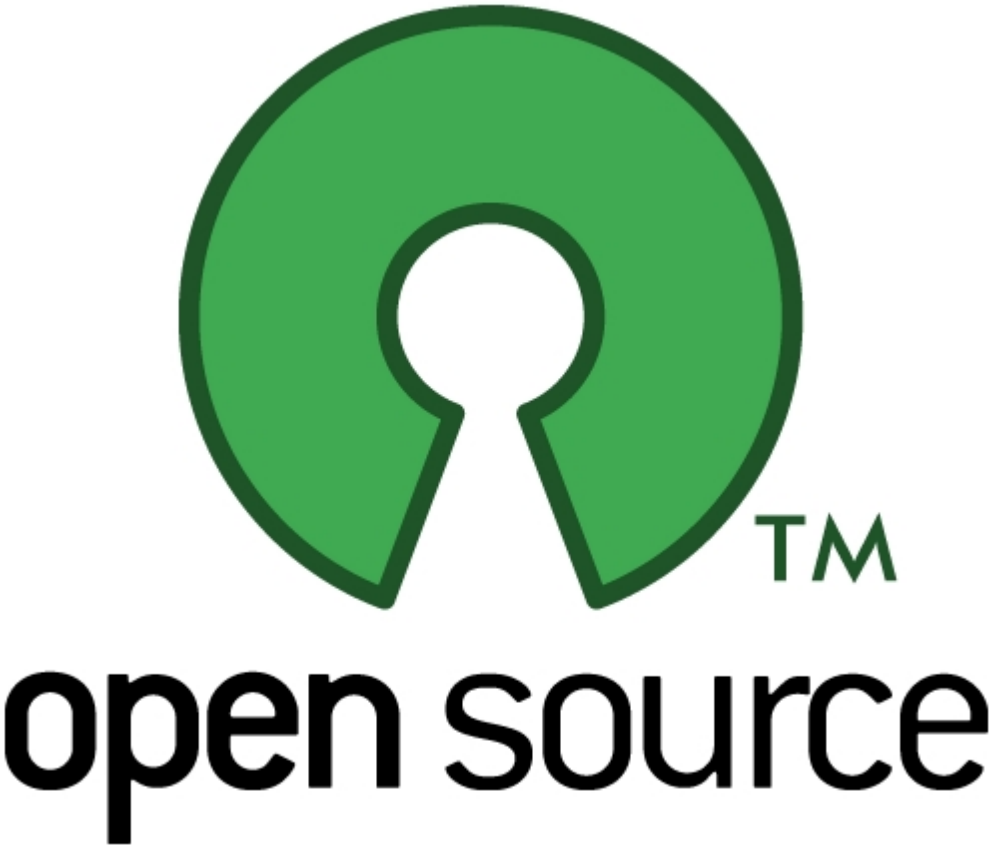
# Kaylen's Library Documentation

# Table of contents

Introduction ..... 3
Welcome ..... 3
Getting Started ..... 4
KTPALGO ..... 4
KTPDatabase ..... 5
Whats next? ..... 5

# Introduction




Initially Developed by: Kaylen Travis Pillay (2017)



## Welcome

Welcome! This library was born from the need of generic methods that would help the development of a final year computer science project. Throughout
the process of the creation of this library an effort was made to keep it as generic as possible, while still having a purpose in the game my group and I
developed.

I have, and will never have the intention on keeping this library closed source and therefore invite you to make it your own! My hope is that I could make
your development life that much more simple, and all I ask is that you improve on it and allow others to use your improvements. The library is very general
at the moment and in the future I would like to come up with the following libraries:

KTPGraphics Library for C++
KTPNetworking Library for C++
KTPGame Library for C++

So stay tuned!

If you'd like to get a hold of me for any reason [ if (!computer science related) ignore ] then you can get me

at info@ktpsolutions.co.za

Thank you and enjoy!



# Getting Started

The following is a brief description of the functions and Classes included in the library



## KTPALGO

### Class KTPAlgo

**Static Functions:**

- **void TestLibraryLinking()**
  This function helps to test if the library has been correctly linked to your C++ project. It displays an appropriate output to the standard output cout (iostream).

- **ktp::TextFileContainer_VEC LoadTextFileData(ktp::FString** filename**)**
  This function collects the data from a text file and places in an easy to use vector. ktp::TextFileContainer_VEC is a standard STL vector.

- **void AutoTypeText(ktp::FString** text**, ktp::UInt** speed**)**
  This function auto types the text you give it at the speed you set it to. This gives the illusion of the computer writing the text to the console one character at a time.

- **void ClearConsole()**
  This function clears the console screen. It works for MAC, LINUX and WINDOWS operating systems.

- **template<class** E**> void GetValidInput(E** e**)**
  This template function gets valid input from the user and places the information into the variable provided. STANDARD INPUT AND OUTPUT are used here. This is a template function and will work with all class types assuming that the '>>' and '<<' operators have been written for those classes.

- **ktp::FString GetIPv4Address_WIN()**
  This function gets the IPv4 address of your wifi adapter to the current network that you are connected to at the moment.

- **template<typename** Key**, typename** Value**>**
  **std::vector< std::pair <**Key, Value**>> ShuffleMap(std::Map<**Key, Value**>** map**)**

  This method shuffles a map and returns a vector of map entries (pairs!). This would allow the user to have access to all the entries in a map which has been shuffled

# KTPDatabase

## Class KTPDatabase

```
          The following class helps to encapsulate a basic database.
     CSV files are
          an inefficient way of store small chunks of data from a
program, however
          it can be a challenge to read from and update information.
This class seeks
          to encapsulate all those tasks into simple method calls
that guarantee to
          complete there goals.
```

- **KTP_Database()**
  KTP_Database constructor. This constructor creates the databases core structure which will allow the user of this database to freely load tables of different data types.

- **ktp::DataBaseTable*** **GetDataBaseTable(ktp::TableName** tablename**)**
  GetDatabaseTable. This returns a pointer to the database table with the specified name.

- **ktp::Flag LoadTableInto_DB(ktp::TableName** tablename, **ktp::Col_LEN** columnLength, **ktp::FString** csvFileName **)**
  Loads a table into the database. Note that all the data will be stored as STRING data-type and is left to the user to convert were necessary.

- **ktp::Flag SaveDataBaseTable(ktp::TableName** tableName, **ktp::FString** csvFileName**)**
  SaveDataBaseTable. This method saves a database table to the file the file that you choose.



# Whats next?

So whats next?

The next step is for you to use and/or improve on the library and distribute freely! Free Software To All.

Lastly I would like to formally thank,

Microsoft - Microsoft Visual Studio 2015 Express used to develop the library
HelpNDoc - HelpNDoc Personal Edition used to create this documentation

Thank you and Enjoy!