day_nu[26];             // 69
long  week_acct_discount;     // 95
char  week_nu[26];            // 99
long  month_acct_discount;    // 125
char  month_nu[26];           // 129
char  year_nu[30];            // 155
char  reserved[22];           // 185
long  month_sales[12];        // 207
unsigned char  qdx;           // 255
} PEXT_REC;                   // 256
```

MESSAGE.QDX (167)

File Type	Remote Messages (such as RPO control checks)
Record Size	1024
Key Length	30

Record Format

```
typedef struct RemoteMessage_
{
    /****BEGIN KEY***/
    unsigned char opcode[2];           //0   BCD
    unsigned char pos_num[2];          //2   BCD
    unsigned char trans_id[3];         //4   BCD
    unsigned char date[3];             //7   BCD//yyymmdd
    unsigned char time[3];             //10  BCD//hhmmss
    unsigned char privilege[2];        //13  BCD
    unsigned char rpo_nu[15];          //15
    /****END KEY***/

    unsigned short cashier_id;         //30
    unsigned char event_type;          //32  1==CC 2==K/B func
    unsigned short event_id;           //33
    unsigned char RPO_action;          //35  0==None 1==Inform Only 2==Mgr Appr Reqr'd
    unsigned short approved_by;        //36
    unsigned char status;              //38  0==New 1==Appr'd by RPO 2==Will Manage @ POS

    /*******
    //TICKET INFO
    /*******
    long ticket_total;                //39
```



```
struct
{
unsigned char tender_name[16];
long tender_amount;
}tender_info[4];           //43
long balance_due;          //123
short items_in_ticket;     //127
unsigned char ticket_type[6]; //129 SALE,RETURN,WIC
long discount_total;       //135
long total_refund_qty;     //139
long total_refund_amount;  //143
long ticket_subtract_amount; //147
long ticket_cancel_amount; //151
short coupons_in_ticket;   //155
unsigned char membercard_number[20]; //157
unsigned char membercard_name[20];   //177
long ticket_points;           //197

//*****
//ITEM INFO
//*****
unsigned char item_number[7]; //201 BCD
char item_name[20];          //208
unsigned char item_dept_name[16]; //228
long item_price;             //244
short item_MSU;              //248
unsigned char linked_item_name[20]; //250

//*****
//ADDITIONAL INFO
//*****
long max_negative_amount;    //270
short max_negative_percent;  //274
long scan_limit;            //276
```

```
long max_amount;                //280
short age_required;              //284
long department_limit;          //286
long limit_count;               //290
unsigned char nu_rpo_buffer[729]; //294
unsigned char Qdx;              //1023

} REMOTE_MESSAGE;
```

HOTCARD.QDX (205)

File Type	WIC Hot Card List
Record Size	256
Key Length	23

Record Format

```
struct CHotCard_  
{  
    char PAN[10];           // 0    BCD Account Number  
    char StateFIPS;         //10   Numeric state code 0-99  
    char KeyFiller[12];     //11   Key filler  
    char LockDate[3];      //23   BCD Effective Date YYMMDD  
    char LockTime[3];      //26   BCD Effective Date HHMMSS  
    char nuDateTime;       //29   not used  
    char Reason[4];        //30   Reason card on list  
    char LastUPD[4];       //34   BCD Date of Last Import File  
    char StateCode[2];     //38   State Code  
    char NU[215];          //40   Not Used  
    char QDXflag;          //255  Quickdex Flag
```

WICITEMS.QDX (206)

File Type	APL Wic Items from the state
Record Size	256
Key Length	23

Record Format

```
typedef struct CWICItem_
{
    char state[1];           // 0   BCD State Code
    unsigned char UPC[7];    // 1   BCD UPC Number
    char Category[2];        // 8   BCD Category
    char SubCategory[2];     //10   BCD Subcategory
    char KeyFiller[11];      //12   Key filler
    char nu_UOM[5];          //23   not used
    long PakSize;            //28   Pack Size
    long BenefitQty;         //32   Quantity of benefit units required to
                                //    redeem this product.
    char BenefitUnit[5];     //36   Descrip. of Unit of Measure for benefits
    long Price;              //41   Reimbursement Price
    char PriceType;          //45   Price Type
                                //    00 - Price not disclosed
                                //    01 - Max Price
                                //    02 - Contract Price
                                //    03 - Cash Value Voucher
                                //    03-25 Reserved for national use
                                //    26-99 Reserved for Private Use
    char StartDate[3];       //46   BCD YYMMDD Date
    char ExpDate[3];         //49   BCD YYMMDD Date
    unsigned char LinkUPC[7]; //52   BCD UPC Number
    char NU[196];            //59   Not Used
    char QDXflag;            //255  Quickdex Flag
}
```

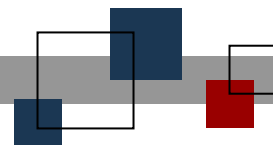
```
} WIC_ITEM;
```

WICCAT.QDX (207)

File Type	WIC Categories
Record Size	256
Key Length	23

Record Format

```
struct CWICCategory_  
{  
    char StateFIPS;           // 0      BCD State Code  
    char Category[2];        // 1      BCD Category  
    char SubCategory[2];     // 3      BCD Subcategory  
    char KeyFiller[18];      // 5      Key filler  
    char UOM[6];             //23     Unit of Measure abbr.  
    char Description[30];    //29     Category description  
    char NU[196];            //59     Not Used  
    char QDXFlag;            //255    Quickdex Flag  
};
```

© StoreNext Retail Technologies LLC 2010

StoreNext Retail Technologies LLC endeavors to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission.

The development of StoreNext products and services is continuous and published information may not be up to date. It is important to check the current position with StoreNext. This document is not part of a contract or license save insofar as may be expressly agreed.