

```

*-----*
*                               *
*                               *
*-----*

```

Internal Tables

```

DATA: wa_zsd_d13_di    TYPE /bic/azsd_d1300,
      wa_0sd_o01_oi    TYPE /bi0/asd_o0100,
      wa_zsd_d04_od    TYPE /bic/azsd_d0400,
      wa_Mat_Unit      TYPE /BIC/PZMAT_UNIT,
      wa_Material       TYPE /BIC/PZMATERIAL.

```

```

*-----*
*                               *
*                               *
*-----*

```

Temporary Variables

```

DATA: l_tabix          TYPE sy-tabix,
      WZSDUKEDQTY      TYPE /bi0/oicopaslqty,
      WZSDUKERQTY      TYPE /bi0/oicopaslqty,
      w1_mat_unit      TYPE /BIC/PZMAT_UNIT-/BIC/ZMAT_UNIT,
      w1_denomintr     TYPE /BIC/PZMAT_UNIT-denomintr,
      w1_numerator     TYPE /BIC/PZMAT_UNIT-numerator,
      w2_mat_unit      TYPE /BIC/PZMAT_UNIT-/BIC/ZMAT_UNIT,
      w2_denomintr     TYPE i,
      w2_numerator     TYPE i.

```

```

DATA : WZSDUKEIQTY    TYPE /bi0/oicopaslqty,
      w1_base_uom      TYPE /bic/pzmaterial-base_uom,
      w1_matl_type     TYPE /bic/pzmaterial-matl_type,
      w1_zpackuvml     TYPE /bic/pzmaterial-/bic/zpackuvml,
      w1_zuntprcs      TYPE /bic/pzmaterial-/bic/zuntprcs,
      w1_zcnv_80z      TYPE /bic/pzmaterial-/BIC/ZCNV_80Z,
      w1_zcs_palt      TYPE /bic/pzmaterial-/BIC/ZCS_PALT,

      w2_zpackuvml     TYPE /bic/pzmaterial-/bic/zpackuvml,
      w2_zuntprcs      TYPE /bic/pzmaterial-/bic/zuntprcs,
      w2_zcnv_80z      TYPE /bic/pzmaterial-/BIC/ZCNV_80Z,
      w2_zcs_palt      TYPE /bic/pzmaterial-/BIC/ZCS_PALT,

      w3_mat_unit      TYPE /BIC/PZMAT_UNIT-/BIC/ZMAT_UNIT,
      w3_zpackuvml     TYPE i,
      w3_zuntprcs      TYPE i,
      w3_zcnv_80z      TYPE STRING,
      w3_zcs_palt      TYPE i,
      w3_denomintr     TYPE i,
      w3_numerator     TYPE i.

```

```

*-----*
*                               *
*                               *
*-----*

```

Constant Definitions

```

CONSTANTS: C_USC       TYPE C LENGTH 3 VALUE 'USC',
           C_ZERO      TYPE I VALUE '0',
           C_1         TYPE STRING VALUE '1.485',
           C_2         TYPE I VALUE '1000',
           C_PAL       TYPE C LENGTH 3 VALUE 'PAL',
           C_GAL       TYPE C LENGTH 3 VALUE 'GAL',
           C_GLL       TYPE C LENGTH 3 VALUE 'GLL',
           C_FERT      TYPE C LENGTH 4 VALUE 'FERT',
           C_HALB      TYPE C LENGTH 4 VALUE 'HALB',

```

```

                C_NUM          TYPE STRING VALUE '.0123456789'.
*-----*
*                               Work-Areas                               *
*-----*
DATA:      rp          TYPE   _ty_s_tg_1.
*-----*
*                               Read all Sales Order Details from ZSD_001 DSO          *
*-----*

CLEAR: wa_zsd_d13_di,
      wa_0sd_o01_oi,
      l_tabix.
If i_0sd_o01_oi[] is initial.
  SELECT * INTO table i_0sd_o01_oi
  FROM /bi0/asd_o0100.
endif.
*-----*
*                               Populate iMat_Unit internal table from ZMAT_UNIT InfoObject      *
*-----*

if iMat_Unit[] is initial.
  SELECT *
        FROM /BIC/PZMAT_UNIT
        INTO table iMat_Unit
        WHERE OBJVERS = 'A'.
endif.
*-----*
*                               Populate iMaterial internal table from ZMATERIAL InfoObject      *
*-----*

if iMaterial[] is initial.
  SELECT *
        FROM /BIC/PZMATERIAL
        INTO table iMaterial
        WHERE OBJVERS = 'A' and
              ( ( MATL_TYPE = C_FERT ) or
                ( MATL_TYPE = C_HALB ) ).
endif.
*-----*
*                               Target Data-Package Processing                               *
*-----*

LOOP AT RESULT_PACKAGE INTO rp.
  l_tabix = sy-tabix.
*-----*
*Check with Order item DSO and get all required fields for open order
*-----*

  read table i_0sd_o01_oi into wa_0sd_o01_oi
    with table key doc_number = rp-DOC_NUMBER
                  s_ord_item = rp-S_ORD_ITEM.

  IF sy-subrc = 0.
    rp-/bic/zmat_plnt = wa_0sd_o01_oi-material.
    rp-customer       = wa_0sd_o01_oi-sold_to.
    MODIFY RESULT_PACKAGE FROM rp INDEX l_tabix.
  ENDIF.
*-----*

```

```

*           End of Sales Order Details - Data Processing           *
*-----*
*           Start of UK Equivalent Qty - Data Processing           *
*           for "UK Equivalent Requirement Qty"                   *
*           And "UK Equivalent Delivery Qty"                       *
*-----*
*           Clear all temporary variables                           *
*-----*

CLEAR : w1_mat_unit, w1_denomintr, w1_numerator,
        w2_mat_unit, w2_denomintr, w2_numerator.

CLEAR : w1_zpackuvml, w1_zuntprcs, w1_zcnv_80z, w1_zcs_palt,
        w1_mat_unit, w1_denomintr, w1_numerator,
        w2_zpackuvml, w2_zuntprcs, w2_zcnv_80z, w2_zcs_palt,
        w2_mat_unit, w2_denomintr, w2_numerator,
        w3_zpackuvml, w3_zuntprcs, w3_zcnv_80z, w3_zcs_palt,
        w3_mat_unit, w3_denomintr, w3_numerator.
*-----*
*           Select UK Unit (USC) attributes from ZMAT_UNIT       *
*-----*

read table iMAT_UNIT
        INTO wa_Mat_Unit
with table blue /bic/zmaterial = rp-/bic/zmaterial
        /BIC/ZMAT_UNIT = C_USC.

IF sy-subrc = 0.
w1_mat_unit   = wa_Mat_Unit-/BIC/ZMAT_UNIT.
w1_denomintr  = wa_Mat_Unit-DENOMINTR.
w1_numerator  = wa_Mat_Unit-NUMERATOR.

w2_mat_unit   = w1_mat_unit.
w2_denomintr  = w1_denomintr.
w2_numerator  = w1_numerator.

IF w2_numerator NE C_ZERO AND w2_denomintr NE C_ZERO.
*-----*
*           Calculation of Delivery Quantity                       *
*-----*
WZSDUKEDQTY = ( w2_denomintr / w2_numerator ) *
              rp-/BIC/ZSDUKEDQY.
*-----*
*           Calculation of Requirement Quantity                   *
*-----*
WZSDUKERQTY = ( w2_denomintr / w2_numerator ) *
              rp-/BIC/ZSDUKERQY.
*-----*

if not ( WZSDUKEDQTY = 0 ).
    rp-/BIC/ZSDUKEDQY = WZSDUKEDQTY.
endif.
if not ( WZSDUKERQTY = 0 ).
    rp-/BIC/ZSDUKERQY = WZSDUKERQTY.
endif.

```

```

        MODIFY RESULT_PACKAGE FROM rp INDEX l_tabix.
    endif.
ENDIF.
*-----*
*           Select attributes from ZMATERIAL           *
*-----*

read table iMaterial
      INTO wa_Material
      with table key /bic/zmaterial = rp-/bic/zmaterial.
IF sy-subrc = 0.
    w1_base_uom    = wa_Material-base_uom.
    w1_matl_type  = wa_Material-matl_type.
    w1_zcnv_80z   = wa_Material-/BIC/ZCNV_80Z.
    w1_zcs_palt   = wa_Material-/bic/zcs_palt.
    w1_zuntprcs   = wa_Material-/bic/zuntprcs.
    w1_zpackuvml  = wa_Material-/bic/zpackuvml.

    w2_zpackuvml  = w1_zpackuvml.
    w2_zuntprcs   = w1_zuntprcs.
    w2_zcnv_80z   = w1_zcnv_80z.
    w2_zcs_palt   = w1_zcs_palt.

    w3_zpackuvml  = w2_zpackuvml.
    w3_zuntprcs   = w2_zuntprcs.
    w3_zcnv_80z   = w2_zcnv_80z.
    w3_zcs_palt   = w2_zcs_palt.
*-----*
*           Data Processing calculation of "UK Equivalent Invoice Qty"           *
*-----*

    IF w1_matl_type = C_FERT.

        IF w3_zpackuvml NE C_ZERO AND w3_zuntprcs NE C_ZERO.
* Records with unit NE "PAL" unit & ZPACKUVML <= 330           *
            IF w3_zpackuvml <= 330 AND NOT rp-SALES_UNIT EQ C_PAL.
                WZSDUKEIQTY = ( w3_zuntprcs / 24 ) * rp-/BIC/ZSDUKEIQY.
* Records with unit EQ "PAL" unit & ZPACKUVML <= 330           *
            ELSEIF w3_zpackuvml <= 330 AND rp-SALES_UNIT EQ C_PAL.
                WZSDUKEIQTY = ( w3_zuntprcs / 24 ) * rp-/BIC/ZSDUKEIQY *
                w3_zcs_palt.
* Records with unit NE "PAL" unit & ZPACKUVML > 330 < 2000     *
            ELSEIF w3_zpackuvml > 330 AND w3_zpackuvml < 2000 AND NOT
rp-SALES_UNIT EQ C_PAL.
                WZSDUKEIQTY = ( w3_zuntprcs / 12 ) * rp-/BIC/ZSDUKEIQY.
* Records with unit EQ "PAL" unit & ZPACKUVML > 330 < 2000     *
            ELSEIF w3_zpackuvml > 330 AND w3_zpackuvml < 2000 AND
rp-SALES_UNIT EQ C_PAL.
                WZSDUKEIQTY = ( w3_zuntprcs / 12 ) * rp-/BIC/ZSDUKEIQY *
                w3_zcs_palt .
* Records with unit NE "PAL" unit & ZPACKUVML >= 2000           *
        
```

```

ELSEIF w3_zpackuvml >= 2000 AND w3_zpackuvml <= 5000 AND
NOT
rp-SALES_UNIT EQ C_PAL.
    WZSDUKEIQTY = ( w3_zuntprcs / 8 ) * rp-/BIC/ZSDUKEIQTY.

* Records with unit EQ "PAL" unit & ZPACKUVML >= 2000 *
ELSEIF w3_zpackuvml >= 2000 AND w3_zpackuvml <= 5000 AND
rp-SALES_UNIT EQ C_PAL.
    WZSDUKEIQTY = ( w3_zuntprcs / 8 ) * rp-/BIC/ZSDUKEIQTY *
    w3_zcs_palt.

ELSE.
    WZSDUKEIQTY = rp-/BIC/ZSDUKEIQTY.
ENDIF.

ELSE.
    WZSDUKEIQTY = rp-/BIC/ZSDUKEIQTY.
ENDIF.

*-----*
* Data Processing calculation of "UK Equivalent Invoice Qty" *
* for records with "GAL" unit & Material type "HALB" *
* and content of "zcnv_80z" field value is a numeric field. *
*-----*
ELSEIF w1_matl_type = C_HALB AND rp-SALES_UNIT = C_GLL.
    IF w2_numerator NE C_ZERO AND w2_denomintr NE C_ZERO.
        WZSDUKEIQTY = ( w2_denomintr / w2_numerator ) *
            rp-/BIC/ZSDUKEIQTY.
    ELSE.
        WZSDUKEIQTY = rp-/BIC/ZSDUKEIQTY.
    ENDIF.
*-----*
ELSE.
    WZSDUKEIQTY = rp-/BIC/ZSDUKEIQTY.
ENDIF.
*-----*
* End of first if statement "IF w3_zpackuvml = 0" *
*-----*
if not ( WZSDUKEIQTY = 0 ).
    rp-/BIC/ZSDUKEIQTY = WZSDUKEIQTY.
endif.
*-----*
* Modify the Final Result Package *
*-----*
MODIFY RESULT_PACKAGE FROM rp INDEX l_tabix.
ELSE.
    rp-/BIC/ZSDUKEIQTY = rp-/BIC/ZSDUKEIQTY.
    MODIFY RESULT_PACKAGE FROM rp INDEX l_tabix.
endif.
ENDLOOP.

```