A to Z of AL11 Operations

By **Swapnaja Bhandare** - May 20, 2019

With A to Z of OLE Excel in ABAP 7.4 we explored a lot about excel file in SAP. GUI_DOWNLOAD/UPLOAD is also very straight forward. But whenever I get requirement to read multiple files from application server, do processing and store multiple files on another server… well it stressed me hard in the beginning. 🙂

The other day one junior asked me, when we have GUI_DOWNLOAD and UPLOAD, then why do we need to save file in AL11. I had to say one line, ***so that you can shut down your laptop and go to sleep*** and when you come in the morning the files would be ready for you to review in AL11. He did not understand my abstract sentence and I am sure, many of you might also be wondering what I wrote.

Shares

Question: Why cannot we do GUI_DOWNLOAD/UPLOAD in background mode?

Answer: Jobs can be scheduled in background mode not only at the current time, but it can also be scheduled for future time. If you schedule a background job which does GUI_DOWNLOAD/UPLOAD, your program would dump or cancel out. The reason being, SAP did not want to take chance. GUI_DOWNLOAD/UPLOAD needs your laptop/computer to be switched on for it to save the data in the presentation server path. Say you scheduled a job which has GUI_DOWNLOAD/UPLOD and *switched off your laptop and went to bed*. When the job actually triggers, **where will it save the file since your laptop is switched off?** *Do you get the catch 22 situation?*

Therefore SAP came with Application File Concept (although not just for background job).

Now, let us do the same exercise. You have OPEN DATASET (for AL11) syntax in your program. You schedule a job. Switch off your laptop and go to sleep. In the middle of the night, the job triggers and it saves the file in Application Server (AL11). It is not dependent on your local laptop or desktop. It is at the Server level and Servers normally never sleep (except at planned outage or downtime). 😛

Also Read: Test Driven Development ABAP

Although the File Directory (AL11) concepts are well known to all of us, *we always Google for the syntax/tcode/FMs* and use them without worrying much about the details and concepts behind the scene (especially lazy programmers like me). 😛

So I thought of doing a favor to myself. Next time I need anything related to T-Code AL11 i.e. SAP Directories, I will not Google. Instead, I will come to this SAPYard page directly where I have listed almost everything we do in AL11. *This should be a one stop page for not only all ABAP developers, but also the Basis Team and the Functional/Business team should also find it enriching.*

In this article we will try to cover everything about AL11. *What AL11 is all about?* **How to create folder in AL11?** *Ways to download files from AL11 and upload multiple files into AL11?* **How to delete files** etc. As the title suggests, this is A to Z of all AL11 Operations. 😛

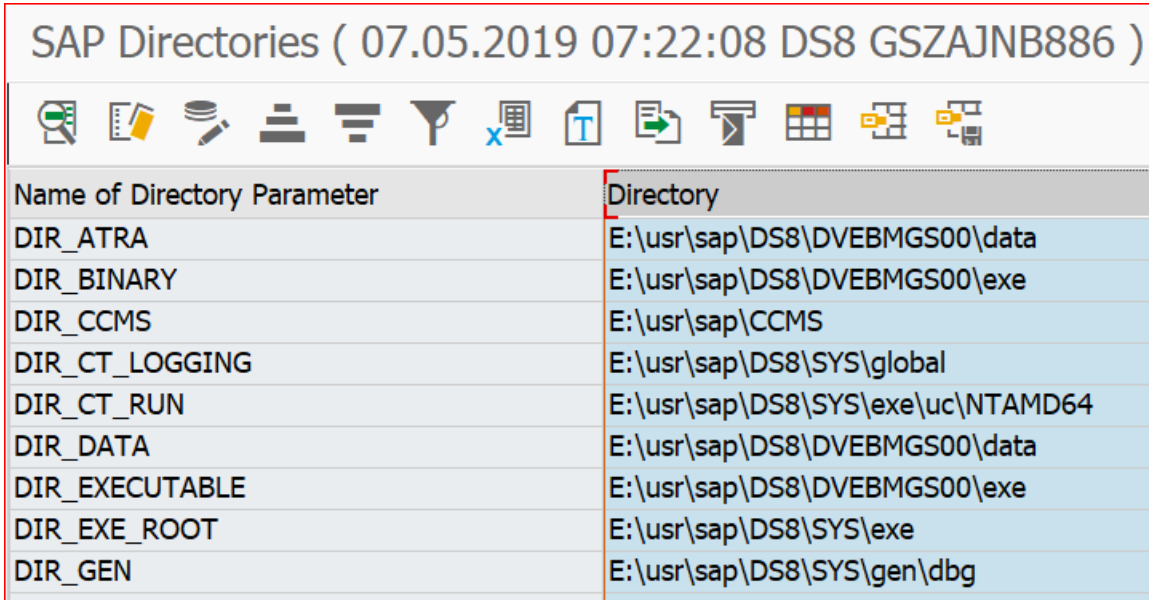
1. **What is AL11 and what is it's use?**

Well, AL11 is used for storing files/data on application server so that anyone (SAP) and even third-party systems (Non-SAP) can pick files from it. It is SAP File Directory as it contains directories, multiple folders and files in them. We can

Shares

upload/download/modify/delete files/folders in t-code AL11.

Below is a quick look of AL11 t-code.



SAP Directories (07.05.2019 07:22:08 DS8 GSZAJNB886)

Name of Directory Parameter	Directory
DIR_ATRA	E:\usr\sap\DS8\DVEBMGS00\data
DIR_BINARY	E:\usr\sap\DS8\DVEBMGS00\exe
DIR_CCMS	E:\usr\sap\CCMS
DIR_CT_LOGGING	E:\usr\sap\DS8\SYS\global
DIR_CT_RUN	E:\usr\sap\DS8\SYS\exe\uc\NTAMD64
DIR_DATA	E:\usr\sap\DS8\DVEBMGS00\data
DIR_EXECUTABLE	E:\usr\sap\DS8\DVEBMGS00\exe
DIR_EXE_ROOT	E:\usr\sap\DS8\SYS\exe
DIR_GEN	E:\usr\sap\DS8\SYS\gen\dbg

It has two columns:

Name of Directory Parameter: Contains the directory names. It can contain multiple folders and files.

Directory: Contains the whole path of the folders.

2. How to create our own directory in AL11?

This is not supposed to be done by any ABAP developer and usually we will not get any authorization to perform this activity.

Just for sake of a complete document, I coaxed one of my Basis friends in this crime. *See I have secret contacts in the SAP Administration and Security Team.* My Basis friend was kind enough to provide the below steps. Now, I am wondering what would be the return price I need to pay for this favor. Just kidding. Tell me what you want. Open offer for you. 😉

He said it is no rocket science

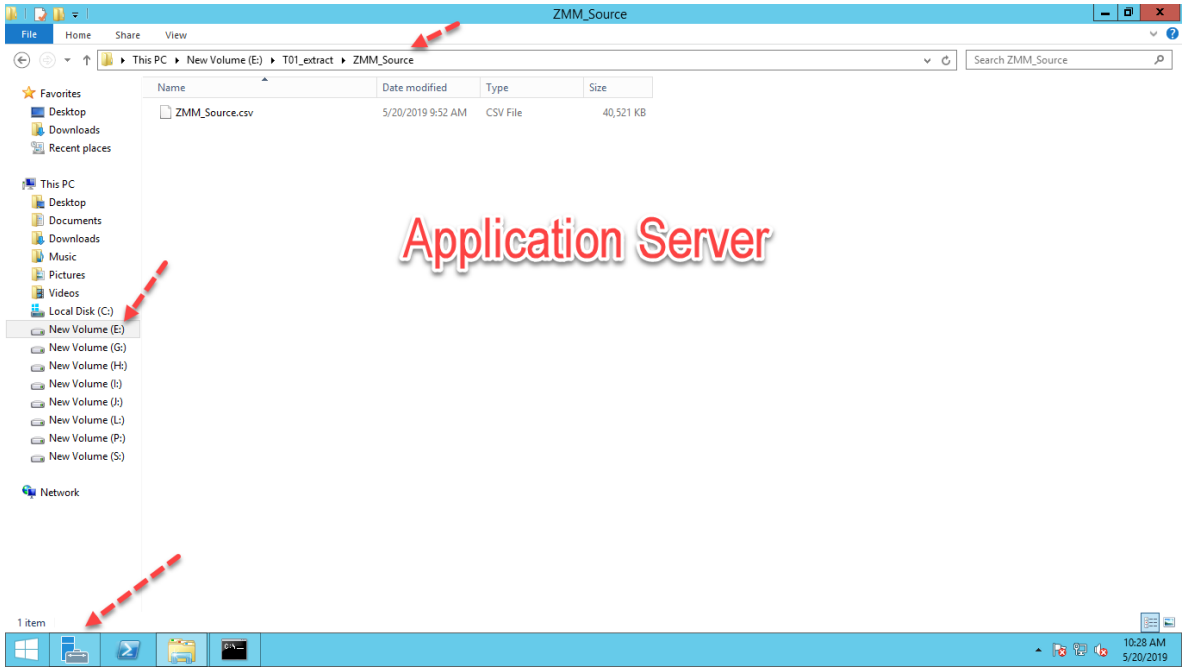
Shares

Directories are nothing but folders on Server System paths.

Basis folks and some privileged ones who have the authorization, log into the **SAP Server System** and create Folders and File. Just like how we create Folders and Files in our Windows/MAC Laptop and PC. As simple as that.

Now suppose we want to create a directory name as 'ZMM_SOURCE' on application server, Basis team will login in that **Server System** and create a folder with the same name which will be our directory.

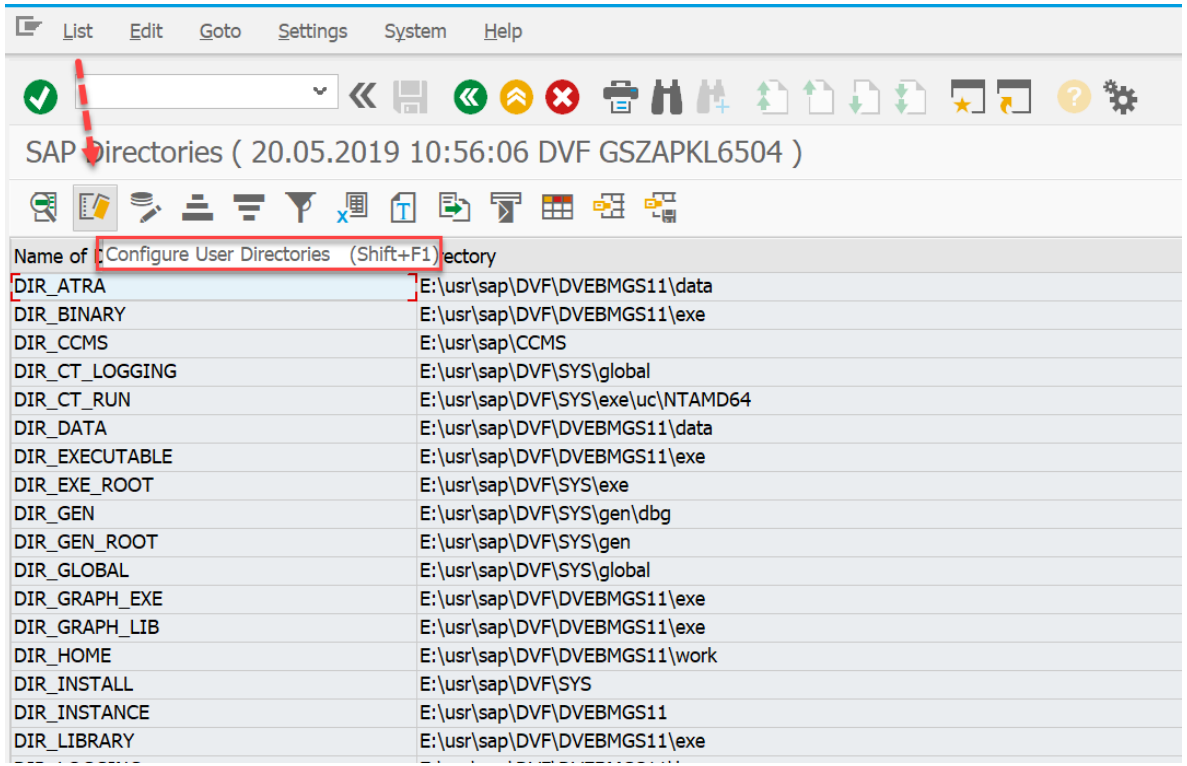
Below is a screenshot of one such SAP Server System. *It looks like normal Windows Folder. Right?* But look closely. It is not part of the Windows operating system. It is another Server. If they create a folder in this Server, we can see that same folder in AL11 t-code in SAP.



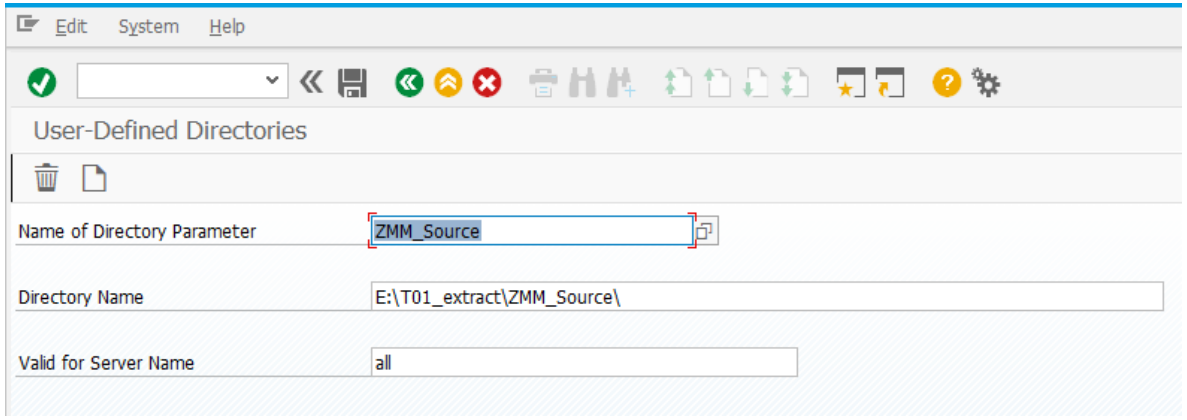
Now we need to configure this directory in AL11.

There is an option in AL11 called name '**Configure User Directories**'

Shares

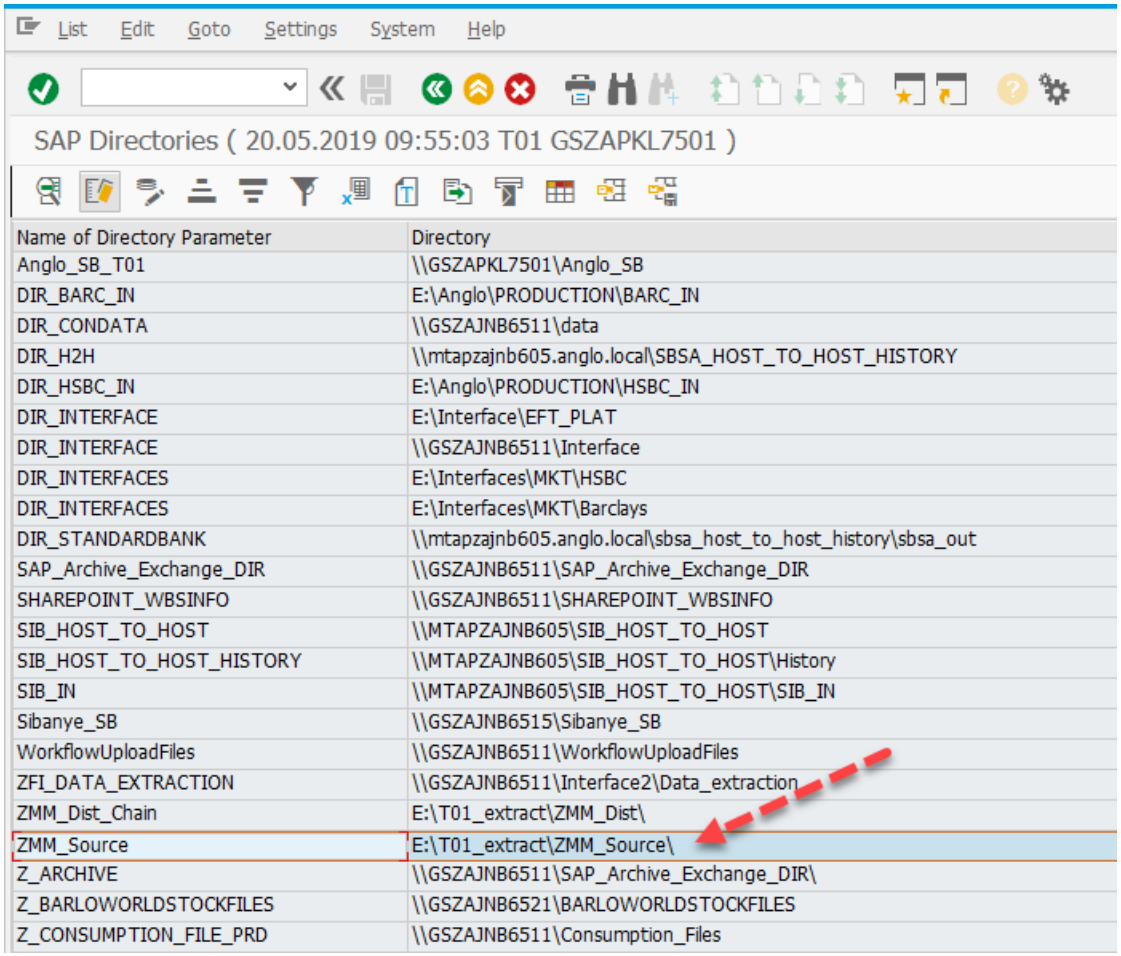


We need to click on that button and need to configure our directory.



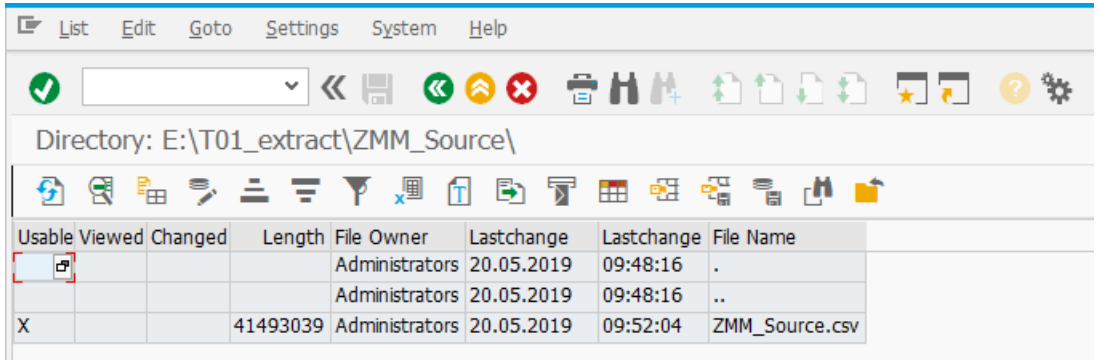
After execution, you can see your directory in AL11.

Shares



Name of Directory Parameter	Directory
Anglo_SB_T01	\\GSZAPKL7501\Anglo_SB
DIR_BARC_IN	E:\Anglo\PRODUCTION\BARC_IN
DIR_CONDATA	\\GSZAJNB6511\data
DIR_H2H	\\mtapzajnb605.anglo.local\SBSA_HOST_TO_HOST_HISTORY
DIR_HSBC_IN	E:\Anglo\PRODUCTION\HSBC_IN
DIR_INTERFACE	E:\Interface\EFT_PLAT
DIR_INTERFACE	\\GSZAJNB6511\Interface
DIR_INTERFACES	E:\Interfaces\MKT\HSBC
DIR_INTERFACES	E:\Interfaces\MKT\Barclays
DIR_STANDARDBANK	\\mtapzajnb605.anglo.local\sbsa_host_to_host_history\sbsa_out
SAP_Archive_Exchange_DIR	\\GSZAJNB6511\SAP_Archive_Exchange_DIR
SHAREPOINT_WBSINFO	\\GSZAJNB6511\SHAREPOINT_WBSINFO
SIB_HOST_TO_HOST	\\MTAPZAJNB605\SIB_HOST_TO_HOST
SIB_HOST_TO_HOST_HISTORY	\\MTAPZAJNB605\SIB_HOST_TO_HOST\History
SIB_IN	\\MTAPZAJNB605\SIB_HOST_TO_HOST\SIB_IN
Sibanye_SB	\\GSZAJNB6515\Sibanye_SB
WorkflowUploadFiles	\\GSZAJNB6511\WorkflowUploadFiles
ZFI_DATA_EXTRACTION	\\GSZAJNB6511\Interface2\Data_extraction
ZMM_Dist_Chain	E:\T01_extract\ZMM_Dist\
ZMM_Source	E:\T01_extract\ZMM_Source\
Z_ARCHIVE	\\GSZAJNB6511\SAP_Archive_Exchange_DIR\
Z_BARLOWORLDSTOCKFILES	\\GSZAJNB6521\BARLOWORLDSTOCKFILES
Z_CONSUMPTION_FILE_PRD	\\GSZAJNB6511\Consumption_Files

If you double click on this, you will see one file in it as I already created it.



Usable	Viewed	Changed	Length	File Owner	Lastchange	Lastchange	File Name
				Administrators	20.05.2019	09:48:16	.
				Administrators	20.05.2019	09:48:16	..
X			41493039	Administrators	20.05.2019	09:52:04	ZMM_Source.csv

Shares

ıs easy as right click and create new folder action of Windows. 😊

Also Check – Free Video Course on HANA Models/Views

3. How to create new folder in directory in AL11?

This is also not part of ABAP work. Only Basis teams have authorization to create it. And this is nothing but creating new subfolder under our main folder in application server. Whereas main folder works like directory and subfolder works like folders/subdirectories in AL11.

Whatever locations on server system, where we are creating folders are called as **Physical Paths** of directories.

If main folder/directory is already configured *then there is no need to configure the sub directory again after addition of subfolders.* Those would be displayed in AL11 directory path automatically.

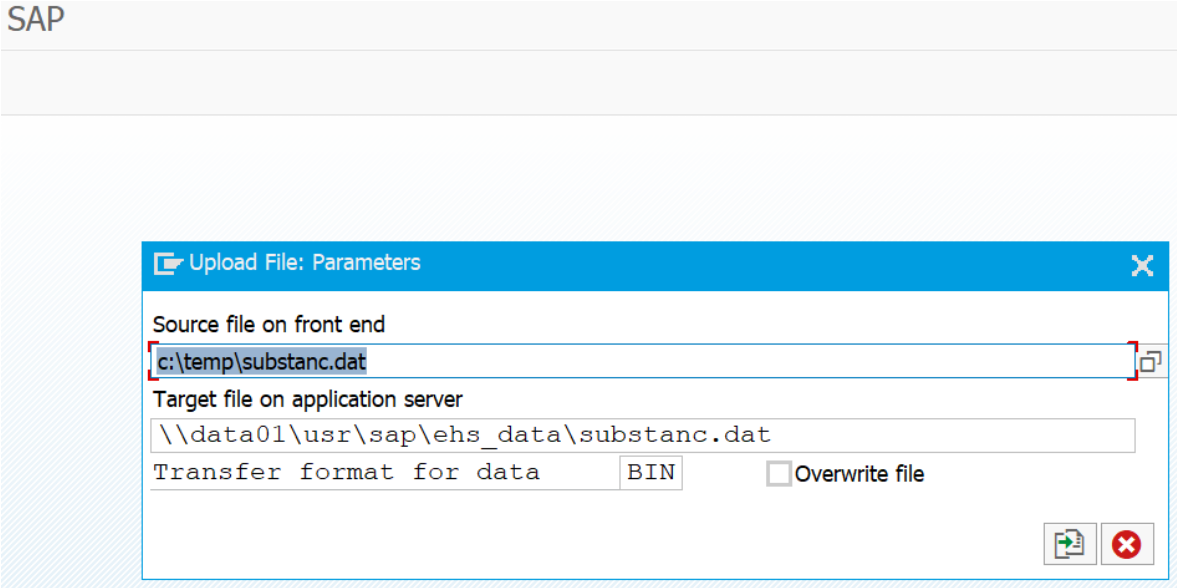
4. How to write/upload files on specific folder of AL11?

There are 2 ways to upload the file in AL11:

i) **Manually via T-Code CG3Z:**

After hitting this T-Code CG3Z, you will get below screen.

Shares

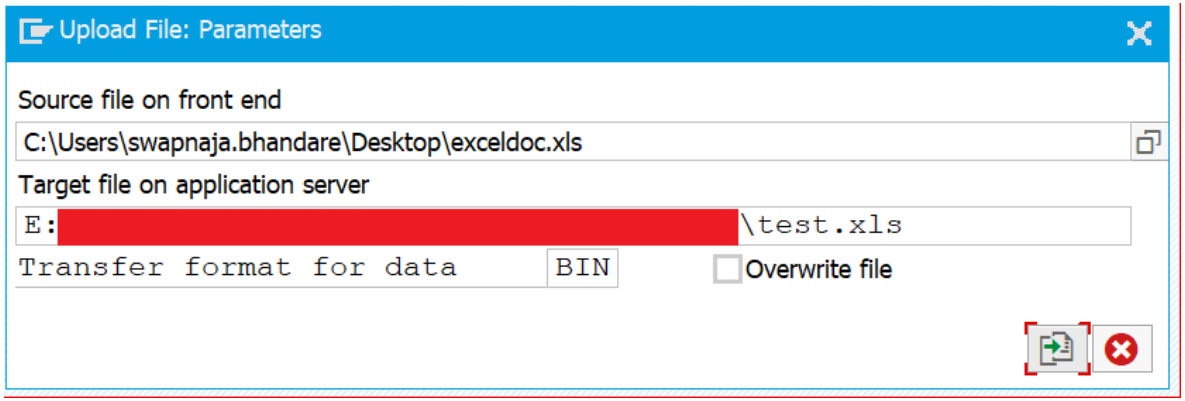


Source file on front end: Select your file from your laptop or desktop system and put it in this parameter.

Target file on application server: Provide path of file location in the AL11 directory.

Overwrite file: If it is checked, whatever file name you provided in target file parameter, if it exists in AL11, it will be overwriting with provided file else it will create new file there.

I have provided below details:



And got below message after clicking on the upload button:

Shares

File C:\Users\swapnaja.bhandare\Desktop\exceldoc.xls transferred to E:\[redacted]\test.xls

Let's check in AL11 folder now. We can see our file test.xls is created here.

Directory: E:\[redacted]

Usable	Viewed	Changed	Length	File Owner	Lastchange	Lastchange	File Name
				Administrators	07.05.2019	17:36:16	.
				Administrators	29.04.2019	18:06:29	..
X			799	Administrators	07.05.2019	17:38:41	A123410.csv
X			1008	Administrators	07.05.2019	17:38:41	ABCHRISTOK01.csv
X			20992	Administrators	08.05.2019	15:32:56	test.xls

Note: We can ONLY read Doc or Txt files directly by double clicking on the file in AL11 but we can't read XLS or XLSX files directly. XLS File will give below error.

File test.xls is not a text file

This error comes because **.XLS is a proprietary format of Microsoft**. So, unless you have some addons installed on SAP Server, you can't read XLS files from Application Server.

Do not worry. That is not the end of the world. If you really want to read the XLS file, you can download the file to your local machine (laptop/desktop) and then you can open and read it. 😀

This brings us to question. How to download AL11 files? Please be patient. It will be covered in the later sections in this article.

Here we are still covering how to create Folders/Files in AL11. Let's look the second way.

ii) **By writing ABAP code:**

As an ABAP Programmer, usually we do not play with t-codes. We want everything should be done by our code. *That gives us more sense of Power*. 🙂

Shares

Sometimes we get requirement to upload multiple files on application server in a single go with some conditions. Let us look into one practical example:

Suppose I have a requirement to collect data from EKKO and EKPO tables and prepare multiple files for each new PO (EBELN) and store them on AL11 path. Each file should have PO number as file_name.

Sounds interesting yeah… Let's do this…

First, we will select data from EKKO and EKPO tables and stores them in one internal table.

```
<code>*** Popoluate tables
SELECT ebeln,
       bukrs,
       bstyp,
       bsart
  INTO TABLE @DATA(gt_ekko)
  FROM ekko
  UP TO 15 ROWS
  WHERE ebeln LIKE '45%'.

IF gt_ekko[] IS NOT INITIAL.

  SELECT ebeln,
         ebelp,
         matnr,
         bukrs
    INTO TABLE @DATA(gt_ekpo)
    FROM ekpo
    FOR ALL ENTRIES IN @gt_ekko
    WHERE ebeln = @gt_ekko-ebeln
    ORDER BY PRIMARY KEY.

ENDIF.</code>
```

THERE IS A TABLE CALLED FILEPATH.

All folders of an application server are stored in the database table **FILEPATH**. We can pass our directory/folder name there and can confirm whether the folder exists or not.

```
<code>CONSTANTS: lc_filename TYPE filepath-pathintern
                 VALUE 'ZMM_BATCH_DATA_FOR_ASPECT'.
```

Shares

```
  FROM filepath
  INTO @DATA(lv_logical_path)
  WHERE pathintern = @lc_filename.</code>
```

As we want to upload multiple files in our folder, we need to call FM **FILE_GET_NAME_USING_PATH** to get full path of that file and then can open it with Dataset concept:

```
<code>CALL FUNCTION 'FILE_GET_NAME_USING_PATH'
          EXPORTING
            logical_path               = lv_logical_path
            file_name                  = lv_file_post
          IMPORTING
            file_name_with_path        = lv_filepath
          EXCEPTIONS
            path_not_found             = 1
            missing_parameter          = 2
            operating_system_not_found = 3
            file_system_not_found      = 4
            OTHERS                     = 5.</code>
```

Now using OPEN DATASET, we can open the file to write fetched data from EKPO into it.

```
<code>"Open the file to write.
 OPEN DATASET lv_filepath FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.  </code>
```

Mode 'Output' is used for writing data in application server file (all dataset concepts are explained in the 5th point below).

Now using TRANSFER and CLOSE DATASET concept, we can pass all relevant data to corresponding files, save them and take exit from the files.

Whole executable code is provided below:

```
<code>REPORT zal11_working.

CONSTANTS: lc_e    TYPE c VALUE 'E',
           lc_s    TYPE c VALUE 'S',
           lc_text TYPE string VALUE 'No specific data found for the input',
           lc_msg  TYPE string VALUE 'The batch data is/are stored at the directory successfully',
           lc_filename  TYPE filepath-pathintern VALUE 'ZMM_BATCH_DATA_FOR_ASPECT'.

DATA: lv_filepath TYPE string.

*** Popoluate tables
```

Shares

```
       bstyp,
       bsart
  INTO TABLE @DATA(gt_ekko)
  FROM ekko
  UP TO 15 ROWS
  WHERE ebeln LIKE '45%'.

IF gt_ekko[] IS NOT INITIAL.

  SELECT ebeln,
         ebelp,
         matnr,
         bukrs
    INTO TABLE @DATA(gt_ekpo)
    FROM ekpo
    FOR ALL ENTRIES IN @gt_ekko
    WHERE ebeln = @gt_ekko-ebeln
    ORDER BY PRIMARY KEY.

ENDIF.


IF gt_ekpo[] IS INITIAL.
  MESSAGE lc_text TYPE lc_s DISPLAY LIKE lc_e.
ELSE.

CONSTANTS: lc_filename  TYPE filepath-pathintern VALUE 'ZMM_BATCH_DATA_FOR_ASPECT'.

  SELECT SINGLE pathintern
  FROM filepath
  INTO @DATA(lv_logical_path)
  WHERE pathintern = @lc_filename.

  LOOP AT gt_ekpo ASSIGNING FIELD-SYMBOL(&lt;fs_ekpo>).

    AT NEW ebeln.

      "Logic to create unique file name with each PO number.
      DATA(lv_file_post) = | { &lt;fs_ekpo>-ebeln } .csv|.

      "Remove spaces.
      CONDENSE lv_file_post NO-GAPS.

      CALL FUNCTION 'FILE_GET_NAME_USING_PATH'
          EXPORTING
            logical_path               = lv_logical_path
            file_name                  = lv_file_post
          IMPORTING
            file_name_with_path        = lv_filepath
          EXCEPTIONS
            path_not_found             = 1
```

Shares

```
            operating_system_not_found = 3
            file_system_not_found      = 4
            OTHERS                     = 5.

      "Open the file to write.
      OPEN DATASET lv_filepath FOR OUTPUT IN TEXT MODE ENCODING DEFAULT.

      "Set intensity of the background color to default.
      FORMAT INTENSIFIED ON.

      "Prepare header of file with comma ',' separated.
      DATA(lv_file) = |EBELN,EBELP,MATNR,BUKRS|.

      "set non-intensified background color.
      FORMAT INTENSIFIED OFF.

      "Transfer header data line by line to file.
      TRANSFER lv_file TO lv_filepath.

      CLEAR lv_file.
    ENDAT.

    "Prepare data of file with comma ',' separated.
    lv_file  = |{ &lt;fs_ekpo>-ebeln } , { &lt;fs_ekpo>-ebelp } , { &lt;fs_ekpo>-matnr } , { &lt;fs_ekpo>-bukrs }|.

    "Remove spaces.
    CONDENSE lv_file.

    "Transfer data line by line to file.
    TRANSFER lv_file TO lv_filepath.

    AT END OF ebeln.
    "CLose the file and save on Application server.
    CLOSE DATASET lv_filepath.
    IF sy-subrc IS INITIAL.
      MESSAGE lc_msg TYPE lc_s.
    ENDIF.
    CLEAR lv_filepath.
    ENDAT.

    CLEAR: lv_file.
  ENDLOOP.
ENDIF.</code>
```

By using COMMAND-BREAK statements (AT END OF), I have achieved to create new file for new PO. The output is like below:

Shares

Directory: E:\Interfaces\ZMM_BATCH_DATA_FOR_ASPECT

Usable	Viewed	Changed	Length	File Owner	Lastchange	Lastchange	File Name
				Administrators	17.05.2019	17:07:58	.
				Administrators	29.04.2019	18:06:29	..
X			54	Administrators	14.05.2019	17:25:07	4500000000.csv
X			73	Administrators	14.05.2019	17:12:43	4500000001.csv
X			54	Administrators	14.05.2019	17:12:43	4500000002.csv
X			54	Administrators	14.05.2019	17:12:43	4500000003.csv
X			54	Administrators	14.05.2019	17:12:43	4500000004.csv
X			54	Administrators	14.05.2019	17:12:43	4500000005.csv
X			64	Administrators	14.05.2019	17:12:43	4500000006.csv
X			170	Administrators	14.05.2019	17:12:43	4500000007.csv
X			54	Administrators	14.05.2019	17:12:43	4500000008.csv
X			54	Administrators	14.05.2019	17:12:43	4500000009.csv
X			54	Administrators	14.05.2019	17:12:43	4500000010.csv
X			54	Administrators	14.05.2019	17:12:43	4500000011.csv
X			83	Administrators	14.05.2019	17:12:43	4500000012.csv
X			73	Administrators	14.05.2019	17:12:43	4500000013.csv
X			54	Administrators	14.05.2019	17:12:43	4500000014.csv

As shown above, these are different files with different POs. If you open any file, you can get header text line as well as values of cells.

SAP Directories

```
Directory:     E:\Interfaces\ZMM_BATCH_DATA_FOR_ASPECT
Name:          4500000006.csv
```

```
EBELN,EBELP,MATNR,BUKRS
4500000006 , 00010 , AM0878285 , X200
```

Also Take – End to End Debugging Course for SAP Functionals

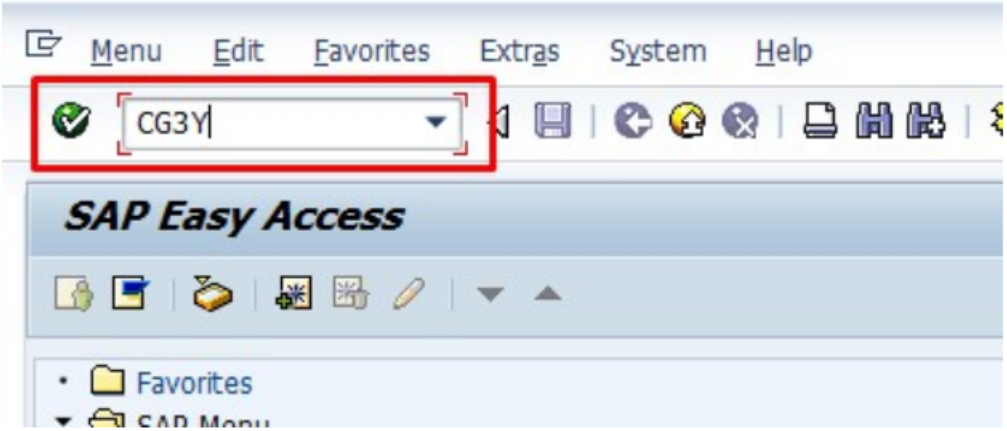
5. How to read/download files from AL11?

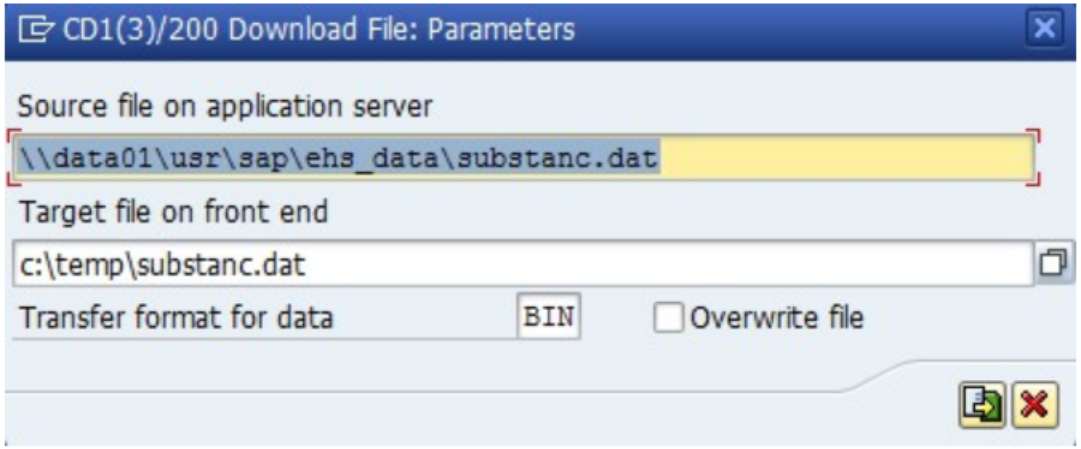
Shares

Above we learned how to write to AL11 file. Now let's learn how to download them. Here we will discuss 4 different approaches:

i) Download via T-Code CG3Y:

After hitting this tcode, you will see below screen:

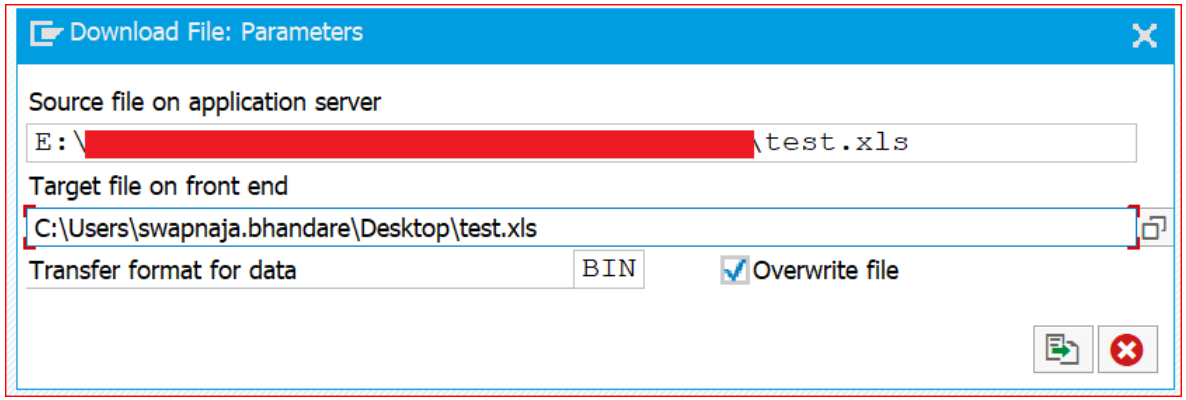




Shares

- **Source file on application server**: Provide the path of file location from AL11 directory which you need to download.
- **Target file on front end:** Provide the file name from your system (can be new file) and put it in this parameter.
- **Overwrite file:** If it is checked, whatever file name you provided in target file parameter, if it exists in system, it will be overwriting with provided file else it will create new file there.
- **Transfer format for data:** Usually ASC format works. Do check your requirement. You might need BIN as well.

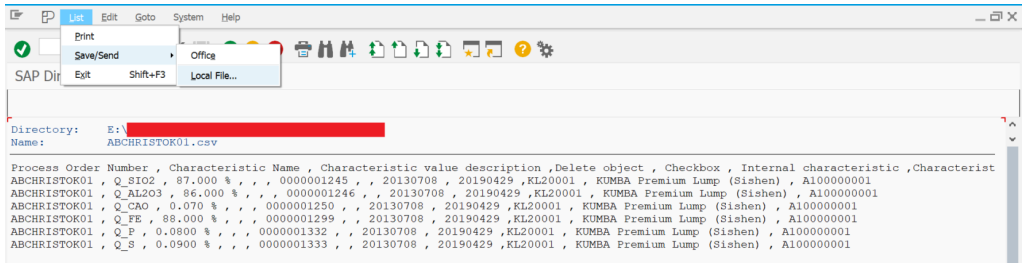
As I wanted to save/download test excel file from application server to my desktop, I have provided below details:



It will create test file on the desktop and if it is already present then will overwrite it.

ii) Download using list menu:

If the file is not so long, you can directly download the content from AL11. Navigate to the file. Then Open the file. On System Menu, go to List->Save/Send/Local file:



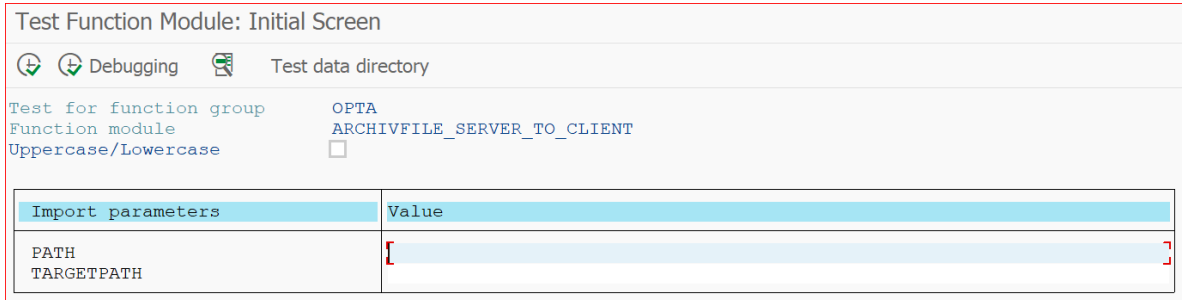
Question: Why is my AL11 exported file truncated?

Shares

Answer: In some case, the exported file is truncated. For List export in AL11, the limit is **512** characters. Apply SAP Note **952766** to increase the limit in AL11.

iii) By using FM ARCHIVFILE_SERVER_TO_CLIENT:

The standard Function module **ARCHIVFILE_SERVER_TO_CLIENT** can be used for downloading file from AL11. Just provide the PATH: the full path to the file on SAP server and the TARGETPATH: the destination path.



iv) By writing ABAP code:

Now if we have complex requirement like reading/downloading multiple files and do some calculation and massaging of data from the AL11 file then we need to create an ABAP program.

For downloading the file from application server, we must know the below concepts:

a) OPEN DATASET:

```
<code>OPEN DATASET dset FOR access IN mode [position]
                                           [os_additions]
                                           [error_handling]. </code>
```

Dset: Contains the file path of AL11.

Access: It has below options –

- **Input:** It opens the file for reading. If the file does not exist, the sy-subrc is set to 8.
- **Output:** Opens the file for writing. If the file exists, the content will be deleted and replaced. If the file does not exist,

Shares

- **Appending:** It opens the file for appending. The new content will be added to the end of existing file or the file will be created if it does not exist. SY-SUBRC = 4 will be triggered if appending is not successful.

- **Update:** It opens the file for changes to the existing content.

Mode: You can specify which mode you want to read the file like binary or text.

ii) TRANSFER DATASET: This statement passes the content of data object dobj to the file specified in dset.

Syntax:

```
<code>TRANSFER dobj TO dset [LENGTH len]
                      [NO END OF LINE]. </code>
```

The file must be open for writes, appends, or changes. If a closed or invalid file is accessed, a handleable exception is raised.

iii) CLOSE DATASET: This statement closes the file specified in dset.

Syntax:

```
<code>CLOSE DATASET dset.</code>
```

If the file is already closed or does not exist, the statement is ignored and the return code sy-subrc is set to 0.

An opened file that was not explicitly closed using CLOSE DATASET is automatically closed when the program is exited.

Did I bore you to death with too much of theories?? ufff.. Even I am tired, so I can understand your plight. 🙂

Let's bring some charm to our learning with the below examples:

In the point 4 above, we created multiple files on our application server now. Let's read them all. Here I have passed only one file for demonstration. You can make it dynamic as well by passing all file names:

```
<code>DATA:v_file LIKE v_excel_string VALUE 'E:\Interfaces\ZMM_BATCH_DATA_FOR_ASPECT\4500000001.csv'.          " name of
```

Shares

```
OPEN DATASET v_file FOR INPUT IN TEXT MODE ENCODING DEFAULT.</code>
```

Open the file in READ mode. We can read each line from the file in a loop and split the data based on delimiter to separate the data and store in internal table:

```
<code>DO.
    READ DATASET v_file INTO wa_tab.
    IF sy-subrc IS INITIAL.
      CLEAR it_string[].
      SPLIT wa_tab AT delimiter INTO TABLE it_string.

      LOOP AT it_string ASSIGNING FIELD-SYMBOL(&lt;fs>).
        ASSIGN COMPONENT sy-tabix OF STRUCTURE wa_tab2 TO FIELD-SYMBOL(&lt;fs_f>).
        IF sy-subrc = 0.
          &lt;fs_f> = &lt;fs>.
        ENDIF.
        AT LAST.
          APPEND wa_tab2 TO it_tab2.
        ENDAT.
      ENDLOOP.

    ELSE.
      EXIT.
    ENDIF.
  ENDDO.  </code>
```

After receiving the data in an internal table, we can CLOSE the file using CLOSE DATSET syntax:

```
<code>CLOSE DATASET v_file.</code>
```

Here I have used ALV factory method to display the results in ALV format. Below is the whole executable program:

```
<code>REPORT zal11_working.

*--- Type structure declaration.
TYPES: BEGIN OF ty_tab1,
         ebeln TYPE string,
         ebelp TYPE string,
         matnr TYPE string,
         bukrs TYPE string,
       END OF ty_tab1.

*--- work area and internal table declaration.
DATA : it_tab TYPE STANDARD TABLE OF alsmex_tabline,
       wa_tab TYPE alsmex_tabline,
       it_tab2 TYPE STANDARD TABLE OF ty_tab1.
```

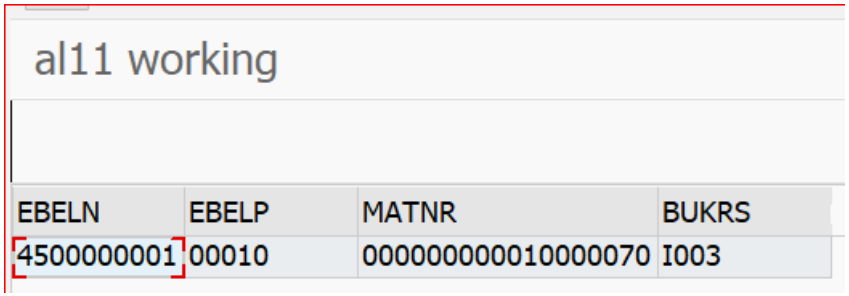
Shares

```
       it_string TYPE TABLE OF string.

*--- variable  declaration
DATA: v_excel_string(2000) TYPE c,
      v_file               LIKE v_excel_string VALUE 'E:\Interfaces\ZMM_BATCH_DATA_FOR_ASPECT\4500000001.csv',         '
      delimiter            TYPE c VALUE ',',
      t_field(1000)        TYPE c OCCURS 0.                                                      " delimiter with defa

*--- Factory method objects
DATA: lr_table TYPE REF TO cl_salv_table,
      lr_columns type ref to cl_salv_columns_table,
      lr_column  type ref to cl_salv_column_table.

*---  read the file from the application server
OPEN DATASET v_file FOR INPUT IN TEXT MODE ENCODING DEFAULT.
IF sy-subrc NE 0.
  WRITE:/ 'error opening file'.
ELSE.
  DO.
    READ DATASET v_file INTO wa_tab.
    IF sy-subrc IS INITIAL.
      CLEAR it_string[].
      SPLIT wa_tab AT delimiter INTO TABLE it_string.

      LOOP AT it_string ASSIGNING FIELD-SYMBOL(&lt;fs>).
        ASSIGN COMPONENT sy-tabix OF STRUCTURE wa_tab2 TO FIELD-SYMBOL(&lt;fs_f>).
        IF sy-subrc = 0.
          &lt;fs_f> = &lt;fs>.
        ENDIF.
        AT LAST.
          APPEND wa_tab2 TO it_tab2.
        ENDAT.
      ENDLOOP.

    ELSE.
      EXIT.
    ENDIF.
  ENDDO.
ENDIF.

CLOSE DATASET v_file.

*------display the data from the internal table

IF it_tab2[] IS NOT INITIAL.
   DELETE it_tab2 INDEX 1. "Delete header line.
ENDIF.

cl_salv_table=>factory(
               IMPORTING r_salv_table = lr_table
               CHANGING t_table = it_tab2 ).
```

Shares

```
lr_columns = lr_table->get_columns( ).

  try.
      lr_column ?= lr_columns->get_column( 'EBELN' ).
      lr_column->set_short_text( 'EBELN' ).

      lr_column ?= lr_columns->get_column( 'EBELP' ).
      lr_column->set_short_text( 'EBELP' ).

      lr_column ?= lr_columns->get_column( 'MATNR' ).
      lr_column->set_short_text( 'MATNR' ).

      lr_column ?= lr_columns->get_column( 'BUKRS' ).
      lr_column->set_short_text( 'BUKRS' ).

    catch cx_salv_not_found.                            "#EC NO_HANDLER
  endtry.

lr_table->display( ).</code>
```

After its execution, you will get below screen:

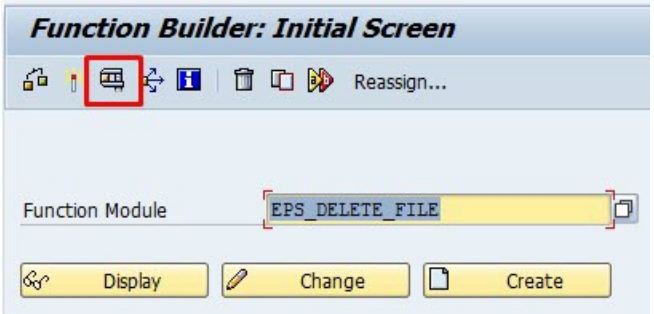


Also Read: How I created my First OData Service in SAP

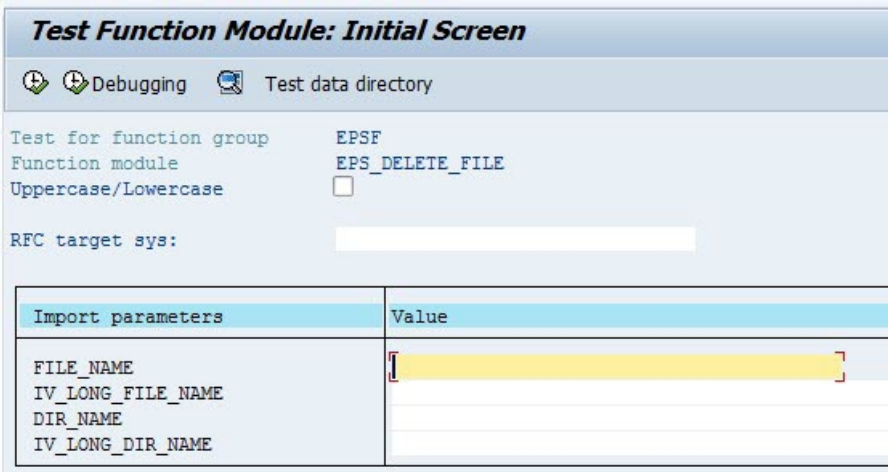
6. **How to delete files from AL11?**

After all you would not want to keep the test files in the application server after testing. So, to delete all unwanted files, you can use below FM: **EPS_DELETE_FILE**

Shares



It has below structures in it:



Enter the File name in the FILE_NAME and the directory path (Excluding the file name) in the DIR_NAME and execute.

Shares

Test Function Module: Initial Screen

Debugging Test data directory

Test for function group EPSF
Function module EPS_DELETE_FILE
Uppercase/Lowercase

RFC target sys:

Import parameters	Value
FILE_NAME	test.xls
IV_LONG_FILE_NAME	
DIR_NAME	E:\Interfaces\ZMM_BATCH_DATA_FOR_ASPECT
IV_LONG_DIR_NAME	

After execution, the file will disappear permanently from the AL11 location. As Spiderman's uncle said, "*With more Power comes more Responsibility*". **So please use it judiciously.**

Directory: E:\Interfaces\ZMM_BATCH_DATA_FOR_ASPECT

Usable	Viewed	Changed	Length	File Owner	Lastchange	Lastchange	File Name
				Administrators	08.05.2019	16:45:32	.
				Administrators	29.04.2019	18:06:29	..
X			799	Administrators	07.05.2019	17:38:41	A123410.csv
X			1008	Administrators	07.05.2019	17:38:41	ABCHRISTOK01.csv

This FM we can use at program level as well if required to delete files in mass.

We can use **DELETE DATASET** statement too at program level to delete files from application server.

Syntax:

```
<code>DELETE DATASET dset.</code>
```

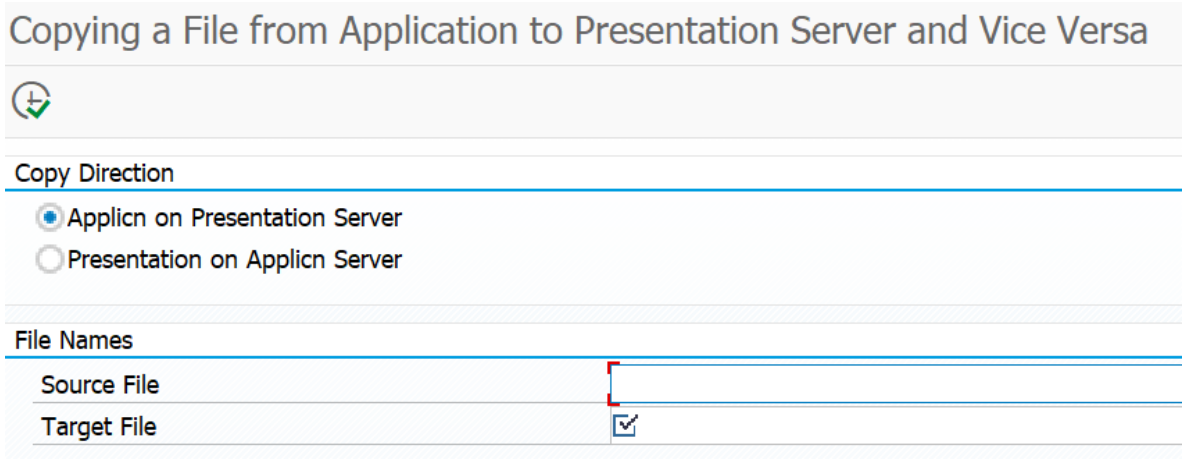
7. Standard SAP Report to upload and download files?

Shares

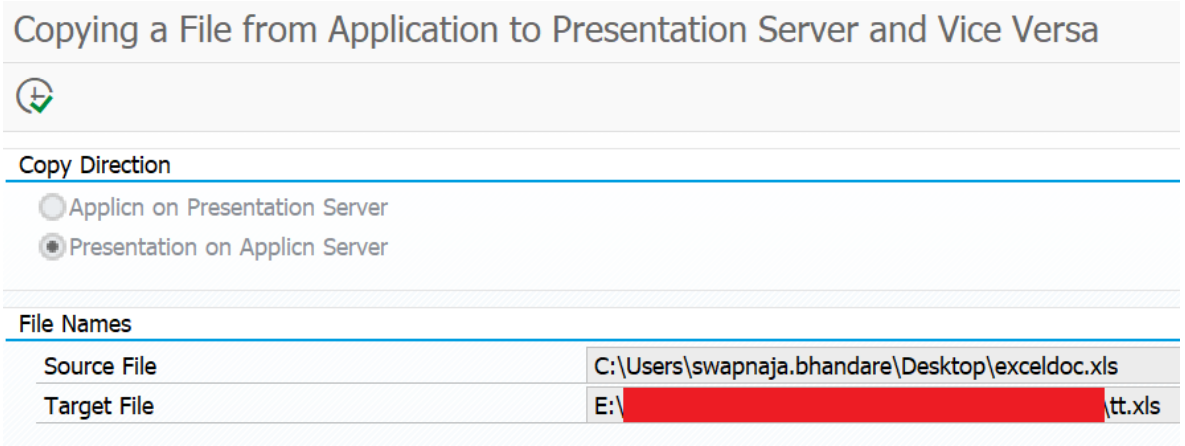
Yes, you read it right. SAP has already provided a report which is used for uploading file from presentation server to application server and vice versa. Awesome yeah ✌

Report name: **CACS_FILE_COPY**

When you execute this report, you will get below selection screen.

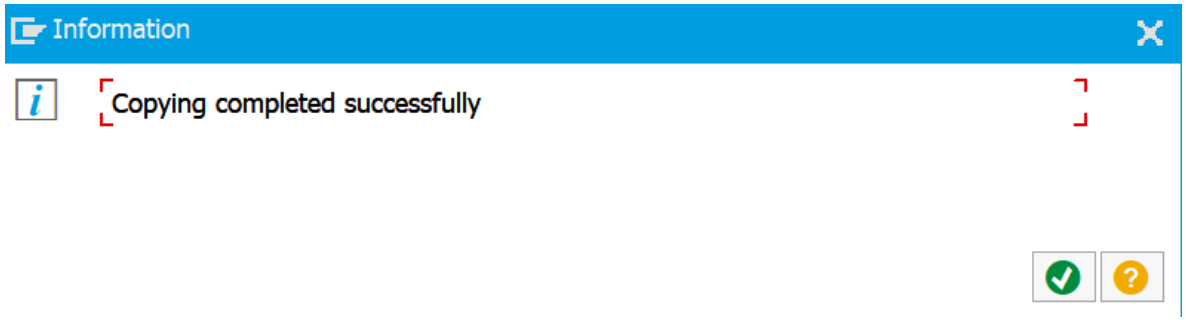


Now I wanted to store a file on application server from my desktop:

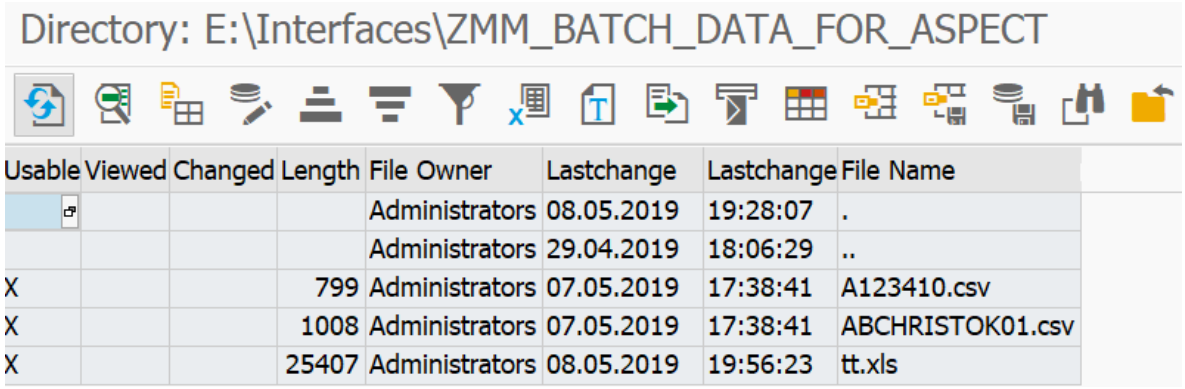


After successful execution, we can get below pop-up message:

Shares



Now let's check in AL11:



Directory: E:\Interfaces\ZMM_BATCH_DATA_FOR_ASPECT

Usable	Viewed	Changed	Length	File Owner	Lastchange	Lastchange	File Name
				Administrators	08.05.2019	19:28:07	.
				Administrators	29.04.2019	18:06:29	..
X			799	Administrators	07.05.2019	17:38:41	A123410.csv
X			1008	Administrators	07.05.2019	17:38:41	ABCHRISTOK01.csv
X			25407	Administrators	08.05.2019	19:56:23	tt.xls

Ohh yeah!!.. So easy and got the file too in AL11.

8. **Get all files from the directory/folder of application server?**

If you want to know what files are present in any AL11 folder, then you can use below 2 FMs:

EPS_GET_DIRECTORY_LISTING or **EPS2_GET_DIRECTORY_LISTING**

Let's see how it works.

Shares

Test Function Module: Initial Screen

Debugging Test data directory

Test for function group	EPSF
Function module	EPS2_GET_DIRECTORY_LISTING
Uppercase/Lowercase	
RFC target sys:	

Import parameters	Value
IV_DIR_NAME	
FILE_MASK	

Tables	Value
DIR_LIST	0 Entries

Here, we have to provide directory/folder path to get list of files present in it.

Shares

Test Function Module: Result Screen

Test for function group EPSF
Function module EPS2_GET_DIRECTORY_LISTING
Uppercase/Lowercase

Runtime: 27,310 Microseconds

RFC target sys:

Import parameters	Value
IV_DIR_NAME	E:\INTERFACES\ZMM_BATCH_DATA_FOR_ASPECT
FILE_MASK	

Export parameters	Value
DIR_NAME	
FILE_COUNTER	62
ERROR_COUNTER	0

Tables	Value
DIR_LIST	0 Entries
Result:	62 Entries

As you see in above snapshot, we got 62 files of our directory. If you check the **DIR_LIST** table, you will get below details:

Shares