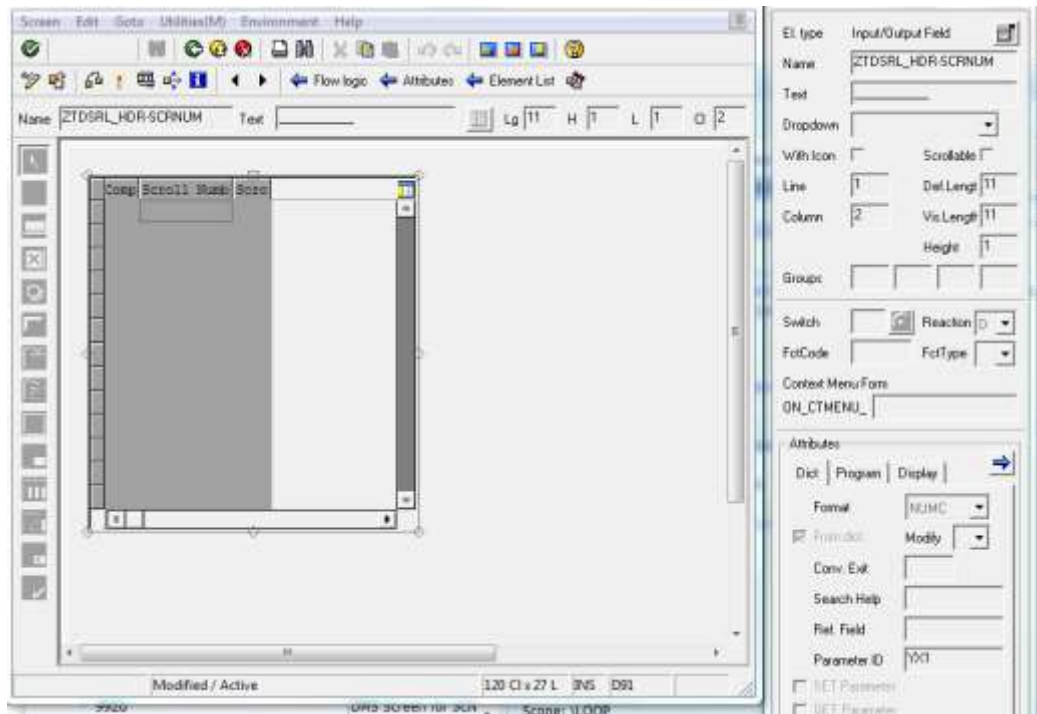


STEP BY STEP DMS CREATION.

Step 1: Add the field which has to be act as an input in the structure **MCDOKOB**.

Step 2: create a screen in function group CV100 for TCode CV01N and CV130 for TCODE CV02N. The screen must be table control and the field must be referring to the **table** where the value is getting stored. Here ZTDSRL_HDR is the table name having SCRNUM as the field. The created screen will be present in OBJECT LINK Tab in CV01N. **The below screen is copied from screen number 1201.** The screen must be a **SUBSCREEN**.



Step 3: In the flow logic of the screen where layout is created copy the below codings.



Step 4: Copy the below codings in the function module. The FM name must start with OBJECT_CHECK_<Table_name>. change the ZTDSRL_HDR to your <Table_name>.

```

*-----
*""Local Interface:
*  IMPORTING
*    VALUE(CALLED_FROM) DEFAULT 1
*    VALUE(LANGUAGE) LIKE SY-LANGU DEFAULT SY-LANGU
*    VALUE(LOCK) DEFAULT SPACE
*    VALUE(SINGLE) DEFAULT SPACE
*    VALUE(I_AUTH_CHK) TYPE C DEFAULT SPACE
*  EXPORTING
*    VALUE(FAULT_TEXT)
*    VALUE(MESSAGE_TYPE)
*    VALUE(NO_AUTH)
*  TABLES
*    OTAB STRUCTURE RMCLOBTX
*  EXCEPTIONS
*    FOREIGN_LOCK
*    OBJECT_NOT_FOUND
*    SYATEM_FAILURE
*-----

*****
*----- A D D E D B Y R I L -----*
TABLES : ztdsrl_hdr.
DATA   : lw_char2 TYPE c VALUE '2'.
DATA   : BEGIN OF lt_objecttab OCCURS 1.
          INCLUDE STRUCTURE cltable.
DATA   : END   OF lt_objecttab.
*----- E N D   O F   C H A N G E -----*

DATA: BEGIN OF lt_tdskey OCCURS 0,
      bukr   TYPE bukr,
      scrnum TYPE zscrnum,
      syyear TYPE zyear.
DATA: END   OF lt_tdskey.
*
CLEAR no_auth.
CLEAR message_type.
CLEAR fault_text.
IF NOT single IS INITIAL.
  READ TABLE otab INDEX 1.
  lt_tdskey = otab-objek.

  SELECT SINGLE * FROM ztdsrl_hdr
    WHERE bukr EQ lt_tdskey-bukr AND
          scrnum EQ lt_tdskey-scrnum AND
          syyear EQ lt_tdskey-syyear.

  IF sy-subrc EQ 0.
    "Auftrag vorhanden

```

```

otab-obtxt = 'TDS Scroll'(001).           "Text vorhanden
MODIFY otab INDEX 1.
IF called_from = lw_char2.               "nur für Klassifizierung
    lt_objecttab-tname = 'ZTDSRL_HDR'.
* RILECC6 Specific changes - Start
*    objecttab-table = vbrk.
    CALL METHOD cl_abap_container_utilities=>fill_container_c
        EXPORTING
            im_value                = ztdsrl_hdr
        IMPORTING
            ex_container            = lt_objecttab-table
        EXCEPTIONS
            illegal_parameter_type = 1
            OTHERS                  = 2.
* RILECC6 Specific changes - End
    APPEND lt_objecttab.
    CALL FUNCTION 'CTMS_OBJECT_AREA'
        TABLES
            objects = lt_objecttab.
    ENDIF.
ELSE.                                     "nicht vorhanden
    otab-
obtxt = 'Scroll not valid for company code'(002).           "Text vorhanden
    RAISE object_not_found.           "Ausnahme auslösen
ENDIF.
ELSE.                                     "Mengenselekt
    REFRESH lt_tdskey.
    LOOP AT otab.
        lt_tdskey = otab-objek.
        APPEND lt_tdskey.
    ENDLOOP.
    CHECK syst-subrc = 0.
    SORT otab BY objek.
    SELECT * FROM ztdsrl_hdr FOR ALL ENTRIES IN lt_tdskey "#EC CI_SUBRC
        WHERE bukrs EQ lt_tdskey-bukrs AND
              scrnum EQ lt_tdskey-scrnum AND
              syyear EQ lt_tdskey-syyear.
    IF sy-subrc = 0.
        lt_tdskey-bukrs = ztdsrl_hdr-bukrs.
        lt_tdskey-scrnum = ztdsrl_hdr-scrnum.
        lt_tdskey-syyear = ztdsrl_hdr-syyear.
        READ TABLE otab WITH KEY lt_tdskey BINARY SEARCH. "#EC CI_SUBRC
        IF sy-subrc = 0.
            otab-obtxt = 'Scroll not valid for company code'(002).
            MODIFY otab INDEX syst-tabix.
        ENDIF.
    ENDIF.
ENDSELECT.
ENDIF.

```

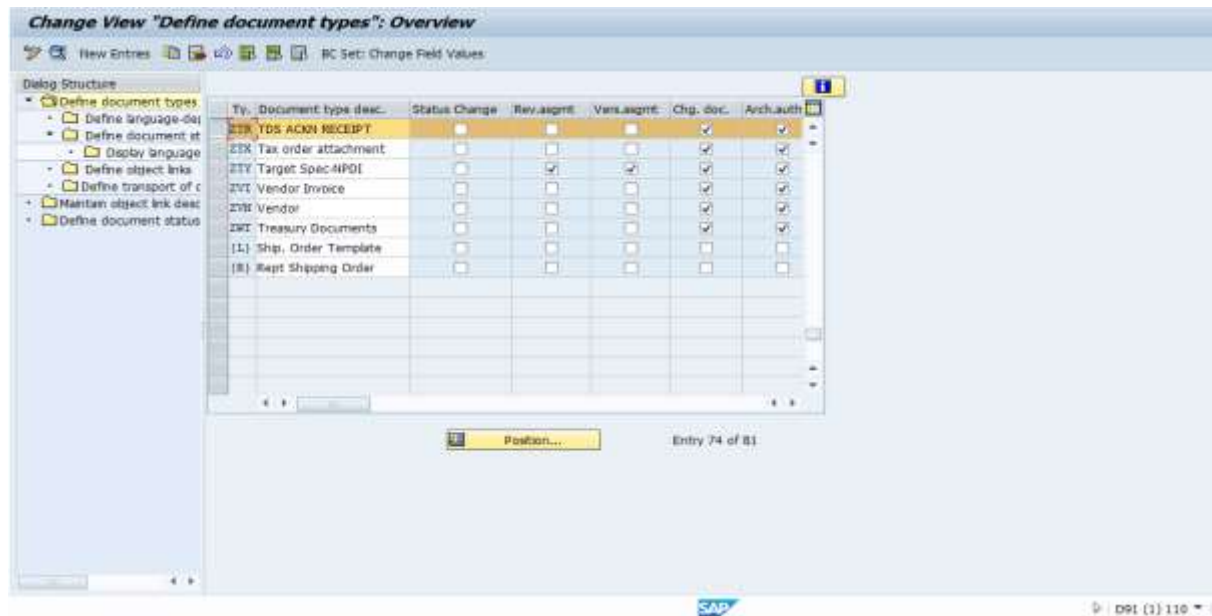
BELOW IS THE CONFIGURATION PART DONE BY FUNCTIONAL:

DMS Configuration and new object link addition in CV01.

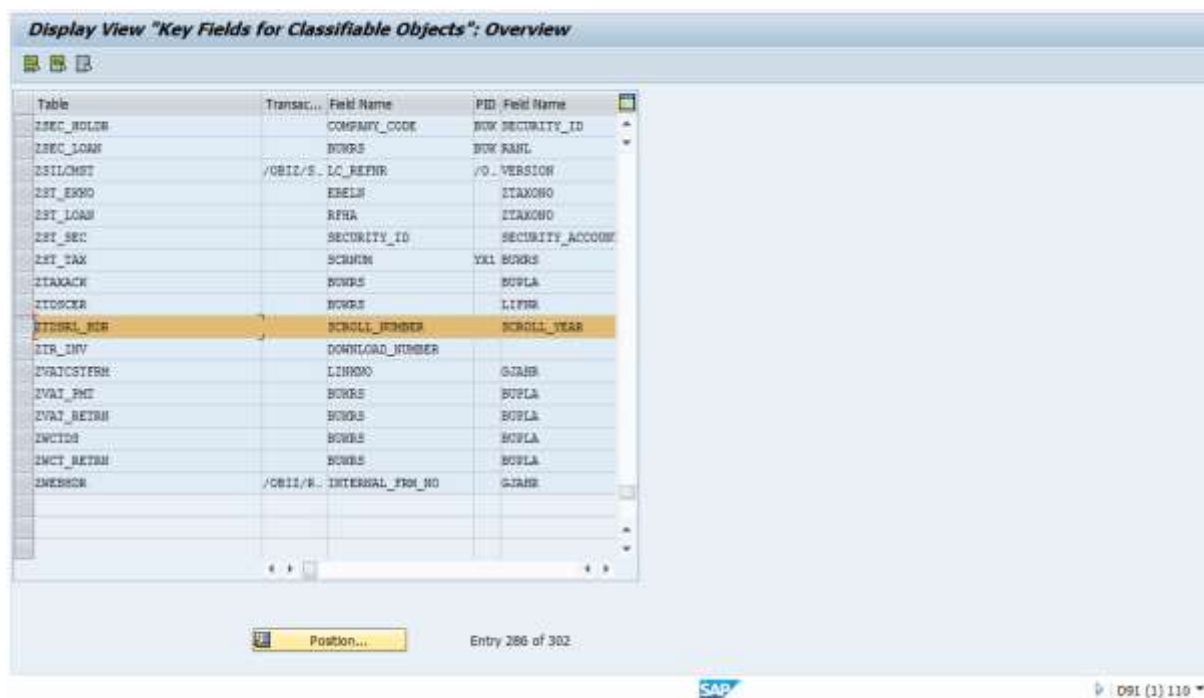
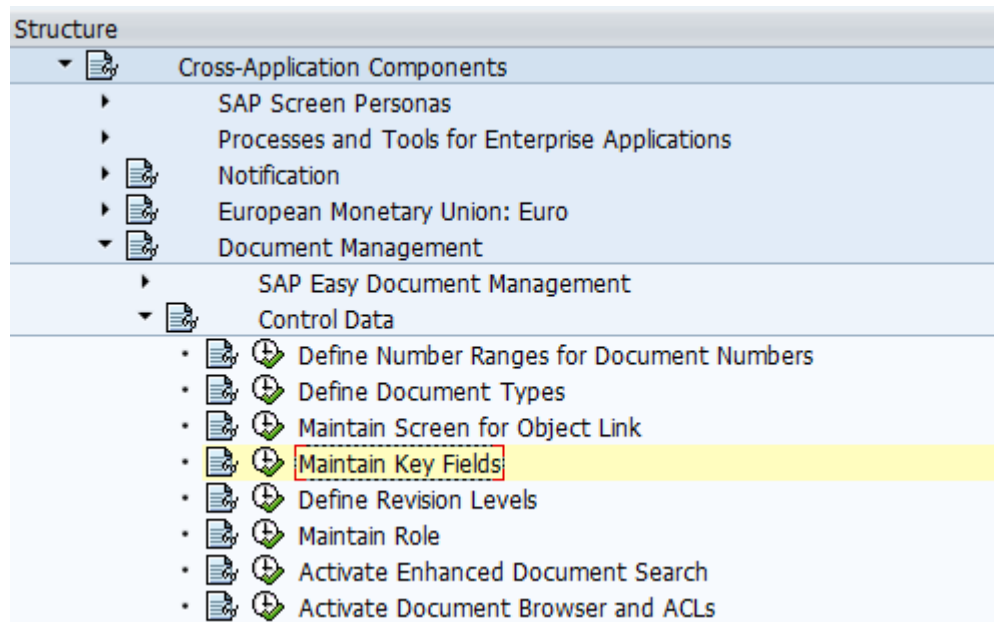
Step 5: Define Document types



Step 6: Taken Already used document type for TDS.



Step 8: Maintain Key Fields (By **ABAP- Workbench Request-** Table and required field to be configured here)



Check Function Group-CV100 and CV130 to get which screen number is not used, take unused screen number.

Step 9: Maintain Screen Object link

Give screen number with the object link.


```

        lw_dms_return          TYPE  bapiret2.

DATA: lv_obj_des              TYPE  char100,
      lv_date                 TYPE  datum,
      lv_count                TYPE  i,
      lv_message              TYPE  char256,
      lv_success              TYPE  char1.

CLEAR lv_success.
DO 30 TIMES.
    SELECT COUNT(*)
      FROM ztdsrl_hdr
     WHERE bukrs EQ p_bukrs AND
           scrnum EQ gv_srno AND
           syyear EQ sy-datum+0(4).
    IF sy-subrc IS INITIAL.
        lv_success = abap_true.
        EXIT.
    ELSE.
        WAIT UP TO 1 SECONDS.
        CONTINUE.
    ENDIF.
ENDDO.

IF lv_success EQ abap_true.
    CONCATENATE p_bukrs
                gv_srno
                sy-datum+0(4)
                INTO
                lv_obj_des.

    lw_documentdata-documenttype = lc_doctyp.
    lw_documentdata-statusintern = lc_staint.

    lw_doc_descrp-language = lc_langu.
    lw_doc_descrp-description = lv_obj_des.
    APPEND lw_doc_descrp TO lt_doc_descrp.
    CLEAR lw_doc_descrp.

    lw_objectlinks-objecttype = lc_objtyp.
    lw_objectlinks-objectkey  = lv_obj_des.
    lw_objectlinks-documentdirection = lc_docdir.
    APPEND lw_objectlinks TO lt_objectlinks.
    CLEAR lw_objectlinks.

    lw_documentfiles-documenttype = lc_doctyp .
    lw_documentfiles-storagecategory = lc_storage.
    lw_documentfiles-wsapplication  = lc_appl.
    lw_documentfiles-docpath        = p_dfile.
    APPEND lw_documentfiles TO lt_documentfiles.
    CLEAR lw_documentfiles.

    CALL FUNCTION 'BAPI_DOCUMENT_CREATE2'

```



```

EXPORTING
    documentdata      = lw_documentdata
    defaultclass      = lc_default
IMPORTING
    documenttype      = lw_doc_type
    documentnumber    = p_gv_dms
    documentpart      = lw_doc_part
    documentversion   = lw_doc_version
    return            = lw_dms_return
TABLES
    documentdescriptions = lt_doc_descr
    objectlinks          = lt_objectlinks
    documentfiles        = lt_documentfiles.

IF lw_dms_return-type EQ sy-abcde+4(1).

    CLEAR:lv_message.
    CALL FUNCTION 'FORMAT_MESSAGE'
    EXPORTING
        id      = lw_dms_return-id
        lang    = sy-langu
        no      = lw_dms_return-number
        v1      = lw_dms_return-message_v1
        v2      = lw_dms_return-message_v2
        v3      = lw_dms_return-message_v3
        v4      = lw_dms_return-message_v4
    IMPORTING
        msg     = lv_message
    EXCEPTIONS
        not_found = 1
        OTHERS   = 2.
    MESSAGE e001(00) WITH 'DMS error-'(040) lv_message.

ENDIF.
IF p_gv_dms IS NOT INITIAL.
    CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'.
ELSE.
    MESSAGE 'DMS failed'(042) TYPE 'E'.
ENDIF.

```