Capgemini

ABAP Part II

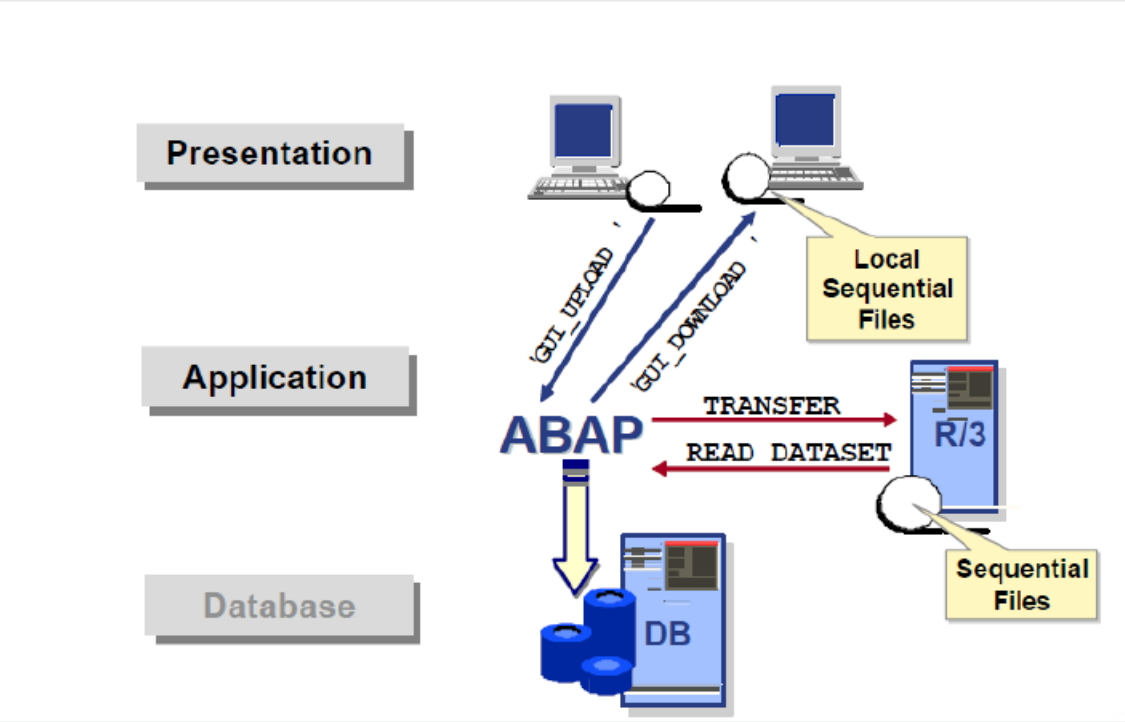
Lesson 04: File Handling

Lesson Objectives

After completing this lesson, participants will be able to know -

- File Handling Application Server
- File Handling Presentation Server

Introduction



Introduction

File is a place where information or data is stored. File handling in simple terms is opening, closing, reading, writing, deleting, copying, renaming the files.

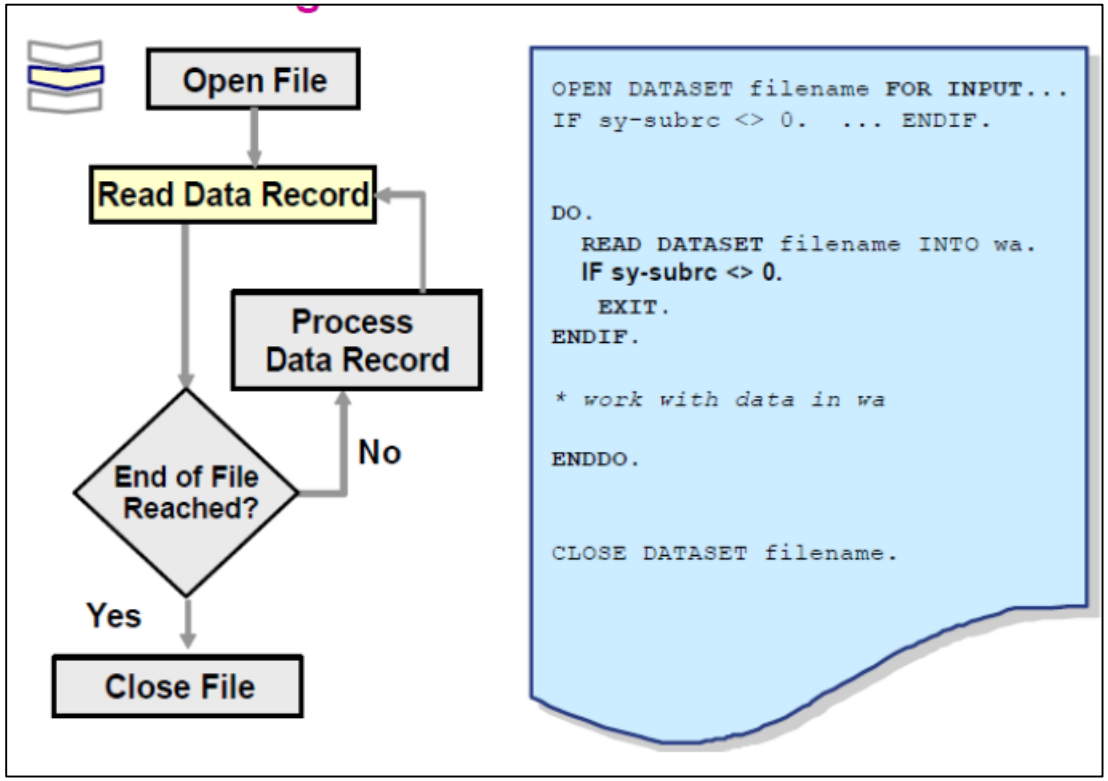
The runtime environment is implemented on the application server, which executes the ABAP programs. ABAP supports the file transfer (data transfer) technique to the application server and the front end hosts.

The interface to the file system on the application server is implemented in the form of ABAP language elements.

You can process sequential files using various file-handling methods and procedures to read data, process data and transfer it into SAP system. These files act as data source and these methods ensure consistency of the data in SAP R/3 system.

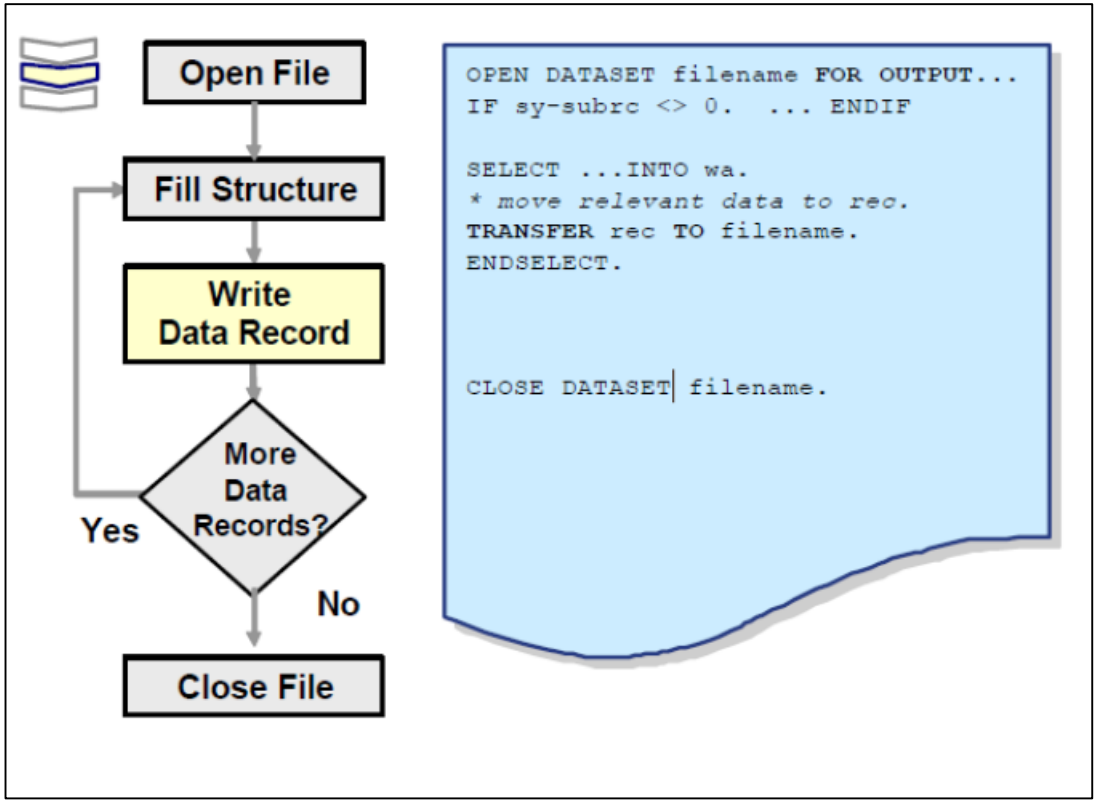
File Handling on Application Server

Overview Diagram – Read File



File Handling on Application Server

Overview Diagram – Write File



File Handling on Application Server

ABAP/4 provides five statements for handling files:

- OPEN DATASET
- CLOSE DATASET
- DELETE DATASET
- READ DATASET
- TRANSFER

File Handling on Application Server

Open Dataset

- Opens the specified file. If you do not use any additions, the file is opened for reading in binary mode. It returns SY-SUBRC = 0 if the file is opened successfully. Otherwise SY-SUBRC = 8.

Syntax

- OPEN DATASET <DSN> [Additions].

File Handling on Application Server

OPEN DATASET <DSN> FOR INPUT.

- This statement tries to open the field in 'read/update' mode (as long as the user has write authorization).If the user does not have write authorization, the system opens the file in 'read' mode. If this fails, an error occurs.

OPEN DATASET <DSN> FOR OUTPUT.

- This statement tries to open the file in 'write/update' mode as long as the user has read authorization. If the authorization is missing, the system opens the file in 'write' mode. If the file already exists, its existing content is deleted. If the file does not exist, the system creates it.

OPEN DATASET <DSN> FOR APPENDING.

- This statement tries to open the file in 'append' mode. If the file is already open, the system moves to the end of the file. When you open a file using FOR APPENDING, attempting to read the file sets SY-SUBRC to 4. The system display the end of the file.
- Note : You can only use one of the additions 1 to 3 in a single statement.

File Handling on Application Server

CLOSE DATASET

- Closes the specified file.
  - Syntax CLOSE DATASET <DSN>.

DELETE DATASET

- Deletes the file specified file. If it deletes the file successfully it returns SY-SUBRC = 0. Otherwise returns SY-SUBRC = 4. The possible reasons for failing are:
  - The file does not exist
  - The file is a directory
  - The file is a program that is currently running

File Handling on Application Server

READ DATASET

- Used to read a record from a file.

Syntax

- READ DATASET DSN INTO F.

Addition : LENGTH LEN.

- The actual length of the data objet read is placed in the field LEN after the read access. LEN must be defined as a variable. A syntax error will occur if you define it as a constant. The following example displays 9.

File Handling on Application Server

TRANSFER

- Used to write a record into a file.

Syntax

- TRANSFER F TO DSN.
  - Transfers the data object f to a sequential file whose name is specified in DSN. DSN can be a field or a literal. You must already have opened the file. . If the specified file is not already open, TRANSFER attempts to open the file FOR OUTPUT IN BINARY MODE. If this is not possible, a runtime error occurs f can be a field, a string, or a structure.

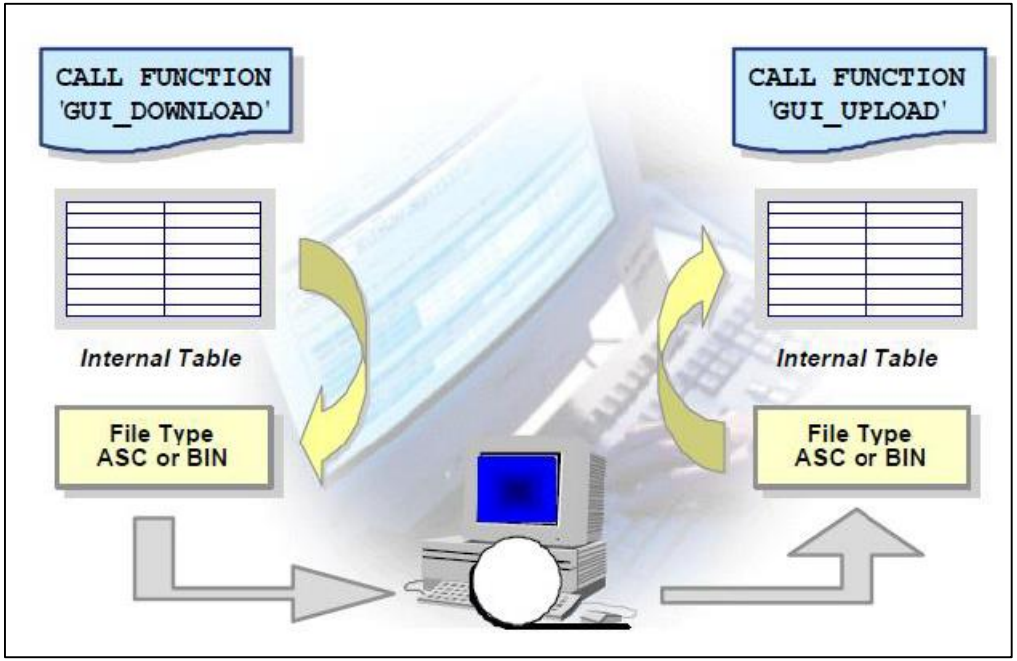
Addition : LENGTH LEN.

- The length of the data object to be written is defined by LEN, where LEN can be either a constant or a variable. If LEN is smaller than the length of the data object f, the system truncates character fields (C, N, D, T, X,P, STRING) on the right. With type I or F fields, unexpected results may occur if LEN is shorter than the default length for the field type

File Handling on Presentation Server

To work with files on the presentation server , SAP provides some special function modules GUI_UPLOAD, for reading from a file, and

GUI_DOWNLOAD, for writing into the file. An internal table must be used as an interface between the program and the function module



File Handling on Presentation Server

GUI Download - Signature

Writing data to a file on the presentation server:

To write data from an internal table to a file on the presentation server, use function module GUI_DOWNLOAD.

The most important parameters that are exported are as follows:

```
CALL FUNCTION 'GUI_DOWNLOAD'
  EXPORTING
*  BIN_FILESIZE
   filename                =
*  FILETYPE                = 'ASC'
*  APPEND                  = ' '
*  WRITE_FIELD_SEPARATOR   = ' '
*  HEADER                  = '00'
*  TRUNC_TRAILING_BLANKS   = ' '
*  WRITE_LF                = 'X'
*  COL_SELECT              = ' '
*  COL_SELECT_MASK         = ' '
*  DAT_MODE                = ' '
*  CONFIRM_OVERWRITE       = ' '
*  NO_AUTH_CHECK           = ' '
*  CODEPAGE                = ' '
*  IGNORE_CERR             = ABAP_TRUE
*  REPLACEMENT             = '#'
*  WRITE_BOM               = ' '
*  TRUNC_TRAILING_BLANKS_EOL = 'X'
*  WK1_N_FORMAT            = ' '
*  WK1_N_SIZE              = ' '
*  WK1_T_FORMAT            = ' '
*  WK1_T_SIZE              = ' '
*  WRITE_LF_AFTER_LAST_LINE = ABAP_TRUE
*  SHOW_TRANSFER_STATUS    = ABAP_TRUE
IMPORTING
*  FILELENGTH              =
TABLES
   data_tab                =
*  FIELDNAMES              =
*EXCEPTIONS
*  FILE_WRITE_ERROR        = 1
*  NO_BATCH                = 2
*  GUI_REFUSE_FILETRANSFER = 3
*  INVALID_TYPE            = 4
*  NO_AUTHORITY            = 5
*  UNKNOWN_ERROR           = 6
*  HEADER_NOT_ALLOWED      = 7
*  SEPARATOR_NOT_ALLOWED   = 8
*  FILESIZE_NOT_ALLOWED    = 9
*  HEADER_TOO_LONG         = 10
*  DP_ERROR_CREATE         = 11
*  DP_ERROR_SEND           = 12
*  DP_ERROR_WRITE          = 13
*  UNKNOWN_DP_ERROR        = 14
*  ACCESS_DENIED           = 15
*  DP_OUT_OF_MEMORY        = 16
*  DISK_FULL               = 17
*  DP_TIMEOUT              = 18
*  FILE_NOT_FOUND          = 19
*  DATAPROVIDER_EXCEPTION  = 20
*  CONTROL_FLUSH_ERROR     = 21
*  OTHERS                  = 22
    .
```

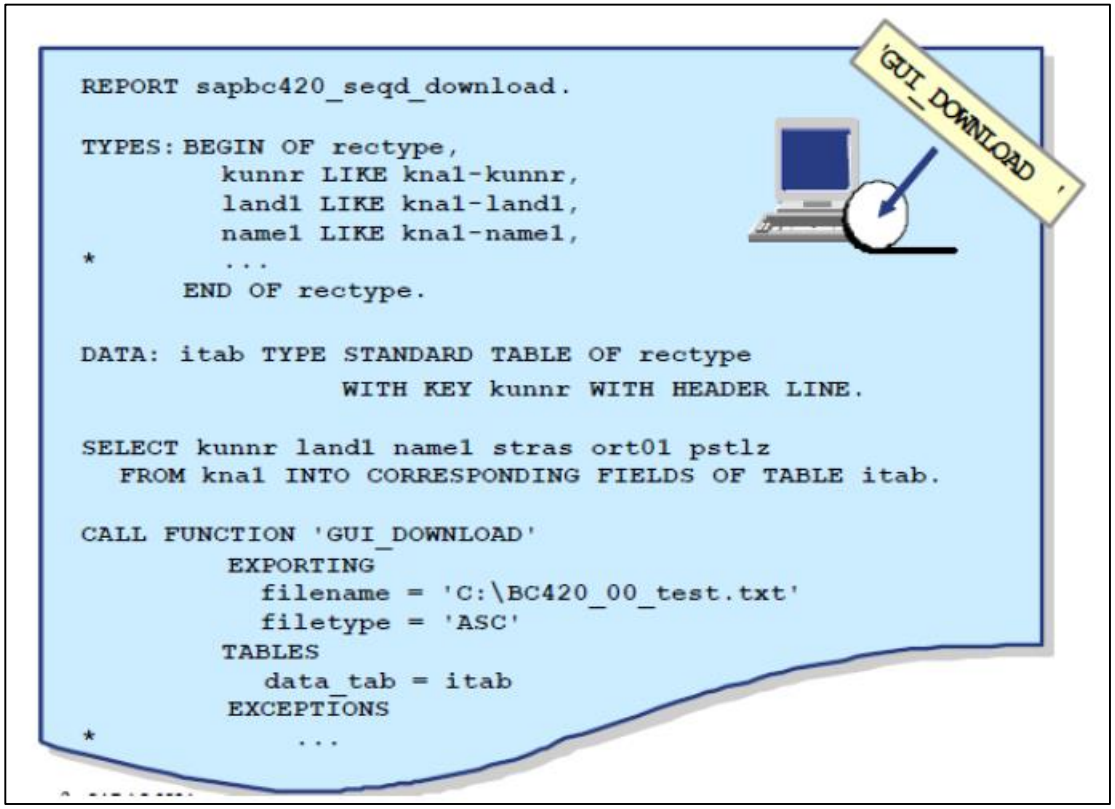
File Handling on Presentation Server

Some of the important parameters that are exported are as follows:

- BIN_FILESIZE - File Length for binary files. A length of zero or the length which is larger than the number of bytes in the internal table (width * number of lines) causes an exception.
- FILENAME - The name of the file that is to be generated on the presentation server(if necessary with predefined path name). If the path doesn’t exist or the file cannot be opened, an exception will be raised.
- APPEND - By default, existing local files are overwritten by new versions. By setting APPEND to 'X', the downloaded data is appended to an existing file. If the file does not yet exist, it is created.
- CONFIRM_OVERWRITE - If this parameter is set, a file is overwritten only after a confirmation by the user
- FILELENGTH - Number of bytes transferred
- Tables Parameter - DATA_TAB
  - The source internal table whose contents are downloaded into a file.

File Handling on Presentation Server

Example of GUI_DOWNLOAD



File Handling on Presentation Server

GUI Upload - Signature

- Reading data from a file on the presentation server:
- To read data from the presentation server into an internal table we use the function module GUI_UPLOAD. The most important parameters that are exported are as follows

```
CALL FUNCTION 'GUI_UPLOAD'
    EXPORTING
      filename                  =
    *  FILETYPE                  = 'ASC'
    *  HAS_FIELD_SEPARATOR       = ' '
    *  HEADER_LENGTH             = 0
    *  READ_BY_LINE              = 'X'
    *  DAT_MODE                  = ' '
    *  CODEPAGE                  = ' '
    *  IGNORE_CERR               = ABAP_TRUE
    *  REPLACEMENT               = '#'
    *  CHECK_BOM                 = ' '
    *  VIRUS_SCAN_PROFILE        =
    *  NO_AUTH_CHECK             = ' '
  IMPORTING
    *  FILELENGTH                =
    *  HEADER                    =
  TABLES
      data_tab                  =
```

File Handling on Presentation Server

GUI Upload - Signature

Some of the exporting parameters

- FILENAME - Name of the file
- FILETYPE - The source file type. Valid values are:
  - 'BIN' : Binary files.
  - 'ASC': ASCII files, text files with end-of-line markers.
  - 'DAT': The file is loaded line by line into the transferred table. Tabs in the file mean a change of field.
- HAS_FIELD_SEPARATOR: Specifies if the fields in the file are separated by a tab. This is necessary if the structure passed contains several components. CR/LF occurs instead of a tab after the last field of a row

File Handling on Presentation Server

GUI Upload - Signature

- Importing Parameter
  - FILELENGTH : Number of bytes transferred.
- Table parameter
  - DATA_TAB : Internal target table, to which the data is loaded.
- Exceptions
  - CONVERSION_ERROR - Errors in the data conversion.
  - FILE_OPEN_ERROR - System cannot open file.
  - FILE-READ_ERROR - System cannot read from file
  - INVALID_TABLE_WIDTH - Invalid table structure
  - INVALID_TYPE - Invalid value for parameter FILETYPE

File Handling on Presentation Server

GUI Upload - Example

```
REPORT sapbc420_seqd_upload.

TYPES: BEGIN OF rectype,
*         ...
       END OF rectype.

DATA: itab TYPE STANDARD TABLE OF rectype
                  WITH KEY kunnr WITH HEADER LINE,
      wa LIKE LINE OF itab.

CALL   FUNCTION 'GUI_UPLOAD'
          EXPORTING
            filename = 'C:\BC420_00_test.txt'
            filetype = 'ASC'
          TABLES
            data_tab = itab
          EXCEPTIONS
*         ...

LOOP AT itab INTO wa.
  WRITE: / wa-kunnr, wa-land1, ...
ENDLOOP.
```

File Archiving

Need: Many a times file cannot be deleted after processing is complete. There are certain implications due to government regulations, taxation (IRS) regulations, FDA requirements, internal organizational requirements, and audit requirements.

Method: As against standard SAP archiving way, one can archive processed file by moving it to a pre-defined folder on the application server. In future if need arise one can retrieve the file from archive folder.

Summary

In this lesson, you have learnt:

- Reading files from Presentation Server
- Reading files from Application Server

Review Question

Question 1: If the dataset is opened in write mode and the If the file already exists, its existing content is deleted.

- True/False

Question 2 A File can be opened for Output and Read at the same time.

- True/False