To download data from Application server files to presentation server

Applies to:

ABAP

Summary

Since it’s not possible to download data to presentation server in background, we first download the data to application server file, then read this application server file and save it to presentation server in foreground, which ideally means each such download program requires another program which takes the file structure into consideration and download it to presentation server.

This code snippet can be used to download data from files on application server to presentation server in foreground.

Author: Raghavendra AY

Company: Infosys Technologies Limited

Created on: 24 Jan 2007

Author Bio

Raghavendra is working as a Software Engineer in Infosys Technologies Limited.He works as a ABAP/4 developer.

Table of Contents

Advantages and limitations.

We have standard transaction CG3Z which can be used to download data from application server to presentation server but most of the custom programs we develop requires to download the data in the same format as it appears in the output list, E.g. when we download a file from application server to presentation server in Excel format using CG3Z the data gets misaligned or is not downloaded in the format necessary.

Using this code snippet we can download most of the files created by such custom programs to presentation server in the same format when the data in the file is in below format.

Each field should be delimited by #.

First line of the Application server file should contain the description of each field.

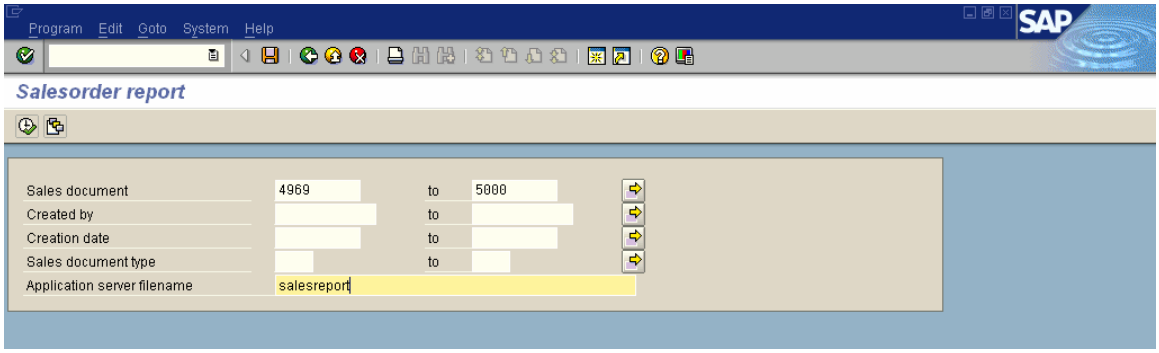
Maximum length of each field is considered to be 40 characters.

Maximum length of the field can be changed by specifying the length in selection screen parameter

Example

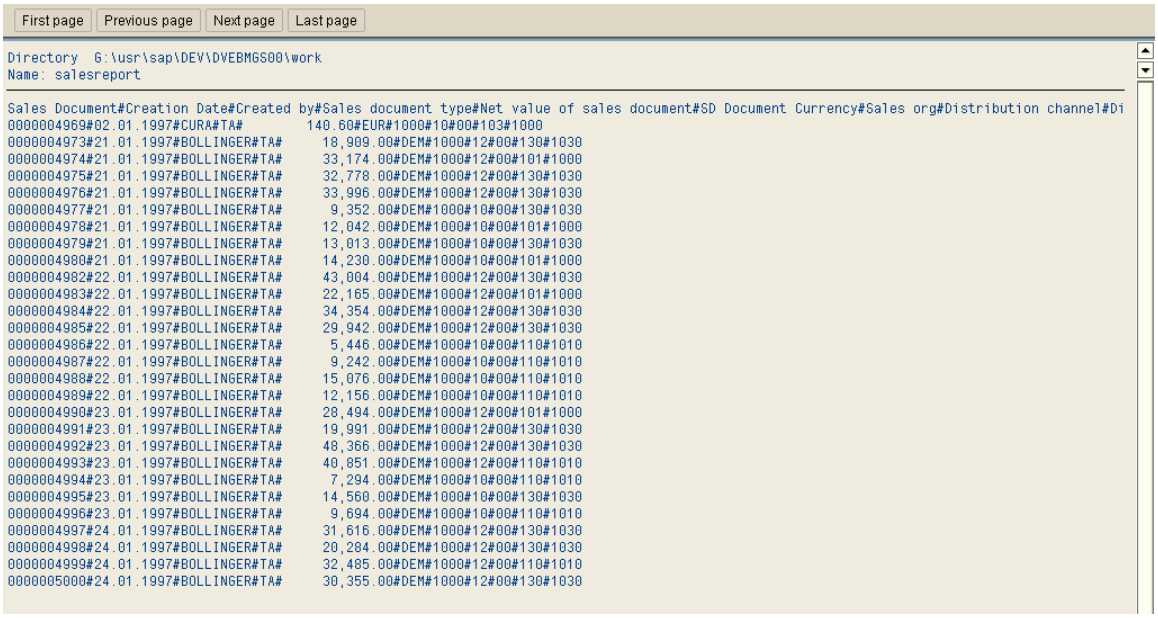
In this example we consider two report programs Sales Report and Purchase order report on execution of which the data will be stored on application server files.

Sales Report

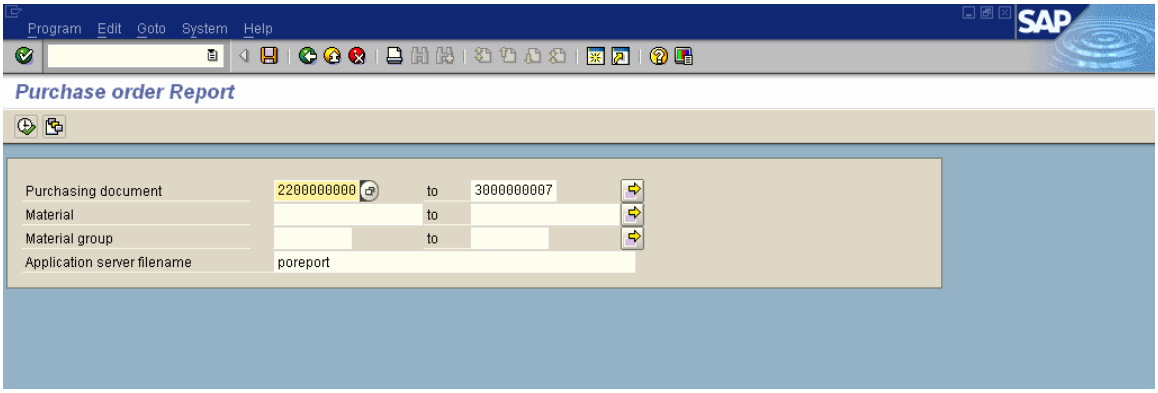


Output of the report will be stored in dataset ‘salesreport’ on application server.

Sales report file structure

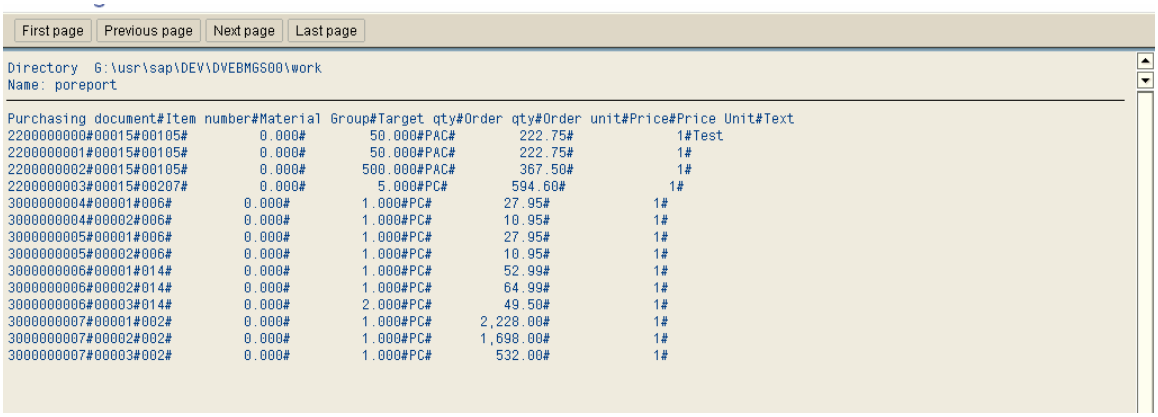


Purchase order report



Output of the report will be stored in dataset ‘poreport’ on application server.

Purchase order file structure



In a typical scenario if we need to download these two application server files to desktop, we need to write Two programs as the file structure is different for these two application server files

Sample code and snapshots

Using the below code we can download such files without the need for different programs when ever file structure varies.

Program documentation

Short text

Download to presentation server

This program can be used to download files from Application server to presentaion server.

1)Maximum length of each field is considered to be 40 characters.

2)Maximum length of the field can be changed by specifying the length in selection screen parameter

3)First line of the Application server file should contain the description of each field.

4)Each field should be delimited by #.

```
*&---------------------------------------------------------------------*
*& Report  ZDOWNLOADFILE                                               *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&                                                                     *
*&---------------------------------------------------------------------*

REPORT ZDOWNLOADFILE
       MESSAGE-ID B1       .

*---------------------------------------------------------------------*
* INCLUDES                                                            *
*---------------------------------------------------------------------*
INCLUDE ZDOWNLOADFILE_TOP.

INCLUDE ZDOWNLOADFILE_FORM.

*---------------------------------------------------------------------*
* EVENT-AT SELECTION-SCREEN                                           *
*---------------------------------------------------------------------*
AT SELECTION-SCREEN ON pa_appl.
  PERFORM check_file_exists USING pa_appl.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR pa_appl.
 PERFORM f4_dxfilename USING pa_appl.
```



```
AT SELECTION-SCREEN ON VALUE-REQUEST FOR pa_pres.
  PERFORM f4_filename USING pa_pres.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR pa_del.
  PERFORM f4_dxfilename USING pa_del.

*---------------------------------------------------------------------*
* EVENT INITIALIZATION                                                *
*---------------------------------------------------------------------*
INITIALIZATION.
  PERFORM initialization.

*---------------------------------------------------------------------*
* EVENT START-OF-SELECTION                                            *
*---------------------------------------------------------------------*
START-OF-SELECTION.

  PERFORM determine_fields.     "Determine number of columns
  PERFORM build_itab.           "Create internal table based on number
                                "of columns
  PERFORM build_header.         "To prepare header for the file
  PERFORM download_data.        "Download data to presentation server
  PERFORM delete_files.         "Delete files from application server


*&---------------------------------------------------------------------*
*&  Include           ZDOWNLOADFILE_TOP                                *
*&---------------------------------------------------------------------*

TYPES: BEGIN OF ty_header,
         text(100) TYPE c,
       END OF ty_header.

DATA: c_fnh_mask type dxfields-filemask value '*.*',
      search_dir type dxfields-longpath value '/sapglobal/users'.

CLASS cl_abap_char_utilities DEFINITION LOAD.

*---------------------------------------------------------------------*
* Internal tables                                                     *
```



```
*---------------------------------------------------------------------*
DATA: gt_header TYPE STANDARD TABLE OF ty_header,
      gt_fieldcat TYPE lvc_t_fcat.



*---------------------------------------------------------------------*
* Work areas                                                          *
*---------------------------------------------------------------------*
DATA: gs_header TYPE ty_header,
      gs_fieldcat TYPE lvc_s_fcat.

DATA: wa_filename TYPE string.

DATA: wa_count(2) TYPE N,
      wa_start(3) TYPE N,
      wa_end(3) TYPE N,
      wa_len(3) TYPE N.

DATA: wa_data(12000) TYPE c,
      wa_off TYPE I,
      itab_appl TYPE REF TO DATA,
      itab_line TYPE REF TO DATA,
      col(2) TYPE c.
DATA: wa_field(30) TYPE c.
DATA: lv_index TYPE sy-tabix.

DATA: gs_adrp    type adrp,
       gs_usr02   type usr02,
       gs_usr21   type usr21,
       gs_char50(50).

CONSTANTS: co_slash(1)   value '/'.

FIELD-SYMBOLS: <itab> TYPE STANDARD TABLE,
               <wa>,
               <fs_line>,
               <wa_line>.

*---------------------------------------------------------------------*
* SELECTION SCREEN                                                    *
```



```
*---------------------------------------------------------------------*
SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE text-f01.
  PARAMETERS: pa_appl LIKE rlgrap-filename OBLIGATORY.
SELECTION-SCREEN END OF BLOCK B1.

SELECTION-SCREEN BEGIN OF BLOCK B2 WITH FRAME TITLE text-f02.
 PARAMETERS: pa_pres LIKE rlgrap-filename OBLIGATORY.
SELECTION-SCREEN END OF BLOCK B2.

SELECTION-SCREEN BEGIN OF BLOCK B3 WITH FRAME TITLE text-f03.
 PARAMETERS: pa_del LIKE rlgrap-filename .
 PARAMETERS: pa_deld AS CHECKBOX.
SELECTION-SCREEN END OF BLOCK B3.

SELECTION-SCREEN BEGIN OF BLOCK B4 WITH FRAME TITLE text-f04.
 PARAMETERS: pa_len(3) TYPE c.
SELECTION-SCREEN END OF BLOCK B4.

*&---------------------------------------------------------------------*
*&  Include           ZDOWNLOADFILE_FORM                             *
*&---------------------------------------------------------------------*

*&---------------------------------------------------------------------*
*&      Form  f4_dxfilename
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_pa_appl  text
*----------------------------------------------------------------------*
form f4_dxfilename  using    p_file.
DATA: wa_file LIKE dxfields-longpath.
 CLEAR: wa_file.

 call function 'F4_DXFILENAME_TOPRECURSION'
  exporting
    i_location_flag       = 'A'
    i_server              = ' '
    i_path                = search_dir
    filemask              = c_fnh_mask
    fileoperation         = 'R'
```



```
  importing
    o_path               = wa_file
  exceptions
    rfc_error            = 1
    error_with_gui       = 2
    others               = 3
          .
 if sy-subrc <> 0.
    message id sy-msgid type sy-msgty number sy-msgno
    with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
 else.
   p_file = wa_file.
 endif.


endform.                    " f4_dxfilename


*&---------------------------------------------------------------------*
*&      Form  f4_filename
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_pa_pres  text
*----------------------------------------------------------------------*
form f4_filename  using    p_data.

    CALL FUNCTION 'F4_FILENAME'
     EXPORTING
       PROGRAM_NAME        = SYST-CPROG
       DYNPRO_NUMBER       = SYST-DYNNR
     IMPORTING
       FILE_NAME           = p_data
             .

endform.                    " f4_filename

*&---------------------------------------------------------------------*
*&      Form  check_file_exists
*&---------------------------------------------------------------------*
```



```
*       text
*---------------------------------------------------------------------*
*      -->P_pa_appl  text
*---------------------------------------------------------------------*
form check_file_exists  using  p_file.
DATA: wa_file LIKE rlgrap-filename.
 wa_file = p_file.
  OPEN DATASET wa_file FOR INPUT IN TEXT MODE ENCODING DEFAULT.
  IF sy-subrc = 8.
    MESSAGE E714
    WITH text-m01 p_file text-m02.
  ELSE.
    CLOSE DATASET p_file.
  ENDIF.

endform.                    " check_file_exists

*&---------------------------------------------------------------------*
*&      Form  build_header
*&---------------------------------------------------------------------*
*       text
*---------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*---------------------------------------------------------------------*
form build_header .

*Select user details
  clear gs_usr21-persnumber.
  select single persnumber
               into (gs_usr21-persnumber)
               from   usr21
               where  bname  = sy-uname.


  concatenate sy-uname
              co_slash
              gs_adrp-name_text(36)
              co_slash
              into gs_char50.
```



```
  condense gs_char50.

  clear gs_usr02-class.
  select single class
               into (gs_usr02-class)
               from   usr02
               where  bname  = sy-uname.

  concatenate gs_char50
              gs_usr02-class
              into gs_char50.
  condense gs_char50.


* Write report technical name
write 'Name: ' TO gs_header+0(10).
gs_header+11(*) = sy-repid.
  APPEND gs_header TO gt_header.
  CLEAR: gs_header.

* Write user data
  write 'User: ' TO gs_header+0(10).
  gs_header+11(*) = gs_char50.
  CONCATENATE gs_header gs_char50 INTO
              gs_header SEPARATED BY SPACE.
  APPEND gs_header TO gt_header.
  CLEAR: gs_header.

* Write System data
  write 'System: ' TO gs_header+0(10).
  write: sy-sysid to gs_header+11(3).
  write: co_slash to gs_header+14(1).
  write: sy-mandt to gs_header+15(3).
  APPEND gs_header TO gt_header.
  CLEAR: gs_header.

* Write System date and time
  write 'System: ' TO gs_header+0(10).
  write  sy-datum to gs_header+11(10).
  write  sy-uzeit to gs_header+22(8).
```



```
  APPEND gs_header TO gt_header.
  CLEAR: gs_header.

* Write Local date and time
  write 'Local: ' TO gs_header+0(10).
  write  sy-datlo to gs_header+11(10).
  write  sy-timlo to gs_header+22(8).
  APPEND gs_header TO gt_header.
  CLEAR: gs_header.
  APPEND gs_header TO gt_header.

endform.                    " build_header

*&---------------------------------------------------------------------*
*&      Form  determine_fields
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form determine_fields .
DATA: wa_data(600) TYPE c.

    CLEAR: wa_count,wa_start,wa_end,wa_data.
    wa_start = 0.
    wa_end = 1.
    OPEN DATASET pa_appl FOR INPUT IN TEXT MODE ENCODING DEFAULT.
    READ DATASET pa_appl INTO wa_data.
    wa_len = STRLEN( wa_data ).
    DO wa_len TIMES.
      IF wa_data+wa_start(wa_end) EQ
         cl_abap_char_utilities=>horizontal_tab.
         wa_count = wa_count + 1.
      ENDIF.
      wa_start = wa_start + 1.
    ENDDO.
   CLOSE DATASET pa_appl.
   wa_count = wa_count + 1.
endform.                    " determine_fields
```



```
*&---------------------------------------------------------------------*
*&      Form  build_itab
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form build_itab .
 DATA: wa_len(4) TYPE c.
   CLEAR: wa_len.
   IF pa_len IS INITIAL.
     wa_len = 40.
   ELSE.
     wa_len = pa_len.
   ENDIF.
    col = 1.
    DO wa_count TIMES.
     CONCATENATE 'FIELD' col INTO wa_field.
     gs_fieldcat-fieldname = wa_field.
     gs_fieldcat-outputlen = wa_len.
     gs_fieldcat-datatype = 'CHAR'.
     gs_fieldcat-col_pos = col.
     col = col + 1.
     APPEND gs_fieldcat TO gt_fieldcat.
     CLEAR: wa_field.
   ENDDO.

*Create the internal table dynamically based on the file structure,
*this table will be used to download data through GUI_DOWNLOAD fm
   CALL METHOD cl_alv_table_create=>create_dynamic_table
     EXPORTING
       it_fieldcatalog          = gt_fieldcat
     IMPORTING
       EP_TABLE                   = itab_appl
           .
*Assign the pointer to the field symbol
  ASSIGN itab_appl->* TO <itab>.
  CREATE DATA itab_line LIKE LINE OF <itab>.
```



```
*Create a work area for the dynamic internal table
  ASSIGN itab_line->* TO <wa_line>.

  col = 1.
    OPEN DATASET pa_appl FOR INPUT IN TEXT MODE ENCODING DEFAULT.
      DO.
       READ DATASET pa_appl INTO wa_data.
        IF sy-subrc <> 0.
          CLOSE DATASET pa_appl.
          RETURN.
        ELSE.
          wa_start = 0.
          lv_index = 1.
         DO wa_count TIMES.
          ASSIGN COMPONENT lv_index OF STRUCTURE <wa_line> TO <fs_line>.
            FIND cl_abap_char_utilities=>horizontal_tab
            IN wa_data+wa_start(*) MATCH OFFSET wa_off.
            IF sy-subrc = 0.
              IF wa_off NE 0.
                <fs_line> = wa_data+wa_start(wa_off).
              ENDIF.
                wa_start = wa_start + wa_off + 1.
                lv_index = lv_index + 1.
            ELSE.
              <fs_line> = wa_data+wa_start(*).
            ENDIF.
          ENDDO.
          APPEND <wa_line> TO <itab>.
          CLEAR: <wa_line>,<fs_line>.
        ENDIF.
      ENDDO.



endform.                    " build_itab

*&---------------------------------------------------------------------*
*&      Form  download_data
```



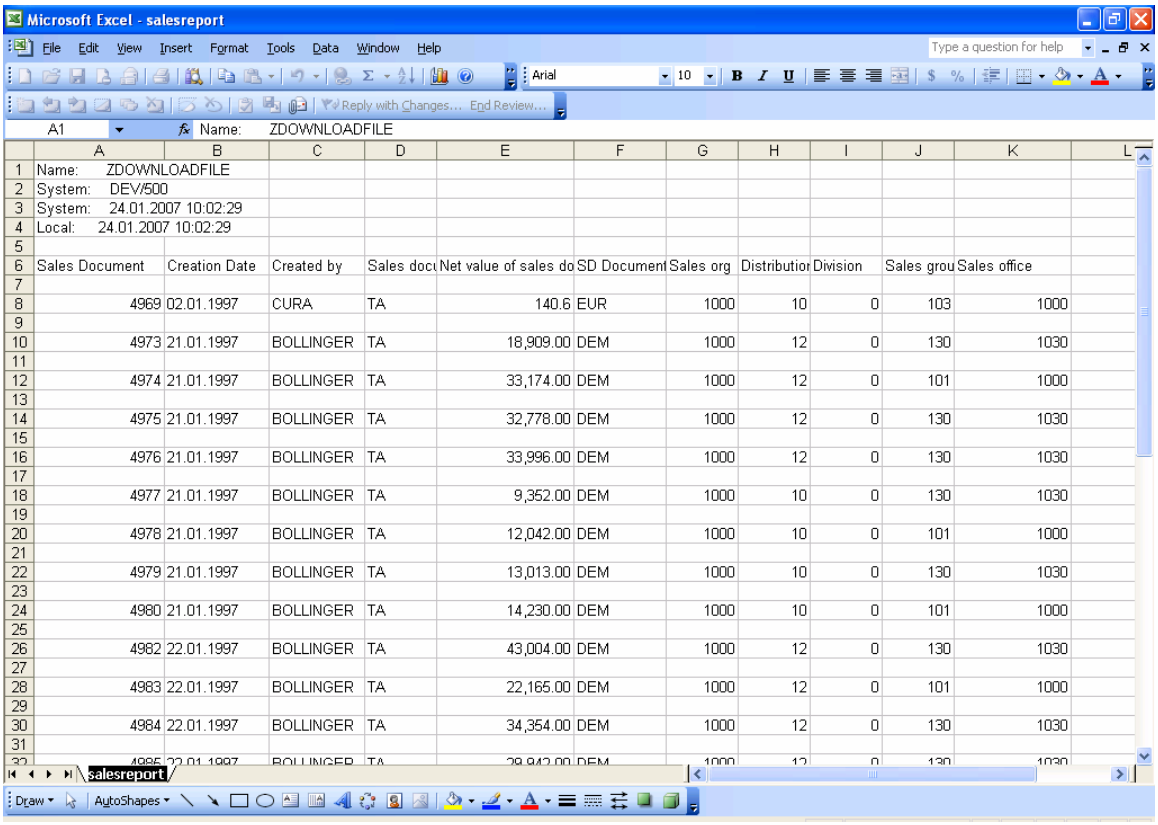
```
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form download_data .

  clear: wa_filename.
  wa_filename = pa_pres.

    call function 'GUI_DOWNLOAD'
    exporting
      filename                  = wa_filename
     filetype                  = 'DAT'
    tables
      data_tab                   = gt_header
    exceptions
   access_denied                = 15
          .

    call function 'GUI_DOWNLOAD'
    exporting
      filename                  = wa_filename
     filetype                  = 'DAT'
     append                     = 'X'
    tables
      data_tab                   = <itab>
    exceptions
   access_denied                = 15
          .

endform.                    " download_data

*&---------------------------------------------------------------------*
*&      Form  initialization
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
```



```
*  <--  p2        text
*---------------------------------------------------------------------*
form initialization .

  REFRESH: gt_header,gt_fieldcat,gt_header.

endform.                    " initialization

*&---------------------------------------------------------------------*
*&      Form  delete_files
*&---------------------------------------------------------------------*
*       text
*---------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*---------------------------------------------------------------------*
form delete_files .

  IF pa_deld EQ 'X'.
    DELETE DATASET pa_appl.
  ENDIF.

  IF pa_del IS NOT INITIAL.
    DELETE DATASET pa_del.
  ENDIF.

endform.                    " delete_files
```


Files downloaded will be as below

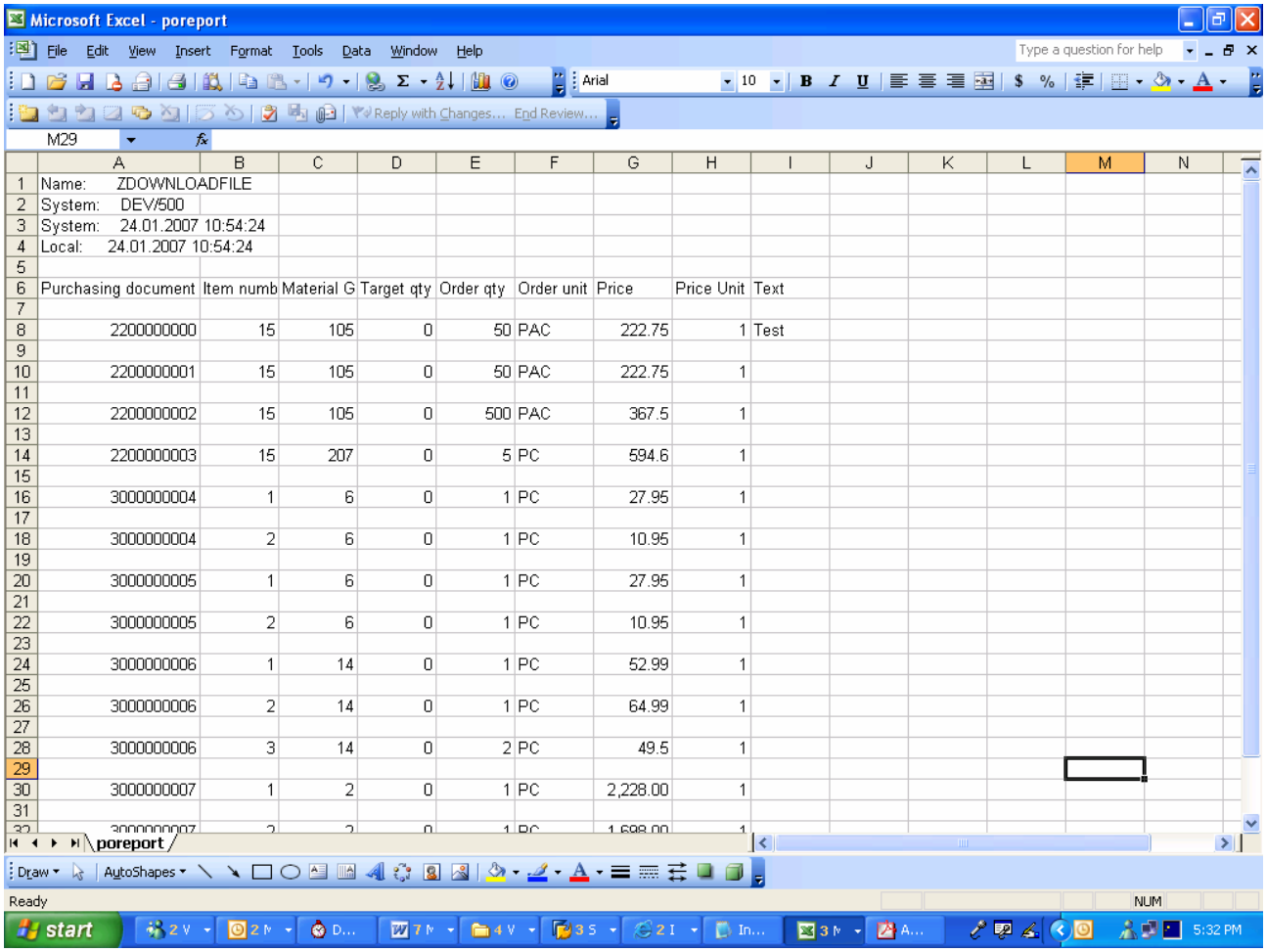
Snap shot of sales report file



Microsoft Excel - salesreport

	A	B	C	D	E	F	G	H	I	J	K
1	Name:	ZDOWNLOADFILE									
2	System:	DEV/500									
3	System:	24.01.2007 10:02:29									
4	Local:	24.01.2007 10:02:29									
5											
6	Sales Document	Creation Date	Created by	Sales doc	Net value of sales do	SD Document	Sales org	Distributior	Division	Sales grou	Sales office
7											
8	4969	02.01.1997	CURA	TA	140.6	EUR	1000	10	0	103	1000
9											
10	4973	21.01.1997	BOLLINGER	TA	18,909.00	DEM	1000	12	0	130	1030
11											
12	4974	21.01.1997	BOLLINGER	TA	33,174.00	DEM	1000	12	0	101	1000
13											
14	4975	21.01.1997	BOLLINGER	TA	32,778.00	DEM	1000	12	0	130	1030
15											
16	4976	21.01.1997	BOLLINGER	TA	33,996.00	DEM	1000	12	0	130	1030
17											
18	4977	21.01.1997	BOLLINGER	TA	9,352.00	DEM	1000	10	0	130	1030
19											
20	4978	21.01.1997	BOLLINGER	TA	12,042.00	DEM	1000	10	0	101	1000
21											
22	4979	21.01.1997	BOLLINGER	TA	13,013.00	DEM	1000	10	0	130	1030
23											
24	4980	21.01.1997	BOLLINGER	TA	14,230.00	DEM	1000	10	0	101	1000
25											
26	4982	22.01.1997	BOLLINGER	TA	43,004.00	DEM	1000	12	0	130	1030
27											
28	4983	22.01.1997	BOLLINGER	TA	22,165.00	DEM	1000	12	0	101	1000
29											
30	4984	22.01.1997	BOLLINGER	TA	34,354.00	DEM	1000	12	0	130	1030
31											

Snap shot of Purchase order report file



Microsoft Excel - poreport

	A	B	C	D	E	F	G	H	I
1	Name: ZDOWNLOADFILE								
2	System: DEV/500								
3	System: 24.01.2007 10:54:24								
4	Local: 24.01.2007 10:54:24								
5									
6	Purchasing document	Item numb	Material G	Target qty	Order qty	Order unit	Price	Price Unit	Text
7									
8	2200000000	15	105	0	50	PAC	222.75	1	Test
9									
10	2200000001	15	105	0	50	PAC	222.75	1	
11									
12	2200000002	15	105	0	500	PAC	367.5	1	
13									
14	2200000003	15	207	0	5	PC	594.6	1	
15									
16	3000000004	1	6	0	1	PC	27.95	1	
17									
18	3000000004	2	6	0	1	PC	10.95	1	
19									
20	3000000005	1	6	0	1	PC	27.95	1	
21									
22	3000000005	2	6	0	1	PC	10.95	1	
23									
24	3000000006	1	14	0	1	PC	52.99	1	
25									
26	3000000006	2	14	0	1	PC	64.99	1	
27									
28	3000000006	3	14	0	2	PC	49.5	1	
29									
30	3000000007	1	2	0	1	PC	2,228.00	1	
31									
32	3000000007	2	2	0	1	PC	1,698.00	1	

Related Content

www.sdn.sap.com

Help.sap.com

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

