SEC Invoice Parser

**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)
* stylesheet, styles used for application, used in the case of User Interface
* 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. Go to *C:\KAUST\sec\application\bin\* folder and double click on the [arabictranslate.exe](https://github.com/premchedella/saudielectric/blob/master/application/bin/arabictranslate.exe) file, application will be launched. Different controlling options and menu options will be explained later sections.

Using Cmd 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 three SEC invoices, viz., KAU1 Branch Feeder, KAU1 Main Feeder, and KAU2 Main Feeder. KAU1 Branch Feeder is bigger in file size than KAU1 Main Feeder and KAU2 Main Feeder.

The SEC invoices are in PDF file format, the parser first converts the PDF file to text data using pdftotxt.exe. Following is the command:

pdftotext.exe -layout -table -f 2 -enc UTF-8 input\_file output\_file

The output file content is stored in the temp folder of application as data.txt. In the above command, *layout* means maintain (as best as possible) the original physical layout of the text, *table* means similar to physical layout option, but optimized for tabular data, with the goal of keeping rows and columns aligned (at the expense of inserting extra whitespace), *f* specifies the first page to convert (2 means 2nd page onwards since 1st page does not contain the valid data fields information), *enc* means encoding type, which is of *UTF-8*, *input\_file* is PDF file and *output\_file* is text file. The text file data is copied to strings data structure, each line of text file represents a row of strings.

## Remove empty lines

The generated text file contains as many as white or empty lines. If the size of the line is 0, which it is an empty line then remove the same from the data.

## Remove Header and Footer

The invoice has header and footer in each page. Each page information in the converted text file starts with the page break “\f” (=0x12). The header is of 4 lines and footer is of one line. Whenever page break found, remove the previous footer (previous page) and header of 4 lines (current page).

## Remove Single value zero

The invoice also has single value zero. There is no use of this zero while subsequent parsing. If the length or size of the line is 1 means it contains zero then remove the same from the strings list.

## Form the table or blocks

Once empty lines, header and footer and single value zeros are removed from the original strings list, check whether line starts with **ﺭﻗﻢ** then a new meter reading starts, copy each line of data from this row to block until it finds another start, whenever another start is found, the block closes. This way the blocks are formed. For the last block, it contains more data as invoice contains more data. Therefore, in the last block, whenever the line starts with **ﻣﺠﻤﻮﻉ**, removes the all others lines including that line from the block. Means all blocks with the uniform block length, but there could be mis-matches because the original invoice itself does not contain.

## VAT, Partial VAT, and non-VAT type invoices

SEC introduced VAT (Value Added Tax) charges from January 2018 onwards; earlier invoices do not contain VAT charges. January 2018 invoice also contains partial VAT means reading from 1st January 2018 and no VAT charge before 1st January 2018. Therefore for understating purpose, the invoices are categorized into 3 different types.

* VAT invoices, the invoices after January 2018 onwards
* Partial VAT Invoices, only for January 2018 invoices
* Non-VAT Invoices, before January 2018 invoices

## KAU1 Branch Feeder

KAU1 Branch Feeder has three different categories of invoices as explained earlier, VAT, Partial VAT, and non-VAT.

### VAT KAU1 Branch Feeder

Following snippet shows the KAU1 Branch Feeder Type1 table or block.

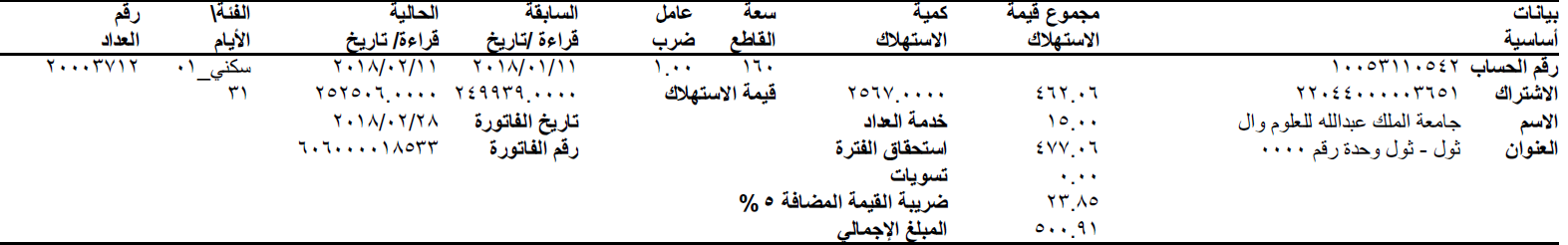


Figure ‑ KAU1 Branch Feeder VAT Type 1 Block

Following snippet shows the KAU1 Branch Feeder Type 5 table or block

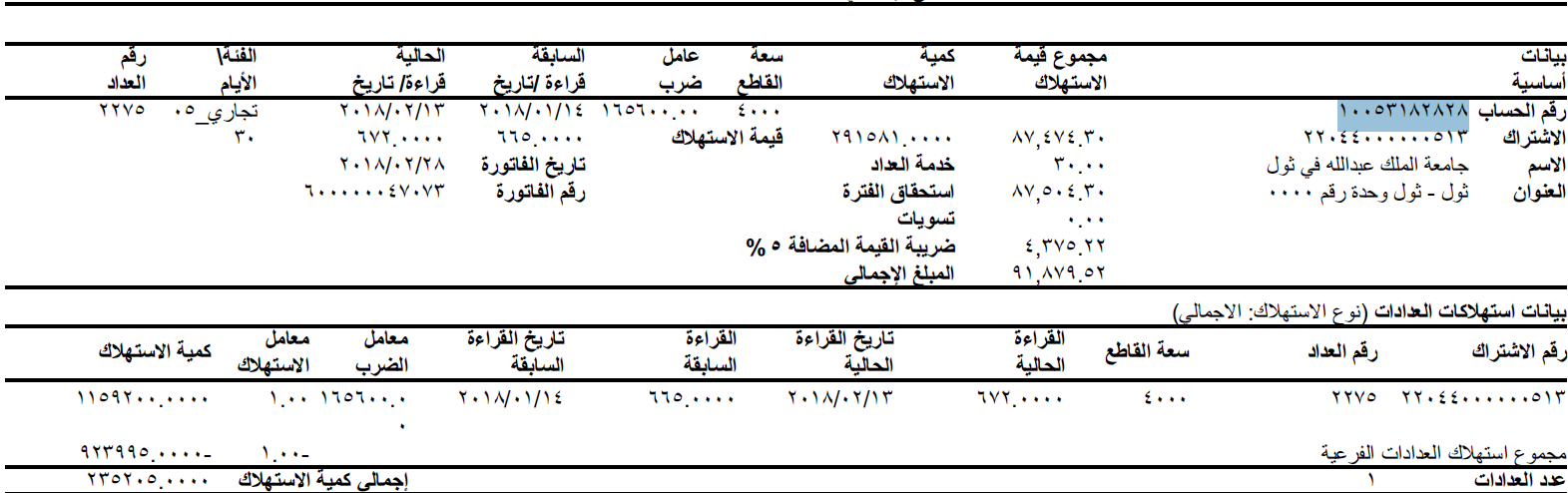


Figure ‑ KAU1 Branch Feeder VAT Type 5 Block

It is assumed that both Type1 and Type 5 fields match in the case of KAU1 Branch Feeder VAT invoices.

Line 3 holds Meter Number at position 0, Type ID and Type at position 1, Reading To (date) at position 2, Reading From (date) at position 3, Multiplication Factor at position 4, Capacity at position 5 and Account Number at position 8. Sometimes the Account Number string and values are interchanged, that time Account Number is positioned at 6. The Type is in Arabic string after \_, the Arabic string is converted to the corresponding English string.

Line 4 contains Number of Reading Days at position 0, Current Meter Reading at position 1, Previous Meter Reading at position 2, Power Consumption at position 5, Power Cost at position 6, and Subscription Number at position 7.

Line 5 has Invoice Date at position 0 and Meter Service at position 5. Line 6 holds Invoice Number at position 0 and duration cost at position 5. Line 7 contains only Settlement at position 1. Line 8 holds only VAT at position 4 or position 5. Line 9 has Total Cost at position 2.

If any one of the fields (Total Cost, Invoice Date, etc.) are not in that position then flag is raised for the whole block as data is **Partial** means user has to add manually if required. If all the fields match with those positions then conversion is **complete**.

### Partial VAT KAU1 Branch Feeder

Following are the snippet of Type 1 and Type 5 of Partial VAT KAU1 Branch Feeder

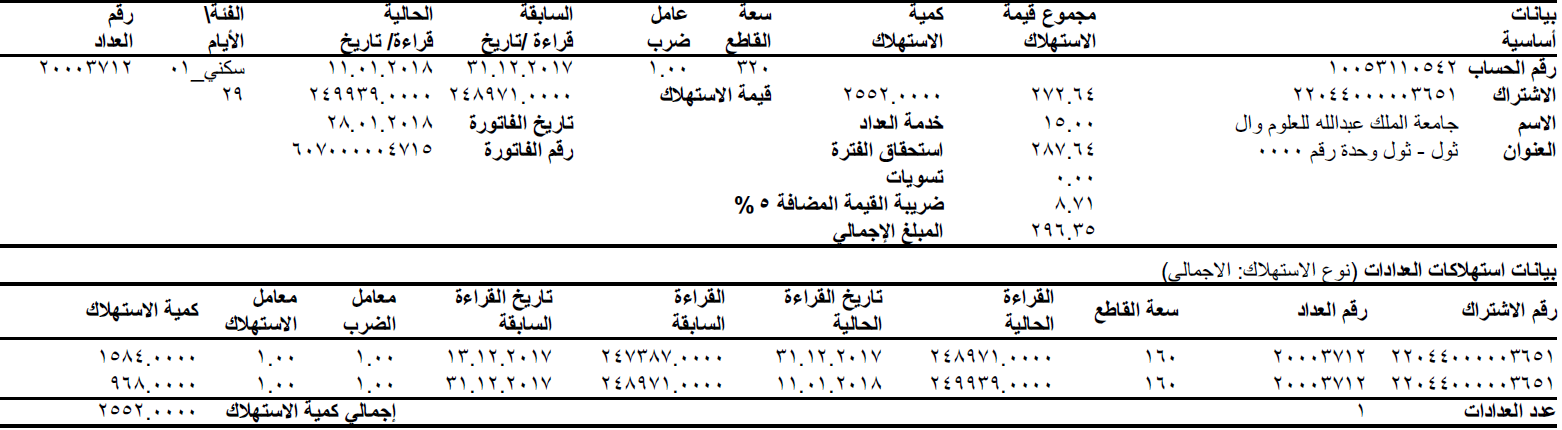


Figure ‑ KAU1 Branch Feeder Partial VAT Type 1 Block

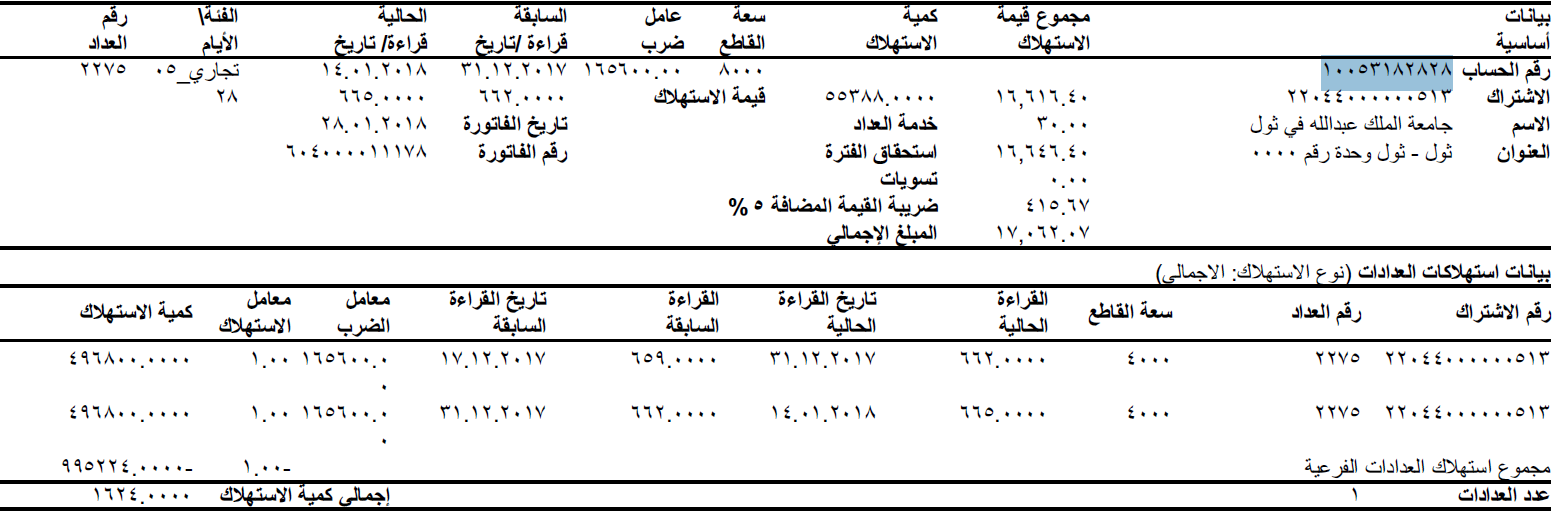


Figure ‑ KAU1 Branch Feeder Partial VAT Type 5 Block

Most of the fields are similar to VAT KAU1 Branch Feeder. Some cases the 5% VAT is not present in the invoice, those cases, the VAT field is kept as **blank**. In the case of Date (Previous Meter Reading Date, Current Meter Reading Date, and Invoice Date) field, format is dd.mm.yyyy, it is not same as VAT KAU1 Branch Feeder, convert the Date to dd/mm/yyyy.

### Non-VAT KAU1 Branch Feeder

Following are the snippet of Type 1 and Type 5 of non-VAT KAU1 Branch Feeder

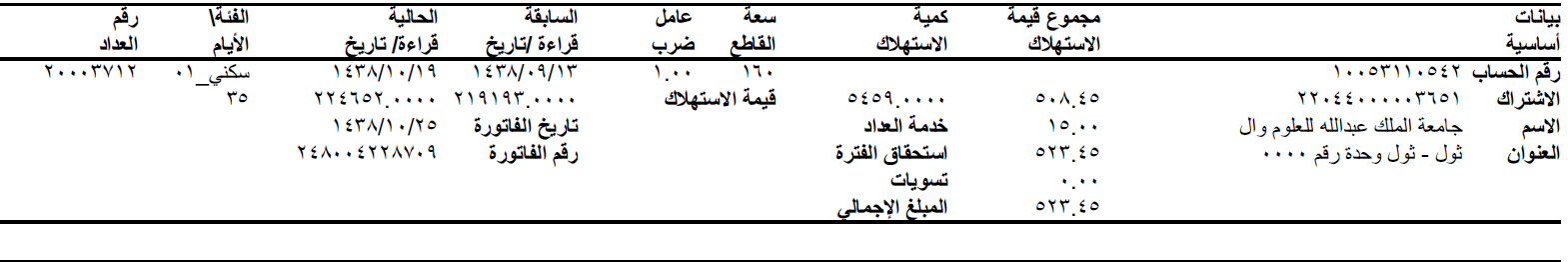


Figure ‑ KAU1 Branch Feeder Non-VAT Type 1 Block

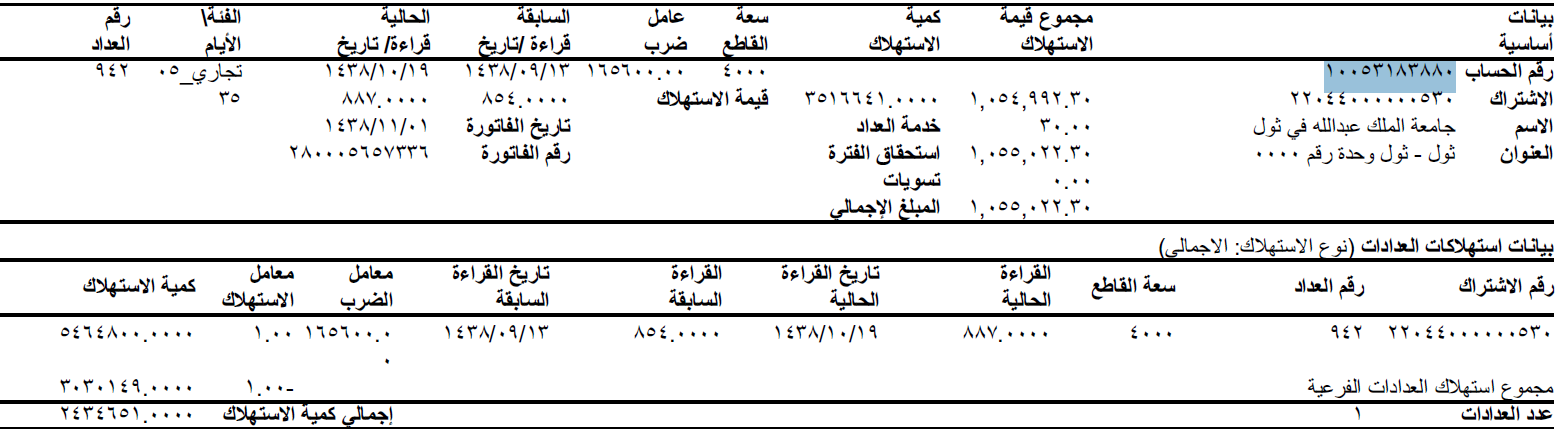


Figure ‑ KAU1 Branch Feeder Non-VAT Type 5 Block

Most of the fields are similar to VAT KAU1 Branch Feeder except VAT field. The blocks does not contain VAT field.

The Date is in Hijri and converted to Georgian Calendar date.

## KAU1 Main Feeder

KAU1 Main Feeder has three different categories of invoices as explained earlier, VAT, Partial VAT, and non-VAT. It contains Type 12 and Type 1 blocks. The Type 1 block is different from KAU1 Branch feeder.

### VAT KAU1 Main Feeder

Following is the snippet of VAT KAU1 Main Feeder for Type 12

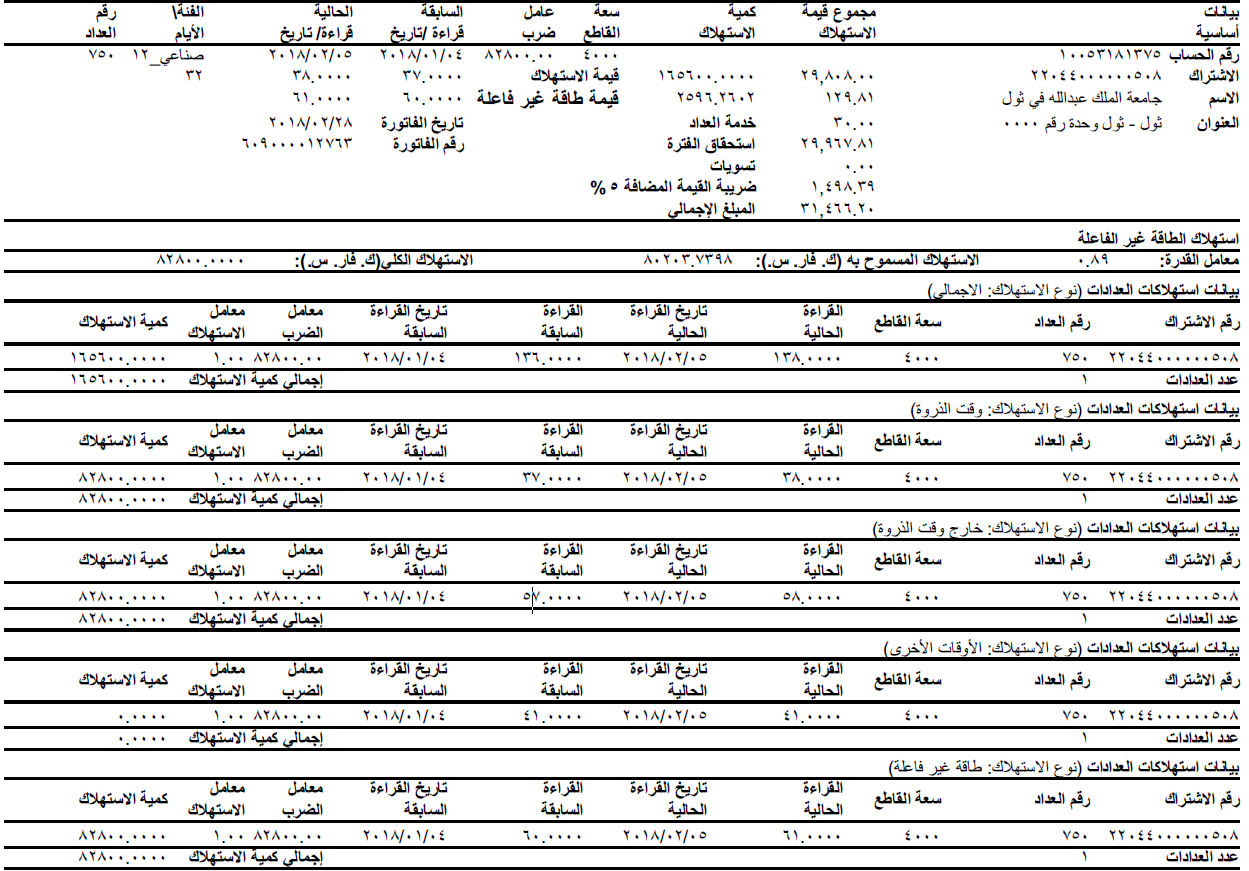


Figure ‑ KAU1 Main Feeder VAT Type 12 Block

Line 3 holds Meter Number at position 0, Type ID and Type at position 1, Reading To (date) at position 2, Reading From (date) at position 3, multiplication factor at position 4, Capacity at position 5 and Account Number at position 8. Sometimes the Account Number string and values are interchanged, that time Account Number position is at 6. The Type is in Arabic string after \_, the Arabic string is converted to the corresponding English string.

Line 4 contains Number of Reading Days at position 0, Active Power Consumption at position 5, Active Power Cost at position 6, and Subscription Number at position 7.

Line 5 has Charged Reactive Power Consumption at position 6 and Reactive Power Cost at position 7. Line 6 has Invoice Date at position 0 and Meter Service at position 5. Line 7 holds Invoice Number at position 0 and Duration Cost at position 5. Line 8 contains only Settlement at position 1. Line 9 holds only VAT at position 4 or position 5. Line 10 has Total Cost at position 2.

Line 11 contains Total Reactive Power Consumption at position 0, Allowed Reactive Power Consumption at position 5, and Power Factor at position 12. Line 15 contains Current Reading at position 4 and Previous Reading at position 6.

Line 35 contains Total Reactive Power Consumption at Position 0, Reactive Power Multiplication Factor at Position 4, Reactive Power Previous Reading at position 4, and Reactive Power Current Reading at position 6.

Following is the snippet of VAT KAU1 Main Feeder for Type 1

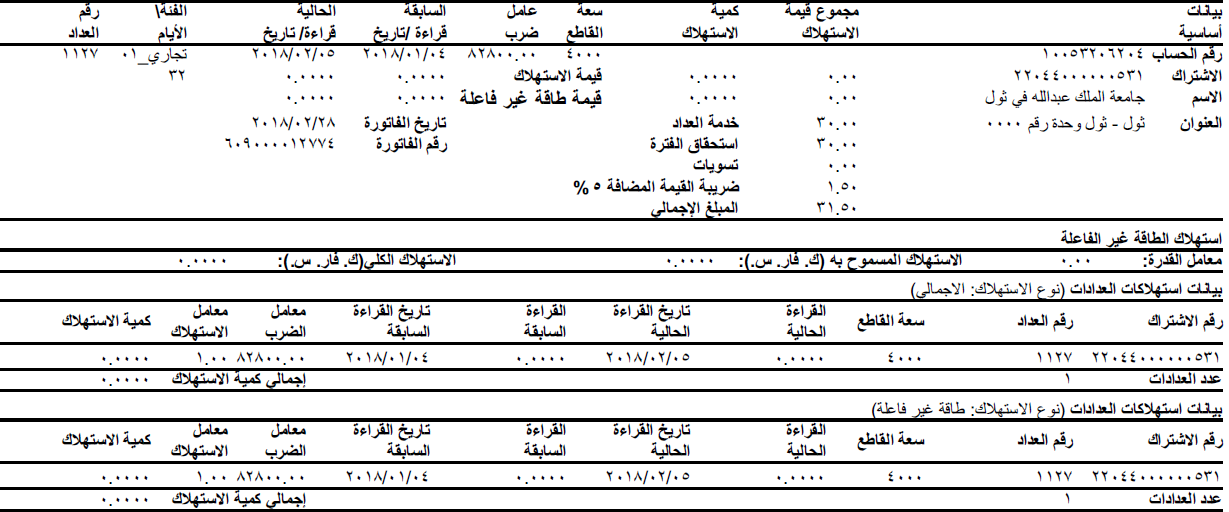


Figure ‑ KAU1 Main Feeder VAT Type 1 Block

Line 3 holds Meter Number at position 0, Type ID and Type at position 1, Reading To (date) at position 2, Reading From (date) at position 3, multiplication factor at position 4, Capacity at position 5 and Account Number at position 8. Sometimes the Account Number string and values are interchanged, that time Account Number position is at 6. The Type is in Arabic string after \_, the Arabic string is converted to the corresponding English string.

Line 4 contains Number of Reading Days at position 0, Active Power Consumption at position 5, Active Power Cost at position 6, and Subscription Number at position 7.

Line 5 has Charged Reactive Power Consumption at position 6 and Reactive Power cost at position 7. Line 6 has Invoice Date at position 0 and Meter Service at position 5. Line 7 holds Invoice Number at position 0 and Duration Cost at position 5. Line 8 contains only Settlement at position 1. Line 9 holds only VAT at position 4 or position 5. Line 10 has Total Cost at position 2.

Line 11 contains Total Reactive Power Consumption at position 0, Allowed Reactive Power Consumption at position 5, and Power Factor at position 12. Line 15 contains Current Reading at position 4 and Previous Reading at position 6.

Line 20 contains Total Reactive Power Consumption at position 0, Reactive Power Multiplication Factor at Position 4, Reactive Power Previous Reading at position 4, and Reactive Power Current Reading at position 6.

### Partial VAT KAU1 Main Feeder

Following is the snippet of Type12 Partial VAT KAU1 Main Feeder

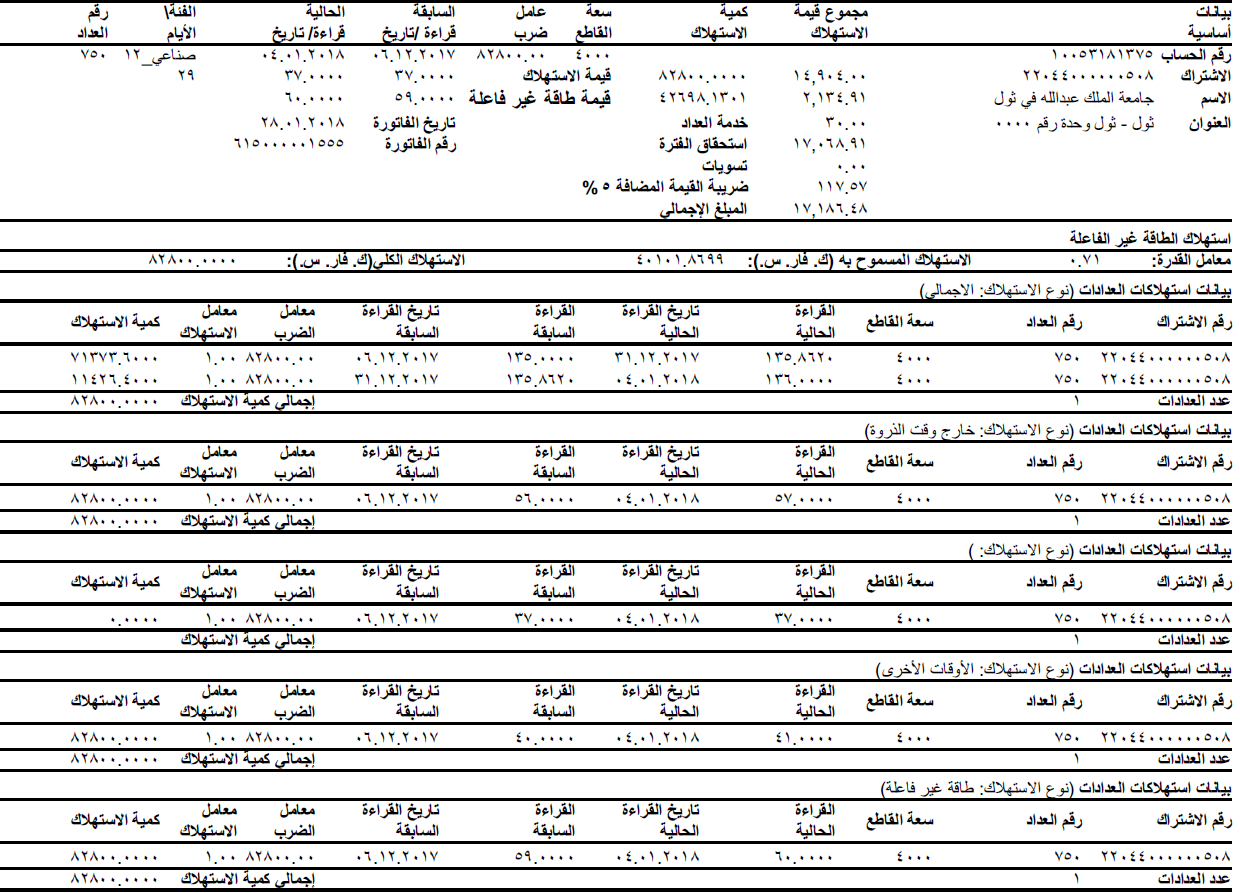


Figure ‑ KAU1 Main Feeder Partial VAT Type 12 Block

Parsing of KAU1 Main Feeder Partial VAT Type 12 Block is same KAU1 Main Feeder VAT Type 12 except for Previous Reading and Current Reading. Line 15 contains Current Reading at position 4 and Line 16 contains Previous Reading at position 6.

Line 36 contains Total Reactive Power Consumption at position 0, Reactive Power Multiplication Factor at Position 4, Reactive Power Previous Reading at position 4, and Reactive Power Current Reading at position 6.

In the case of Date (Previous Meter Reading Date, Current Meter Reading Date, and Invoice Date) field, format is dd.mm.yyyy, it is not same as VAT KAU1 Branch Feeder, convert the Date to dd/mm/yyyy.

Following is the snippet of Type1 Partial VAT KAU1 Main Feeder



Figure ‑ KAU1 Main Feeder Partial VAT Type 1 Block

Parsing of KAU1 Main Feeder Partial VAT Type 1 Block is same KAU1 Main Feeder VAT Type 1 except for Previous Reading and Current Reading. Line 15 contains Current Reading at position 4 and Line 16 contains Previous Reading at position 6.

Line 21 contains Total Reactive Power Consumption at position 0, Reactive Power Multiplication Factor at Position 4, Reactive Power Previous Reading at position 4, and Reactive Power Current Reading at position 6.

In the case of Date (Previous Meter Reading Date, Current Meter Reading Date, and Invoice Date) field, format is dd.mm.yyyy, it is not same as VAT KAU1 Branch Feeder, convert the Date to dd/mm/yyyy.

### Non-VAT KAU1 Main Feeder

Following is the snippet of Non-VAT KAU1 Main Feeder Type 12. Parsing of KAU1 Main Feeder non-VAT Type 12 is same VAT Type 12 except VAT. There is no VAT field.

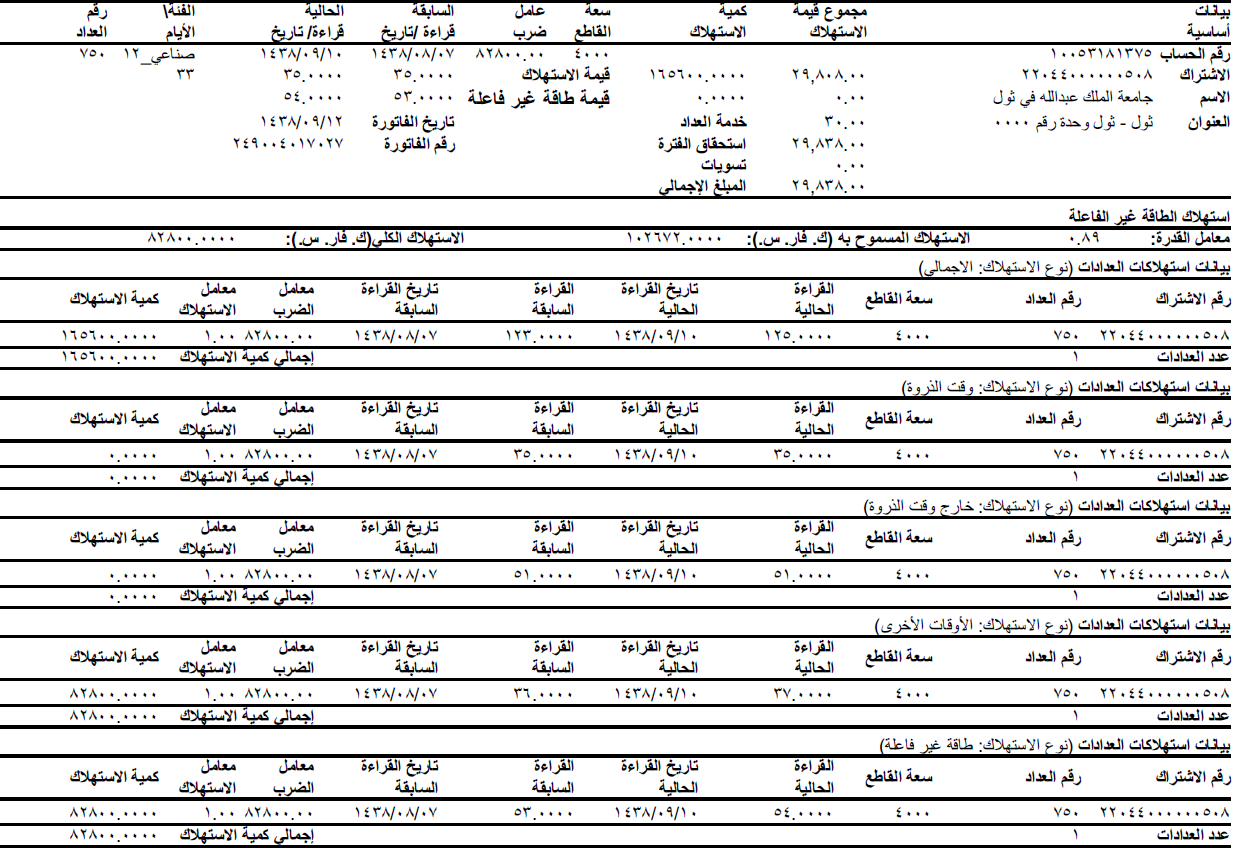


Figure ‑ KAU1 Main Feeder Non VAT Type 12 Block

Parsing of KAU1 Main Feeder non-VAT Type 1 is same VAT Type 1 except VAT. There is no VAT field.

The Date is in Hijri and converted to Georgian Calendar date.

## KAU2 Main Feeder

The KAU2 Main Feeder invoice is similar to KAU1 Main Feeder invoice and contains only Type12. KAU1 Main Feeder is re-used here.

# Launch

The application can be launched in two ways.

## Command Prompt

Follow the following steps:

1. Open the command prompt from the Start button of the Windows PC or Desktop or Laptop.
2. Navigate to the folder where the application has been copied during the installation as mentioned in section 2.
3. Navigate to the bin directory, let us say user is in *C:\KAUST\sec\application\bin\*
4. Type arabictranslate.exe and command line parameters and then press enter key.

Following are the command line parameters. Each parameter is separated by a space.

arabictranslate.exe file\_type month year input\_file output\_file

file\_type is the either one of KAU1 Main Feeder (0), KAU2 Main Feeder (1), or KAU1 Branch Feeder (2). It is an integer value (0, 1, and 2).

month is January (0), February (1), etc., and hold value between 0 to 11 including. It’s an integer value, (0, 1, 2, .. 11).

year is an integer value of 4 digits.

input\_file, is the input file for this application with the extension .pdf

output\_file, is the output file with the extension .csv.

## Explorer

Navigate to the folder of bin in Windows Explorer (or any file manager) and double click the executable file arabictranslate.exe.

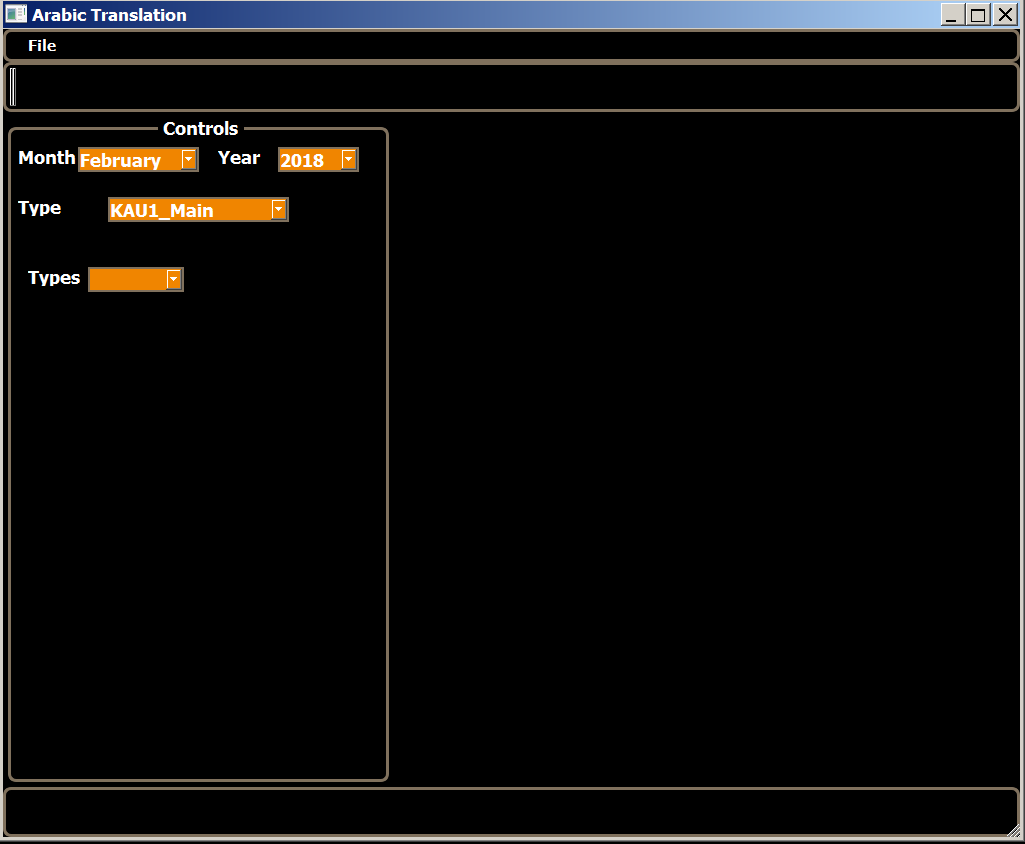


Figure ‑ Application Initial Screen

### File Open

User has select Month, Year, and File Type appropriately otherwise results may not be correct. Once these are selected then File->Open, open dialogue box is displayed, user has select the correct pdf file based on the month, year and file type. Whenever user selected the open file then application process the pdf file and displays the Subscription Number on the right side as shown in the following figure.

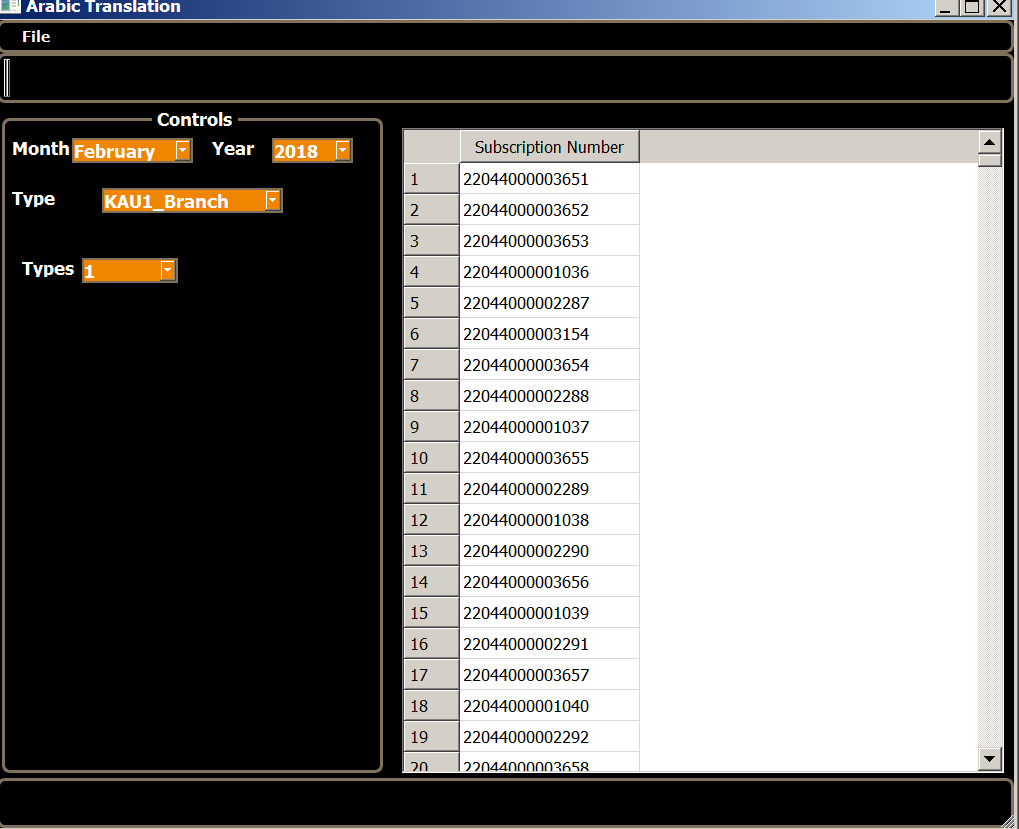


Figure ‑ Application after processing the pdf file

### File Save

User can select different types using combo box, Types. The data fields can be saved in the .csv file by selecting File->Save As Csv option. If the Type is KAU1\_Branch then all the necessary fields are written into .csv file, means each value is separate by “,” (comma). If the Type is KAU1\_Main or KAU2\_Main many fields are written into the .csv file with each value is separated by “,” (comma).

# Arabic to English Conversions

Most of the field’s data is in Arabic and need to convert this Arabic data to the corresponding English values. Following are some of the utilities:

## Numerals

Most of the digits are in numerals. Whenever numeral string is found like Account Number, Invoice Number, etc., take each Arabic character (1632 to 1641) to the corresponding English values 0 to 9.

For Date field, include character (47) in the corresponding string as “/”. For fields like Settlement, Duration Cost, etc., include character (46) in the corresponding string as “.”.

## Date

Date field may be dd/mm/yyyy or mm/dd/yyyy, convert either type of the date to uniform dd/mm/yyyy.

## Type

Invoices include Type and Type ID in Arabic string. First those needs be supported. Once Type Id is separated, the Arabic string of Type needs to be converted to the corresponding English strings. The different English Type strings are Industrial, Residential, Commercial, Medical, Charity, and Educational.

## Hijri to Georgian Conversion

For Non-VAT invoices the date is in Hijri and converted this Hijri date to Georgian calendar date and date is in format dd/mm/yyyy.