**Table of Contents**

[1 Introduction 3](#_Toc515882264)

[2 Installation 3](#_Toc515882265)

[3 SEC Invoices 4](#_Toc515882266)

[3.1 Remove empty lines 4](#_Toc515882267)

[3.2 Remove Header and Footer 4](#_Toc515882268)

[3.3 Remove Single value zero 4](#_Toc515882269)

[3.4 Form the table or blocks 5](#_Toc515882270)

[3.5 VAT, Partial VAT, and non-VAT type invoices 5](#_Toc515882271)

[3.6 KAU1 Branch Feeder 5](#_Toc515882272)

[3.6.1 VAT KAU1 Branch Feeder 5](#_Toc515882273)

[3.6.2 Partial VAT KAU1 Branch Feeder 6](#_Toc515882274)

[3.6.3 Non-VAT KAU1 Branch Feeder 7](#_Toc515882275)

[3.7 KAU1 Main Feeder 8](#_Toc515882276)

[3.7.1 VAT KAU1 Main Feeder 8](#_Toc515882277)

[3.7.2 Partial VAT KAU1 Main Feeder 10](#_Toc515882278)

[3.7.3 Non-VAT KAU1 Main Feeder 11](#_Toc515882279)

[3.8 KAU2 Main Feeder 12](#_Toc515882280)

[4 Launch 12](#_Toc515882281)

[4.1 Command Prompt 12](#_Toc515882282)

[4.2 Explorer 13](#_Toc515882283)

[4.2.1 File Open 14](#_Toc515882284)

[4.2.2 File Save 15](#_Toc515882285)

[5 Arabic to English Conversions 15](#_Toc515882286)

[5.1 Numerals 16](#_Toc515882287)

[5.2 Date 16](#_Toc515882288)

[5.3 Type 16](#_Toc515882289)

[5.4 Hijri to Georgian Conversion 16](#_Toc515882290)

# Introduction

The document discusses the SEC (Saudi Electric Company) invoice (energy consumption bill) parsing application. KAUST Facilities and Management team receives energy consumption bill through SEC. The energy consumption bill data is different for different types like industrial, commercial, charity, medical, residential, etc. usage. The energy consumption bill is more than 500 pages of PDF (Portable Document Format) document and mostly the data in Arabic language.

The document contains energy consumption data fields like subscription number, meter readings (to and from), meter reading dates (to and from), invoice number, invoice date, cost, value added tax, total cost, etc., for each type inside KAUST like villas, town houses, academic buildings, and other facilities. In the case of industry, the bill contains active power consumption, re-active power consumption etc. fields.

Storing energy consumption data to a digital form is very time consuming, error prone by entering the data values manually. This process is repetitive on every month. Once the data is stored in the digital form, different reports can be generated and viewed.

Develop an application that runs either in UI (User Interface) or command line mode. The application is aimed to develop a software package to digitize the energy consumption bills. The output will be a CSV (Comma Separated Value) file. The CSV file data contains the different energy consumption field’s data in English numerals.

The developed application will be packaged along with the necessary third party executables (in this case pdftotxt.exe (from XPDF)), DLLs (Dynamic Link Libraries), and system files. The application runs on the desktop that has Windows 7 or Windows 8 Operating Systems.

# Installation

The application is developed in QT and C++. The tool is internally uses pdftotxt.exe. The application is bundled in a zip file contains following sub folders:

* bin, contains the binary (executable) files and other related executables (QT)
* tools, contains pdftotxt.exe.
* temp, a directory used for temporary purpose.

The necessary application folder is stored at github and can be downloaded. Download the application folder from the github to the desired directory in a Windows computer, say C:\KAUST\sec.

Using Cmd (command prompt application) prompt user can go to C:\KAUST\sec\application\bin\ and then execute the [arabictranslate.exe](https://github.com/premchedella/saudielectric/blob/master/application/bin/arabictranslate.exe) application along with command line parameters. The command line parameters will be explained in the later section.

There is no need user to download any other files. All the required libraries or dlls or executables are bundled and placed in the application folder.

# SEC Invoices

KAUST receives SEC invoices in the form of .pdf file. The SEC invoice file (.pdf) has mainly two kinds of invoices. One invoice has many (more) fields and other invoice has less number of fields. The many fields invoice is called **bigger invoice** and less number of fields is called **smaller invoice**.

Most of the times all bigger invoices are in one file, all smaller invoices in another file. Sometimes the smaller and bigger invoices are available in the same file.

The earlier executable takes type (KAU 1 or 2) an input parameter which says whether the input file is bigger or smaller invoices file. Based on this input, the parser parses and generates the data accordingly. Whenever the input file contains both smaller and bigger invoices, the executable is not able to generate all the invoices since there is a mismatch for the input parameter, either it generates all the bigger invoices with **Complete** and all the smaller invoices with **Partial** data or it generates all the smaller invoices with **Complete** and all the bigger invoices with **Partial** data. To solve this problem, a new kind of parser is required.

Each invoice start with **الاسم** phrase. Whenever the line starts with this phrase then it is considered a block or invoice.

To differentiate between smaller and bigger invoices, **: طاقة غير فاعلة)** phrased to used. If the invoice contains this phrase then invoice is called bigger invoice otherwise the invoice is called smaller invoice. In the case of bigger invoices, most of the cases the differentiated phrase is available in the last 6th line. Whenever this phrase is found in the last 6 line then the corresponding invoice is called bigger invoice otherwise it is called smaller invoice.

# Execution

Generate the invoices data with following options:

## Option 1

Following is the command line instruction:

**arabictranslate.exe 7 2021 D:\KAUST\projects\tests\misc\saudi\_electric\jul\_2021\KAU2\_18\_Feeders\_Jul\_21.pdf**

Following are the parameters:

1. Invoice Month
2. Invoice Year
3. Invoice file (with .pdf)

The application generates output file(s) automatically in the same folder where the input .pdf file exists.

Assume that name of the input file is **KAU\_Main.pdf**. If the input invoice file contains only bigger invoices then it generates **KAU\_Main\_bigger.csv**. If the input invoice file contains only smaller invoices then it generates **KAU\_Main\_smaller.csv**. If the input invoice file contains both bigger and smaller invoices then it generates both **KAU\_Main\_bigger.csv** and **KAU\_Main\_smaller.csv** files.

Also, there is a need to generate the summary of the file with the file name **KAU\_Main\_summary.csv** irrespective of the invoices type, which means this file is always generated.

## Option 2

Following is the command line instruction:

**arabictranslate.exe 7 2021 D:\KAUST\projects\tests\misc\saudi\_electric\jul\_2021\KAU2\_18\_Feeders\_Jul\_21.pdf**

Following are the parameters:

1. Invoice Month
2. Invoice Year
3. Invoice file (with .pdf)

The application generates only one output file in the same folder where the input .pdf file exists.

Assume that name of the input file is **KAU\_Main.pdf**. The output file name is **KAU\_Main.xlsx.**  The first tab/worksheet in the .xlsx contains **summary** of the invoice file, the second tab/worksheet contains the **bigger invoices** data if bigger invoices present in the invoice pdf file, and the third tab/worksheet contains the **smaller invoices** data if smaller invoices present in the invoice pdf file.

Option 1 requires less changes in the parser and front end when compare to option 2.

# Summary

The invoices pdf file first page contains the summary of all individual invoices. There is a need to generate summary information also.

Following is the first page of the invoice file.

Need to know which fields needs to write and the corresponding English name for the same.

