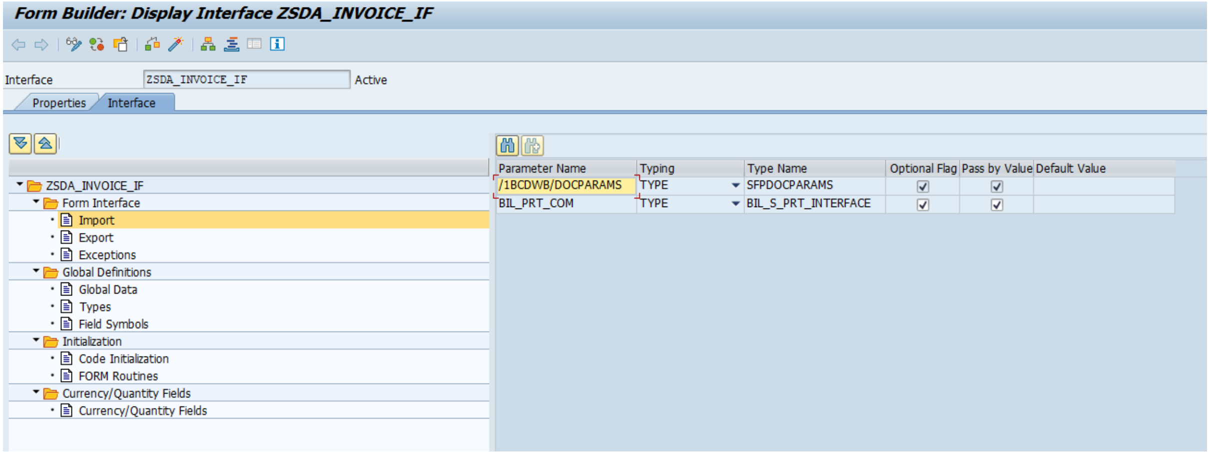
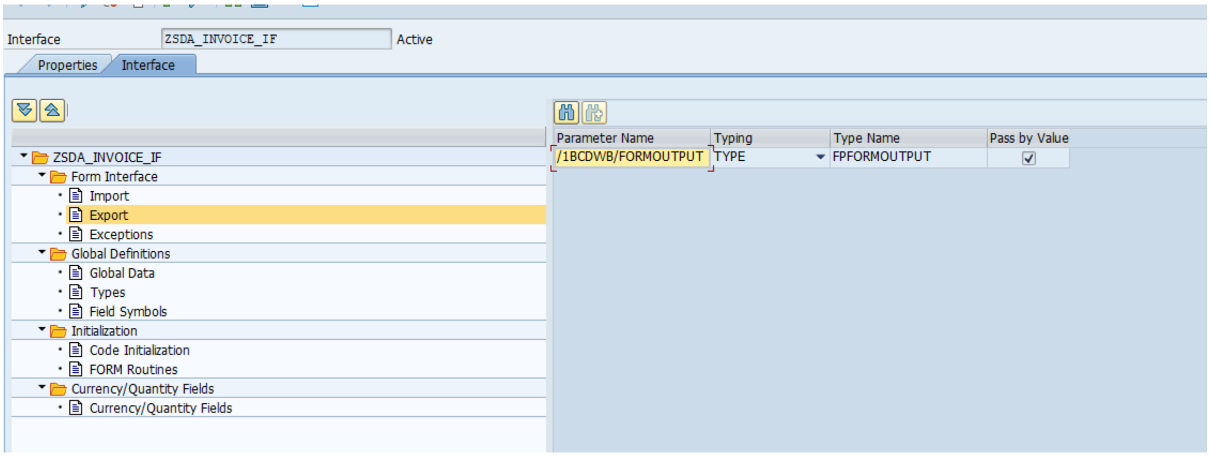


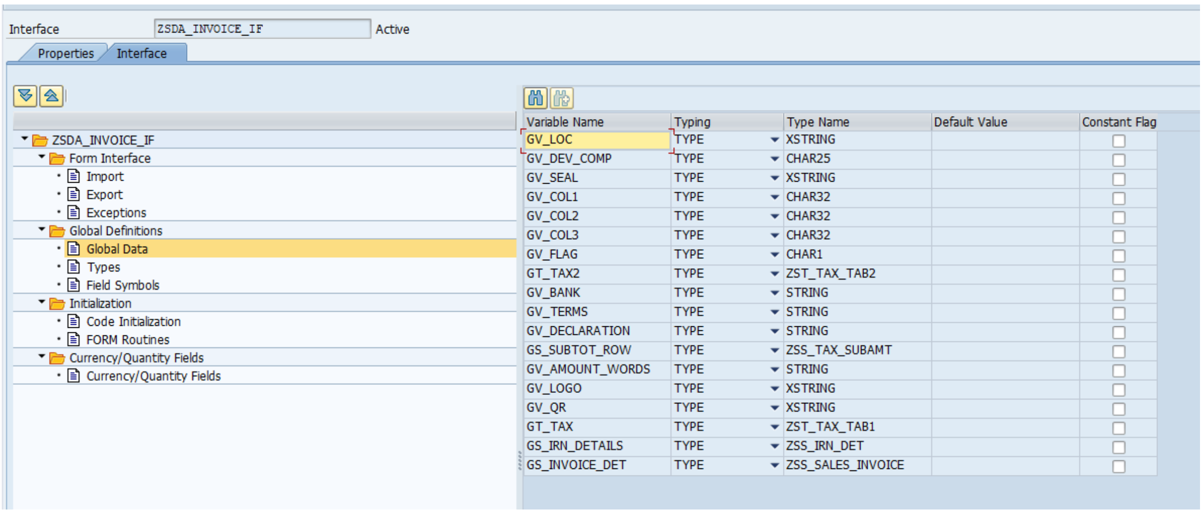
Interface ZSDA_INVOICE_IF Active

Parameter Name	Typing	Type Name	Optional Flag	Pass by Value	Default Value
/1BCDWB/DOCPARAMS	TYPE	SFPDOCPARAMS	☑	☑	
BIL_PRT_COM	TYPE	BIL_S_PRT_INTERFACE	☑	☑	



Interface ZSDA_INVOICE_IF Active

Parameter Name	Typing	Type Name	Pass by Value
/1BCDWB/FORMOUTPUT	TYPE	FPFORMOUTPUT	☑



Interface ZSDA_INVOICE_IF Active

Variable Name	Typing	Type Name	Default Value	Constant Flag
GV_LOC	TYPE	XSTRING		□
GV_DEV_COMP	TYPE	CHAR25		□
GV_SEAL	TYPE	XSTRING		□
GV_COL1	TYPE	CHAR32		□
GV_COL2	TYPE	CHAR32		□
GV_COL3	TYPE	CHAR32		□
GV_FLAG	TYPE	CHAR1		□
GT_TAX2	TYPE	ZST_TAX_TAB2		□
GV_BANK	TYPE	STRING		□
GV_TERMS	TYPE	STRING		□
GV_DECLARATION	TYPE	STRING		□
GS_SUBTOT_ROW	TYPE	ZSS_TAX_SUBAMT		□
GV_AMOUNT_WORDS	TYPE	STRING		□
GV_LOGO	TYPE	XSTRING		□
GV_QR	TYPE	XSTRING		□
GT_TAX	TYPE	ZST_TAX_TAB1		□
GS_IRN_DETAILS	TYPE	ZSS_IRN_DET		□
GS_INVOICE_DET	TYPE	ZSS_SALES_INVOICE		□

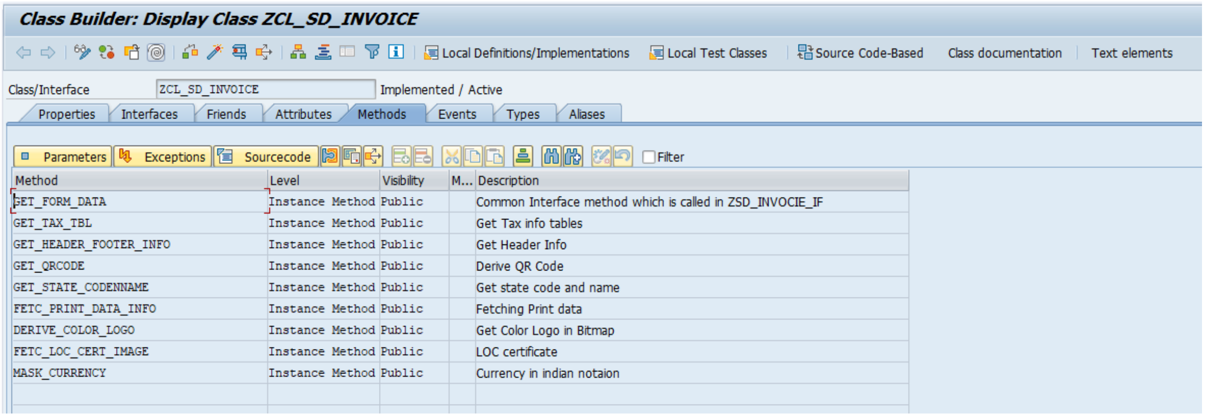
```
TYPES: TYPES: BEGIN OF TY_TAX,
         SR_NO      TYPE CHAR3,
         DESC       TYPE CHAR60,
         SAC_HSN    TYPE STEUC,
         QTY        TYPE CHAR10,
         UNIT_PR    TYPE CHAR20,
         DICOUNT    TYPE CHAR10,
         TAXABLE_AMT TYPE CHAR20,
         GST_RATE   TYPE CHAR10,
         CGST      TYPE CHAR20,
         SGST      TYPE CHAR20,
         TOT_AMT   TYPE CHAR20  ,
       END OF TY_TAX,
       BEGIN OF TY_TAX1,
         SR_NO     TYPE CHAR3,
         DESC      TYPE CHAR60,
         SAC_HSN   TYPE STEUC,
         QTY       TYPE CHAR10,
         UNIT_PR   TYPE CHAR20,
         DICOUNT   TYPE CHAR10,
         TAXABLE_AMT TYPE CHAR20,
         GST_RATE  TYPE CHAR10,
         IGST      TYPE CHAR20,
         TOT_AMT   TYPE CHAR20  ,
       END OF TY_TAX1,
       BEGIN OF TY_TAX2,
         SR_NO     TYPE CHAR3,
         DESC      TYPE CHAR60,
         SAC_HSN   TYPE STEUC,
         QTY       TYPE CHAR10,
         UNIT_PR   TYPE CHAR20,
         DICOUNT   TYPE CHAR10,
         TAXABLE_AMT TYPE CHAR20,
         GST_RATE  TYPE CHAR10,
         IGST      TYPE CHAR20,
         TOT_AMT   TYPE CHAR20,
         TOT_INRAMT   TYPE CHAR20,
       END OF TY_TAX2,
       gty_tax TYPE STANDARD TABLE OF TY_TAX INITIAL SIZE 0,
       gty_tax1 TYPE STANDARD TABLE OF TY_TAX1 INITIAL SIZE 0,
       gty_tax2 TYPE STANDARD TABLE OF TY_TAX2 INITIAL SIZE 0.
```

_________________--------------

CODE INITILISATION

```
*----------------------------------------------------------------------*
*    Program   :  Updates Tables (   )    Process data (  )
*                 Outputs List   (   )    Validation   (  )
*
*    Module                       : SD
*    WRICEF Object Id             : SD_DV020
*    Functional Specifications : FS_SD_DV020_SALES_INVOCIE
*    Technical Spec No            : TS_SD_DV020_SALES_INVOCIE
*    Functional Consultant        : Parul Singh
*    Date Created (dd-mm-yyyy) : 01-12-2024
*    Technical Consultant         : Pravesh Patil
*    Client                       : upGrad
*    Project Name                 : upGrad
*    Company                      : TechWit Business Solutions
```

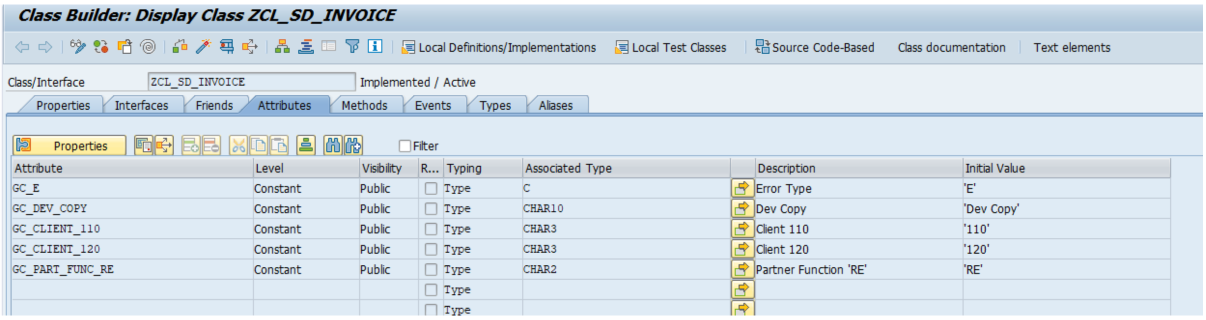
```
*----------------------------------------------------------------------*
 "Call interface method for layout values and tax table valuse
 DATA(LO_OBJECT) = NEW ZCL_SD_INVOICE( ).
 LO_OBJECT->GET_FORM_DATA(
   EXPORTING
     IS_BIL_INVOICE2     = BIL_PRT_COM
   IMPORTING
     ET_TAX1             = GT_TAX
     ES_INVOICE_DET      = GS_INVOICE_DET
     ES_IRN_DETAILS      = GS_IRN_DETAILS
     ES_SUBTOT_ROW       = GS_SUBTOT_ROW
     EV_QR               = GV_QR
     EV_AMOUNT_WORDS     = GV_AMOUNT_WORDS
     EV_DECLARATION      = GV_DECLARATION
     EV_TERMS            = GV_TERMS
     EV_BANK             = GV_BANK
     EV_COL1             = GV_COL1
     EV_COL2             = GV_COL2
     EV_COL3             = GV_COL3
     EV_LOGO             = GV_LOGO
     EV_SEAL             = GV_SEAL
     EV_DEV_COPY         = GV_DEV_COMP
     EV_LOC              = GV_LOC ).
```



Class Builder: Display Class ZCL_SD_INVOICE

Class/Interface ZCL_SD_INVOICE Implemented / Active

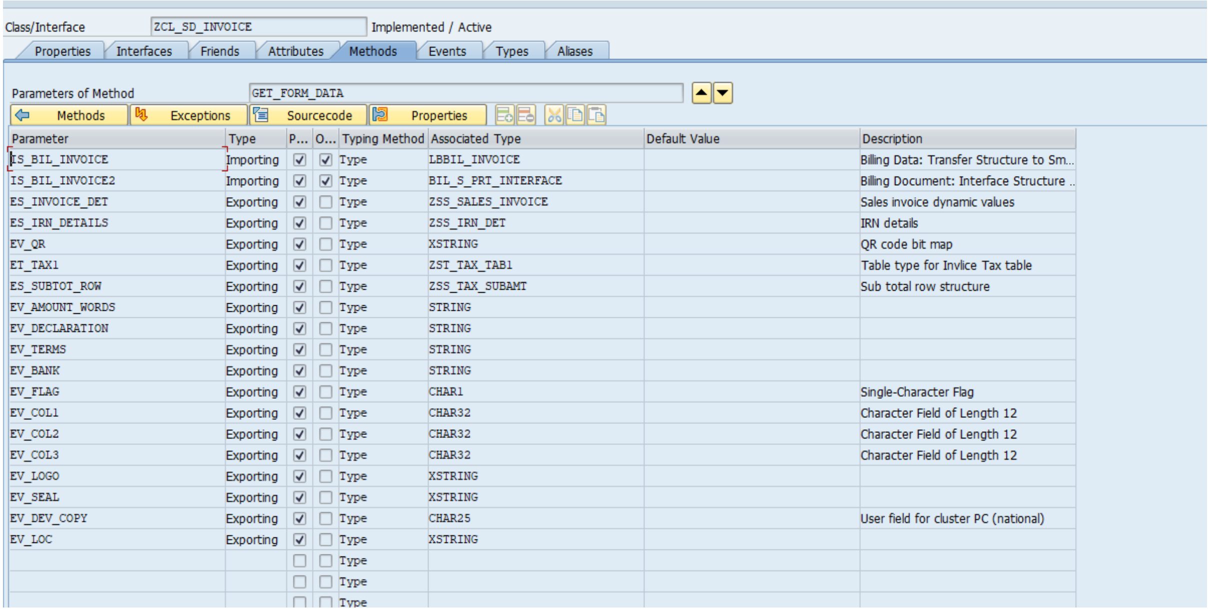
Method	Level	Visibility	M...	Description
GET_FORM_DATA	Instance Method	Public		Common Interface method which is called in ZSD_INVOCIE_IF
GET_TAX_TBL	Instance Method	Public		Get Tax info tables
GET_HEADER_FOOTER_INFO	Instance Method	Public		Get Header Info
GET_QRCODE	Instance Method	Public		Derive QR Code
GET_STATE_CODENNAME	Instance Method	Public		Get state code and name
FETC_PRINT_DATA_INFO	Instance Method	Public		Fetching Print data
DERIVE_COLOR_LOGO	Instance Method	Public		Get Color Logo in Bitmap
FETC_LOC_CERT_IMAGE	Instance Method	Public		LOC certificate
MASK_CURRENCY	Instance Method	Public		Currency in indian notaion



Class Builder: Display Class ZCL_SD_INVOICE

Class/Interface ZCL_SD_INVOICE Implemented / Active

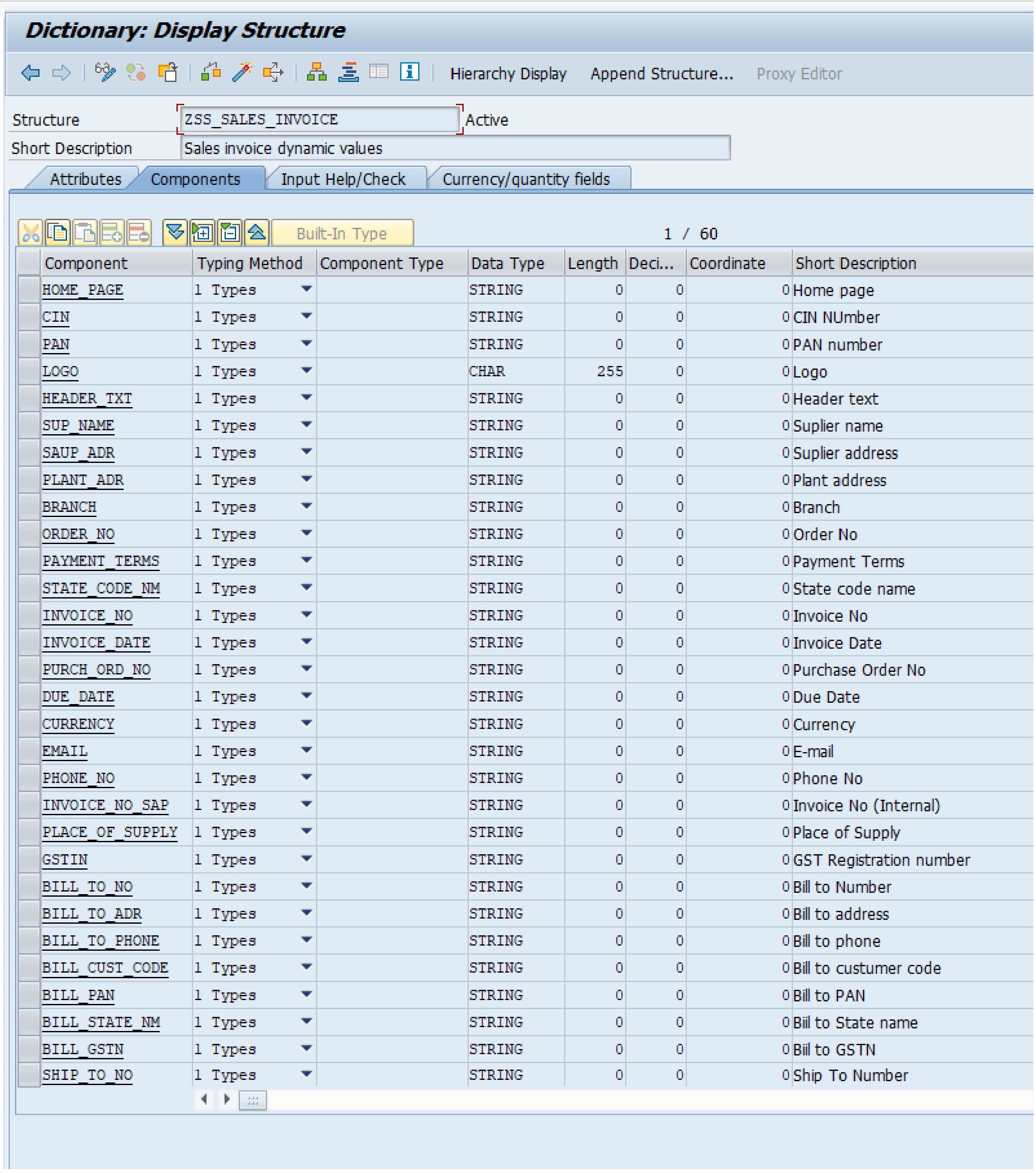
Attribute	Level	Visibility	R...	Typing	Associated Type	Description	Initial Value
GC_E	Constant	Public	☐	Type	C	Error Type	'E'
GC_DEV_COPY	Constant	Public	☐	Type	CHAR10	Dev Copy	'Dev Copy'
GC_CLIENT_110	Constant	Public	☐	Type	CHAR3	Client 110	'110'
GC_CLIENT_120	Constant	Public	☐	Type	CHAR3	Client 120	'120'
GC_PART_FUNC_RE	Constant	Public	☐	Type	CHAR2	Partner Function 'RE'	'RE'
			☐	Type			
			☐	Type			



Class/Interface ZCL_SD_INVOICE Implemented / Active

Parameters of Method GET_FORM_DATA

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
IS_BIL_INVOICE	Importing	☑	☑	Type	LBBIL_INVOICE		Billing Data: Transfer Structure to Sm...
IS_BIL_INVOICE2	Importing	☑	☑	Type	BIL_S_PRT_INTERFACE		Billing Document: Interface Structure ..
ES_INVOICE_DET	Exporting	☑	□	Type	ZSS_SALES_INVOICE		Sales invoice dynamic values
ES_IRN_DETAILS	Exporting	☑	□	Type	ZSS_IRN_DET		IRN details
EV_QR	Exporting	☑	□	Type	XSTRING		QR code bit map
ET_TAX1	Exporting	☑	□	Type	ZST_TAX_TAB1		Table type for Invlice Tax table
ES_SUBTOT_ROW	Exporting	☑	□	Type	ZSS_TAX_SUBAMT		Sub total row structure
EV_AMOUNT_WORDS	Exporting	☑	□	Type	STRING		
EV_DECLARATION	Exporting	☑	□	Type	STRING		
EV_TERMS	Exporting	☑	□	Type	STRING		
EV_BANK	Exporting	☑	□	Type	STRING		
EV_FLAG	Exporting	☑	□	Type	CHAR1		Single-Character Flag
EV_COL1	Exporting	☑	□	Type	CHAR32		Character Field of Length 12
EV_COL2	Exporting	☑	□	Type	CHAR32		Character Field of Length 12
EV_COL3	Exporting	☑	□	Type	CHAR32		Character Field of Length 12
EV_LOGO	Exporting	☑	□	Type	XSTRING		
EV_SEAL	Exporting	☑	□	Type	XSTRING		
EV_DEV_COPY	Exporting	☑	□	Type	CHAR25		User field for cluster PC (national)
EV_LOC	Exporting	☑	□	Type	XSTRING		
		□	□	Type			
		□	□	Type			
		□	□	Type			

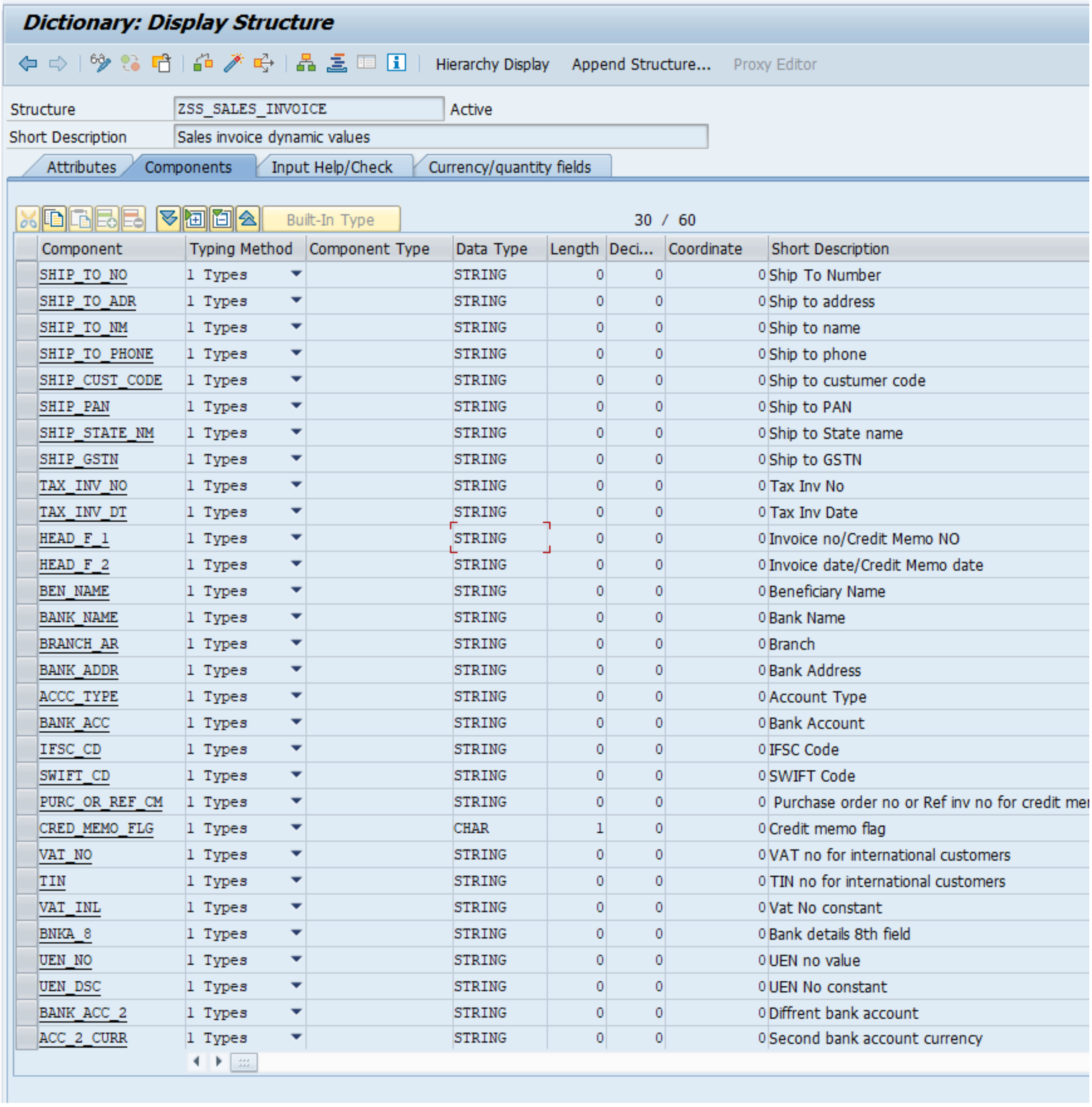


Structure: ZSS_SALES_INVOICE Active

Short Description: Sales invoice dynamic values

1 / 60

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description
HOME_PAGE	1 Types		STRING	0	0	0	Home page
CIN	1 Types		STRING	0	0	0	CIN NUmber
PAN	1 Types		STRING	0	0	0	PAN number
LOGO	1 Types		CHAR	255	0	0	Logo
HEADER_TXT	1 Types		STRING	0	0	0	Header text
SUP_NAME	1 Types		STRING	0	0	0	Suplier name
SAUP_ADR	1 Types		STRING	0	0	0	Suplier address
PLANT_ADR	1 Types		STRING	0	0	0	Plant address
BRANCH	1 Types		STRING	0	0	0	Branch
ORDER_NO	1 Types		STRING	0	0	0	Order No
PAYMENT_TERMS	1 Types		STRING	0	0	0	Payment Terms
STATE_CODE_NM	1 Types		STRING	0	0	0	State code name
INVOICE_NO	1 Types		STRING	0	0	0	Invoice No
INVOICE_DATE	1 Types		STRING	0	0	0	Invoice Date
PURCH_ORD_NO	1 Types		STRING	0	0	0	Purchase Order No
DUE_DATE	1 Types		STRING	0	0	0	Due Date
CURRENCY	1 Types		STRING	0	0	0	Currency
EMAIL	1 Types		STRING	0	0	0	E-mail
PHONE_NO	1 Types		STRING	0	0	0	Phone No
INVOICE_NO_SAP	1 Types		STRING	0	0	0	Invoice No (Internal)
PLACE_OF_SUPPLY	1 Types		STRING	0	0	0	Place of Supply
GSTIN	1 Types		STRING	0	0	0	GST Registration number
BILL_TO_NO	1 Types		STRING	0	0	0	Bill to Number
BILL_TO_ADR	1 Types		STRING	0	0	0	Bill to address
BILL_TO_PHONE	1 Types		STRING	0	0	0	Bill to phone
BILL_CUST_CODE	1 Types		STRING	0	0	0	Bill to custumer code
BILL_PAN	1 Types		STRING	0	0	0	Bill to PAN
BILL_STATE_NM	1 Types		STRING	0	0	0	Bil to State name
BILL_GSTN	1 Types		STRING	0	0	0	Bil to GSTN
SHIP_TO_NO	1 Types		STRING	0	0	0	Ship To Number



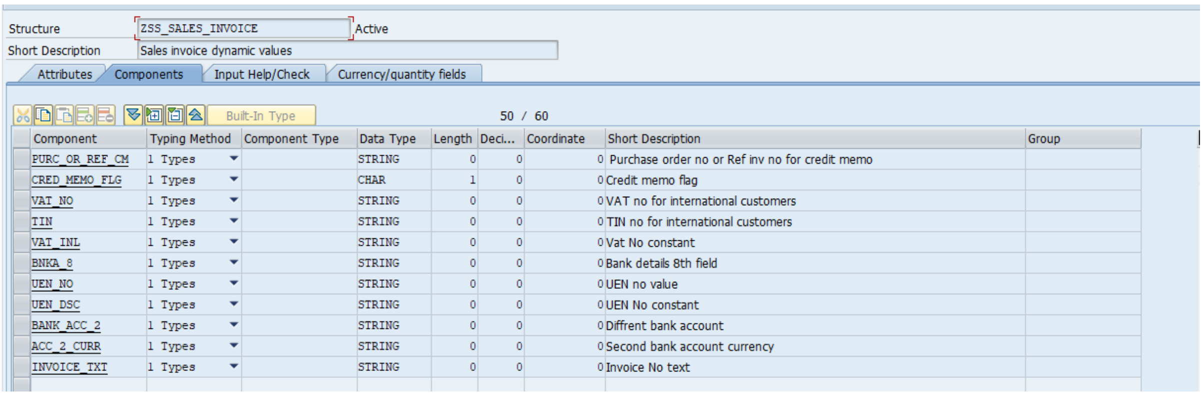
Dictionary: Display Structure

Structure ZSS_SALES_INVOICE Active

Short Description Sales invoice dynamic values

30 / 60

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description
SHIP_TO_NO	1 Types		STRING	0	0	0	Ship To Number
SHIP_TO_ADR	1 Types		STRING	0	0	0	Ship to address
SHIP_TO_NM	1 Types		STRING	0	0	0	Ship to name
SHIP_TO_PHONE	1 Types		STRING	0	0	0	Ship to phone
SHIP_CUST_CODE	1 Types		STRING	0	0	0	Ship to custumer code
SHIP_PAN	1 Types		STRING	0	0	0	Ship to PAN
SHIP_STATE_NM	1 Types		STRING	0	0	0	Ship to State name
SHIP_GSTN	1 Types		STRING	0	0	0	Ship to GSTN
TAX_INV_NO	1 Types		STRING	0	0	0	Tax Inv No
TAX_INV_DT	1 Types		STRING	0	0	0	Tax Inv Date
HEAD_F_1	1 Types		STRING	0	0	0	Invoice no/Credit Memo NO
HEAD_F_2	1 Types		STRING	0	0	0	Invoice date/Credit Memo date
BEN_NAME	1 Types		STRING	0	0	0	Beneficiary Name
BANK_NAME	1 Types		STRING	0	0	0	Bank Name
BRANCH_AR	1 Types		STRING	0	0	0	Branch
BANK_ADDR	1 Types		STRING	0	0	0	Bank Address
ACCC_TYPE	1 Types		STRING	0	0	0	Account Type
BANK_ACC	1 Types		STRING	0	0	0	Bank Account
IFSC_CD	1 Types		STRING	0	0	0	IFSC Code
SWIFT_CD	1 Types		STRING	0	0	0	SWIFT Code
PURC_OR_REF_CM	1 Types		STRING	0	0	0	Purchase order no or Ref inv no for credit mer
CRED_MEMO_FLG	1 Types		CHAR	1	0	0	Credit memo flag
VAT_NO	1 Types		STRING	0	0	0	VAT no for international customers
TIN	1 Types		STRING	0	0	0	TIN no for international customers
VAT_INL	1 Types		STRING	0	0	0	Vat No constant
BNKA_8	1 Types		STRING	0	0	0	Bank details 8th field
UEN_NO	1 Types		STRING	0	0	0	UEN no value
UEN_DSC	1 Types		STRING	0	0	0	UEN No constant
BANK_ACC_2	1 Types		STRING	0	0	0	Diffrent bank account
ACC_2_CURR	1 Types		STRING	0	0	0	Second bank account currency

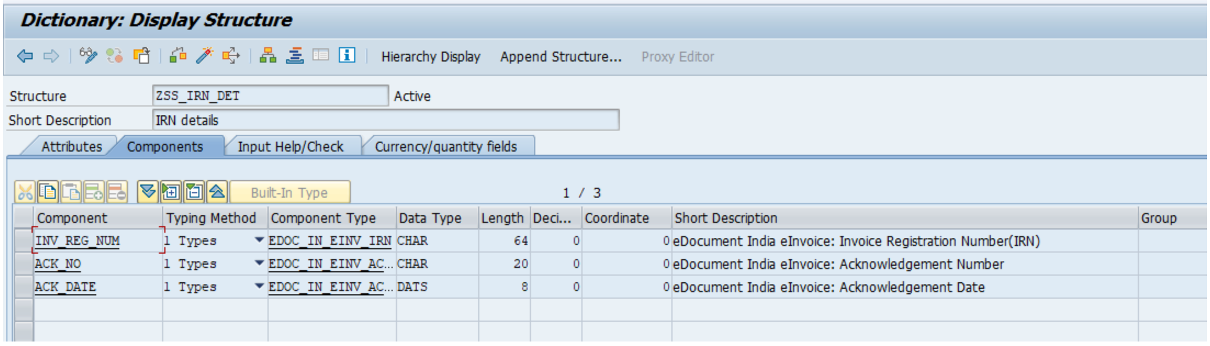


Structure ZSS_SALES_INVOICE Active

Short Description Sales invoice dynamic values

50 / 60

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description	Group
PURC_OR_REF_CM	1 Types		STRING	0	0	0	Purchase order no or Ref inv no for credit memo	
CRED_MEMO_FLG	1 Types		CHAR	1	0	0	Credit memo flag	
VAT_NO	1 Types		STRING	0	0	0	VAT no for international customers	
TIN	1 Types		STRING	0	0	0	TIN no for international customers	
VAT_INL	1 Types		STRING	0	0	0	Vat No constant	
BNKA_8	1 Types		STRING	0	0	0	Bank details 8th field	
UEN_NO	1 Types		STRING	0	0	0	UEN no value	
UEN_DSC	1 Types		STRING	0	0	0	UEN No constant	
BANK_ACC_2	1 Types		STRING	0	0	0	Diffrent bank account	
ACC_2_CURR	1 Types		STRING	0	0	0	Second bank account currency	
INVOICE_TXT	1 Types		STRING	0	0	0	Invoice No text	



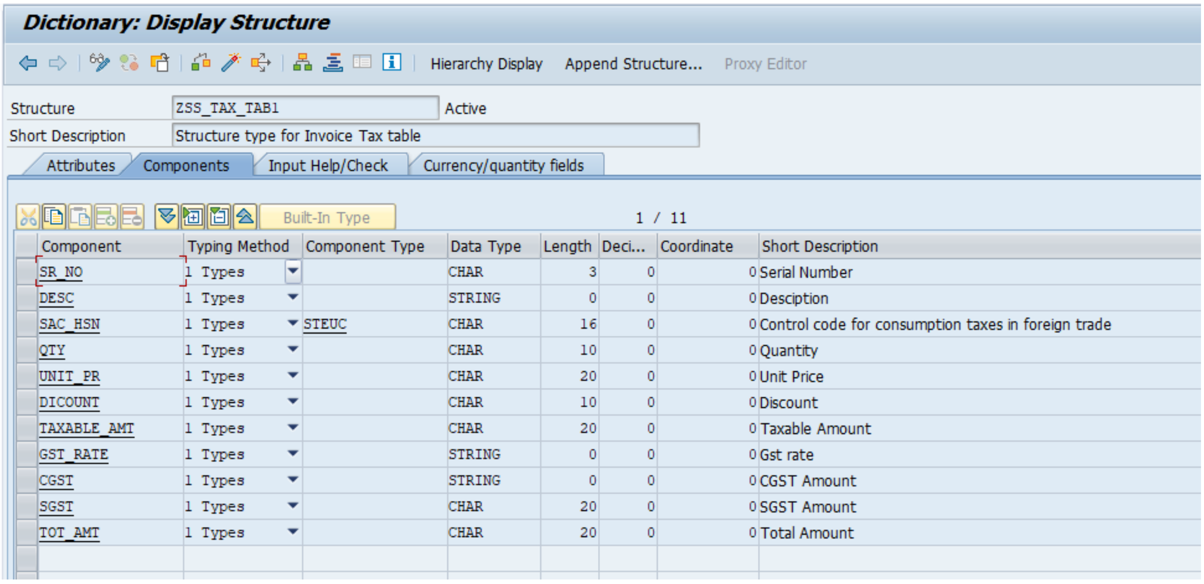
Dictionary: Display Structure

Structure ZSS_IRN_DET Active

Short Description IRN details

1 / 3

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description	Group
INV_REG_NUM	1 Types	EDOC_IN_EINV_IRN	CHAR	64	0	0	eDocument India eInvoice: Invoice Registration Number(IRN)	
ACK_NO	1 Types	EDOC_IN_EINV_AC...	CHAR	20	0	0	eDocument India eInvoice: Acknowledgement Number	
ACK_DATE	1 Types	EDOC_IN_EINV_AC...	DATS	8	0	0	eDocument India eInvoice: Acknowledgement Date	



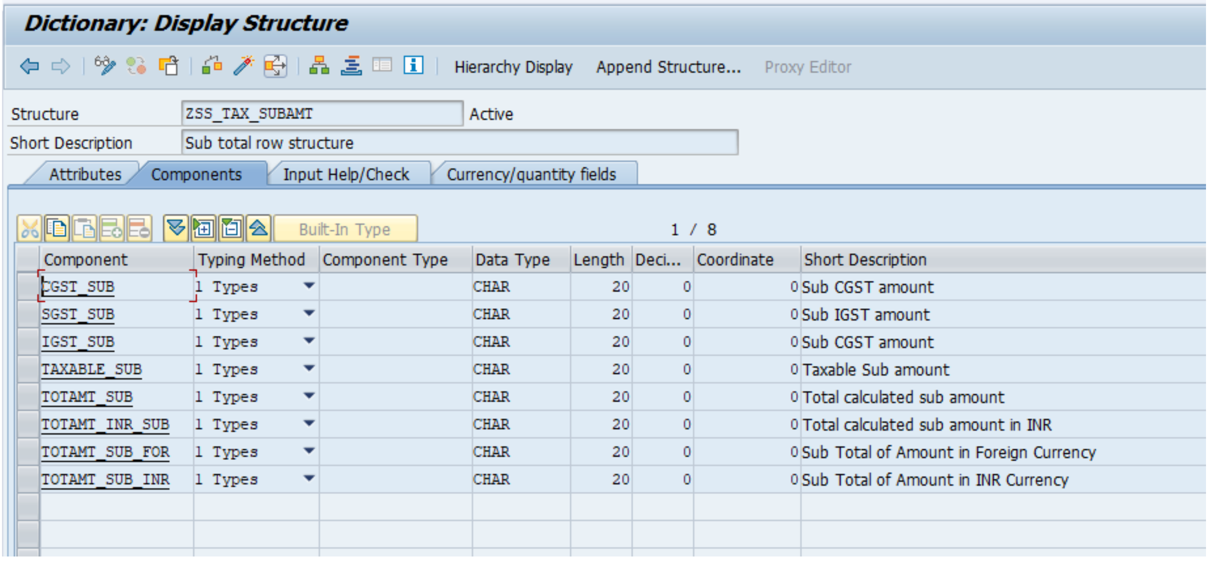
Dictionary: Display Structure

Structure ZSS_TAX_TAB1 Active

Short Description Structure type for Invoice Tax table

1 / 11

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description
SR_NO	1 Types		CHAR	3	0	0	Serial Number
DESC	1 Types		STRING	0	0	0	Description
SAC_HSN	1 Types	STEUC	CHAR	16	0	0	Control code for consumption taxes in foreign trade
QTY	1 Types		CHAR	10	0	0	Quantity
UNIT_PR	1 Types		CHAR	20	0	0	Unit Price
DICOUNT	1 Types		CHAR	10	0	0	Discount
TAXABLE_AMT	1 Types		CHAR	20	0	0	Taxable Amount
GST_RATE	1 Types		STRING	0	0	0	Gst rate
CGST	1 Types		STRING	0	0	0	CGST Amount
SGST	1 Types		CHAR	20	0	0	SGST Amount
TOT_AMT	1 Types		CHAR	20	0	0	Total Amount



Dictionary: Display Structure

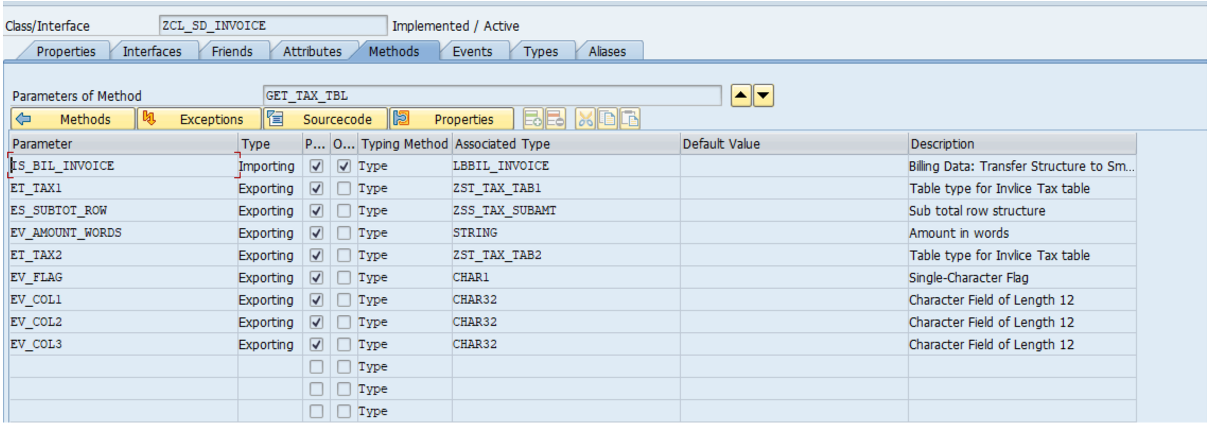
Structure ZSS_TAX_SUBAMT Active

Short Description Sub total row structure

1 / 8

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description
CGST_SUB	1 Types		CHAR	20	0	0	Sub CGST amount
SGST_SUB	1 Types		CHAR	20	0	0	Sub IGST amount
IGST_SUB	1 Types		CHAR	20	0	0	Sub CGST amount
TAXABLE_SUB	1 Types		CHAR	20	0	0	Taxable Sub amount
TOTAMT_SUB	1 Types		CHAR	20	0	0	Total calculated sub amount
TOTAMT_INR_SUB	1 Types		CHAR	20	0	0	Total calculated sub amount in INR
TOTAMT_SUB_FOR	1 Types		CHAR	20	0	0	Sub Total of Amount in Foreign Currency
TOTAMT_SUB_INR	1 Types		CHAR	20	0	0	Sub Total of Amount in INR Currency

```
  METHOD get_form_data.
*--------------------------------------------------------------------*
*     IDENTITY
*--------------------------------------------------------------------*
*     Program    :  Updates Tables (   )     Process data (   )
*                   Outputs List    (   )     Validation    (   )
*
*     Module                      :  SD
*     WRICEF Object Id            :  SD_DV020
*     Functional Specifications  :  FS_SD_DV020_SALES_INVOCIE
*    Technical Spec No         : TS_SD_DV020_SALES_INVOCIE
*    Functional Consultant     : Parul Singh
*    Date Created (dd-mm-yyyy) : 01-12-2024
*    Technical Consultant      : Pravesh Patil
*    Client                    : upGrad
*    Project Name              : upGrad
*    Company                   : TechWit Business Solutions
*---------------------------------------------------------------------*
    "The following line will decide if print is water mark as DEV COPY or n
o water mark
    IF sy-mandt EQ gc_client_110 OR sy-mandt EQ gc_client_120.
      ev_dev_copy = gc_dev_copy.
    ENDIF.

    "Fetch Print data information for more print data
    me->fetc_print_data_info( EXPORTING is_bil_invoice2 = is_bil_invoice2
                              IMPORTING es_bil_invoice  = DATA(ls_bil_invoi
ce) ).


    "Fetch header data,Supplier data, Bill&Ship info,IRN details, Declarati
on , Bank details and Other Terms
    me->get_header_footer_info(
    EXPORTING
        is_bil_invoice     = ls_bil_invoice     "Passing Print data info wh
ich got from previous method
        is_bil_invoice2    = is_bil_invoice2
      IMPORTING
        es_irn_details     = es_irn_details
        es_invoice_det     = es_invoice_det
        ev_declaration     = ev_declaration
        ev_terms           = ev_terms
        ev_bank            = ev_bank
        ev_seal            = ev_seal ).

    "Call Tax Table data, Sub row, Amount in words and Dynamic Columns
    me->get_tax_tbl(
    EXPORTING
      is_bil_invoice  = ls_bil_invoice
    IMPORTING
      et_tax1         = et_tax1
      es_subtot_row   = es_subtot_row
      ev_amount_words = ev_amount_words
      ev_col1         = ev_col1
      ev_col2         = ev_col2
      ev_col3         = ev_col3 )..

    "QR Code
    DATA(lv_bill_no) = ls_bil_invoice-hd_gen-bil_number.
    me->get_qrcode( EXPORTING iv_bil_invoice = lv_bill_no
                    IMPORTING ev_qrbitmap    = ev_qr ).

    "Logo
    me->derive_color_logo( EXPORTING iv_xlogo = es_invoice_det-logo
                           IMPORTING ev_logo = ev_logo ).

    " LOC image
    me->fetc_loc_cert_image( EXPORTING iv_fkdat = is_bil_invoice2-
head_detail-vbdkr-fkdat  "++PATILP14.11.2025 iv_fkdat
                             IMPORTING ev_loc =  ev_loc  ).


  ENDMETHOD.
```

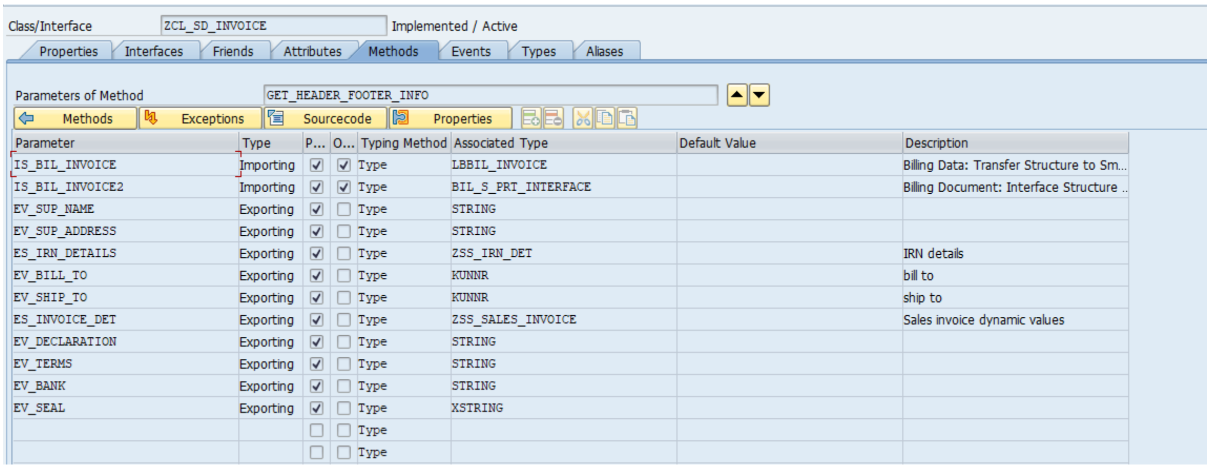


Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
IS_BIL_INVOICE	Importing	☑	☑	Type	LBBIL_INVOICE		Billing Data: Transfer Structure to Sm...
ET_TAX1	Exporting	☑	□	Type	ZST_TAX_TAB1		Table type for Invlice Tax table
ES_SUBTOT_ROW	Exporting	☑	□	Type	ZSS_TAX_SUBAMT		Sub total row structure
EV_AMOUNT_WORDS	Exporting	☑	□	Type	STRING		Amount in words
ET_TAX2	Exporting	☑	□	Type	ZST_TAX_TAB2		Table type for Invlice Tax table
EV_FLAG	Exporting	☑	□	Type	CHAR1		Single-Character Flag
EV_COL1	Exporting	☑	□	Type	CHAR32		Character Field of Length 12
EV_COL2	Exporting	☑	□	Type	CHAR32		Character Field of Length 12
EV_COL3	Exporting	☑	□	Type	CHAR32		Character Field of Length 12
		□	□	Type			
		□	□	Type			
		□	□	Type			

```
  METHOD get_tax_tbl.
*----------------------------------------------------------------------*
*    Program    :  Updates Tables (   )    Process data (  )
*                 Outputs List    (   )    Validation    (  )
*
*    Module                         : SD
*    WRICEF Object Id               : SD_DV020
*    Functional Specifications      : FS_SD_DV020_SALES_INVOCIE
*    Technical Spec No              : TS_SD_DV020_SALES_INVOCIE
*    Functional Consultant          : Parul Singh
*    Date Created (dd-mm-yyyy)      : 01-12-2024
*    Technical Consultant           : Pravesh Patil
*    Client                         : upGrad
*    Project Name                   : upGrad
*    Company                        : TechWit Business Solutions
*---------------------------------------------------------------------*

    DATA: lv_qty_line            TYPE char10,
          lv_unit_line           TYPE char20,
          lv_unit_price          TYPE netwr,
          lv_gst_tax_line       TYPE char10,
          lv_cgst_line          TYPE char20,
          lv_igst_line          TYPE char20,
          lv_sgst_line          TYPE char20,
          lv_amt_line           TYPE char20,
          lv_tot_line           TYPE char20,
          lv_tot_line_inr       TYPE char20,
          lv_tot_tot            TYPE char20,
          lv_t_cgst             TYPE kwert,
          lv_t_sgst             TYPE kwert,
          lv_t_igst             TYPE kwert,
          lv_t_total_amount     TYPE kwert,
          lv_t_total_amount_inr TYPE kwert,
          lv_t_amount           TYPE kwert.

    DATA: lv_words(130) TYPE c,
          lv_words_i    TYPE spell,
          lv_tot_amt    TYPE pc207-betrg.

    DATA: ls_tax1   TYPE zss_tax_tab1,
          ls_komk   TYPE komk,
          ls_komp   TYPE komp,
          ls_komk_e TYPE komk,
          ls_komp_e TYPE komp.

    DATA: lt_komv     TYPE STANDARD TABLE OF  komv,
          lt_komvd    TYPE STANDARD TABLE OF  komvd,
          lt_mat_dsc  TYPE STANDARD TABLE OF tline,
          lt_bill_dsc TYPE STANDARD TABLE OF tline.

    DATA(lt_gen) = is_bil_invoice-it_gen.
    DATA(lt_price) = is_bil_invoice-it_price.
    DATA(lv_srno) = 0.
    DATA(ls_komkinr) = is_bil_invoice-hd_komk.
    DATA(lv_exchange_rt) = ls_komkinr-kurrf.
    DATA(lv_currency) = ls_komkinr-waerk..

    LOOP AT lt_gen INTO DATA(ls_gen).
      "SR No
      lv_srno = lv_srno + 1.

      "Discription and Matarial text and Billing text
      DATA(lv_bill_ins) = CONV tdobname( ls_gen-bil_number && ls_gen-
itm_number ).
      DATA(lv_desc) = CONV string( '' ).
***      IF lv_bill_ins IS NOT INITIAL.
***        CALL FUNCTION 'READ_TEXT'
***          EXPORTING
***            client                  = sy-mandt
***            id                      = '0001'
***            language                = sy-langu
***            name                    = lv_bill_ins
***            object                  = 'VBBP'
***          TABLES
***            lines                     = lt_mat_dsc
***          EXCEPTIONS
***            id                        = 1
***            language                  = 2
***            name                      = 3
***            not_found                 = 4
***            object                    = 5
***            reference_check           = 6
***            wrong_access_to_archive = 7
***            OTHERS                    = 8.
***        IF sy-subrc <> 0.
***        ELSEIF lt_mat_dsc IS NOT INITIAL.
***          LOOP AT lt_mat_dsc INTO DATA(ls_mat).
***            lv_desc = lv_desc && ls_mat-
tdline && cl_abap_char_utilities=>newline.
***          ENDLOOP.
***        ENDIF.

      lv_desc = lv_desc && ls_gen-short_text .

      CALL FUNCTION 'READ_TEXT'
        EXPORTING
          client                  = sy-mandt
          id                      = 'TX05'
          language                = sy-langu
          name                    = lv_bill_ins
          object                  = 'VBBP'
        TABLES
          lines                   = lt_bill_dsc
        EXCEPTIONS
          id                      = 1
          language                = 2
          name                    = 3
          not_found               = 4
          object                  = 5
          reference_check         = 6
          wrong_access_to_archive = 7
          OTHERS                  = 8.
      IF sy-subrc <> 0.
* Implement suitable error handling here
      ELSEIF lt_bill_dsc IS NOT INITIAL.
        LOOP AT lt_bill_dsc INTO DATA(ls_bill).
          lv_desc = lv_desc && cl_abap_char_utilities=>newline && ls_bill-
tdline.
        ENDLOOP.
      ENDIF.
***      ENDIF.



      SELECT SINGLE werks FROM vbrp INTO @DATA(lv_werks)
        WHERE vbeln = @ls_gen-bil_number
        AND   posnr = @ls_gen-itm_number.
      IF sy-subrc = 0.
        "HSN/SAC
        SELECT SINGLE steuc INTO @DATA(lv_hsn) FROM marc
          WHERE matnr = @ls_gen-material
          AND   werks = @lv_werks.
        IF sy-subrc = 0.ENDIF.
      ENDIF.

      "Qunatity
      CLEAR lv_qty_line.
      DATA(lv_val) = trunc( ls_gen-fkimg ).
      WRITE lv_val TO lv_qty_line.
      CONDENSE lv_qty_line.

      "Unit Price
      READ TABLE lt_price INTO DATA(ls_price) WITH KEY bil_number = ls_gen-
bil_number
                                                  itm_number = ls_gen-
itm_number.
      IF ls_gen-fkimg IS NOT INITIAL AND sy-subrc EQ 0.
        lv_unit_price = ls_price-netwr / ls_gen-fkimg.
        CLEAR lv_unit_line.

        me->mask_currency(
         EXPORTING
           iv_amount     = lv_unit_price               " Net Value in Docume
nt Currency
           IMPORTING
           ev_masked_amt = lv_unit_line  ).
        CONDENSE lv_unit_line.
      ENDIF.

      "GST Rate
      CLEAR: ls_komk, ls_komp.

      DATA(ls_head)   = is_bil_invoice-hd_gen.

      CALL FUNCTION 'RV_PRICE_PRINT_REFRESH'
        TABLES
          tkomv = lt_komv.

      REFRESH lt_komv.

      ls_komk_e-belnr = ls_head-bil_number.
      ls_komk_e-knumv = ls_head-kond_numb.
      ls_komp_e-kposn = ls_gen-itm_number.

      CALL FUNCTION 'RV_PRICE_PRINT_ITEM'
        EXPORTING
          comm_head_i = ls_komk_e
          comm_item_i = ls_komp_e
          language    = sy-langu
        IMPORTING
          comm_head_e = ls_komk
          comm_item_e = ls_komp
*         ET_QTYCNVFACT       =
        TABLES
          tkomv       = lt_komv
          tkomvd      = lt_komvd.

      IF lt_komv IS NOT INITIAL.
        BREAK patilp.
        READ TABLE lt_komv INTO DATA(ls_komv)
                                WITH KEY kschl = 'JOCG' kposn = ls_gen-
itm_number.
        IF sy-subrc = 0 AND ls_komv-kbetr <> 0.
          DATA(lv_cgst_tax) = CONV kbetr( ls_komv-kbetr / 10 ).
          DATA(lv_tax) = lv_cgst_tax.
        ENDIF.
        READ TABLE lt_komv INTO ls_komv
                                  WITH KEY kschl = 'JOSG' kposn = ls_gen-
itm_number.
        IF sy-subrc = 0 AND ls_komv-kbetr <> 0.
          DATA(lv_sgst_tax) = CONV kbetr( ls_komv-kbetr / 10 ).
          lv_tax = lv_cgst_tax + lv_sgst_tax.
        ENDIF.
        READ TABLE lt_komv INTO ls_komv
                                WITH KEY kschl = 'JOIG' kposn = ls_gen-
itm_number.
        IF sy-subrc = 0 AND ls_komv-kbetr <> 0.
          DATA(lv_igst_tax) = CONV kbetr( ls_komv-kbetr / 10 ).
          lv_tax = lv_igst_tax.
        ENDIF.

        CLEAR lv_gst_tax_line.
        IF lv_tax <> 0.
          WRITE lv_tax TO lv_gst_tax_line.
          SPLIT lv_gst_tax_line AT '.' INTO lv_gst_tax_line DATA(lv_extra).
          IF lv_gst_tax_line IS NOT INITIAL.
            CONCATENATE lv_gst_tax_line '%' INTO lv_gst_tax_line.
            CONDENSE lv_gst_tax_line.
          ENDIF.
        ELSE.
          CONCATENATE lv_gst_tax_line '0%' INTO lv_gst_tax_line.
          CONDENSE lv_gst_tax_line.
        ENDIF.
      ENDIF.

      "Reading TAX conditions
      CALL FUNCTION 'RV_PRICE_PRINT_REFRESH'
        TABLES
          tkomv = lt_komv.

      REFRESH lt_komv.
      ls_komk-belnr = ls_head-bil_number.
      ls_komk-knumv = ls_head-kond_numb.
      ls_komp-kposn = ls_gen-itm_number.

      CALL FUNCTION 'RV_PRICE_PRINT_ITEM'
        EXPORTING
          comm_head_i = ls_komk_e
          comm_item_i = ls_komp_e
          language    = sy-langu
        IMPORTING
          comm_head_e = ls_komk
          comm_item_e = ls_komp
*         ET_QTYCNVFACT       =
        TABLES
          tkomv       = lt_komv
          tkomvd      = lt_komvd.

      "CGST
      DATA(ls_it_kond) = is_bil_invoice-it_kond.
      CLEAR: ls_komk, ls_komp, lv_tax.

      IF lt_komv IS NOT INITIAL.
        "CGST
        READ TABLE lt_komv INTO ls_komv
        WITH KEY kschl = 'JOCG' kposn = ls_gen-itm_number.
        IF sy-subrc = 0.
          DATA(lv_cgst_amt) = ls_komv-kwert.

          me->mask_currency(
        EXPORTING
          iv_amount     = CONV netwr( ls_komv-
kwert )              " Net Value in Document Currency
        IMPORTING
          ev_masked_amt = lv_cgst_line              " Amount in indian notat
ion
      ).
          CONDENSE lv_cgst_line.
        ENDIF.
        "SGST
        READ TABLE lt_komv INTO ls_komv
        WITH KEY kschl = 'JOSG' kposn = ls_gen-itm_number.
        IF sy-subrc = 0.
          DATA(lv_sgst_amt) = ls_komv-kwert.

          me->mask_currency(
        EXPORTING
          iv_amount     = CONV netwr( ls_komv-
kwert )              " Net Value in Document Currency
        IMPORTING
          ev_masked_amt = lv_sgst_line  ).
          CONDENSE lv_sgst_line.
        ENDIF.
        "IGST
        READ TABLE lt_komv INTO ls_komv
        WITH KEY kschl = 'JOIG' kposn = ls_gen-itm_number.
        IF sy-subrc = 0.
          DATA(lv_igst_amt) = ls_komv-kwert.

          me->mask_currency(
        EXPORTING
          iv_amount     = CONV netwr( ls_komv-
kwert )              " Net Value in Document Currency
        IMPORTING
          ev_masked_amt = lv_igst_line  ).
          CONDENSE lv_igst_line.
        ELSE.
          lv_igst_line = |0.00|.
        ENDIF.

      ENDIF.
      DATA(ls_org)    = is_bil_invoice-hd_org.

      CLEAR lv_amt_line.
      DATA(lv_valtaxable) = ls_price-netwr.

      me->mask_currency(
        EXPORTING
          iv_amount     = lv_valtaxable                " Net Value in Docum
ent Currency
        IMPORTING
          ev_masked_amt = lv_amt_line             " Amount in indian notati
on
      ).
      CONDENSE lv_amt_line.

      "Total amount
      DATA(lv_total_amount) =  CONV  netwr(  ls_price-netwr + ls_price-
mwsbp ) .

      DATA: lv_amount TYPE bf_wrbtr.
      lv_amount = lv_total_amount.
      lv_total_amount = lv_amount.

      CLEAR : lv_tot_line,lv_tot_line_inr.

      me->mask_currency(
        EXPORTING
          iv_amount     = lv_total_amount                " Net Value in Doc
ument Currency
        IMPORTING
          ev_masked_amt = lv_tot_line            " Amount in indian notatio
n
      ).
      CONDENSE lv_tot_line.

      "Total Amount in INR
      DATA(lv_tot_amt1) =  CONV  netwr( ls_price-netwr + ls_price-mwsbp ).
      DATA(lv_totinr) = CONV  netwr( lv_tot_amt1 *  lv_exchange_rt ).

      me->mask_currency(
        EXPORTING
          iv_amount     = lv_totinr                 " Net Value in Document 
Currency
        IMPORTING
          ev_masked_amt = lv_tot_line_inr              " Amount in indian no
tation
      ).
      CONDENSE lv_tot_line_inr.

      "Fill tax table
      IF lv_cgst_line IS NOT INITIAL.
        ls_tax1 = VALUE #( sr_no = lv_srno
                           desc  = lv_desc
                           sac_hsn = lv_hsn
                           qty = lv_qty_line
                           unit_pr = lv_unit_line
                           taxable_amt = lv_amt_line
                           gst_rate = lv_gst_tax_line
                           cgst = lv_cgst_line
                           sgst = lv_sgst_line
                           tot_amt = lv_tot_line
        ).
        APPEND ls_tax1 TO et_tax1.
      ELSE.
        ls_tax1 = VALUE #( sr_no = lv_srno
                         desc  = lv_desc
                         sac_hsn = lv_hsn
                         qty = lv_qty_line
                         unit_pr = lv_unit_line
                         taxable_amt = lv_amt_line
                         gst_rate = lv_gst_tax_line
                         cgst  = lv_igst_line
*                         SGST = LV_TOT_LINE
                         sgst  = COND #( WHEN is_bil_invoice-hd_gen-
bil_waerk <> 'INR' THEN lv_tot_line )
                         tot_amt = lv_tot_line_inr
      ).
        APPEND ls_tax1 TO et_tax1.
      ENDIF.

      "Taxable amount
      lv_t_amount = lv_t_amount + lv_valtaxable.
      "Total For Taxes
      "CGST
      lv_t_cgst = lv_t_cgst + lv_cgst_amt.
      "SGST
      lv_t_sgst = lv_t_sgst + lv_sgst_amt.
      "IGST
      lv_t_igst = lv_t_igst + lv_igst_amt.
      "Total Amount
      lv_t_total_amount = lv_t_total_amount + lv_total_amount.
      "Total INR Amount
      lv_t_total_amount_inr = lv_t_total_amount_inr + lv_totinr.

      CLEAR: ls_tax1,lv_tax,  lv_hsn, lv_qty_line,lv_unit_price, lv_unit_li
ne.
    ENDLOOP.

    "Deciding name of laste three columns based on CGST/SGST or IGST values
    IF lv_cgst_line IS NOT INITIAL.
      ev_col1 = |CGST|.
      ev_col2 = |SGST|.
      ev_col3 = |Total Amount|.
    ELSE.
      ev_col1 = |IGST|.
      ev_col3 = |Total Amount|.
      IF is_bil_invoice-hd_gen-bil_waerk <> 'INR' .
        ev_col2 = |Total Amount| && ' (' && is_bil_invoice-hd_gen-
bil_waerk && ')'.
        ev_col3 = |Total Amount (INR)|.
      ENDIF.
    ENDIF.

    "*****************************************************Sub Total row and
 Amount in words
    "Fill sub total row
    IF lv_cgst_line IS NOT INITIAL.
      me->mask_currency(
        EXPORTING
          iv_amount     = CONV netwr( lv_t_amount )              " Net Valu
e in Document Currency
        IMPORTING
          ev_masked_amt = es_subtot_row-
taxable_sub             " Amount in indian notation
      ).
      CONDENSE es_subtot_row-taxable_sub .

      me->mask_currency(
              EXPORTING
                iv_amount     = CONV netwr( lv_t_cgst )              " Net
Value in Document Currency
              IMPORTING
                ev_masked_amt = es_subtot_row-
cgst_sub             " Amount in indian notation
            ).
      CONDENSE es_subtot_row-cgst_sub .


      me->mask_currency(
               EXPORTING
                 iv_amount     = CONV netwr( lv_t_sgst )              " Net
 Value in Document Currency
```

```
               IMPORTING
                 ev_masked_amt = es_subtot_row-
sgst_sub             " Amount in indian notation
             ).
      CONDENSE es_subtot_row-sgst_sub .

      me->mask_currency(
               EXPORTING
                 iv_amount     = CONV netwr( lv_t_total_amount )
   " Net Value in Document Currency
               IMPORTING
                 ev_masked_amt = es_subtot_row-
totamt_sub            " Amount in indian notation
             ).
      CONDENSE es_subtot_row-totamt_sub .
    ELSE.

      me->mask_currency(
              EXPORTING
                iv_amount     = CONV netwr( lv_t_amount )                " Ne
t Value in Document Currency
              IMPORTING
                ev_masked_amt = es_subtot_row-
taxable_sub             " Amount in indian notation
            ).
      CONDENSE es_subtot_row-taxable_sub .

      me->mask_currency(
             EXPORTING
               iv_amount     = CONV netwr( lv_t_igst )                " Net V
alue in Document Currency
             IMPORTING
               ev_masked_amt = es_subtot_row-
cgst_sub             " Amount in indian notation
           ).
      CONDENSE es_subtot_row-cgst_sub .

      IF is_bil_invoice-hd_gen-bil_waerk <> 'INR' .
        me->mask_currency(
               EXPORTING
                 iv_amount     = CONV netwr( lv_t_total_amount )
   " Net Value in Document Currency
               IMPORTING
                 ev_masked_amt = es_subtot_row-
sgst_sub             " Amount in indian notation
             ).
        CONDENSE es_subtot_row-sgst_sub .
      ENDIF.

      me->mask_currency(
             EXPORTING
               iv_amount     = CONV netwr( lv_t_total_amount_inr )
     " Net Value in Document Currency
```

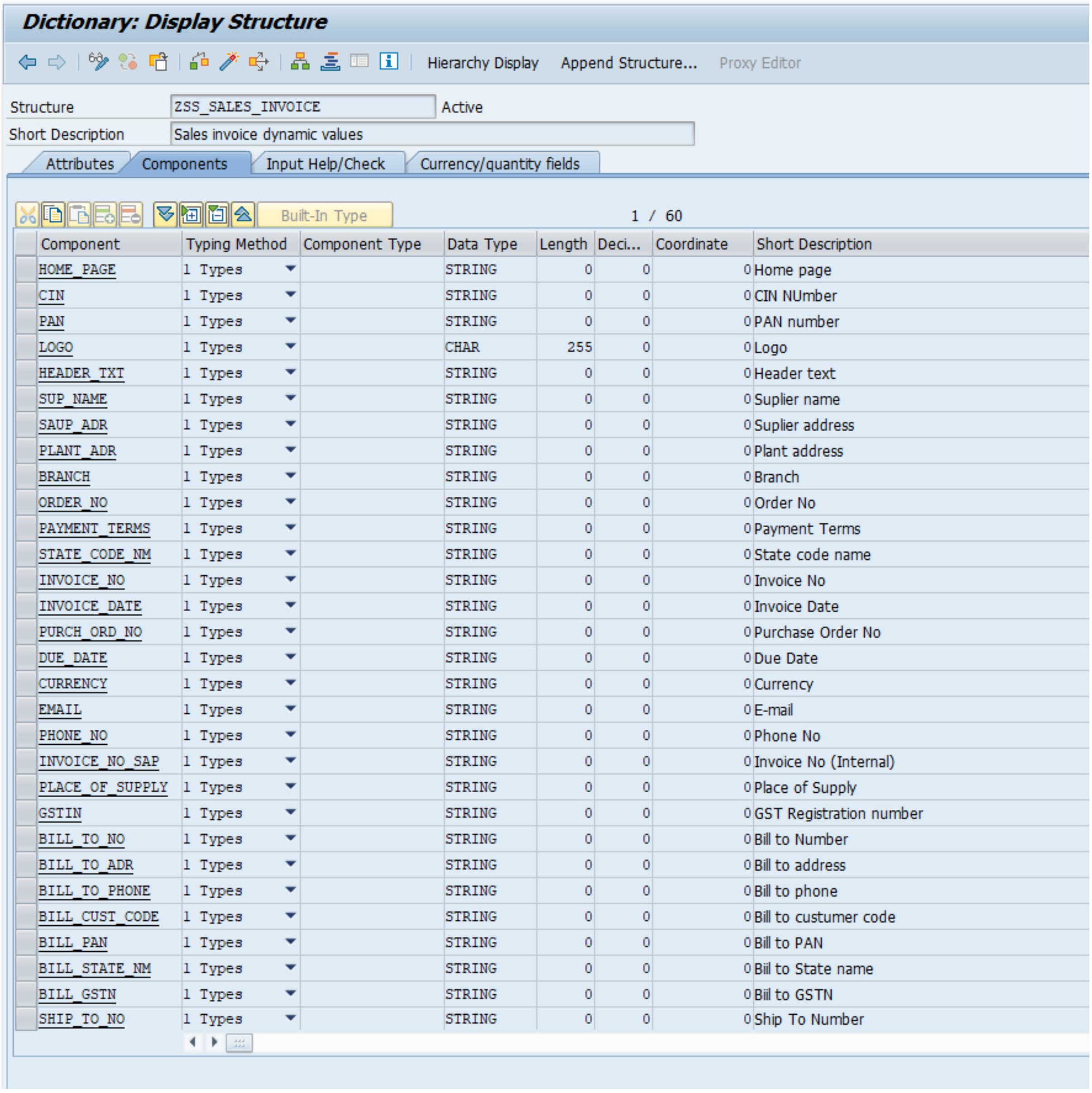
```
             IMPORTING
               ev_masked_amt = es_subtot_row-
totamt_sub            " Amount in indian notation
           ).
      CONDENSE es_subtot_row-totamt_sub .


    ENDIF.

    CLEAR: lv_tot_tot.

    me->mask_currency(
           EXPORTING
             iv_amount     = CONV netwr( lv_t_total_amount  )
" Net Value in Document Currency
           IMPORTING
             ev_masked_amt = lv_tot_tot            " Amount in indian notat
ion
         ).
    CONDENSE lv_tot_tot .

    "Fill sub total row
    IF is_bil_invoice-hd_gen-bil_waerk <> |INR|.
      es_subtot_row-totamt_sub_for = lv_tot_tot.
    ELSE.
      es_subtot_row-totamt_sub_inr = lv_tot_tot.
    ENDIF.

    "Amount in words
*    lv_tot_amt = trunc( lv_t_total_amount ).
    lv_tot_amt =  lv_t_total_amount.
    IF ls_head-bil_waerk = 'INR'.  " for India
      CALL FUNCTION 'HR_IN_CHG_INR_WRDS'
        EXPORTING
          amt_in_num          = lv_tot_amt
        IMPORTING
          amt_in_words        = lv_words
        EXCEPTIONS
          data_type_mismatch = 1
          OTHERS             = 2.
      IF sy-subrc <> 0.
* Implement suitable error handling here
      ENDIF.

      TRANSLATE lv_words TO LOWER CASE.
      CONDENSE lv_words.
      TRANSLATE lv_words+0(1) TO UPPER CASE.
      DATA(lv_amount_words) = CONV string( lv_words ).

    ELSE.                          " For other countries
      CALL FUNCTION 'SPELL_AMOUNT'
        EXPORTING
          amount    = lv_tot_amt
          currency  = ls_head-bil_waerk
          language  = sy-langu
        IMPORTING
          in_words  = lv_words_i
        EXCEPTIONS
          not_found = 1
          too_large = 2
          OTHERS    = 3.
      IF sy-subrc <> 0.
*   Implement suitable error handling here
      ENDIF.
      IF lv_words_i-decword IS INITIAL.
        lv_words = lv_words_i-word.
      ELSE.
        CONCATENATE lv_words_i-word 'AND' lv_words_i-decword
        INTO lv_words SEPARATED BY space.
      ENDIF.

      TRANSLATE lv_words TO LOWER CASE.
      CONDENSE lv_words.
      TRANSLATE lv_words+0(1) TO UPPER CASE.

      DATA(lv_waers) = CONV waers_curc( is_bil_invoice-hd_gen-bil_waerk ).
      SELECT SINGLE ktext
        FROM tcurt
        INTO @DATA(lv_curr_word)
        WHERE spras EQ 'E'
        AND waers EQ @lv_waers.
      IF sy-subrc EQ 0.
        TRANSLATE   lv_curr_word TO UPPER CASE.
        CONCATENATE lv_curr_word lv_words INTO lv_words SEPARATED BY space.
      ENDIF.
      lv_amount_words = lv_words.
    ENDIF.
    ev_amount_words = lv_amount_words.
  ENDMETHOD.
```



Class/Interface ZCL_SD_INVOICE Implemented / Active

Parameters of Method GET_HEADER_FOOTER_INFO

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
IS_BIL_INVOICE	Importing	☑	☑	Type	LBBIL_INVOICE		Billing Data: Transfer Structure to Sm...
IS_BIL_INVOICE2	Importing	☑	☑	Type	BIL_S_PRT_INTERFACE		Billing Document: Interface Structure ..
EV_SUP_NAME	Exporting	☑	□	Type	STRING		
EV_SUP_ADDRESS	Exporting	☑	□	Type	STRING		
ES_IRN_DETAILS	Exporting	☑	□	Type	ZSS_IRN_DET		IRN details
EV_BILL_TO	Exporting	☑	□	Type	KUNNR		bill to
EV_SHIP_TO	Exporting	☑	□	Type	KUNNR		ship to
ES_INVOICE_DET	Exporting	☑	□	Type	ZSS_SALES_INVOICE		Sales invoice dynamic values
EV_DECLARATION	Exporting	☑	□	Type	STRING		
EV_TERMS	Exporting	☑	□	Type	STRING		
EV_BANK	Exporting	☑	□	Type	STRING		
EV_SEAL	Exporting	☑	□	Type	XSTRING		
		□	□	Type			
		□	□	Type			

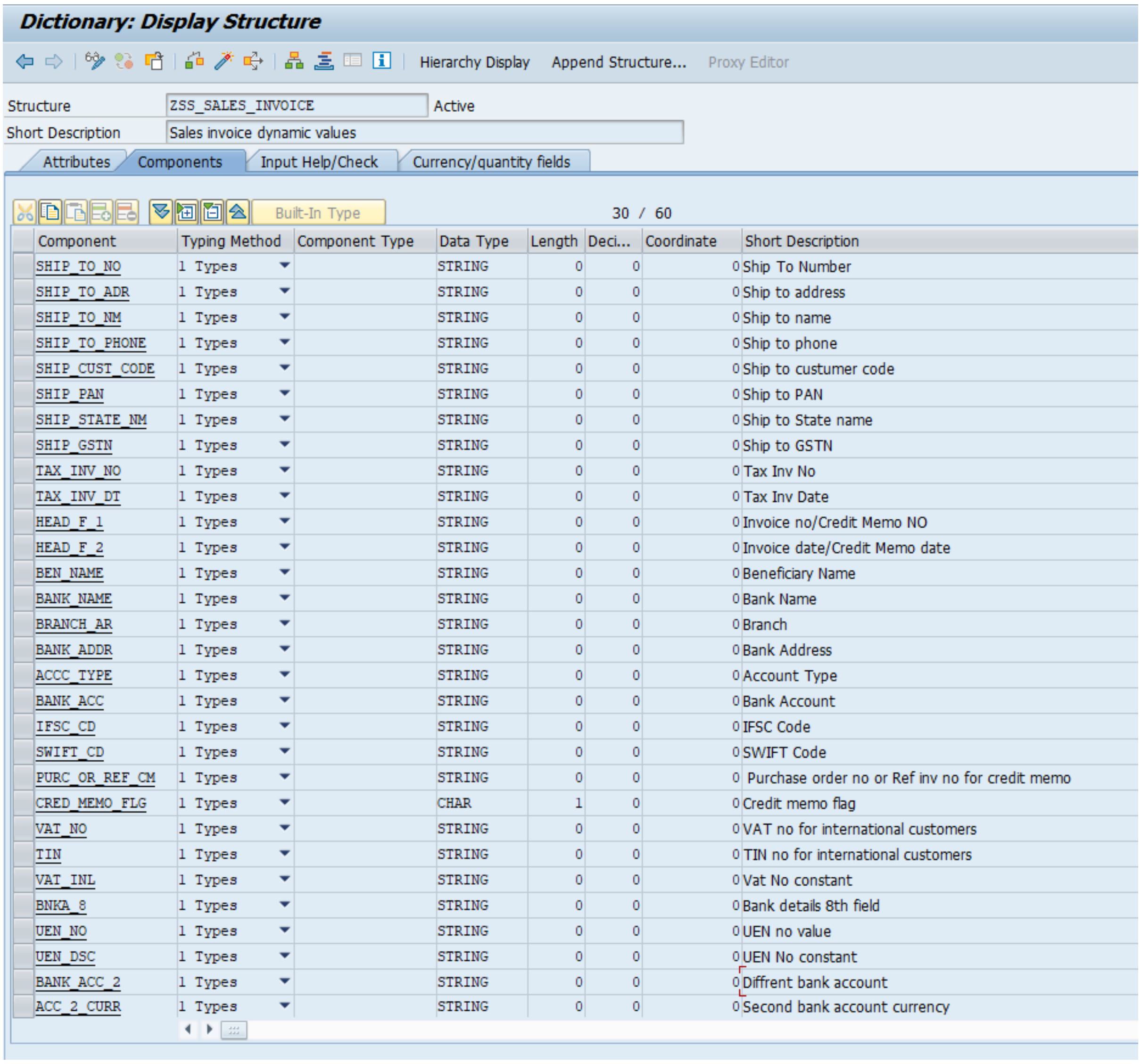


Dictionary: Display Structure

Structure	ZSS_SALES_INVOICE	Active
Short Description	Sales invoice dynamic values	

1 / 60

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description
HOME_PAGE	1 Types		STRING	0	0	0	Home page
CIN	1 Types		STRING	0	0	0	CIN NUmber
PAN	1 Types		STRING	0	0	0	PAN number
LOGO	1 Types		CHAR	255	0	0	Logo
HEADER_TXT	1 Types		STRING	0	0	0	Header text
SUP_NAME	1 Types		STRING	0	0	0	Suplier name
SAUP_ADR	1 Types		STRING	0	0	0	Suplier address
PLANT_ADR	1 Types		STRING	0	0	0	Plant address
BRANCH	1 Types		STRING	0	0	0	Branch
ORDER_NO	1 Types		STRING	0	0	0	Order No
PAYMENT_TERMS	1 Types		STRING	0	0	0	Payment Terms
STATE_CODE_NM	1 Types		STRING	0	0	0	State code name
INVOICE_NO	1 Types		STRING	0	0	0	Invoice No
INVOICE_DATE	1 Types		STRING	0	0	0	Invoice Date
PURCH_ORD_NO	1 Types		STRING	0	0	0	Purchase Order No
DUE_DATE	1 Types		STRING	0	0	0	Due Date
CURRENCY	1 Types		STRING	0	0	0	Currency
EMAIL	1 Types		STRING	0	0	0	E-mail
PHONE_NO	1 Types		STRING	0	0	0	Phone No
INVOICE_NO_SAP	1 Types		STRING	0	0	0	Invoice No (Internal)
PLACE_OF_SUPPLY	1 Types		STRING	0	0	0	Place of Supply
GSTIN	1 Types		STRING	0	0	0	GST Registration number
BILL_TO_NO	1 Types		STRING	0	0	0	Bill to Number
BILL_TO_ADR	1 Types		STRING	0	0	0	Bill to address
BILL_TO_PHONE	1 Types		STRING	0	0	0	Bill to phone
BILL_CUST_CODE	1 Types		STRING	0	0	0	Bill to custumer code
BILL_PAN	1 Types		STRING	0	0	0	Bill to PAN
BILL_STATE_NM	1 Types		STRING	0	0	0	Bill to State name
BILL_GSTN	1 Types		STRING	0	0	0	Bill to GSTN
SHIP_TO_NO	1 Types		STRING	0	0	0	Ship To Number

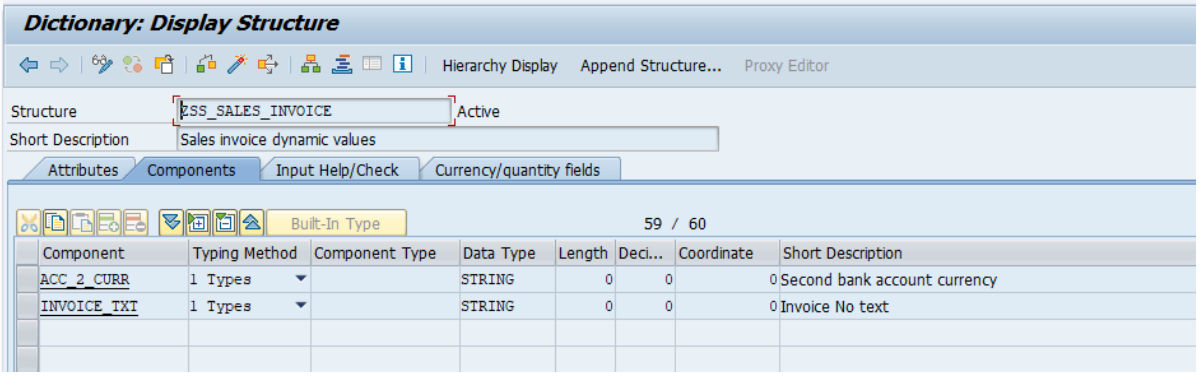


Structure ZSS_SALES_INVOICE Active

Short Description Sales invoice dynamic values

30 / 60

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description
SHIP_TO_NO	1 Types		STRING	0	0	0	Ship To Number
SHIP_TO_ADR	1 Types		STRING	0	0	0	Ship to address
SHIP_TO_NM	1 Types		STRING	0	0	0	Ship to name
SHIP_TO_PHONE	1 Types		STRING	0	0	0	Ship to phone
SHIP_CUST_CODE	1 Types		STRING	0	0	0	Ship to custumer code
SHIP_PAN	1 Types		STRING	0	0	0	Ship to PAN
SHIP_STATE_NM	1 Types		STRING	0	0	0	Ship to State name
SHIP_GSTN	1 Types		STRING	0	0	0	Ship to GSTN
TAX_INV_NO	1 Types		STRING	0	0	0	Tax Inv No
TAX_INV_DT	1 Types		STRING	0	0	0	Tax Inv Date
HEAD_F_1	1 Types		STRING	0	0	0	Invoice no/Credit Memo NO
HEAD_F_2	1 Types		STRING	0	0	0	Invoice date/Credit Memo date
BEN_NAME	1 Types		STRING	0	0	0	Beneficiary Name
BANK_NAME	1 Types		STRING	0	0	0	Bank Name
BRANCH_AR	1 Types		STRING	0	0	0	Branch
BANK_ADDR	1 Types		STRING	0	0	0	Bank Address
ACCC_TYPE	1 Types		STRING	0	0	0	Account Type
BANK_ACC	1 Types		STRING	0	0	0	Bank Account
IFSC_CD	1 Types		STRING	0	0	0	IFSC Code
SWIFT_CD	1 Types		STRING	0	0	0	SWIFT Code
PURC_OR_REF_CM	1 Types		STRING	0	0	0	Purchase order no or Ref inv no for credit memo
CRED_MEMO_FLG	1 Types		CHAR	1	0	0	Credit memo flag
VAT_NO	1 Types		STRING	0	0	0	VAT no for international customers
TIN	1 Types		STRING	0	0	0	TIN no for international customers
VAT_INL	1 Types		STRING	0	0	0	Vat No constant
BNKA_8	1 Types		STRING	0	0	0	Bank details 8th field
UEN_NO	1 Types		STRING	0	0	0	UEN no value
UEN_DSC	1 Types		STRING	0	0	0	UEN No constant
BANK_ACC_2	1 Types		STRING	0	0	0	Diffrent bank account
ACC_2_CURR	1 Types		STRING	0	0	0	Second bank account currency



Structure ZSS_SALES_INVOICE Active

Short Description Sales invoice dynamic values

59 / 60

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description
ACC_2_CURR	1 Types		STRING	0	0	0	Second bank account currency
INVOICE_TXT	1 Types		STRING	0	0	0	Invoice No text

```
  METHOD get_header_footer_info.
*-----------------------------------------------------------------------*
*    Program    :  Updates Tables (   )     Process data (  )
*                  Outputs List    (   )    Validation    (  )
*
*    Module                      : SD
*    WRICEF Object Id            : SD_DV020
*    Functional Specifications : FS_SD_DV020_SALES_INVOCIE
*    Technical Spec No           : TS_SD_DV020_SALES_INVOCIE
*    Functional Consultant     : Parul Singh
*    Date Created (dd-mm-yyyy) : 01-12-2024
*    Technical Consultant      : Pravesh Patil
*    Client                    : upGrad
*    Project Name              : upGrad
*    Company                   : TechWit Business Solutions
*---------------------------------------------------------------------*
*Pravesh Patil | 08.07.2025 | PP08.07.2025 | Added to fetch sez number dynamic based on billing date
*----------------------------------------------------------------------*
*Amritansu Das | 06.11.2025 | Added correction for reference date
*----------------------------------------------------------------------*
    CONSTANTS: cv_1810         TYPE char4  VALUE '1810',
               cv_5            TYPE c VALUE '5',
               cv_m            TYPE c VALUE 'M',
               cv_o            TYPE c VALUE 'O',
               cv_p            TYPE c VALUE 'P',
               cv_u            TYPE c VALUE 'U',
               cv_c            TYPE c VALUE 'C',
               cv_upgrad       TYPE string VALUE 'upGrad Education Private Limited',
               cv_outside_ind  TYPE string  VALUE '97 - Outside India',
               cv_cin_upgrad   TYPE char30 VALUE 'U80902MH2012PTC258559',
               cv_na           TYPE char2 VALUE 'NA',
               cv_tax_inv      TYPE char25 VALUE 'TAX INVOICE',
               cv_bos          TYPE char25 VALUE 'BILL OF SUPPLY',
               cv_credit_memo  TYPE char25 VALUE 'CREDIT MEMO',
               cv_debit_memo   TYPE char25 VALUE 'DEBIT MEMO',
               cv_performa     TYPE char25 VALUE 'PROFORMA INVOICE',
               cv_inv_no       TYPE char25 VALUE 'Invoice No:',
               cv_inv_dt       TYPE char25 VALUE 'Invoice Date:',
               cv_cred_memo_no TYPE char25 VALUE 'Credit Memo No:',
               cv_cred_memo_dt TYPE char25 VALUE 'Credit Memo Date:',
               cv_purc_ord_no  TYPE char20 VALUE 'Purchase Order No: ',
               cv_reference_cm TYPE char20 VALUE 'Ref Inv No: '.

    DATA: lv_logo_bukrs TYPE char255,
          lv_spart      TYPE spart.

    DATA: lt_lines    TYPE STANDARD TABLE OF string,
          it_payterms TYPE TABLE OF ttext.

    DATA: ls_addr      TYPE szadr_addr1_complete,
          wa_item      TYPE lbbil_it_gen,
          wa_bill      TYPE msr_s_lbbil_it,
          wa_price     TYPE lbbil_it_price,
          ls_suppaddr  TYPE  szadr_printform_table,
          ls_compcdddr TYPE  szadr_printform_table,
          ls_suppgst   TYPE  stcd3,
          ls_billaddr  TYPE  szadr_printform_table,
          ls_shipaddr  TYPE  szadr_printform_table,
          ls_kopf      TYPE  vbdkr.
    DATA: lt_decla_lines TYPE STANDARD TABLE OF tline,
          lt_terms_lines TYPE STANDARD TABLE OF tline,
          lt_bank        TYPE STANDARD TABLE OF tline,
          lv_declaration TYPE string,
          lv_terms       TYPE string,
          lv_bank        TYPE string.


    DATA(it_item)   = is_bil_invoice-it_gen.
    DATA(gs_it_gen) = is_bil_invoice-it_gen.
    DATA(gs_head)   = is_bil_invoice-hd_gen.
    DATA(gs_org)    = is_bil_invoice-hd_org.
    DATA(ls_ref)    = is_bil_invoice-hd_ref.
    DATA(it_addr)   = is_bil_invoice-hd_adr.

************************ Header Part
******* Header Text

    IF gs_org-salesorg EQ cv_1810.
      DATA(lv_header) = COND string( WHEN gs_head-
bil_vbtype+0(1) = cv_5 THEN cv_tax_inv
                                     WHEN gs_head-
bil_vbtype+0(1) = cv_m THEN cv_tax_inv
                                     WHEN gs_head-
bil_vbtype+0(1) = cv_o THEN cv_credit_memo
                                     WHEN gs_head-
bil_vbtype+0(1) = cv_p THEN cv_debit_memo
                                     WHEN gs_head-
bil_vbtype+0(1) = cv_u THEN  cv_performa
                                     ).
    ENDIF.

    DATA(lv_bukrs) = gs_org-comp_code.
    DATA(lv_vkorg) = gs_org-salesorg.
    lv_spart =  gs_org-division.
    DATA(lv_vtweg) = gs_org-distrb_channel.

    SELECT SINGLE *
       FROM zsd_inv_div
       INTO @DATA(ls_inv_div)
       WHERE sales_org = @lv_vkorg
       AND   dist_chnl = @lv_vtweg
       AND   division  = @lv_spart.
    IF sy-subrc EQ 0.
    ENDIF.


**************IRN details
    SELECT SINGLE *
      FROM vbrk
      INTO @DATA(ls_vbrk)
      WHERE vbeln = @gs_head-bil_number.
    IF sy-subrc EQ 0.
```

```
      SELECT SINGLE
        qrcode ,
        inv_reg_num ,
        ack_no ,
        ack_date
        FROM edoineinv
        INTO  @DATA(gs_irndetails)
        WHERE inv_num = @ls_vbrk-xblnr.
*      IF SY-SUBRC EQ 0.
      IF gs_irndetails-ack_date IS INITIAL.
        CLEAR gs_irndetails-ack_date.
        gs_irndetails-ack_date = cv_na.
      ENDIF.
      IF gs_irndetails-ack_no IS INITIAL.
        gs_irndetails-ack_no = cv_na.
      ENDIF.
      IF gs_irndetails-inv_reg_num IS INITIAL.
        gs_irndetails-inv_reg_num = cv_na.
      ENDIF.
      es_irn_details = CORRESPONDING #( gs_irndetails ).
*      ENDIF.
    ENDIF.

    "Error Billing document is not released to accounting at Print Preview
    IF ls_vbrk-buchk <> cv_c AND ls_vbrk-fkart <> |F5|.
      MESSAGE TEXT-002 TYPE gc_e DISPLAY LIKE gc_e.
    ENDIF.

**************Logo Name in SE74
    IF lv_spart = 14.
      lv_logo_bukrs = |1840_SD|.
    ELSEIF lv_spart = 40 .
      lv_logo_bukrs = |1820_SD|.
    ELSEIF lv_spart = 41 .
***      LV_LOGO_BUKRS = |ZREKRUT_LOGO|.
      lv_logo_bukrs = |1810_41_SD|.
    ELSE.
      lv_logo_bukrs = |1810_SD|.
    ENDIF.

*************Branch
    DATA(lv_bupla) = CONV j_1bbranc_( is_bil_invoice2-head_detail-vbdkr-
bupla ).
    SELECT SINGLE name
      FROM j_1bbranch
      WHERE branch = @lv_bupla
      AND   bukrs  = @lv_bukrs
      INTO @DATA(lv_branch).
    IF sy-subrc EQ 0.ENDIF.
************supplier name - Plant address

    READ TABLE it_item INTO DATA(ls_item) INDEX 1.
    SELECT SINGLE werks FROM vbrp
```

```
      INTO @DATA(lv_werks)
      WHERE vbeln = @ls_item-bil_number
      AND   posnr = @ls_item-itm_number.
    IF sy-subrc EQ 0.
      SELECT SINGLE adrnr, land1, regio
        FROM t001w
        INTO (@DATA(lv_adrnr), @DATA(lv_cntry), @DATA(lv_reg))
        WHERE werks = @lv_werks.
      IF sy-subrc EQ 0.ENDIF.
    ENDIF.

    "Check if salse address is available then update same instead plant add
ress
    SELECT SINGLE vgbel
      FROM vbrp
      INTO @DATA(lv_vgbel)
      WHERE vbeln = @gs_head-bil_number.
    IF sy-subrc EQ 0 AND lv_vgbel IS NOT INITIAL.
      SELECT SINGLE vkbur
        FROM vbak
        INTO @DATA(lv_vkbur)
        WHERE vbeln = @lv_vgbel.
      IF lv_vkbur IS NOT INITIAL.
        SELECT SINGLE adrnr
          FROM tvbur
          INTO @lv_adrnr
          WHERE vkbur = @lv_vkbur.
      ENDIF.
    ENDIF.

    IF gs_head-bil_waerk = 'INR'.
      SELECT SINGLE * FROM vbrp
        INTO @DATA(ls_vbrp)
        WHERE vbeln EQ @ls_item-bil_number.
      IF sy-subrc EQ 0 AND
        ( ls_vbrp-taxm1 = '3' AND ( ls_vbrp-taxm2 = '3' OR ls_vbrp-
taxm3 = '3' OR ls_vbrp-taxm4 = '3' ) ).

        DATA(lv_aland) = ls_vbrp-aland.
        DATA(lv_wkreg) = ls_vbrp-wkreg.
        SELECT SINGLE legal_state_code, bezei
           INTO @DATA(ls_sup_place)
           FROM j_1istatecdm
           WHERE land1 EQ @lv_aland
           AND std_state_code EQ @lv_wkreg.
      ELSE.
        SELECT c~legal_state_code, c~bezei FROM vbpa
            AS a INNER JOIN kna1 AS b ON a~kunnr = b~kunnr
            INNER JOIN j_1istatecdm AS c ON b~land1 EQ c~land1
            AND b~regio EQ c~std_state_code
            WHERE vbeln = @ls_item-bil_number
            AND parvw = 'RE'
            ORDER BY c~bezei
            INTO @ls_sup_place UP TO 1 ROWS.
        ENDSELECT.
        IF sy-subrc EQ 0.ENDIF.
      ENDIF.

      DATA(gv_sup_place) = ls_sup_place-
legal_state_code && | - | && ls_sup_place-bezei.
    ELSE.
      gv_sup_place = cv_outside_ind.
    ENDIF.

    "Update Header to Bill Of Supply based on MAtarial Tax classification
    IF ls_vbrk-fkart <> |F5|.
      lv_header = COND #( WHEN ls_vbrp-taxm1 EQ '1' THEN cv_bos
                          ELSE lv_header ).
    ENDIF.

    CALL FUNCTION 'ADDRESS_INTO_PRINTFORM'
      EXPORTING
        address_type                   = '1'
        address_number                 = lv_adrnr
        number_of_lines                = 15
        street_has_priority            = 'X'
        iv_country_name_separate_line  = 'X'
      IMPORTING
        address_printform_table        = ls_suppaddr
      EXCEPTIONS
        address_blocked                = 1
        person_blocked                 = 2
        contact_person_blocked         = 3
        addr_to_be_formated_is_blocked = 4
        OTHERS                         = 5.
    IF sy-subrc <> 0.
* Implement suitable error handling here
    ENDIF.


    IF line_exists( ls_suppaddr[ line_type = 'L' address_line = 'INDIA' ] )
.
      DELETE ls_suppaddr WHERE line_type = 'R'.
      DELETE ls_suppaddr WHERE line_type = 'L'.
    ENDIF.

    LOOP AT ls_suppaddr INTO DATA(wa_suppaddr).
      CONCATENATE ls_plantadr wa_suppaddr-
address_line INTO DATA(ls_plantadr)
       SEPARATED BY  space."GC_NEWLINE.
    ENDLOOP.

    "company code adress
    SELECT SINGLE adrnr
    FROM t001 INTO @DATA(lv_adrnr1)
    WHERE bukrs = @lv_bukrs.
    CALL FUNCTION 'ADDRESS_INTO_PRINTFORM'
      EXPORTING
        address_type                   = '1'
        address_number                 = lv_adrnr1
        number_of_lines                = 15
        street_has_priority            = 'X'
        iv_country_name_separate_line  = 'X'
      IMPORTING
        address_printform_table        = ls_compcdddr
      EXCEPTIONS
        address_blocked                = 1
        person_blocked                 = 2
        contact_person_blocked         = 3
        addr_to_be_formated_is_blocked = 4
        OTHERS                         = 5.
    IF sy-subrc <> 0.
* Implement suitable error handling here
    ENDIF.


    DATA(lv_sup_name) = COND string( WHEN line_exists( ls_compcdddr[ 1 ] )
THEN ls_compcdddr[ 1 ]-address_line ).
    IF ls_compcdddr IS NOT INITIAL.
      DELETE ls_compcdddr INDEX 1.
    ENDIF.

    IF line_exists( ls_compcdddr[ line_type = 'L' address_line = 'INDIA' ]
).
      DELETE ls_compcdddr WHERE line_type = 'R'.
      DELETE ls_compcdddr WHERE line_type = 'L'.
    ENDIF.

    LOOP AT ls_compcdddr INTO DATA(wa_compadr).
      CONCATENATE ls_compadr wa_compadr-address_line INTO DATA(ls_compadr)
       SEPARATED BY  space. "GC_NEWLINE.
    ENDLOOP.


    "fetch the phone num, homepage, supp mail id
    CALL FUNCTION 'ADDR_GET_COMPLETE'
      EXPORTING
        addrnumber              = gs_org-salesorg_adr
      IMPORTING
        addr1_complete          = ls_addr
      EXCEPTIONS
        parameter_error         = 1
        address_not_exist       = 2
        internal_error          = 3
        wrong_access_to_archive = 4
        address_blocked         = 5
        OTHERS                  = 6.
    IF sy-subrc <> 0.
* Implement suitable error handling here
```

```
    ENDIF.
    READ TABLE ls_addr-adtel_tab INTO DATA(wa_adtel) INDEX 1.
    READ TABLE ls_addr-adsmtp_tab INTO DATA(wa_adsmtp) INDEX 1.
    READ TABLE ls_addr-adrct_tab INTO DATA(wa_adrct) INDEX 1.


    "supplier GST

    CALL FUNCTION 'J_1IG_GET_PLANT_DETAILS'
      EXPORTING
        im_werks = lv_werks
        im_bukrs = gs_org-comp_code
      IMPORTING
*       EX_BRANCH       =
        ex_gstin = ls_suppgst.

    "fetch the bill to party address
** Getting address number from adr table
    READ TABLE it_addr INTO DATA(gs_hd_adr) WITH KEY partn_role = 'RE'.
    DATA(gv_bp) = gs_hd_adr-partn_numb.
*    EV_BILL_TO = GS_HD_ADR-PARTN_NUMB.
    DATA(gvbpadr) = gs_hd_adr-addr_no. " use to print
    CALL FUNCTION 'ADDRESS_INTO_PRINTFORM'
      EXPORTING
        address_type                   = '1'
        address_number                 = gs_hd_adr-addr_no
        number_of_lines                = 15
        street_has_priority            = 'X'
        iv_country_name_separate_line  = 'X'
      IMPORTING
        address_printform_table        = ls_billaddr
      EXCEPTIONS
        address_blocked                = 1
        person_blocked                 = 2
        contact_person_blocked         = 3
        addr_to_be_formated_is_blocked = 4
        OTHERS                         = 5.
    IF sy-subrc <> 0.
* Implement suitable error handling here
    ENDIF.
    DELETE ls_billaddr WHERE line_type = 'A'.
    IF line_exists( ls_billaddr[ line_type = 'L' address_line = 'INDIA' ] )
.
      DELETE ls_billaddr WHERE line_type = 'R'.
      DELETE ls_billaddr WHERE line_type = 'L'.
    ENDIF.

    LOOP AT ls_billaddr INTO DATA(wa_billaddr).
      CONCATENATE ls_bill wa_billaddr-address_line INTO DATA(ls_bill)
      SEPARATED BY space." gc_newline.
    ENDLOOP.

    "Bill to PAN
    SELECT SINGLE regio, j_1ipanno
      FROM kna1
      WHERE kunnr EQ @gv_bp
      INTO @DATA(ls_bill_pan).

    "Bill to GSTIN
    SELECT SINGLE taxnum
         FROM dfkkbptaxnum
         WHERE partner EQ @gv_bp
         AND taxtype EQ 'IN3'
         INTO @DATA(lv_bill_gstin).

    "Bill to State code and name
    me->get_state_codenname( EXPORTING
                              iv_regio = ls_bill_pan-regio
                             IMPORTING
                              ev_state_codenm = DATA(lv_bill_statenmcd) ).


    "fetch the shill to party address
    READ TABLE it_addr INTO gs_hd_adr WITH KEY partn_role = 'WE'.
    DATA(gv_sp) = gs_hd_adr-partn_numb.

    DATA(gvspadr) = gs_hd_adr-addr_no. "use to print


    CALL FUNCTION 'ADDRESS_INTO_PRINTFORM'
      EXPORTING
        address_type                   = '1'
        address_number                 = gs_hd_adr-addr_no
        number_of_lines                = 15
        street_has_priority            = 'X'
        iv_country_name_separate_line  = 'X'
      IMPORTING
*       ADRSWA_OUT                     =
        address_printform_table        = ls_shipaddr
      EXCEPTIONS
        address_blocked                = 1
        person_blocked                 = 2
        contact_person_blocked         = 3
        addr_to_be_formated_is_blocked = 4
        OTHERS                         = 5.
    IF sy-subrc <> 0.
* Implement suitable error handling here
    ENDIF.
    DELETE ls_shipaddr WHERE line_type = 'A'.
    IF line_exists( ls_shipaddr[ line_type = 'L' address_line = 'INDIA' ] )
.
      DELETE ls_shipaddr WHERE line_type = 'R'.
      DELETE ls_shipaddr WHERE line_type = 'L'.
    ENDIF.

*READ TABLE ls_shipaddr INTO wa_shipaddr INDEX 1.
```

```
LOOP AT ls_shipaddr INTO DATA(wa_shipaddr).
  CONCATENATE ls_ship wa_shipaddr-address_line INTO DATA(ls_ship)
   SEPARATED BY space.
ENDLOOP.

"Ship to PAN
SELECT SINGLE j_1ipanno, regio
  FROM kna1
  WHERE kunnr EQ @gv_sp
  INTO @DATA(ls_ship_pan).
IF sy-subrc EQ 0.ENDIF.

"Ship to GSTIN
SELECT SINGLE taxnum
     FROM dfkkbptaxnum
     WHERE partner EQ @gv_sp
     AND taxtype EQ 'IN3'
     INTO @DATA(lv_ship_gstin).
IF sy-subrc EQ 0.ENDIF.

"Ship to State code and name
me->get_state_codenname( EXPORTING
                          iv_regio = ls_ship_pan-regio
                         IMPORTING
                          ev_state_codenm = DATA(lv_ship_statenmcd) ).

"fetch the buyer GST
READ TABLE it_addr INTO gs_hd_adr WITH KEY partn_role = 'RE'.

SELECT SINGLE land1 INTO @DATA(lv_land1) FROM kna1
  WHERE kunnr = @gs_hd_adr-partn_numb.
IF sy-subrc EQ 0.ENDIF.

SELECT SINGLE high
  FROM tvarvc
  INTO @DATA(lv_taxtype)
  WHERE name = 'ZSD_CUST_TAX_TYPE' AND low = @lv_land1.
IF sy-subrc EQ 0.ENDIF.

IF lv_taxtype IS NOT INITIAL .

  SELECT SINGLE taxnum
    FROM dfkkbptaxnum
    INTO @DATA(lv_taxnum)
    WHERE partner = @gs_hd_adr-partn_numb AND
          taxtype = @lv_taxtype.
  IF sy-subrc EQ 0.ENDIF.
ENDIF.
"duedate
DATA: ls_comwa TYPE vbco3,
      lt_pos   TYPE TABLE OF vbdpr.
ls_comwa-vbeln = gs_head-bil_number.
CALL FUNCTION 'RV_BILLING_PRINT_VIEW'
```

```
      EXPORTING
        comwa                        = ls_comwa
      IMPORTING
        kopf                         = ls_kopf
      TABLES
        pos                          = lt_pos
      EXCEPTIONS
        terms_of_payment_not_in_t052 = 1
        OTHERS                       = 2.
    IF sy-subrc <> 0.
* Implement suitable error handling here
    ENDIF.

    "payment terms
    DATA(lv_other_terms) = is_bil_invoice-hd_gen_descript-name_paymterm.

    "company pan
    SELECT SINGLE paval FROM t001z INTO @DATA(lv_pan)
     WHERE party = 'J_1I02' AND bukrs = @gs_org-comp_code.
    IF sy-subrc EQ 0.ENDIF.

*      SELECT SINGLE xblnr FROM vbrk INTO @DATA(lv_xblnr) --
commeneted by amritansu for fetching date
*      WHERE vbeln = @gs_head-bil_number.

    SELECT SINGLE zuonr, xblnr FROM vbrk INTO @DATA(ls_vbrk1)  " ++ amritans
u for fetching date
      WHERE vbeln = @gs_head-bil_number.
*    IF sy-subrc = 0 AND lv_xblnr IS NOT INITIAL. --amritansu
    IF sy-subrc = 0 AND ls_vbrk1-xblnr IS NOT INITIAL.
      DATA(lv_zounr) = CONV xblnr( ls_vbrk1-
zuonr ).            " ++ amritansu for fetching date
      SELECT SINGLE fkdat, xblnr FROM vbrk INTO @DATA(ls_vbrk2)  " ++ amrita
nsu for fetching date
       WHERE xblnr = @lv_zounr.
    ENDIF.

    "PO NUMBER
    DATA(gt_po) = is_bil_invoice-it_refpurord.
    READ TABLE gt_po INTO DATA(gs_po) INDEX 1.
    IF sy-subrc EQ 0.
    ENDIF.

    "SEAL
    "Get State code and name
    me->get_state_codenname( EXPORTING
                              iv_bukrs = lv_bukrs
                              iv_bupla = lv_bupla
                             IMPORTING
                              ev_state_codenm = DATA(lv_statenmcd) ).


    "CIN Constant for now
```

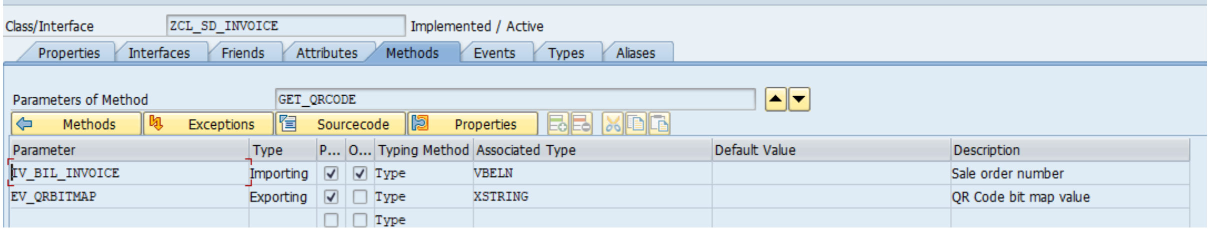
```
   SELECT SINGLE paval
     FROM t001z
     INTO @DATA(lv_cin_no)
     WHERE bukrs = @lv_bukrs
     AND party   = 'Z_CIN'.

   "Invocie/Invoice date or Credit memo or credit memo date
   IF gs_head-bil_vbtype+0(1) = cv_o.
     DATA(lv_head_f1) = cv_cred_memo_no.
     DATA(lv_head_f2) = cv_cred_memo_dt.
   ELSE.
     lv_head_f1 = cv_inv_no.
     lv_head_f2 = cv_inv_dt.
   ENDIF.

   "Credit memo reference number based on document number
   "Purchase order number for normal invocie / reference number for credit
memo

   DATA(lv_po_or_ref_inv) = cv_purc_ord_no.
   IF gs_head-bil_vbtype+0(1) = cv_o.
     DATA(lv_cred_memo_flg) = abap_true.
     SELECT SINGLE aubel
       FROM vbrp
       INTO @DATA(lv_aubel)
       WHERE vbeln EQ @gs_head-bil_number.
     IF sy-subrc EQ 0 AND lv_aubel IS NOT INITIAL.
       SELECT SINGLE xblnr, erdat
         FROM vbak
         INTO (@DATA(lv_cm_ref),@DATA(lv_refcre_dt))
         WHERE vbeln = @lv_aubel.
       IF sy-subrc EQ 0 AND lv_cm_ref IS NOT INITIAL.
*          GS_PO-PURCH_NO_C = LV_CM_REF.
         lv_po_or_ref_inv = cv_reference_cm.
       ENDIF.
     ENDIF.
   ELSE.
     lv_cred_memo_flg = 'Y'.
     CONDENSE lv_po_or_ref_inv.
   ENDIF.


   "++ Pravesh Patil added if ref details not found
   IF lv_aubel IS NOT INITIAL.
     SELECT SINGLE bstkd, bstdk
       FROM vbkd
       INTO @DATA(ls_vbkd)
       WHERE vbeln = @lv_aubel.
     IF sy-subrc EQ 0.
       IF lv_cm_ref IS INITIAL.
         lv_cm_ref = ls_vbkd-bstkd.
       ENDIF.
```

```
        IF ls_vbrk2-fkdat IS INITIAL.
          ls_vbrk2-fkdat = ls_vbkd-bstdk.
        ENDIF.
      ENDIF.
    ENDIF.

    "Assigning all required data in common structure to fill form layout
    es_invoice_det = VALUE #(
                             home_page   = COND #( WHEN ls_inv_div-
website IS NOT INITIAL THEN ls_inv_div-website
                                                     ELSE cv_na )
                           cin           = COND #( WHEN lv_cin_no IS NOT IN
ITIAL THEN lv_cin_no ELSE cv_cin_upgrad )
                           pan           = COND #( WHEN lv_pan IS NOT INITI
AL THEN lv_pan ELSE cv_na )
                           logo          = lv_logo_bukrs
                           header_txt    = lv_header
                           head_f_1      = lv_head_f1
                           head_f_2      = lv_head_f2
                           sup_name      = COND #( WHEN lv_sup_name IS NOT
INITIAL THEN lv_sup_name ELSE cv_na )
                           saup_adr      = COND #( WHEN ls_compadr  IS NOT
INITIAL THEN ls_compadr  ELSE cv_na )
                           plant_adr     = COND #( WHEN ls_plantadr IS NOT
INITIAL THEN ls_plantadr ELSE cv_na )
                           branch        = COND #( WHEN lv_branch IS NOT IN
ITIAL THEN lv_branch ELSE cv_na )
                          order_no       = COND #( WHEN ls_ref-
order_numb IS NOT INITIAL THEN ls_ref-order_numb ELSE cv_na )
                          payment_terms  = COND #( WHEN lv_other_terms IS N
OT INITIAL THEN lv_other_terms  ELSE cv_na )
                          state_code_nm  = COND #( WHEN lv_statenmcd IS NOT
 INITIAL THEN lv_statenmcd ELSE cv_na )
*                          invoice_no     = COND #( WHEN  lv_xblnr IS NOT I
NITIAL THEN lv_xblnr ELSE cv_na ) --amritansu
                          invoice_no     = COND #( WHEN  ls_vbrk1-
xblnr IS NOT INITIAL THEN ls_vbrk1-xblnr ELSE cv_na ) "++amritansu
                          invoice_date   = gs_head-bil_date+6(2) && |-
| && gs_head-bil_date+4(2) && |-|  && gs_head-bil_date+0(4) "20241221

                          purch_ord_no   = COND #( WHEN gs_po-
purch_no_c IS NOT INITIAL THEN gs_po-purch_no_c  ELSE cv_na )
                          due_date       = ls_kopf-
payment_due_date+6(2) && |-| && ls_kopf-payment_due_date+4(2) && |-
| && ls_kopf-payment_due_date+0(4)
                          currency       =  COND #( WHEN gs_head-
bil_waerk IS NOT INITIAL THEN gs_head-bil_waerk  ELSE cv_na )
                           email         =   COND #( WHEN ls_inv_div-
email IS NOT INITIAL THEN ls_inv_div-email ELSE cv_na )
                           phone_no      =  COND #( WHEN wa_adtel-adtel-
tel_number IS NOT INITIAL THEN wa_adtel-adtel-tel_number ELSE cv_na )
                           invoice_no_sap  = COND #( WHEN gs_head-
bil_number IS NOT INITIAL THEN gs_head-bil_number ELSE cv_na )
```

```
                           place_of_supply = COND #( WHEN gv_sup_place IS N
OT INITIAL THEN gv_sup_place ELSE cv_na )
                           gstin           = COND #( WHEN ls_suppgst IS NOT
 INITIAL THEN ls_suppgst ELSE cv_na )
                           bill_to_no      = COND #( WHEN gv_bp IS NOT INIT
IAL THEN gv_bp ELSE cv_na )
                           bill_to_adr     = COND #( WHEN ls_bill IS NOT IN
ITIAL THEN ls_bill ELSE cv_na )
                           bill_cust_code  = COND #( WHEN gv_bp IS NOT INIT
IAL THEN gv_bp ELSE cv_na )
                           bill_pan       = COND #( WHEN ls_bill_pan-
j_1ipanno IS NOT INITIAL THEN ls_bill_pan-j_1ipanno ELSE cv_na )
                           bill_state_nm  = COND #( WHEN lv_bill_statenmcd
IS NOT INITIAL THEN lv_bill_statenmcd ELSE cv_na )
                           bill_gstn      = COND #( WHEN lv_bill_gstin IS N
OT INITIAL THEN lv_bill_gstin ELSE cv_na )
                           ship_to_no     = COND #( WHEN gv_sp IS NOT INITI
AL THEN gv_sp ELSE cv_na )
                           ship_to_adr    = COND #( WHEN ls_ship IS NOT INI
TIAL THEN ls_ship ELSE cv_na )
                           ship_cust_code =  COND #( WHEN gv_sp IS NOT INIT
IAL THEN gv_sp ELSE cv_na )
                           ship_pan       = COND #( WHEN ls_ship_pan-
j_1ipanno IS NOT INITIAL THEN ls_ship_pan-j_1ipanno ELSE cv_na )
                           ship_state_nm  = COND #( WHEN lv_ship_statenmcd
IS NOT INITIAL THEN lv_ship_statenmcd  ELSE cv_na )
                           ship_gstn  = COND #( WHEN lv_ship_gstin IS NOT I
NITIAL THEN lv_ship_gstin ELSE cv_na )
*                           TAX_INV_NO = COND #( WHEN LV_CM_REF IS NOT INIT
IAL THEN LV_CM_REF ELSE CV_NA )
                           tax_inv_no = lv_cm_ref
*                           tax_inv_dt = COND #( WHEN lv_refcre_dt IS NOT I
NITIAL THEN
*                                                     lv_refcre_dt+6(2) &&
|-| && lv_refcre_dt+4(2) && |-| && lv_refcre_dt+0(4)
*                                                 ELSE cv_na ) -amritansu

                          tax_inv_dt = COND #( WHEN ls_vbrk2-
fkdat IS NOT INITIAL THEN
                                                ls_vbrk2-fkdat+6(2) && |-
| && ls_vbrk2-fkdat+4(2) && |-| && ls_vbrk2-fkdat+0(4)
                                                 ELSE cv_na ) "++amritansu
                           purc_or_ref_cm = lv_po_or_ref_inv
                           cred_memo_flg  = lv_cred_memo_flg ).

*****************************Footer Part
**    IF lv_spart = |40| OR lv_spart = |41|.        "++PP13.05.2025 Stamp a
dded as per business
    DATA(gv_seal) = CONV thead-tdname( gs_org-
salesorg && '_LOGO_SD' ).  "Graphics
    DATA(lv_seal_name) = CONV char255( gv_seal ).
    me->derive_color_logo(
      EXPORTING
```

```
        iv_xlogo = lv_seal_name                " Char255
      IMPORTING
        ev_logo  = ev_seal
    ).
**    ENDIF.

***************Declaration logic
    SELECT SINGLE taxkd
      FROM knvi
      INTO @DATA(lv_taxkd)
      WHERE kunnr = @gs_hd_adr-partn_numb
        AND taxkd = '3'.
    IF sy-subrc = 0.
      DATA(gv_sez_flag) = 'X'.
    ELSE.
      gv_sez_flag = space.
    ENDIF.
    IF gv_sez_flag IS NOT INITIAL.
      DATA(lv_so10_decl) = CONV thead-tdname( |ZSD_SEZ_DECL| ).
    ENDIF.

*< SOA | PP08.07.2025 | Added to fetch sez number dynamic based on billing date >
    "Get declaration number from set table based on fkdat
    SELECT SINGLE * FROM setleaf
      INTO @DATA(ls_setleaf)
      WHERE setname = 'ZSD_SEZ_LUTNO'
      AND valfrom < @ls_vbrk-fkdat
      AND valto > @ls_vbrk-fkdat.
    IF sy-subrc EQ 0.
      SELECT SINGLE descript
        FROM setlinet INTO @DATA(lv_sez_num)
        WHERE setname = 'ZSD_SEZ_LUTNO'
        AND lineid EQ @ls_setleaf-lineid
        AND langu = 'E'.
    ENDIF.
*< EOA | PP08.07.2025 | Added to fetch sez number dynamic based on billing date >

    IF lv_so10_decl IS NOT INITIAL.
      CALL FUNCTION 'READ_TEXT'
        EXPORTING
          client                  = sy-mandt
          id                      = 'ST'
          language                = sy-langu
          name                    = lv_so10_decl
          object                  = 'TEXT'
        TABLES
          lines                   = lt_decla_lines
        EXCEPTIONS
          id                      = 1
          language                = 2
          name                    = 3
          not_found               = 4
          object                  = 5
          reference_check         = 6
          wrong_access_to_archive = 7
          OTHERS                  = 8.
      IF sy-subrc <> 0.ENDIF.
      IF  lt_decla_lines IS NOT INITIAL.
        LOOP AT lt_decla_lines INTO DATA(ls_lines).
          lv_declaration =  lv_declaration && ls_lines-tdline.
        ENDLOOP.
*        REPLACE '&SEZ_YEAR&' IN lv_declaration WITH  'AD270324178879C'. --
PP08.07.2025 Change for Sez text
        REPLACE '&SEZ_YEAR&' IN lv_declaration WITH  lv_sez_num. "++PP08.07
.2025 Change for Sez text
      ENDIF.
      ev_declaration = lv_declaration.
    ENDIF.

    IF is_bil_invoice-hd_gen-dlv_land <> |IN|.
      CLEAR ev_declaration.
      ev_declaration = TEXT-001.
    ENDIF.

**********Other terms and conditions based on sales org, distrib chanl and
division wrt user registered or not for tax
    "deriving so10 for other terms
    IF  lv_bukrs EQ |1810| AND ls_vbrk-taxk1 <> 1.  "If registered for TAX
      DATA(gv_terms) = CONV thead-
tdname( |TCREG_{ lv_vkorg }_{ lv_vtweg }_{ lv_spart }| ).
      CALL FUNCTION 'READ_TEXT'
        EXPORTING
          client                  = sy-mandt
          id                      = 'ST'
          language                = sy-langu
          name                    = gv_terms
          object                  = 'TEXT'
        TABLES
          lines                   = lt_terms_lines
        EXCEPTIONS
          id                      = 1
          language                = 2
          name                    = 3
          not_found               = 4
          object                  = 5
          reference_check         = 6
          wrong_access_to_archive = 7
          OTHERS                  = 8.
      IF sy-subrc <> 0.
        gv_terms = |TCREG_{ lv_vkorg }_{ lv_vtweg }_ALL| .
        CALL FUNCTION 'READ_TEXT'
          EXPORTING
            client                  = sy-mandt
            id                      = 'ST'
            language                = sy-langu
            name                    = gv_terms
            object                  = 'TEXT'
          TABLES
            lines                   = lt_terms_lines
          EXCEPTIONS
            id                      = 1
            language                = 2
            name                    = 3
            not_found               = 4
            object                  = 5
            reference_check         = 6
            wrong_access_to_archive = 7
            OTHERS                  = 8.
      ENDIF.

    ELSEIF lv_bukrs EQ |1810| AND ls_vbrk-
taxk1 EQ 1. "If not registered for TAX
      gv_terms = |TCUREG_{ lv_vkorg }_{ lv_vtweg }_{ lv_spart }|.
      CALL FUNCTION 'READ_TEXT'
        EXPORTING
          client                  = sy-mandt
          id                      = 'ST'
          language                = sy-langu
          name                    = gv_terms
          object                  = 'TEXT'
        TABLES
          lines                   = lt_terms_lines
        EXCEPTIONS
          id                      = 1
          language                = 2
          name                    = 3
          not_found               = 4
          object                  = 5
          reference_check         = 6
          wrong_access_to_archive = 7
          OTHERS                  = 8.
      IF sy-subrc <> 0.
        gv_terms = |TCUREG_{ lv_vkorg }_{ lv_vtweg }_ALL| .
        CALL FUNCTION 'READ_TEXT'
          EXPORTING
            client                  = sy-mandt
            id                      = 'ST'
            language                = sy-langu
            name                    = gv_terms
            object                  = 'TEXT'
          TABLES
            lines                   = lt_terms_lines
          EXCEPTIONS
            id                      = 1
            language                = 2
            name                    = 3
            not_found               = 4
            object                  = 5
            reference_check         = 6
            wrong_access_to_archive = 7
            OTHERS                  = 8.
      ENDIF.
    ENDIF.

    "Fill other terms
    IF  lt_terms_lines IS NOT INITIAL.
      LOOP AT lt_terms_lines INTO DATA(ls_terms).
        IF sy-tabix EQ 1.
          lv_terms = lv_terms  && ls_terms-tdline.
        ELSEIF ls_terms-tdformat = '*'.
          lv_terms = lv_terms && cl_abap_char_utilities=>newline && ls_term
s-tdline.
        ELSE.
          lv_terms = |{ lv_terms } { ls_terms-tdline }|.
        ENDIF.
      ENDLOOP.
    ENDIF.

    ev_terms = lv_terms.
    IF lt_terms_lines IS INITIAL.
      ev_terms = cv_na.
    ENDIF.

    "Bank details
    es_invoice_det-ben_name = cv_upgrad.  "upGrad Education Private Limited

    DATA(lv_accid) = CONV char5( ls_inv_div-acc_key ).
    SELECT SINGLE *
    FROM t012k
    INTO @DATA(ls_t012l)
    WHERE bukrs = @lv_vkorg
    AND   hbkid  = @ls_inv_div-house_bank
    AND   hktid  = @ls_inv_div-acc_key .
    IF sy-subrc EQ 0.
      es_invoice_det-bank_acc = ls_t012l-bankn.
    ENDIF.

    SELECT SINGLE *
      FROM bnka
      INTO @DATA(ls_bnka)
      WHERE banks = 'IN'
      AND   bankl = @ls_inv_div-bank_key.
    IF sy-subrc EQ 0.
      es_invoice_det-bank_name = ls_bnka-banka.
      es_invoice_det-branch_ar = ls_bnka-brnch.
      es_invoice_det-bank_addr = |{ ls_bnka-stras } { ls_bnka-ort01 }|.
      es_invoice_det-accc_type = |Current|.
      es_invoice_det-ifsc_cd = ls_inv_div-bank_key.
      es_invoice_det-swift_cd = ls_bnka-swift.
    ENDIF.
```

```
*    IF LS_VBRK-
TAXK1 EQ 1. "If customer is unregistred for TAX then no bank details
*      ES_INVOICE_DET-BEN_NAME = CV_NA.
*      ES_INVOICE_DET-BANK_NAME = CV_NA.
*      ES_INVOICE_DET-BRANCH_AR = CV_NA.
*      ES_INVOICE_DET-BANK_ADDR = CV_NA.
*      ES_INVOICE_DET-ACCC_TYPE = CV_NA.
*      ES_INVOICE_DET-BANK_ACC = CV_NA.
*      ES_INVOICE_DET-IFSC_CD = CV_NA.
*      ES_INVOICE_DET-SWIFT_CD = CV_NA.
*    ENDIF.

    DATA(gv_bank1) = CONV thead-tdname( gs_head-bil_number ).
    DATA(gv_bank) = CONV thead-tdname( 'ZSD_INV_BANK_' && gs_org-
salesorg ).
    IF gv_bank1 IS NOT INITIAL.

      CALL FUNCTION 'READ_TEXT'
        EXPORTING
          client                  = sy-mandt
          id                      = 'ZBNK'
          language                = sy-langu
          name                    = gv_bank1
          object                  = 'VBBK'
        TABLES
          lines                   = lt_bank
        EXCEPTIONS
          id                      = 1
          language                = 2
          name                    = 3
          not_found               = 4
          object                  = 5
          reference_check         = 6
          wrong_access_to_archive = 7
          OTHERS                  = 8.
      IF sy-subrc <> 0.ENDIF.
    ENDIF.

    IF lt_bank IS NOT INITIAL.
      DATA(gv_flag_text) = 'X'.
    ELSE.
      gv_flag_text = ' '.
    ENDIF.

    IF gv_flag_text IS INITIAL.
      CALL FUNCTION 'READ_TEXT'
        EXPORTING
          client                  = sy-mandt
          id                      = 'ST'
          language                = sy-langu
          name                    = gv_bank
          object                  = 'TEXT'
        TABLES
          lines                   = lt_bank
        EXCEPTIONS
          id                      = 1
          language                = 2
          name                    = 3
          not_found               = 4
          object                  = 5
          reference_check         = 6
          wrong_access_to_archive = 7
          OTHERS                  = 8.
      IF sy-subrc <> 0.ENDIF.
    ENDIF.

    IF lt_bank IS NOT INITIAL.
      LOOP AT lt_bank INTO DATA(ls_bank).
        lv_bank = |Bank Details:{ cl_abap_char_utilities=>newline }|.
        IF ls_terms-tdformat = '*'.
          lv_bank = lv_bank && ls_bank-
tdline && cl_abap_char_utilities=>newline.
        ELSE.
          lv_bank = lv_bank && ls_bank-tdline.
        ENDIF.
      ENDLOOP.
      ev_bank =  lv_bank.
    ENDIF.

  ENDMETHOD.
```

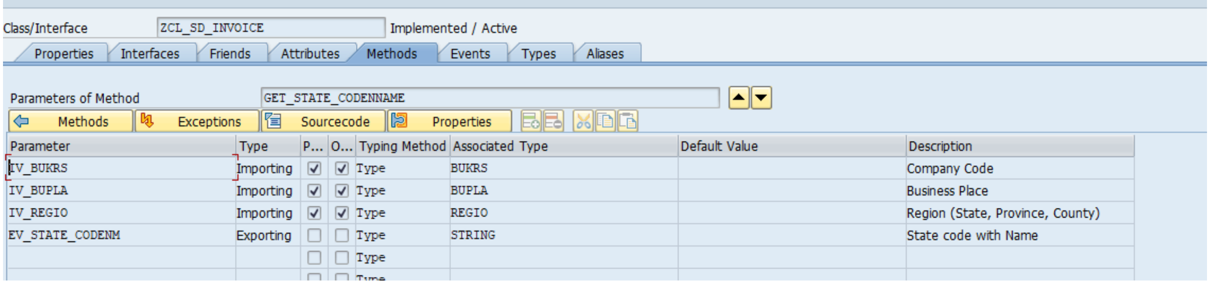


Class/Interface ZCL_SD_INVOICE Implemented / Active

Parameters of Method GET_QRCODE

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
IV_BIL_INVOICE	Importing	☑	☑	Type	VBELN		Sale order number
EV_QRBITMAP	Exporting	☑	□	Type	XSTRING		QR Code bit map value
		□	□	Type			

```
  METHOD GET_QRCODE.
*-----------------------------------------------------------------------*
*    Program    :  Updates Tables (   )     Process data (  )
*               Outputs List    (   )     Validation    (  )
*
*    Module                    : SD
*    WRICEF Object Id          : SD_DV020
*    Functional Specifications : FS_SD_DV020_SALES_INVOCIE
*    Technical Spec No         : TS_SD_DV020_SALES_INVOCIE
*    Functional Consultant     : Parul Singh
*    Date Created (dd-mm-yyyy) : 01-12-2024
*    Technical Consultant      : Pravesh Patil
*    Client                    : upGrad
*    Project Name              : upGrad
*    Company                   : TechWit Business Solutions
*----------------------------------------------------------------------*
    CONSTANTS: LC_A TYPE C VALUE 'A',
               LC_H TYPE C VALUE 'H'.
    DATA(LV_SALES_ORDER)   = IV_BIL_INVOICE.

***************QR Code
    SELECT SINGLE XBLNR
      FROM VBRK
      INTO @DATA(LV_XBLNR)
      WHERE VBELN = @LV_SALES_ORDER.
    IF SY-SUBRC EQ 0.
      SELECT SINGLE
            QRCODE ,
            INV_REG_NUM ,
            ACK_NO ,
            ACK_DATE
            FROM EDOINEINV
            INTO  @DATA(LS_IRNDETAILS)
            WHERE INV_NUM = @LV_XBLNR.
      IF SY-SUBRC EQ 0.

        IF LS_IRNDETAILS-QRCODE IS NOT INITIAL.
          CL_RSTX_BARCODE_RENDERER=>QR_CODE(
            EXPORTING
              I_MODULE_SIZE       = 20
              I_MODE              = LC_A
              I_ERROR_CORRECTION  = LC_H
              I_BARCODE_TEXT      = LS_IRNDETAILS-QRCODE
            IMPORTING
              E_BITMAP            = DATA(LV_BITQRCODE)
                 ) .

          EV_QRBITMAP = LV_BITQRCODE.
        ENDIF.
      ENDIF.
    ENDIF.

  ENDMETHOD.
```

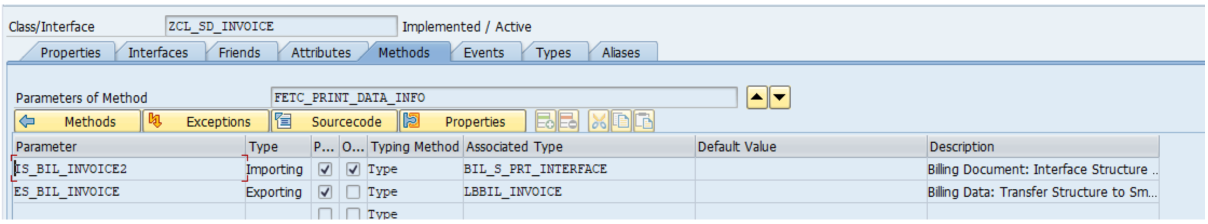


Class/Interface ZCL_SD_INVOICE Implemented / Active

Parameters of Method GET_STATE_CODENAME

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
IV_BUKRS	Importing	☑	☑	Type	BUKRS		Company Code
IV_BUPLA	Importing	☑	☑	Type	BUPLA		Business Place
IV_REGIO	Importing	☑	☑	Type	REGIO		Region (State, Province, County)
EV_STATE_CODENM	Exporting	□	□	Type	STRING		State code with Name
		□	□	Type			
		□	□	Type			

```
  METHOD GET_STATE_CODENNAME.
*----------------------------------------------------------------------*
*     Program    :  Updates Tables (   )     Process data (  )
*                Outputs List    (   )     Validation    (  )
*
*     Module                        : SD
*     WRICEF Object Id              : SD_DV020
*     Functional Specifications : FS_SD_DV020_SALES_INVOCIE
*     Technical Spec No             : TS_SD_DV020_SALES_INVOCIE
*     Functional Consultant       : Parul Singh
*     Date Created (dd-mm-yyyy) : 01-12-2024
*     Technical Consultant        : Pravesh Patil
*     Client                      : upGrad
*     Project Name                : upGrad
*     Company                     : TechWit Business Solutions
*----------------------------------------------------------------------*
    DATA : LS_ADDR_BUPLA TYPE ADDR1_VAL.
    IF IV_REGIO IS NOT INITIAL.
      DATA(LV_REG) = IV_REGIO.
    ENDIF.

    IF IV_BUKRS IS NOT INITIAL AND IV_BUPLA IS NOT INITIAL.
      CALL FUNCTION 'GET_BUPLA_ADDRESS'
        EXPORTING
          BUKRS   = IV_BUKRS
          BUPLA   = IV_BUPLA
        IMPORTING
          ADRESSE = LS_ADDR_BUPLA.
      LV_REG = LS_ADDR_BUPLA-REGION.
    ENDIF.


    IF LV_REG IS NOT INITIAL.
      SELECT SINGLE * FROM J_1ISTATECDM
       INTO @DATA(LS_STATECDM)
       WHERE LAND1         EQ 'IN'
       AND   STD_STATE_CODE = @LV_REG.
      IF SY-SUBRC EQ 0.
        EV_STATE_CODENM = LS_STATECDM-
LEGAL_STATE_CODE && | - | && LS_STATECDM-BEZEI.
      ENDIF.
    ENDIF.
  ENDMETHOD.
```



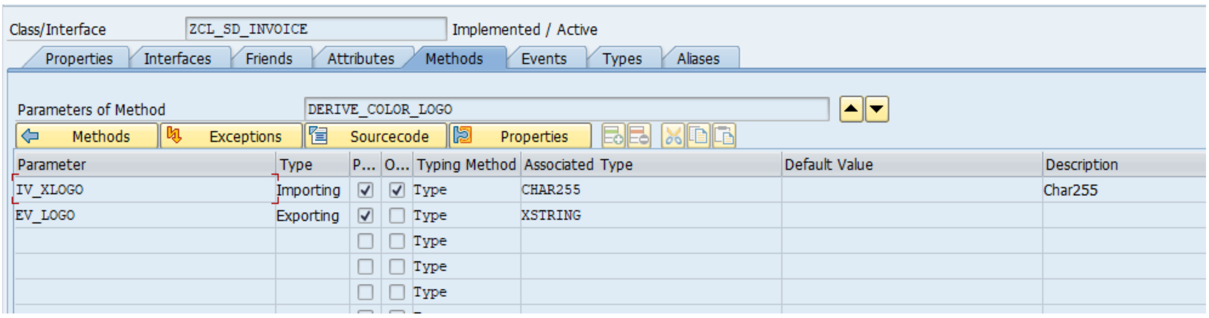
Class/Interface ZCL_SD_INVOICE Implemented / Active

Parameters of Method FETC_PRINT_DATA_INFO

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
IS_BIL_INVOICE2	Importing	☑	☑	Type	BIL_S_PRT_INTERFACE		Billing Document: Interface Structure ..
ES_BIL_INVOICE	Exporting	☑	□	Type	LBBIL_INVOICE		Billing Data: Transfer Structure to Sm...
		□	□	Type			

```
  METHOD FETC_PRINT_DATA_INFO.
*----------------------------------------------------------------------*
*     Program    :  Updates Tables (    )     Process data (  )
*                   Outputs List    (    )    Validation    (  )
*
*     Module                      : SD
*     WRICEF Object Id            : SD_DV020
*     Functional Specifications : FS_SD_DV020_SALES_INVOCIE
*     Technical Spec No           : TS_SD_DV020_SALES_INVOCIE
*     Functional Consultant       : Parul Singh
*     Date Created (dd-mm-yyyy) : 01-12-2024
*     Technical Consultant        : Pravesh Patil
*    Client                        : upGrad
*    Project Name                  : upGrad
*    Company                       : TechWit Business Solutions
*----------------------------------------------------------------------*
    DATA: LS_PRINT_DATA_TO_READ TYPE LBBIL_PRINT_DATA_TO_READ,
          LS_BIL_INVOICE        TYPE LBBIL_INVOICE.

    DATA(LV_VBELN) = CONV NA_OBJKEY( IS_BIL_INVOICE2-HEAD_DETAIL-VBDKR-
VBELN ).
    DATA(LV_SPRAS) = CONV NA_SPRAS( IS_BIL_INVOICE2-HEAD_DETAIL-VBDKR-
SPRAS_VKO ).
    DATA(LV_KUNWE) = CONV CHAR10( IS_BIL_INVOICE2-HEAD_DETAIL-VBDKR-
KUNWE ).

    "Fill all flags to get All Print data
    LS_PRINT_DATA_TO_READ = VALUE #( HD_GEN  = ABAP_TRUE
                                         HD_ADR  = ABAP_TRUE
                                         HD_GEN_DESCRIPT  = ABAP_TRUE
                                         HD_ORG  = ABAP_TRUE
                                         HD_PART_ADD  = ABAP_TRUE
                                         HD_KOND  = ABAP_TRUE
                                         HD_FIN  = ABAP_TRUE
                                         HD_REF  = ABAP_TRUE
                                         HD_TECH  = ABAP_TRUE
                                         IT_GEN  = ABAP_TRUE
                                         IT_ADR  = ABAP_TRUE
                                         IT_PRICE  = ABAP_TRUE
                                         IT_KOND  = ABAP_TRUE
                                         IT_REF  = ABAP_TRUE
                                         IT_REFDLV  = ABAP_TRUE
                                         IT_REFORD  = ABAP_TRUE
                                         IT_REFPURORD  = ABAP_TRUE
                                         IT_REFVAG  = ABAP_TRUE
                                         IT_REFVG2  = ABAP_TRUE
                                         IT_REFVKT  = ABAP_TRUE
                                         IT_TECH  = ABAP_TRUE
                                         IT_FIN  = ABAP_TRUE
                                         IT_CONFITM  = ABAP_TRUE
                                         IT_CONFBATCH  = ABAP_TRUE
                                         MSR_HD  = ABAP_TRUE
                                         MSR_IT  = ABAP_TRUE ).

*Read print data from FM
    CALL FUNCTION 'LB_BIL_INV_OUTP_READ_PRTDATA'
      EXPORTING
        IF_BIL_NUMBER         = LV_VBELN
        IF_PARVW              = GC_PART_FUNC_RE
        IF_PARNR              = LV_KUNWE
        IF_LANGUAGE           = LV_SPRAS
        IS_PRINT_DATA_TO_READ = LS_PRINT_DATA_TO_READ
      IMPORTING
        ES_BIL_INVOICE        = LS_BIL_INVOICE
      EXCEPTIONS
```

```
        RECORDS_NOT_FOUND      = 1
        RECORDS_NOT_REQUESTED  = 2
        OTHERS                 = 3.
    IF SY-SUBRC <> 0.
    ENDIF.

    ES_BIL_INVOICE = LS_BIL_INVOICE.
  ENDMETHOD.
```

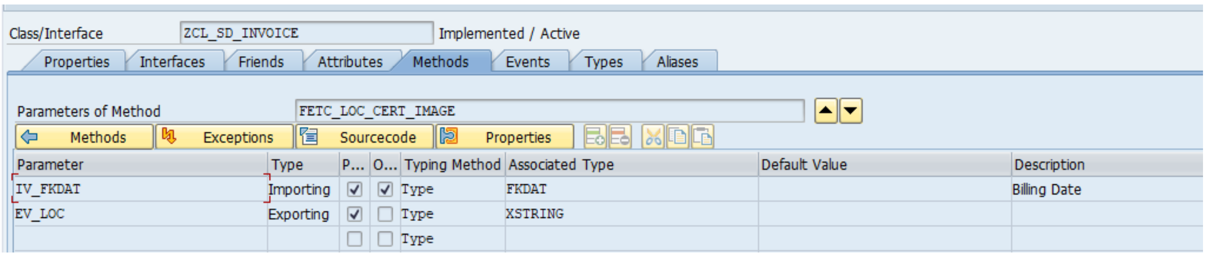


Class/Interface ZCL_SD_INVOICE Implemented / Active

Parameters of Method DERIVE_COLOR_LOGO

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
IV_XLOGO	Importing	☑	☑	Type	CHAR255		Char255
EV_LOGO	Exporting	☑	☐	Type	XSTRING		
		☐	☐	Type			
		☐	☐	Type			
		☐	☐	Type			

```
  METHOD DERIVE_COLOR_LOGO.
*----------------------------------------------------------------------*
*    Program   :  Updates Tables (   )    Process data (  )
*                Outputs List   (   )    Validation   (  )
*
*    Module                    : SD
*    WRICEF Object Id          : SD_DV020
*    Functional Specifications : FS_SD_DV020_SALES_INVOCIE
*    Technical Spec No         : TS_SD_DV020_SALES_INVOCIE
*    Functional Consultant     : Parul Singh
*    Date Created (dd-mm-yyyy) : 01-12-2024
*    Technical Consultant      : Pravesh Patil
*    Client                    : upGrad
*    Project Name              : upGrad
*    Company                   : TechWit Business Solutions
*---------------------------------------------------------------------*
    "Get upGrad Logo in Bitmap
    DATA(LV_LOGO) = CONV TDOBNAME( IV_XLOGO ).
    CALL METHOD CL_SSF_XSF_UTILITIES=>GET_BDS_GRAPHIC_AS_BMP
      EXPORTING
        P_OBJECT       = 'GRAPHICS'
        P_NAME         = LV_LOGO " Name of the logo as in se78
        P_ID           = 'BMAP'
        P_BTYPE        = 'BCOL' "( whether the image is in color or black a
nd white )
      RECEIVING
        P_BMP          = DATA(LV_FIELD)
      EXCEPTIONS
        NOT_FOUND      = 1
        INTERNAL_ERROR = 2
        OTHERS         = 3.

    EV_LOGO = LV_FIELD.
  ENDMETHOD.
```

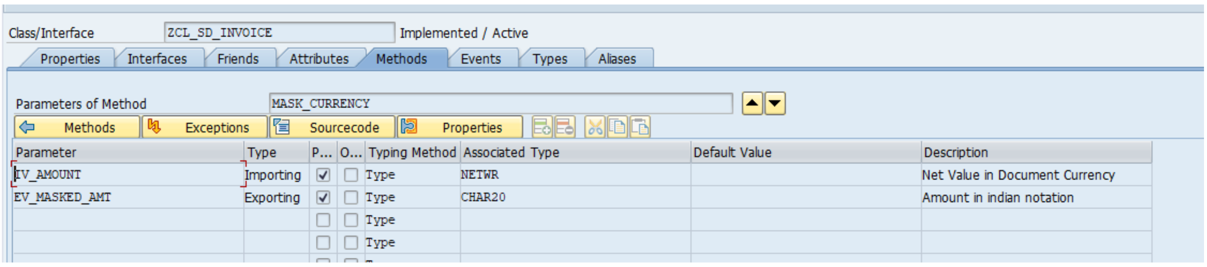


Class/Interface ZCL_SD_INVOICE Implemented / Active

Parameters of Method FETC_LOC_CERT_IMAGE

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
IV_FKDAT	Importing	☑	☑	Type	FKDAT		Billing Date
EV_LOC	Exporting	☑	□	Type	XSTRING		
		□	□	Type			

```
  METHOD fetc_loc_cert_image.
*----------------------------------------------------------------------*
*    Program    :  Updates Tables (   )     Process data (  )
*                Outputs List   (   )     Validation   (  )
*
*    Module                      : SD
*    WRICEF Object Id            : SD_DV020
*    Functional Specifications : FS_SD_DV020_SALES_INVOCIE
*    Technical Spec No           : TS_SD_DV020_SALES_INVOCIE
*    Functional Consultant       : Parul Singh
*    Date Created (dd-mm-yyyy) : 01-12-2024
*    Technical Consultant        : Pravesh Patil
*    Client                      : upGrad
*    Project Name                : upGrad
*    Company                     : TechWit Business Solutions
*---------------------------------------------------------------------*
    CALL METHOD cl_mime_repository_api=>if_mr_api~get_api
      RECEIVING
        r_mr_api = DATA(lr_mr_api). " MIME REPOSITORY API

    IF iv_fkdat  <= |20250331|.
      DATA(lv_url) = 'SAP/PUBLIC/LOC_CERTIFICATE/LDC_2024-25_LAND.jpeg'.
    ELSE.
      lv_url = 'SAP/PUBLIC/LOC_CERTIFICATE/LDC_FY_2025-26.jpg'.
    ENDIF..

    CALL METHOD lr_mr_api->get
      EXPORTING
*       I_URL              = 'SAP/PUBLIC/LOC_CERTIFICATE/LDC_2024-
25.jpeg'  "Object url
*       I_URL              = 'SAP/PUBLIC/LOC_CERTIFICATE/LDC_2024-
25_LAND.jpeg'  "Object url
        i_url              = lv_url
      IMPORTING
        e_content          = DATA(logo)        "OBJECT CONTENTS
        e_mime_type        = DATA(mime_type)   "MIME TYPE
      EXCEPTIONS
        parameter_missing  = 1
        error_occured      = 2
        not_found          = 3
        permission_failure = 4
        OTHERS             = 5.
    ev_loc = logo.
  ENDMETHOD.
```



Class/Interface ZCL_SD_INVOICE Implemented / Active

Parameters of Method MASK_CURRENCY

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
IV_AMOUNT	Importing	☑	□	Type	NETWR		Net Value in Document Currency
EV_MASKED_AMT	Exporting	☑	□	Type	CHAR20		Amount in indian notation
		□	□	Type			
		□	□	Type			

```
  METHOD MASK_CURRENCY.
*-----------------------------------------------------------------------*
*     Program    :  Updates Tables (    )     Process data (  )
*                  Outputs List    (    )    Validation    (  )
*
*     Module                        : SD
*     WRICEF Object Id              : SD_DV020
*     Functional Specifications : FS_SD_DV020_SALES_INVOCIE
*     Technical Spec No             : TS_SD_DV020_SALES_INVOCIE
*     Functional Consultant         : Parul Singh
*     Date Created (dd-mm-yyyy) : 01-12-2024
*     Technical Consultant          : Pravesh Patil
*     Client                        : upGrad
*     Project Name                  : upGrad
*     Company                       : TechWit Business Solutions
*----------------------------------------------------------------------*
    DATA : LV_AMOUNT TYPE CHAR20,
           LV_MASK   TYPE CHAR25.

    LV_AMOUNT = IV_AMOUNT.
    SPLIT LV_AMOUNT AT '.' INTO DATA(LV_NUMBER) DATA(LV_DEC) .

    CONDENSE:LV_DEC, LV_NUMBER, LV_AMOUNT.
    DATA(LV_LEN) = STRLEN( LV_NUMBER ).

    LV_MASK = SWITCH #( LV_LEN
     WHEN 12
     THEN | { LV_AMOUNT+0(1) },{ LV_AMOUNT+1(2) },{ LV_AMOUNT+3(2) },{ LV_A
MOUNT+5(2) },{ LV_AMOUNT+7(2) },{ LV_AMOUNT+9(3) }.{ LV_AMOUNT+13(2) } |
     WHEN 11
     THEN  | { LV_AMOUNT+0(2) },{ LV_AMOUNT+2(2) },{ LV_AMOUNT+4(2) },{ LV_
AMOUNT+6(2) },{ LV_AMOUNT+8(3) }.{ LV_AMOUNT+12(2) } |
     WHEN 10
     THEN  | { LV_AMOUNT+0(1) },{ LV_AMOUNT+1(2) },{ LV_AMOUNT+3(2) },{ LV_
AMOUNT+5(2) },{ LV_AMOUNT+7(3) }.{ LV_AMOUNT+11(2) } |
     WHEN 9
     THEN  | { LV_AMOUNT+0(2) },{ LV_AMOUNT+2(2) },{ LV_AMOUNT+4(2) },{ LV_
AMOUNT+6(3) }.{ LV_AMOUNT+10(2) } |
     WHEN 8
     THEN  | { LV_AMOUNT+0(1) },{ LV_AMOUNT+1(2) },{ LV_AMOUNT+3(2) },{ LV_
AMOUNT+5(3) }.{ LV_AMOUNT+9(2) } |
     WHEN 7
     THEN  | { LV_AMOUNT+0(2) },{ LV_AMOUNT+2(2) },{ LV_AMOUNT+4(3) }.{ LV_
AMOUNT+8(2) } |
     WHEN 6
     THEN  | { LV_AMOUNT+0(1) },{ LV_AMOUNT+1(2) },{ LV_AMOUNT+3(3) }.{ LV_
AMOUNT+7(2) } |
     WHEN 5
     THEN  | { LV_AMOUNT+0(2) },{ LV_AMOUNT+2(3) }.{ LV_AMOUNT+6(2) } |
     WHEN 4
     THEN  | { LV_AMOUNT+0(1) },{ LV_AMOUNT+1(3) }.{ LV_AMOUNT+5(2) } |
     WHEN 3
     THEN  | { LV_AMOUNT+0(3) }.{ LV_AMOUNT+4(2) } |
     WHEN 2
     THEN  | { LV_AMOUNT+0(2) }.{ LV_AMOUNT+3(2) } |
     WHEN 1
     THEN  | { LV_AMOUNT+0(1) }.{ LV_AMOUNT+2(2) } |
     WHEN 0
     THEN  | 0.{ LV_AMOUNT+0(2) } |

    ).

    CONDENSE LV_MASK.
    EV_MASKED_AMT = LV_MASK.
  ENDMETHOD.
```