# CONTENTS Date 24/07/25

**Definition for Tables**

1. K\_MEAL
2. K\_PRO
3. K\_REV
4. K\_WK\_VAT
5. PDVAT\_TB
6. PDITEM<n> where n is a shop code
7. COMBTB<n> where n is a shop code
8. ACODES
9. BCODES
10. COMP\_PRO
11. OPT\_PRO
12. P\_CHOICE
13. ST\_ITEMS
14. APP\_COMB
15. APP\_PROD
16. GROUP\_TB
17. MISC\_SEC
18. COMB\_EXT
19. PROD\_EXT
20. SHOPS\_TB

**FILES Required by Uncle Sams for each shop and each day of week**

1. MP<ddmmyy>.CSV for date dd/mm/yy
2. PD<ddmmyy>.CSV for date dd/mm/yy
3. RV<ddmmyy>.CSV for date dd/mm/yy
4. K\_WK\_VAT.CSV for the week

**FILES provided to the shop at the start of a week**

1. PDITEM<n>.CSV where n is a shop code eg PDITEM15.CSV for Crawley Shop
2. COMBTB<n>.CSV where n is a shop code eg COMBTB15.CSV for Crawley Shop
3. ACODES.CSV (\*)
4. BCODES.CSV (\*)
5. COMP\_PRO.CSV (\*)
6. OPT\_PRO.CSV (\*)
7. P\_CHOICE.CSV (\*)
8. PDVAT\_TB.CSV (\*)

1. ST\_ITEMS.CSV (\*)
2. APP\_COMB.CSV (\*)
3. APP\_PROD.CSV (\*)
4. GROUP\_TB.CSV (\*)
5. MISC\_SEC.CSV (\*)
6. COMB\_EXT.CSV (\*)
7. PROD\_EXT.CSV (\*)
8. SHOPS\_TB.CSV (\*)

# Definition for Table K\_MEAL

### Field Name Type

* PRODNUMB Integer

Product code.

* TAKEAWAY Integer

Total number of this product sold as part of takeaway meals on the given date.

* EATIN Integer

Total number of this product sold as part of eatin meals on the given date.

# Remarks

1. Table K\_Meal is a template for MP<ddmmyy>, where dd/mm/yy is the given date. We store the number of meals sold on date dd/mm/yy in table MP<dd/mm/yy>.
2. Sales of meal products are also added to the cumulative sales of the corresponding products in table PD<ddmmyy>, where dd/mm/yy is the given date. Table K\_PRO is a template for PD<ddmmyy>.
3. There is a record for each product at the shop in table K\_Meal.

# Definition for Table K\_PRO

### Field Name Type

* PRODNUMB Integer

If COMBO is False then PRODNUMB is a product code.

If COMBO is True then PRODNUMB is a Combination product code.

* COMBO Boolean

See definition for PRODNUMB.

* TAKEAWAY Integer

Total number of this product sold as takeaway on the given date.

* EATIN Integer

Total number of this product sold as eatin on the given date.

* WASTE Integer

Total number of this product entered as Cooked Waste on the given date.

* STAFF Integer

Total number of this product entered as Crew Food on the given date.

* OPTION Integer

Total number of this product chosen as optional product for products (not for combination products) on the given date.

# Remarks

1. Table K\_PRO is a template for PD<ddmmyy>, where dd/mm/yy is the given date.

We store the number of products sold on date dd/mm/yy in table PD<ddmmyy>.

1. There is a record for each product at the shop in table PD<ddmmyy>.

There is a record for each combination product at the shop in table PD<ddmmyy>.

# Definition for Table K\_REV

### Field Name Type

* TCASHVAL Long

Amount of cash (in pence) in drawer recorded by POS on the given date.

* TCHQVAL Long

Amount of cheques (in pence) in drawer recorded by POS on the given date.

* TCARDVAL Long

Amount of credit/debit card payments in pence recorded by POS on the given date.

* TONACCOUNT Long

Amount of “on account” payments in pence recorded by POS on the given date.

* TSTAFFVAL Long

Value of crew food in pence recorded by POS on the given date.

* TWASTEVAL Long

Value of cooked waste in pence recorded by POS on the given date.

* TCOUPVAL Long

Value of coupons in pence recorded by POS on the given date.

* TPAYOUTVA Long

Amount of paid outs in pence recorded by POS on the given date.

Whenever the cashier has done a paid out, the system will add the amount of that paid out to TPAYOUTVA and deduct the amount of that paid out from TCASHVAL

* TTOKENVAL Long

Value of vouchers in pence recorded by POS on the given date.

* TDISCNTVA Long

Amount of combination discount in pence recorded by POS on the given date.

A = Total amount due for all the products in the order in pence

B = Total amount due for all the combination products in the order in pence

Combination discount for an order = A – B

# Definition for Table K\_REV

### Field Name Type

* TTOKENNOVR Long

Voucher overage in pence recorded by POS on the given date.

* TGOLARGENU Integer

Number of go large recorded by POS on the given date.

* TMEAL\_DISCNT Long

Amount of discount for meals in pence recorded by POS on the given date.

Total discount given to the products (Burger, Drink, Fries) ordered as part of a meal.

* ACTCASH Long

Actual amount of cash taken by POS in pence on the given date.

* ACTCHQ Long

Actual amount of cheques taken by POS in pence on the given date.

* ACTCARD Long

Actual amount of credit/debit card payments taken by the card machine in pence on the given date.

* VAT Long

Total amount of VAT due in pence on the given date.

* XPV Long

Extended product value in pence on the given date.

Data in this column is not required.

# Remarks

1. Table K\_REV is a template for RV<ddmmyy>, where dd/mm/yy is the given date.

For example, Table RV210725 is for date 21/07/25.

1. There is only one record in table RV<ddmmyy>.

# Key

POS – Point of Sales System

# Definition for Table K\_WK\_VAT

### Field Name Type

* VAT\_CLASS Byte

Vat class.

* VAT\_RATE Single

Vat rate for this vat class.

* TOT\_VAT\_1 Single

Total amount of VAT due in pound for all the products which were sold at this VAT rate on Monday. This figure does not include any VAT due for the Deleted orders, Crew food and Cooked waste.

* TOT\_VAT\_2 Single

Total amount of VAT due in pound for all the products which were sold at this VAT rate on Tuesday. This figure does not include any VAT due for the Deleted orders, Crew food and Cooked waste.

Similarily for TOT\_VAT\_<n> where n=3,4,5,6,7.

* TOT\_VAT\_3 Single
* TOT\_VAT\_4 Single
* TOT\_VAT\_5 Single
* TOT\_VAT\_6 Single
* TOT\_VAT\_7 Single
* T\_VAL\_EXCLVAT\_1 Single

Total value of products excluding VAT in pound sold at this VAT rate on Monday. This figure does not include Deleted orders, Crew food and Cooked waste.

* T\_VAL\_EXCLVAT\_2 Single

Total value of products excluding VAT in pound sold at this VAT rate on Tuesday. This figure does not include Deleted orders, Crew food and Cooked waste.

Similarily for T\_VAL\_EXCLVAT\_<n> where n=3,4,5,6,7.

* T\_VAL\_EXCLVAT\_3 Single
* T\_VAL\_EXCLVAT\_4 Single

# Definition for Table K\_WK\_VAT

### Field Name Type

* T\_VAL\_EXCLVAT\_5 Single
* T\_VAL\_EXCLVAT\_6 Single
* T\_VAL\_EXCLVAT\_7 Single

# Remarks

1. Table K\_Wk\_Vat stores values of VAT at different rates on each day of the week. There is one record for each VAT rate.
2. Table PdVat\_Tb stores the details of VAT rates. There is one record for each VAT rate.
3. We initialize table K\_Wk\_Vat when we start a new week. For each record in table PDVat\_Tb, we add a record to table K\_Wk\_Vat.
4. We update table K\_Wk\_Vat on each day of the week.

# Definition for Table PDVAT\_TB

### Field Name Type

* VAT\_CLASS Byte

Vat class.

* VAT\_RATE Single

Vat rate for this vat class.

* VAT\_DESC Text, 20

Description of this vat class.

# Remarks

1. There are 2 records in table PdVat\_Tb

VAT\_CLASS VAT\_RATE VAT\_DESC

0 0 Zero rate

1 20 Standard rate

If there is another vat rate, we just have to add a record to table PdVat\_Tb.

# Definition for Table PDITEM<n>

### Field Name Type

* PRODNUMB Integer

Product code.

* PRODNAME Text, 16

Product name.

* EAT\_VAT\_CLASS Byte

Vat class for this product in an Eatin order.

We can find out the VAT rate by looking up the VAT class in table PDVat\_Tb. Use this VAT rate to calculate the non-Vat price of this product in an Eatin order.

* TAKE\_VAT\_CLASS Byte

Vat class for this product in a Takeaway order.

We can find out the VAT rate by looking up the VAT class in table PDVat\_Tb. Use this VAT rate to calculate the non-Vat price of this product in a Takeaway order.

* READBACK\_ORD Byte

Read Back sorting order.

Read Back sorting method:

Set Meal Product to True if the given product is ordered as a standard meal or kids meal.

Items in the order are sorted in ascending order of ReadBack\_ord and then Meal Product. After the items in an order have been sorted by Read Back sorting method, they are displayed in the Read Back Box.

There are 3 sections in the Read Back Box.

Items with readback\_ord between 1 and 85 inclusive appear in section 1 of the Read Back Box.

Items with readback\_ord between 86 and 170 inclusive appear in section 2 of the Read Back Box.

Items with readback\_ord between 171 and 255 inclusive appear in section 3 of the Read Back Box.

In each section, meal products are displayed before single products.

Lowest number appears highest on the list of that section.

If the number is 0 then that item will follow the previously rung in item (used for ‘extras’).

# Definition for Table PDITEM<n>

### Field Name Type

We have set up the Read Back sorting order in the following way

Top section for

1. Main courses
2. Nuggets

Middle section for

1. Fries
2. Onion Rings
3. Mushrooms
4. Desserts

Bottom section for

1. Hot drinks
2. Shakes
3. Fizzy

* MEAL\_ONLY Boolean

True if this product can only be sold as part of meal.

* MEAL\_CODE Byte

1. – This product can’t be orderd as part of meal
2. - This product can be ordered as part of Standard Meal
3. - This product can be ordered as part of Kids Meal

* MEAL\_DRINK Byte

1. - This product is not a drink or This drink is not available for meals
2. – This product is a drink for Standard Meal
3. – This product is a drink for Kids Meal

* T\_DRINK\_CD Integer

Product code of trade up drink.

It is the product code of the corresponding large drink if this product is a drink and can go large. Otherwise set it to 0.

* VATPR Integer

Current price including VAT in pence.

This is standard price.

* DC\_VATPR Integer

Current discounted price for meal product including VAT in pence

This is standard price.

# Definition for Table PDITEM<n>

### Field Name Type

* VATPR\_2 Integer

Current price including VAT in pence.

This is price band 2.

Similarly for VATPR\_<n> where n=3,4,5,6.

* DC\_VATPR\_2 Integer

Current discounted price for meal product including VAT in pence

This is price band 2.

Similarly for DC\_VATPR\_<n> where n=3,4,5,6.

* VATPR\_3 Integer
* DC\_VATPR\_3 Integer
* VATPR\_4 Integer
* DC\_VATPR\_4 Integer
* VATPR\_5 Integer
* DC\_VATPR\_5 Integer
* VATPR\_6 Integer
* DC\_VATPR\_6 Integer

# Remarks

1. Table PdItem<n> is for the shop with shop code <n>.

For example, PDItem15 is for Crawley shop,

PDItem9 is for Broadwater shop.

1. It contains all the products available at shop <n>. There is one record for each product.

# Definition for Table COMBTB<n>

### Field Name Type

* COMBONUMB Integer

Combination product code.

* DESC Text, 16

Name of this combination product.

* T\_COMB\_NUM Integer

The combination product code for the corresponding trade up combination product. Set it to 0 if there is no corresponding trade up combination product.

* EAT\_VAT\_CLASS Byte

Vat class for this combination product in an Eatin order.

We can find out the VAT rate by looking up the VAT class in table PDVat\_Tb. Use this VAT rate to calculate the non-Vat price of this combination product in an Eatin order.

* TAKE\_VAT\_CLASS Byte

Vat class for this combination product in a Takeaway order.

We can find out the VAT rate by looking up the VAT class in table PDVat\_Tb. Use this VAT rate to calculate the non-Vat price of this combination product in a Takeaway order.

* VATPR Integer

Current price including vat in pence of this combination product. This is standard price.

* T\_VATPR Integer

Current price including vat in pence of the corresponding trade up combination product.

Set it to 0 if there is no corresponding trade up combination product.

This is standard price.

* VATPR\_2 Integer

Current price including vat in pence of this combination product.

This is price band 2.

Similarly for VATPR\_<n> where n=3,4,5,6.

* T\_VATPR\_2 Integer

Current price including vat in pence of the corresponding trade up combination product.

Set it to 0 if there is no corresponding trade up combination product.

This is price band 2.

Similarly for T\_VATPR\_<n> where n=3,4,5,6.

# Definition for Table COMBTB<n>

### Field Name Type

* VATPR\_3 Integer
* T\_VATPR\_3 Integer
* VATPR\_4 Integer
* T\_VATPR\_4 Integer
* VATPR\_5 Integer
* T\_VATPR\_5 Integer
* VATPR\_6 Integer
* T\_VATPR\_6 Integer

# Remarks

1. Table CombTb<n> is for the shop with shop code <n>.

For example, CombTb15 is for Crawley shop,

CombTb9 is for Broadwater shop.

1. It contains all the combination products available at shop <n>. There is one record for a combination product.

# Definition for Table ACODES

### Field Name Type

* PRODNUMB Integer

Product code.

* ST\_CODENUM Integer

Code of a ‘A’ code stock component for the product with Product Code = PRODNUMB

* QTY Single

Quantity of this ‘A’ code stock component for this product.

# Remarks

1. If a product has <n> ‘A’ code stock components, then add <n> records to table Acodes. Each of these records represents a ‘A’ code stock component for the product.

# Definition for Table BCODES

### Field Name Type

* PRODNUMB Integer

Product code.

* ST\_CODENUM Integer

Code of a ‘B’ code stock component for the product with Product Code = PRODNUMB

* QTY Single

Quantity of this ‘B’ code stock component for this product.

# Remarks

1. If a product has <n> ‘B’ code stock components, then add <n> records to table BCodes. Each of these records represents a ‘B’ code stock component for the product.

# Definition for Table COMP\_PRO

### Field Name Type

* COMBONUMB Integer

Combination product code.

* PRODNUMB Integer

Product code of a compulsory product for this combination product.

* T\_PRODNUMB Integer

Set it to 0 if there is no trade up combination product for this combination product. Otherwise, it is the product code which will replace the product with the product code PRODNUMB for the corresponding trade up combination product.

# Remarks

1. There may be more than 1 record for a combination product.

For example, if there are 3 compulsory products for a given combination product, then there are 3 records for the given combination product.

# Definition for Table OPT\_PRO

### Field Name Type

* COMBONUMB Integer

Combination product code.

* PRODNUMB Integer

Product code of an optional product for this combination product.

* T\_PRODNUMB Integer

Set it to 0 if there is no trade up combination product for this combination product. Otherwise, it is the product code which will replace the product with the product code PRODNUMB for the corresponding trade up combination product.

# Remarks

1. There may be more than 1 record for a combination product.

For example, if there are 3 optional products for a given combination product, then there are 3 records for the given combination product.

# Definition for Table P\_CHOICE

### Field Name Type

* PRODNUMB Integer

Product code.

* OPT\_PRODNUMB Integer

Product code of an optional product for the product with product code PRODNUMB.

# Remarks

1. We add records to table P\_Choice for each product that has optional products. If the product has <n> optional products, then we add <n> records to table P\_Choice for the product. Each of these records represents an optional product for the product.

# Definition for Table ST\_ITEMS

### Field Name Type

* CODEALPH Text, 1

A stock item code consists of two parts, a letter and then followed by an integer.

CODEALPH is the first part of a stock item code.

* ST\_CODENUM Integer

Second part of a stock item code. It is a 4-digit number.

* ITEM Text, 18

Description for this stock item.

# Remarks

1. Table ST\_Items contains records of A and B codes stock items.

# Definition for Table APP\_COMB

### Field Name Type

* COMBONUMB Integer

Combination product code.

* DESC Text, 16

Name of this combination product.

* GROUP\_ID Byte

Section number under which this combination product is printed on the report

“App Product Prices – Band <n>”, where n = 1,2, … ,6.

GROUP\_ID = 0 means that this combination product is not printed on the report

“App Product Prices – Band <n>”.

* GROUP\_SUB\_ID Integer

The order in which this combination product is printed in section <GROUP\_ID> on the report

“App Product Prices – Band <n>”.

Lowest number appears highest in the list.

# Remarks

1. Table App\_Comb contains all the combination products at Uncle Sams. There is one record for each combination product.

# Definition for Table APP\_PROD

### Field Name Type

* PRODNUMB Integer

Product code.

* PRODNAME Text, 16

Product name.

* GROUP\_ID Byte

Section number under which this product is printed on the report

“App Product Prices – Band <n>” where n = 1,2,..,6.

* GROUP\_SUB\_ID Integer

The order in which this product is printed in section <GROUP\_ID> on the report

“App Product Prices – Band <n>”.

Lowest number appears highest in the list.

* MEAL\_ID Byte

Section number under which the meal price of this product is printed on the report

“App Product Prices – Band <n>”.

MEAL\_ID = 0 means that the meal price of this product is not printed on the report

“App Product Prices – Band <n>”.

* MEAL\_SUB\_ID Integer

The order in which this product is printed in section <MEAL\_ID> on the report

“App Product Prices – Band <n>”.

Lowest number appears highest in the list.

* DOUBLE\_PDNUMB Integer

If PRODNUMB is a single product, then DOUBLE\_PDNUMB is the corresponding double product for this product.

* TRIPLE\_PDNUMB Integer

If PRODNUMB is a single product, then TRIPLE\_PDNUMB is the corresponding triple product for this product.

# Remarks

1. Table App\_Prod contains all the products (not combination products) at Uncle Sams.There is one record for each product.
2. Meal price of a single product

= Discounted price for this single product ( eg Chesse Burger ) +

Discounted price for Regular Fries +

Discounted price for Fizzy Reg (BASE)

We have set product “49 Cola” as the Fizzy Reg (BASE)

1. Meal price of a double product

= Discounted price for this double product ( eg Double Chesse Burger ) +

Discounted price for Regular Fries +

Discounted price for Fizzy Reg (BASE)

1. Meal price of a triple product

= Discounted price for this triple product ( eg Triple Chesse Burger ) +

Discounted price for Regular Fries +

Discounted price for Fizzy Reg (BASE)

1. Meal price of a kid hamburger

= Discounted price for this kid hamburger ( eg kid chesse burger ) +

Discounted price for Regular Fries +

Discounted price for Fizzy Kids (BASE)

We have set product “89 Cola small” as the Fizzy Kids (BASE)

# Definition for Table GROUP\_TB

### Field Name Type

* GROUP\_ID Byte

Section number on the report “App Product Prices – Band <n>” where n = 1,2,..,6.

* GROUP\_NAME Text, 16

Section name, eg “Chicken”, “Fries and Sides”.

* SOURCE\_TYPE Text, 1

If SOURCE\_TYPE = ”P”, then this section is for the products (not the combination products),

eg Cheese burger, Chicken AllStar.

If SOURCE\_TYPE = ”C”, then this section is for the combination products,

eg “Sharing platter”.

* MEAL\_GROUP Byte

Section number for meal products. If MEAL\_GROUP is greater than 0, then we list the meal prices of all the products with MEAL\_ID (Field of table App\_Prod) = MEAL\_GROUP under section number <MEAL\_GROUP>.

# Remarks

1. Table Group\_TB does not contain all the sections on the report “App Product Prices – Band <n>” .

There are 2 more sections:

Section 10 – Miscellaneous

Section 11 – Class Prices of Fries and Drinks

Section 11 is included on the report for us to check the data.

# Definition for Table MISC\_SEC

### Field Name Type

* PRODNUMB Integer

Product Code. We set PRODNUMB to 0 if this item is not a product in our product file.

* PRODNAME Text, 16

Product name. We set PRODNAME to blank if this item is not a product in our product file.

* ITEM\_DESC Text, 20

If PRODNUMB > 0 then ITEM\_DESC is same as PRODNAME.

ITEM\_DESC is the description printed in column “Product Name” in

Section 10 – Miscellaneous. We print the price of this product for the chosen price band in column “Price”.

If PRODNUMB = 0 then ITEM\_DESC is not a product name. We need to calculate the figure and print it in column “Price”.

Eg, for ITEM\_DESC = “Shake Instead”, we need to calculate the figure for Shake Instead.

Shake Instead = Shake Reg – Fizzy Reg (BASE)

* SEQ\_ORDER Integer

The order in which this item is printed in Section 10 – Miscellaneous on the report

“App Product Prices – Band <n>”, where n = 1,2,…,6.

Lowest number appears highest in the list.

# Remarks

1. Table MISC\_SEC contains all the items to be printed in Section 10 - Miscellaneous on the report “App Product Prices – Band <n>”, where n = 1,2,…,6.

# Definition for Table COMB\_EXT

### Field Name Type

* COMBONUMB Integer

Combination product code.

* DESC Text, 16

Name of this combination product.

* NAME\_ON\_MENU Text, 30

Name shown on the menu.

* PROD\_DESC Text, 255

Description of this combination product.

# Remarks

1. Table Comb\_Ext contains all the combination products at Uncle Sams. There is one record for each combination product.

# Definition for Table PROD\_EXT

### Field Name Type

* PRODNUMB Integer

Product code.

* PRODNAME Text, 16

Name of this product, eg “DBL CHEESEBURGER”.

* NAME\_ON\_MENU Text, 30

Name shown on the menu, eg “Double Cheese Burger”.

* COOK\_ZONE Byte

Cooking zone.

Code Cooking zone

1. Product is not shown on the Cooking Zone
2. PREP
3. DRINKS
4. FRIER

* PROD\_DESC Text, 255

Description of this product.

# Remarks

1. Table Prod\_Ext contains all the products at Uncle Sams. There is one record for each product.

# Definition for Table SHOPS\_TB

### Field Name Type

* SHOP\_CODE Byte

Shop code.

* SHOP\_NAME Text, 10

Shop Name, eg “Crawley”.

* SHOP\_ABBREV Text, 2

Abbreviation of the shop name, eg “CW” for Crawley shop.

* FRANCHISEE Text, 25

Name of the franchisee.

* FULL\_NAME Text, 32

Name of the company, eg “T/A Uncle Sams Hamburger Express”.

* ADDRESS\_1 Text, 22

First line of the shop address.

* ADDRESS\_2 Text, 20

Second line of the shop address.

* ADDRESS\_3 Text, 17

Third line of the shop address.

* POST\_CODE Text, 8

Postcode of the shop address.

# Remarks

1. Table Shops\_Tb contains records of the shops which have your POS System.

----- END -----