\*&---------------------------------------------------------------------\*  
\*& Report  ZDATASET\_READ\_DATA\_10  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zdataset\_read\_data\_10.  
" Read data from table MARC and upload to the application server  
"Data: Plant data for Material: Material, Plant and Purchasing Group  
"Declaration  
"Data Declaration  
TYPES: BEGIN OF t\_marc,  
  matnr TYPE marc-matnr,  
  werks TYPE marc-werks,  
  ekgrp TYPE marc-ekgrp,  
  END OF t\_marc,  
  t\_marc\_tab TYPE STANDARD TABLE OF t\_marc.  
  
DATA:  
      wa\_marc TYPE t\_marc,  
      it\_marc TYPE t\_marc\_tab.  
  
SELECT-OPTIONS s\_matnr FOR wa\_marc-matnr.  
PARAMETERS: p\_path TYPE dxfields-longpath.  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_path.  
  PERFORM get\_file.  
  
START-OF-SELECTION.  
  PERFORM get\_data.  
  PERFORM open\_dataset.  
  PERFORM transfer\_data.  
  PERFORM close\_dataset.  
  
\*END-OF-SELECTION.  
\*  PERFORM display\_data.  
\*&---------------------------------------------------------------------\*  
\*&      Form  GET\_FILE  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM get\_file .  
  CALL FUNCTION 'F4\_DXFILENAME\_TOPRECURSION'  
   EXPORTING  
     i\_location\_flag       = 'A'  
\*     I\_SERVER              = '?'  
\*     i\_path                = ''  
     filemask              = '\*.\*'  
     fileoperation         = 'R'  
   IMPORTING  
\*     O\_LOCATION\_FLAG       =  
\*     O\_SERVER              =  
     o\_path                = p\_path  
\*     ABEND\_FLAG            =  
\*   EXCEPTIONS  
\*     RFC\_ERROR             = 1  
\*     ERROR\_WITH\_GUI        = 2  
\*     OTHERS                = 3  
            .  
  IF sy-subrc <> 0.  
\* Implement suitable error handling here  
  ENDIF.  
ENDFORM.                    " GET\_FILE  
\*&---------------------------------------------------------------------\*  
\*&      Form  OPEN\_DATASET  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM open\_dataset .  
  OPEN DATASET p\_path FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.  
  IF sy-subrc NE 0.  
    MESSAGE 'not able to open dataset' TYPE 'E'.  
  ENDIF.  
ENDFORM.                    " OPEN\_DATASET  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  CLOSE\_DATASET  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM close\_dataset .  
  CLOSE DATASET p\_path.  
ENDFORM.                    " CLOSE\_DATASET  
\*&---------------------------------------------------------------------\*  
\*&      Form  DISPLAY\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM display\_data .  
  LOOP AT it\_marc INTO wa\_marc.  
    WRITE: / wa\_marc-matnr,  
             wa\_marc-werks,  
             wa\_marc-ekgrp.  
  ENDLOOP.  
ENDFORM.                    " DISPLAY\_DATA  
\*&---------------------------------------------------------------------\*  
\*&      Form  GET\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM get\_data .  
  SELECT matnr  
         werks  
         ekgrp  
    FROM marc  
    INTO TABLE it\_marc  
    WHERE matnr IN s\_matnr.  
ENDFORM.                    " GET\_DATA  
\*&---------------------------------------------------------------------\*  
\*&      Form  TRANSFER\_DATA  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM transfer\_data .  
  DATA: l\_data TYPE string.  
  LOOP AT it\_marc INTO wa\_marc.  
    CONCATENATE wa\_marc-matnr  
                wa\_marc-werks  
                wa\_marc-ekgrp  
           INTO l\_data  
           SEPARATED BY ','.  
    TRANSFER l\_data TO p\_path.  
  ENDLOOP.  
ENDFORM.                    " TRANSFER\_DATA