El kibambo

Official manual • Version 2.1

By El CONCEPT

Table of contents

Technical details

- Creator: Michael KYUNGU ILUNGA
- Developed by: El CONCEPT
- First version: Beta (v. 1.0) / 2021
- Last version: 2.1 (January 30, 2022)
- Reposit:

  ☞ bit.ly/3Rn98xF

  ☞ https://github.com/elconcept15/elkibambo/tree/main/winforms
- Written in: C#
- Type: .NET Class Library
- Target .NET framework: >= 4.5
- Supported languages: French and English

Overview

```
This document has been automatically translated from French
```

El Kibambo is a .NET library for quickly and easily implementing search functionality on data grids in Windows Forms programs.

With El Kibambo, the search can be done on several columns of the data grid (DataGridView) and the comparison is not limited to equality.

El Kibambo is currently in version 2.1, this is the first alpha version.

In this manual, we will illustrate it with the programming languages C# and VB.NET.

El Kibambo supports the following comparison operators:

- == (equality) ;
- != (different) ;
- > (strictly superior) ;
- < (strictly inferior) ;
- >= (greater than or equal to) ;
- <= (less or equal) ;
- LIKE: an attempt by El Kibambo to use the legendary token search like the SQL « LIKE » operator.

The El Kibambo library has as classes:

- **`Kibambo`**: it is the main class of the library. It is thanks to her that research is implemented.
- **`KibamboColumn`**: represents each column of the grid containing the data.
- **`KibamboException`**: in case of error, El Kibambo raises exceptions of this type. This class has a `KibamboErrorCode` type attribute allowing us to better know the type of error.

El Kibambo uses the following enumerations:

- **`KibamboErrorCode`**: each `KibamboException` type object has the `ErrorCode` attribute of `KibamboErrorCode` type giving us more details on the type of error.
- **`KibamboColumnType`**: each object of type `KibamboColumn` has a property of type `KibamboColumnType` indicating the type of value of the column. By the way, a column can have numeric values or strings, and the way the comparison is done for numeric values and strings is not the same.
- **`KibamboComparisonType`** : allows you to specify the type of comparison (case-sensitive or insensitive, etc.). Each `Kibambo` object has a `ComparisonType` attribute allowing to specify the type of comparison.

El Kibambo also defines a delegate, the `KibamboDelegate` delegate. [...]:

```
public delegate void KibamboDelegate(DataGridViewRow row);
```

```
Public Delegate Sub KibamboDelegate(row As DataGridViewRow)
```

1. Enumerations

1.1. KibamboErrorCode

Overview:

```
public enum KibamboErrorCode
{
    ColumnNotRecognized = 5,
    ColumnsNotSpecified = 10,
    ComparisonTypeNotRecognized = 15,
    DgvOfDataMustBeDifferentThanDgvOfSearch = 20,
    MoreColumnsThanDgvData = 25,
    OperatorNotRecognized = 30,
    ValueMustBeNumeric = 35,
    YouMustSpecifyAtLeastOneOperator = 40
}
```

```
Public Enum KibamboErrorCode
    ColumnNotRecognized = 5
    ColumnsNotSpecified = 10
    ComparisonTypeNotRecognized = 15
    DgvOfDataMustBeDifferentThanDgvOfSearch = 20
    MoreColumnsThanDgvData = 25
    OperatorNotRecognized = 30
    ValueMustBeNumeric = 35
    YouMustSpecifyAtLeastOneOperator = 40
End Enum
```

This enumeration has the following values:

- **`ColumnNotRecognized`**: with El Kibambo, each column of the datagrid is passed as an object of type `KibamboColumn` when creating the `Kibambo` object. To be specific, you can pass the `Kibambo` class an array of `KibamboColumn` objects, representing the grid columns containing the data. During the search, if such a column is used, El Kibambo will look for all the lines for which such a column satisfies the search. This error is triggered in case El Kibambo cannot find a column. This is the case, for example, of a non-existent column in the grid containing the data, or deleted/renamed (`Name` property of a `DataGridViewColumn` object) after instantiation of the `Kibambo` object.
- **`ColumnsNotSpecified`**: this error can occur during initialization (`Kibambo.Initialize()`) in case we did not specify columns when instantiating `Kibambo` object and `Kibambo.AutoCompleteColumns` property is `false`.
- **`ComparisonTypeNotRecognized`**: this type of error is internal to the library. By default, the comparison type is case-insensitive and ignores diacritics (`KibamboComparisonType.IgnoreDiacriticSignsAndCase`). However, in

the event of a serious internal error and El Kibambo fails to determine the type of comparison (which is really very less likely), then this error can occur.

- **`DgvOfDataMustBeDifferentThanDgvOfSearch`**: occurs if we pass the same `DataGridView` as the grid containing the data and also as the search grid to the same `Kibambo` object.
- **`MoreColumnsThanDgvData`**: occurs if we pass more `KibamboColumn` objects to the `Kibambo` object than actually contains the grid containing the data.
- **`OperatorNotRecognized`**: this error occurs in the event that, for some reason, El Kibambo fails to recognize the comparison operator during the search (which is still extremely very less likely).
- **`ValueMustBeNumeric`**: This error is caused by the user entering a non-numeric value for a numeric type column.
- **`YouMustSpecifyAtLeastOneOperator`**: occurs if we pass an empty array for comparison operators when constructing a `KibamboColumn` object.

1.2. KibamboColumnType

Overview:

```
public enum KibamboColumnType
{
    String = 5,
    Numeric = 10
}
```

```
Public Enum KibamboColumnType
  [String] = 5
  Numeric = 10
End Enum
```

For the moment, El Kibambo only supports two types of values:

- **`String`**: for character strings;
- **`Numeric`**: for numeric values.

So El Kibambo can be used with other types using `KibamboColumnType.String`.

But why ...? You should know that the comparison between numeric and non-numeric (character strings) is diametrically opposed. If with numerics it is the order of magnitude that is used, with character strings this is not the case. Thus, “10” < “2” gives true but “10 < 2” gives false!

1.3. KibamboComparisonType

Overview:

```
public enum KibamboComparisonType
{
    CaseSensitive = 5,
    IgnoreCase = 10,
    IgnoreDiacriticSigns = 15,
    IgnoreDiacriticSignsAndCase = 20
}
```

```
Public Enum KibamboComparisonType
    CaseSensitive = 5
    IgnoreCase = 10
    IgnoreDiacriticSigns = 15
    IgnoreDiacriticSignsAndCase = 20
End Enum
```

This enumeration has as values:

- **`CaseSensititve`**: for ordinal comparison of case-sensitive character strings.
- **`IgnoreCase`**: for case-insensitive ordinal comparison of character strings.
- **`IgnoreDiacriticSigns`**: for a comparison ignoring diacritics, mainly accents.
- **`IgnoreDiacriticSignsAndCase`** : works like `IgnoreDiacriticSigns`, with the advantage that the comparison is case insensitive.

2. The classes

2.1. KibamboException

Overview:

```
public class KibamboException : Exception
{
    public KibamboException(KibamboErrorCode errorCode);
    public KibamboErrorCode ErrorCode { get; }
    public string Message { get; }
    public override string ToString();
}
```

```
Public Class KibamboException Inherits Exception
    Public Sub New(errorCode As KibamboErrorCode)
    Public ReadOnly Property ErrorCode As KibamboErrorCode
    Public ReadOnly Property Message As String
    Public Overrides Function ToString() As String
End Class
```

Inheriting from the base exception class `Exception`, this class defines two properties:

- **`Message`**: accessible in read-only mode, it provides a message detailing the error. This property overrides the one defined in the base class.

```
string KibamboException.Message { get; }
```

```
ReadOnly Property KibamboException.Message As String
```

- **`ErrorCode`**: contains the error code. It is of type `KibamboErrorCode`.

```
KibamboErrorCode KibamboException.ErrorCode { get; }
```

```
ReadOnly Property KibamboException.ErrorCode As KibamboErrorCode
```

The `KibamboException` class redefines the basic `ToString()` method by adding the error message to the end of the character string:

```
public override string ToString()
{
    return base.ToString() + Environment.NewLine + "Kibambo Error Message: 
" + Message;
}
```

```
Public Overrides Function ToString() As String
    Return MyBase.ToString() + Environment.NewLine + "Kibambo Error 
Message: " + Message
End Function
```

2.2. KibamboColumn

The `KibamboColumn` class is used to represent each column of the grid containing the data.

Overview:

```
public class KibamboColumn
{
    public static readonly string[] SupportedOperators;

    public KibamboColumn(string name, KibamboColumnType columnType);
    public KibamboColumn(string name, string header);
    public KibamboColumn(string name, string header, KibamboColumnType 
columnType);
    public KibamboColumn(string name, string header, KibamboColumnType 
columnType, string headerColumnType);
    public KibamboColumn(string name, string header, KibamboColumnType 
columnType, string headerColumnType, string[] operators);

    public KibamboColumnType ColumnType { get; set; }
    public string Header { get; set; }
    public string HeaderColumnType { get; set; }
    public string Name { get; set; }
    public string[] Operators { get; set; }

    public override string ToString();
}
```

```
Public Class KibamboColumn
    Public Shared ReadOnly SupportedOperators As String()

    Public Sub New(name As String, columnType As KibamboColumnType)
    Public Sub New(name As String, header As String)
    Public Sub New(name As String, header As String, columnType As
KibamboColumnType)
    Public Sub New(name As String, header As String, columnType As
KibamboColumnType, headerColumnType As String)
    Public Sub New(name As String, header As String, columnType As
KibamboColumnType, headerColumnType As String, operators() As String)

    Public Property ColumnType As KibamboColumnType
    Public Property Header As String
    Public Property HeaderColumnType As String
    Public Property Name As String
    Public Property Operators As String()

    Public Overrides Function ToString() As String
End Class
```

a. Properties

The `KibamboColumn` class has as properties: `Name`, `Header`, `ColumnType`, `HeaderColumnType`, `Operators`.

- **`Name`**: refers to the `DataGridViewColumn.Name` property of the grid column containing the data that the `KibamboColumn` object represents.

```
string KibamboColumn.Name { get; set; }
```

```
Property KibamboColumn.Name As String
```

- **`Header`**: this is the title of the column to be displayed by El Kibambo.

```
string KibamboColumn.Header { get; set; }
```

```
Property KibamboColumn.Header As String
```

- **`ColumnType`**: represents the type of the column.

```
KibamboColumnType KibamboColumn.ColumnType { get; set; }
```

```
Property KibamboColumn.ColumnType As KibamboColumnType
```

- **`HeaderColumnType`**: the type of column to be displayed by El Kibambo in the search grid.

```
string KibamboColumn.HeaderColumnType { get; set; }
```

```
Property KibamboColumn.HeaderColumnType As String
```

- **`Operators`**: represents an array of comparison operator signs to use for the column. As a reminder, El Kibambo only supports eight operators (==, !=, <=, >, >=, !=, LIKE). You must specify which carriers El Kibambo supports. The presence of an unsupported operator raises a `KibamboException` with an error code of `KibamboErrorCode.OperatorNotRecognized`. Specifying an empty array also throws a `KibamboException` with an error code of `KibamboErrorCode.YouMustSpecifyAtLeastOneOperator`.

```
string[] KibamboColumn.Operators { get; set; }
```

```
Property KibamboColumn.Operators As String()
```

- **`SupportedOperators`**: constant containing an array of comparison operators supported by the El Kibambo library.

```
string[] KibamboColumn.SupportedOperators
```

```
KibamboColumn.SupportedOperators As String()
```

b. Constructors

The `KibamboColumn` class has the following constructors:

```
KibamboColumn(string name, KibamboColumnType columnType)

KibamboColumn(string name, string header)

KibamboColumn(string name, string header, KibamboColumnType columnType)

KibamboColumn(string name, string header, KibamboColumnType columnType,
string headerColumnType)

KibamboColumn(string name, string header, KibamboColumnType columnType,
string headerColumnType, string[] operators)
```

```
KibamboColumn(name As String, columnType As KibamboColumnType)

KibamboColumn(name As String, header As String)

KibamboColumn(name As String, header As String, columnType As
KibamboColumnType)

KibamboColumn(name As String, header As String, columnType As
KibamboColumnType, headerColumnType As String)

KibamboColumn(name As String, header As String, columnType As
KibamboColumnType, headerColumnType As String, operators As String())
```

Let's detail each parameter:

- **`name`**: value to assign to the `KibamboColumn.Name` property;
- **`header`**: value to assign to the `KibamboColumn.Header` property;
- **`columnType`**: value to assign to the `KibamboColumn.ColumnType` property;
- **`headerColumnType`**: value to assign to the `KibamboColumn.HeaderColumnType` property;
- **`operators`**: value to assign to the `KibamboColumn.Operators` property.

By using a constructor that does not take an array of comparison operators as a parameter, all comparison operators supported by El Kibambo will be automatically used.

c. Methods

The `KibamboColumn` class just redefines a method: « ToString() ».

```
public override string ToString()
{
    return Header;
}
```

```
Public Overrides Function ToString() As String
    Return Header
End Function
```

2.3. Kibambo

The `Kibambo` class is the very heart of the El Kibambo library. Overview:

```
public class Kibambo
{
    public Kibambo(DataGridView dgvData, DataGridView dgvSearch, bool
autoCompleteColumns = false);
    public Kibambo(DataGridView dgvData, DataGridView dgvSearch,
KibamboColumn[] columns, bool autoCompleteColumns = false);
    public Kibambo(DataGridView dgvData, DataGridView dgvSearch,
KibamboDelegate searchHandler, bool autoCompleteColumns = false);
    public Kibambo(DataGridView dgvData, DataGridView dgvSearch,
KibamboDelegate searchHandler, KibamboDelegate cancelSearchHandler, bool
autoCompleteColumns = false);
    public Kibambo(DataGridView dgvData, DataGridView dgvSearch,
KibamboColumn[] columns, KibamboDelegate searchHandler, bool
autoCompleteColumns = false);
    public Kibambo(DataGridView dgvData, DataGridView dgvSearch,
KibamboColumn[] columns, KibamboDelegate searchHandler, KibamboDelegate
cancelSearchHandler, bool autoCompleteColumns = false);

    public bool AutoCompleteColumns { get; set; }
    public KibamboDelegate CancelSearchHandler { get; set; }
    public KibamboColumn[] Columns { get; set; }
    public KibamboComparisonType ComparisonType { get; set; }
    public List RowsSatisfied { get; }
    public List RowsUnsatisfied { get; }
    public KibamboDelegate SearchHandler { get; set; }
    public string TitleField { get; set; }
    public string TitleType { get; set; }
    public string TitleOperator { get; set; }
    public string TitleValue { get; set; }

    public void CancelSearch();
    public void Initialize();
    public void Search();
}
```

```
Public Class Kibambo
    Public Sub New(dgvData As DataGridView, dgvSearch As DataGridView,
Optional autoCompleteColumns As Boolean = False)
    Public Sub New(dgvData As DataGridView, dgvSearch As DataGridView,
columns() As KibamboColumn, Optional autoCompleteColumns As Boolean =
False)
    Public Sub New(dgvData As DataGridView, dgvSearch As DataGridView,
searchHandler As KibamboDelegate, Optional autoCompleteColumns As Boolean =
False)
    Public Sub New(dgvData As DataGridView, dgvSearch As DataGridView,
searchHandler As KibamboDelegate, cancelSearchHandler As KibamboDelegate,
Optional autoCompleteColumns As Boolean = False)
    Public Sub New(dgvData As DataGridView, dgvSearch As DataGridView,
columns() As KibamboColumn, searchHandler As KibamboDelegate, Optional
autoCompleteColumns As Boolean = False)
    Public Sub New(dgvData As DataGridView, dgvSearch As DataGridView,
columns() As KibamboColumn, searchHandler As KibamboDelegate,
cancelSearchHandler As KibamboDelegate, Optional autoCompleteColumns As
Boolean = False)

    Public Property AutoCompleteColumns As Boolean
    Public Property CancelSearchHandler As KibamboDelegate
    Public Property Columns As KibamboColumn()
    Public Property ComparisonType As KibamboComparisonType
    Public ReadOnly Property RowsSatisfied As List(Of DataGridViewRow)
    Public ReadOnly Property RowsUnsatisfied As List(Of DataGridViewRow)
    Public Property SearchHandler As KibamboDelegate
    Public Property TitleHead As String
    Public Property TitleType As String
    Public Property TitleOperator As String
    Public Property TitleValue As String

    Public Sub CancelSearch()
    Public Sub Initialize()
    Public Sub Search()
End Class
```

a. Constructors

The Kibambo class has the following constructors:

```
Kibambo(DataGridView dgvData, DataGridView dgvSearch, [bool
autoCompleteColumns = false])

Kibambo(DataGridView dgvData, DataGridView dgvSearch, KibamboColumn[]
columns, [bool autoCompleteColumns = false])
```

```
Kibambo(DataGridView dgvData, DataGridView dgvSearch, KibamboDelegate
searchHandler, [bool autoCompleteColumns = false])

Kibambo(DataGridView dgvData, DataGridView dgvSearch, KibamboColumn[]
columns, KibamboDelegate searchHandler, [bool autoCompleteColumns = false])

Kibambo(DataGridView dgvData, DataGridView dgvSearch, KibamboDelegate
searchHandler, KibamboDelegate cancelSearchHandler, [bool
autoCompleteColumns = false])

Kibambo(DataGridView dgvData, DataGridView dgvSearch, KibamboColumn[]
columns, KibamboDelegate searchHandler, KibamboDelegate
cancelSearchHandler, [bool autoCompleteColumns = false])
```

```
Kibambo(dgvData As DataGridView, dgvSearch As DataGridView,
[autoCompleteColumns As Boolean = False])

Kibambo(dgvData As DataGridView, dgvSearch As DataGridView, columns As
KibamboColumn(), [autoCompleteColumns As Boolean = False])

Kibambo(dgvData As DataGridView, dgvSearch As DataGridView, searchHandler
As KibamboDelegate, [autoCompleteColumns As Boolean = False])

Kibambo(dgvData As DataGridView, dgvSearch As DataGridView, columns As
KibamboColumn(), searchHandler As KibamboDelegate, [autoCompleteColumns As
Boolean = False])

Kibambo(dgvData As DataGridView, dgvSearch As DataGridView, searchHandler
As KibamboDelegate, cancelSearchHandler As KibamboDelegate,
[autoCompleteColumns As Boolean = False])

Kibambo(dgvData As DataGridView, dgvSearch As DataGridView, columns As
KibamboColumn(), searchHandler As KibamboDelegate, cancelSearchHandler As
KibamboDelegate, [autoCompleteColumns As Boolean = False])
```

Let's detail each parameter:

- **`dgvData`**: the `DataGridView` containing the data ;
- **`dgvSearch`**: the `DataGridView` which will allow to set up the search functionalities ;
- **`autoCompleteColumns`**: indicates whether we want El Kibambo to auto-complete unspecified columns or not ;
- **`columns`**: array of `KibamboColumn` objects referencing each column of the grid containing the data;
- **`searchHandler`**: delegate of `KibamboDelegate` type to call during the search, i.e. when calling the `Kibambo.Search()` method;
- **`cancelSearchHandler`**: `KibamboDelegate` type delegate called by El Kibambo when you want to cancel a search, i.e. when calling the `Kibambo.CancelSearch()` method.

Note the presence of a new type: the `KibamboDelegate` delegate. His statement in the meanders of El Kibambo is as follows:

```
delegate void
ElKibambo.KibamboDelegate(System.Windows.Forms.DataGridViewRow row)
```

```
Delegate Sub ElKibambo.KibamboDelegate(row As
System.Windows.Forms.DataGridViewRow)
```

Quite simply a method which returns nothing and which receives as the only parameter an object of type `DataGridViewRow` representing of course a row of the grid containing the data.

But why ? See you when studying the `Kibambo.Search()` and `Kibambo.CancelSearch()` methods!

It should be noted:

- If the `Kibambo.AutoCompleteColumns` property is `true`, using constructors that do not take an array of `KibamboColumn` objects as a parameter is equivalent to letting the Kibambo class use the default parameters for each column of the grid containing the data. Also note that El Kibambo will consider that the column is of type `KibamboColumnType.String`. However, if `Kibambo.AutoCompleteColumns` is `false` and we don't specify any columns, El Kibambo throws a `KibamboException` type error with the error code `KibamboErrorCode.ColumnsNotSpecified`.
- If the value of `dgvData` and that of `dgvSearch` is the same, i.e. we pass the same data grid to the constructor, for the data and for the search, El Kibambo raises a `KibamboException` with the error code `KibamboErrorCode.DgvOfDataMustBeDifferentThanDgvOfSearch`.
- During initialization (`Kibambo.Initialize()`), El Kibambo proceeds to configure the search grid. For example, the following properties are disabled (value passed to `false`): `AllowDrop`, `AllowUserToAddRows`, `AllowUserToDeleteRows`, `AllowUserToOrderColumns`, `AllowUserToResizeRows`, `RowHeadersVisible`.

b. Properties

AutoCompleteColumns

```
bool Kibambo.AutoCompleteColumns { get; set; }
```

```
Property Kibambo.AutoCompleteColumns As Boolean
```

This property tells El Kibambo whether or not to autocomplete unspecified columns. Thus, you can not specify columns and leave the task to El Kibambo to generate them automatically based on the columns of the grid containing the data. You can also specify only part of columns (if necessary, for example if some columns require certain comparison operators and others do not), and let El Kibambo complete the unspecified columns.

Columns

```
KibamboColumn[] Kibambo.Columns { get; set; }
```

```
Property Kibambo.Columns As KibamboColumn()
```

The `Columns` property provides read and write access to the `KibamboColumn` objects representing each column of the grid containing the data.

Thanks to this property, you can specify only certain columns of the grid containing the data if, for example, you want to search only certain columns and not others (of course, the `Kibambo.AutoCompleteColumns` property must be `false` in this case).

ComparisonType

```
KibamboComparisonType Kibambo.ComparisonType { get; set; }
```

```
Property Kibambo.ComparisonType As KibamboComparisonType
```

Allows you to specify the type of comparison to use during the search. The default value for this property is « `KibamboComparisonType.IgnoreDiacriticSignsAndCase` ».

RowsSatisfied and RowsUnsatisfied

```
List<DataGridViewRow> Kibambo.RowsSatisfied { get; }
List<DataGridViewRow> Kibambo.RowsUnsatisfied { get; }
```

```
ReadOnly Property Kibambo.RowsSatisfied As List (Of DataGridView)
ReadOnly Property Kibambo.RowsUnsatisfied As List (Of DataGridView)
```

After a search using the `Kibambo.Search()` method, the `Kibambo.RowsSatisfied` property will contain all the rows that satisfied the search criteria and `Kibambo.RowsUnsatisfied` all the rows that did not satisfy the search criteria.

SearchHandler and CancelSearchHandler

```
KibamboDelegate Kibambo.SearchHandler { get; set; }
KibamboDelegate Kibambo.CancelSearchHandler { get; set; }
```

```
Property Kibambo.SearchHandler As KibamboDelegate
Property Kibambo.CancelSearchHandler As KibamboDelegate
```

These two properties are of type `KibamboDelegate` and represent the methods called respectively (if specified) when calling the `Kibambo.Search()` and `Kibambo.CancelSearch()` methods.

TitleField, TitleType, TitleOperator, TitleValue

These properties represent the four headers of the data grid that will implement the search. It is respectively: “field”, “type”, “operator”, and “value”. They just allow you to modify the default values if necessary.

```
string Kibambo.TitleField { get; set; }
string Kibambo.TitleType { get; set; }
string Kibambo.TitleOperator { get; set; }
string Kibambo.TitleValue { get; set; }
```

```
Property Kibambo.TitleField As String
Property Kibambo.TitleType As String
Property Kibambo.TitleOperator As String
Property Kibambo.TitleValue As String
```

c. Methods

Search()

```
void Kibambo.Search()
```

```
Sub Kibambo.Search()
```

This method is called to perform the search with El Kibambo. During the search, all rows that have satisfied the search criteria are added to a collection accessible through the `Kibambo.RowsSatisfied` property. Those that are not satisfied are put in another collection accessible through the `Kibambo.RowsUnsatisfied` property. When searching, if the search delegate, `Kibambo.SearchHandler` has been specified, each row that does not meet the search criteria will be passed to the processing method.

CancelSearch()

```
void Kibambo.CancelSearch()
```

```
Sub Kibambo.CancelSearch()
```

When calling this method, if the `Kibambo.CancelSearchHandler` property is not `null` (that is, a callback function has been specified) and the `Kibambo.RowsUnsatisfied` collection is not empty, the rows that do not meet the search criteria are passed to the `Kibambo.CancelSearchHandler` delegate in turn.

It is therefore reasonable to reverse what was done in the search delegate. For example, if during the search we hid the lines that did not meet the search criteria, it seems logical that in the `Kibambo.CancelSearchHandler` delegate, we must redisplay them.

Note that this method empties the `Kibambo.RowsSatisfied` and `Kibambo.RowsUnsatisfied` collections.

Initialize()

```
void Kibambo.Initialize()
```

```
Sub Kibambo.Initialize()
```

This method is called to initialize/reinitialize El Kibambo. This method should be called just after constructing the Kibambo object or after modifying its properties.

More information

- You can find examples of use on the playlist of El Kibambo on the Youtube channel of El CONCEPT.

  ☞ bit.ly/3AKqqie

  ☞ https://youtube.com/playlist?list=PL7i4WkgFSFMlT75YwyJVNduON6Tzu8upT

- Library download link:

  ☞ bit.ly/3yVcdh6

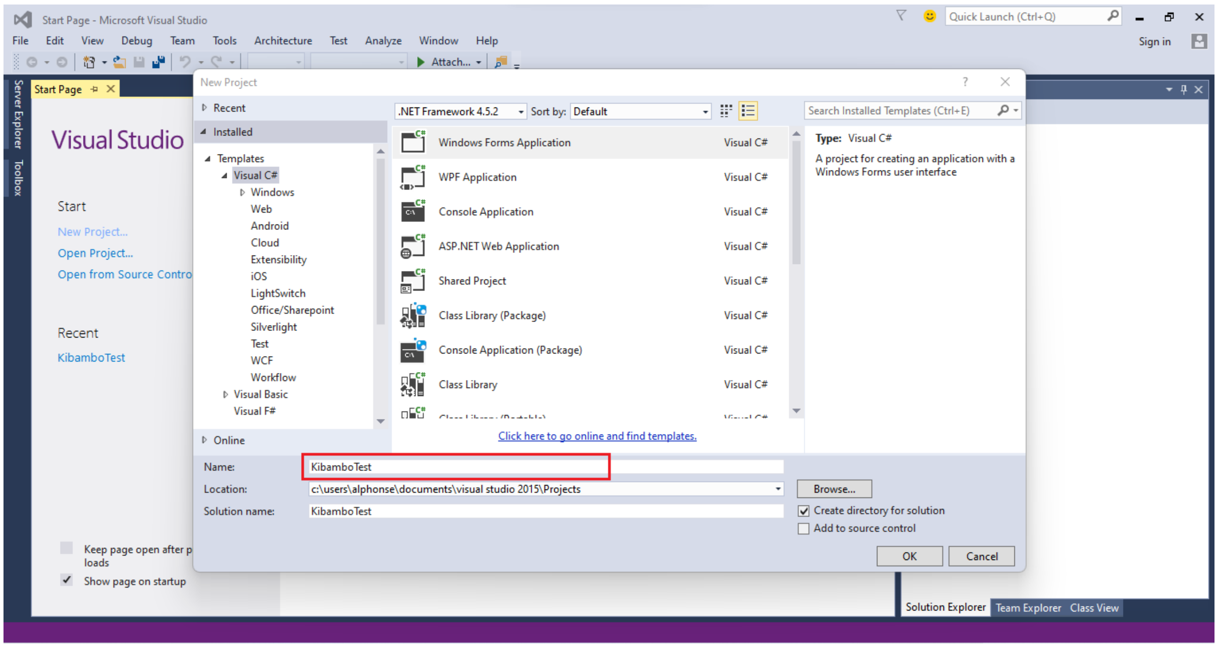
  ☞ https://github.com/elconcept15/elkibambo/blob/main/winforms/elkibambo.dll

- When needed, [...] :
  - Email : elconcept15@gmail.com
  - Contact :
    - +243 97 24 38 801
    - +243 85 57 99 756
  - Facebook page: https://web.facebook.com/elconcept15

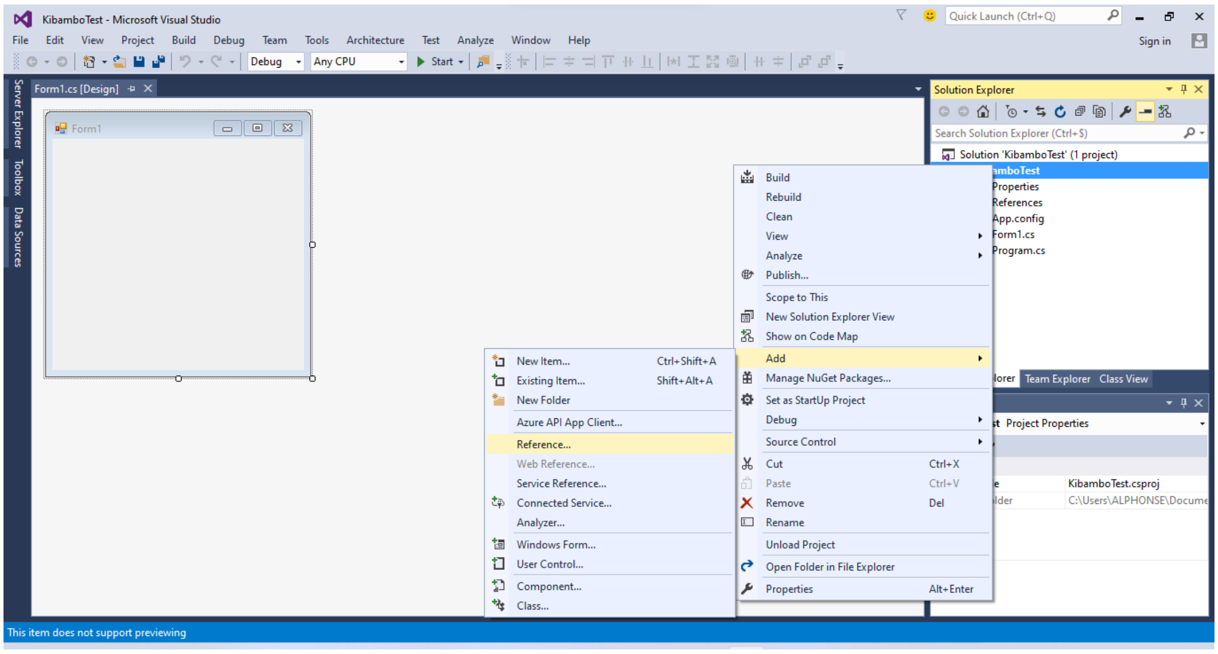
Implementation - C#

In this section, we will create a C# Windows Forms program demonstrating the implementation of the El Kibambo library. We will use as IDE « Microsoft Visual Studio Enterprise 2015 ».

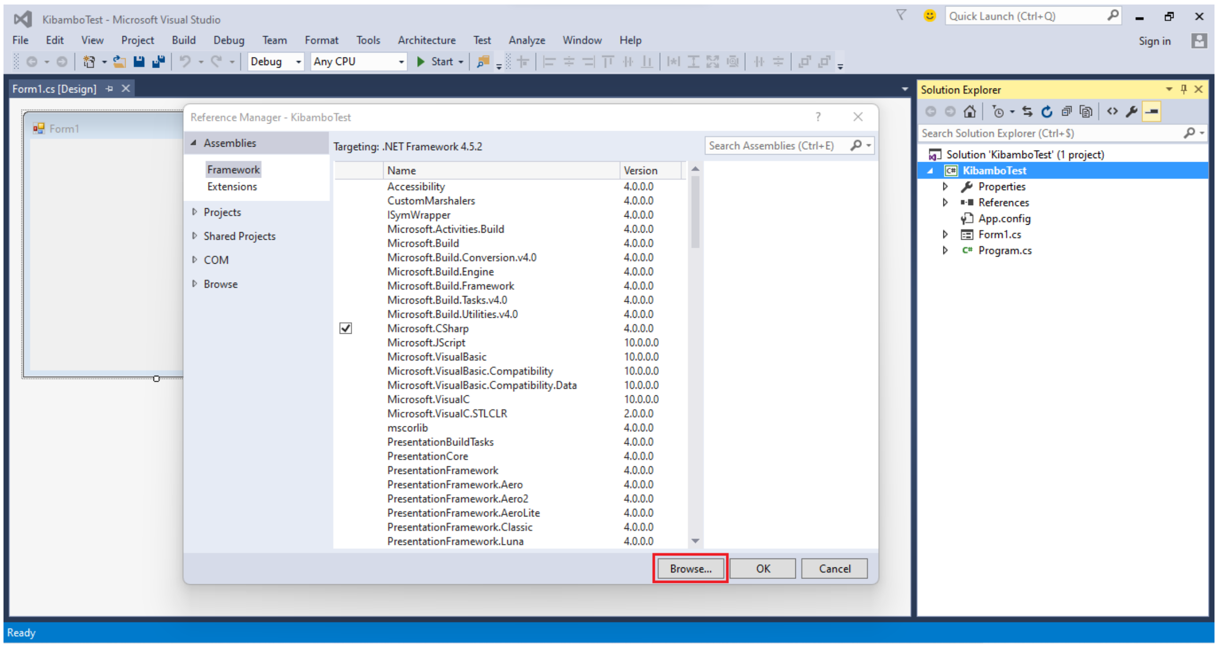
Let's create a new C# Windows Forms project that we name « KibamboTest ».


New Project
.NET Framework 4.5.2
Sort by: Default
Installed
Templates
Visual C#
Windows Forms Application
WPF Application
Console Application
ASP.NET Web Application
Shared Project
Class Library (Package)
Console Application (Package)
Class Library
Type: Visual C#
A project for creating an application with a Windows Forms user interface
Name: KibamboTest
Location: c:\users\alphonse\documents\visual studio 2015\Projects
Solution name: KibamboTest
Create directory for solution
OK
Cancel

Now let's add the reference to the El Kibambo library. To do this, right-click on the name of the project, choose the « Add » option, then « Reference... ».



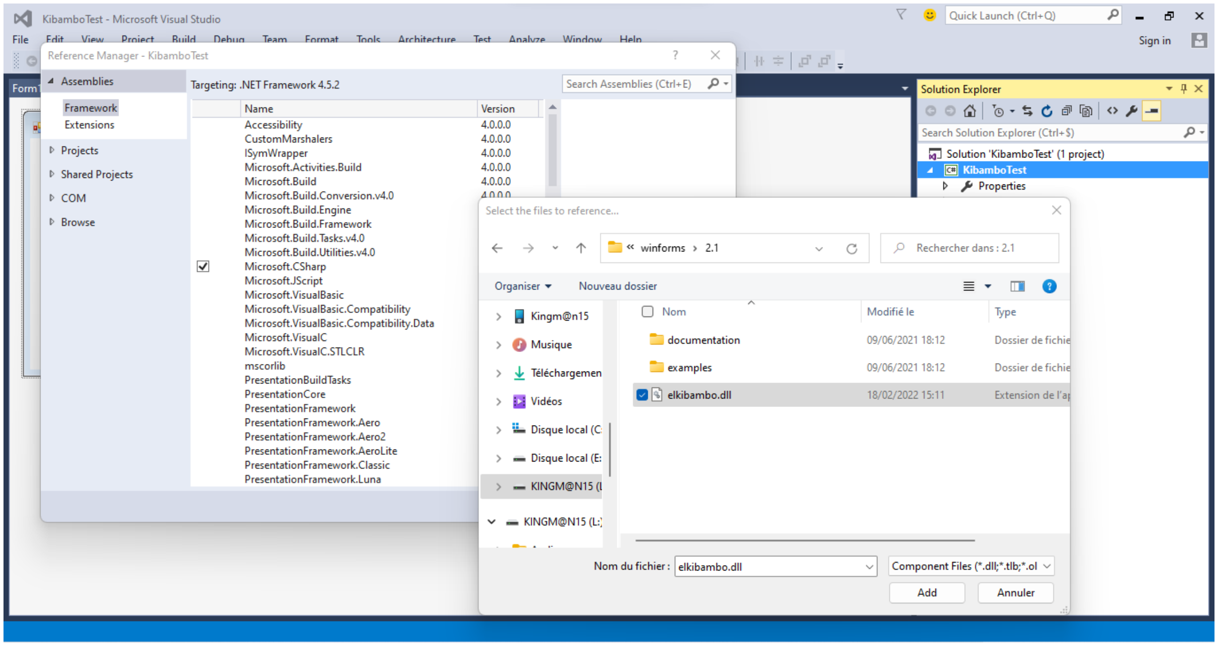
By clicking on the « Reference... » option, Visual Studio will show you an interface similar to:


KibamboTest - Microsoft Visual Studio
File Edit View Project Build Debug Team Format Tools Architecture Test Analyze Window Help
Debug
Any CPU
Start
Quick Launch (Ctrl+Q)
Sign in
Form1.cs [Design]
Form1
Reference Manager - KibamboTest
Assemblies
Framework
Extensions
Projects
Shared Projects
COM
Browse
Search Assemblies (Ctrl+E)
Browse...
OK
Cancel
Solution Explorer
Search Solution Explorer (Ctrl+$)
Solution 'KibamboTest' (1 project)
KibamboTest
Properties
References
App.config
Form1.cs
Program.cs
Ready

Targeting: .NET Framework 4.5.2

	Name	Version
	Accessibility	4.0.0.0
	CustomMarshalers	4.0.0.0
	ISymWrapper	4.0.0.0
	Microsoft.Activities.Build	4.0.0.0
	Microsoft.Build	4.0.0.0
	Microsoft.Build.Conversion.v4.0	4.0.0.0
	Microsoft.Build.Engine	4.0.0.0
	Microsoft.Build.Framework	4.0.0.0
	Microsoft.Build.Tasks.v4.0	4.0.0.0
	Microsoft.Build.Utilities.v4.0	4.0.0.0
✓	Microsoft.CSharp	4.0.0.0
	Microsoft.JScript	10.0.0.0
	Microsoft.VisualBasic	10.0.0.0
	Microsoft.VisualBasic.Compatibility	10.0.0.0
	Microsoft.VisualBasic.Compatibility.Data	10.0.0.0
	Microsoft.VisualC	10.0.0.0
	Microsoft.VisualC.STLCLR	2.0.0.0
	mscorlib	4.0.0.0
	PresentationBuildTasks	4.0.0.0
	PresentationCore	4.0.0.0
	PresentationFramework	4.0.0.0
	PresentationFramework.Aero	4.0.0.0
	PresentationFramework.Aero2	4.0.0.0
	PresentationFramework.AeroLite	4.0.0.0
	PresentationFramework.Classic	4.0.0.0
	PresentationFramework.Luna	4.0.0.0

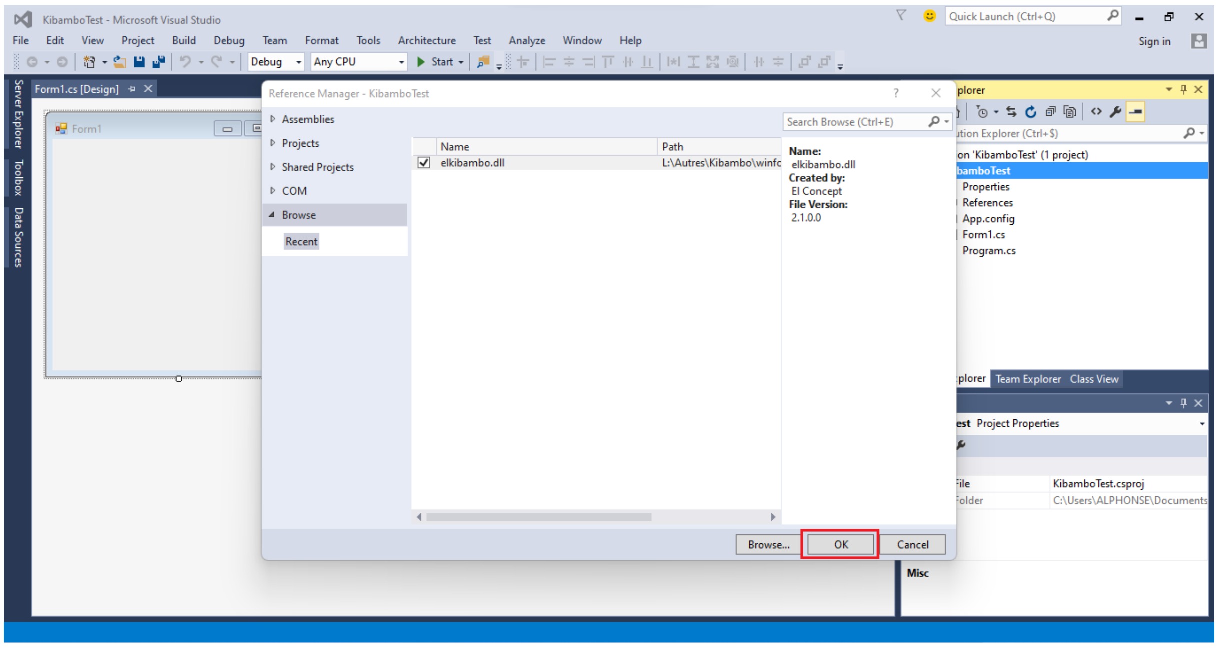
Press the « Browse... » button, this will display the file explorer so you can choose the dll file from the library.



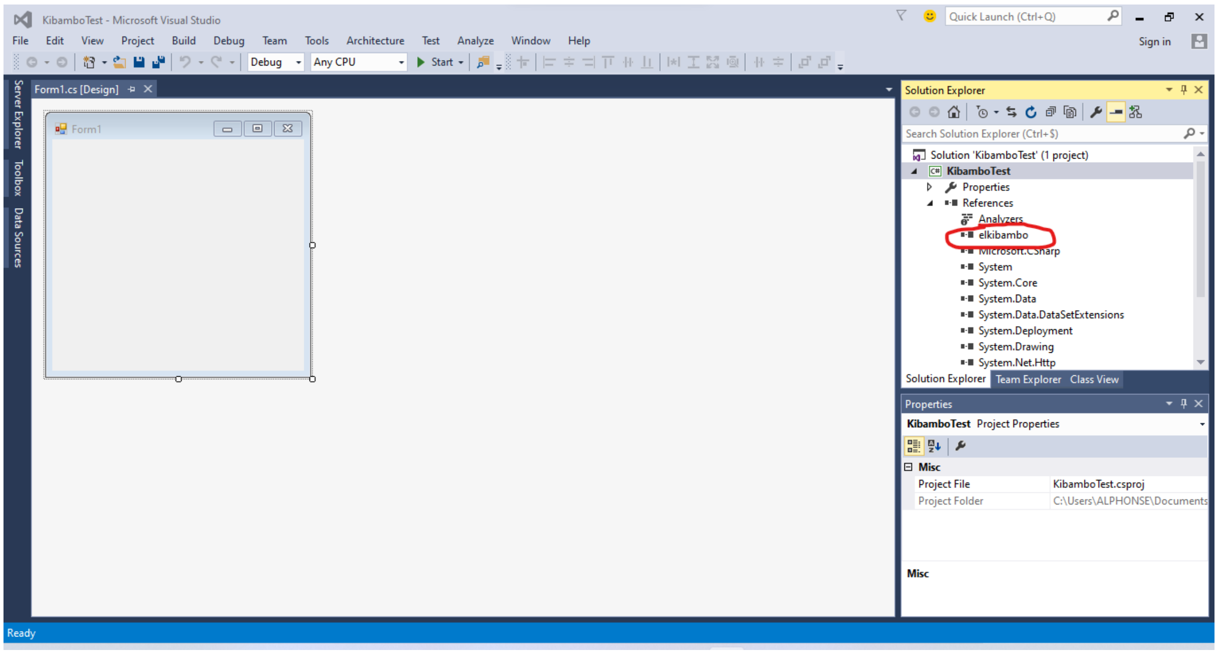
	Nom	Modifié le	Type
	documentation	09/06/2021 18:12	Dossier de fichie
	examples	09/06/2021 18:12	Dossier de fichie
✓	elkibambo.dll	18/02/2022 15:11	Extension de l'ap

Nom du fichier : elkibambo.dll
Component Files (*.dll;*.tlb;*.ol

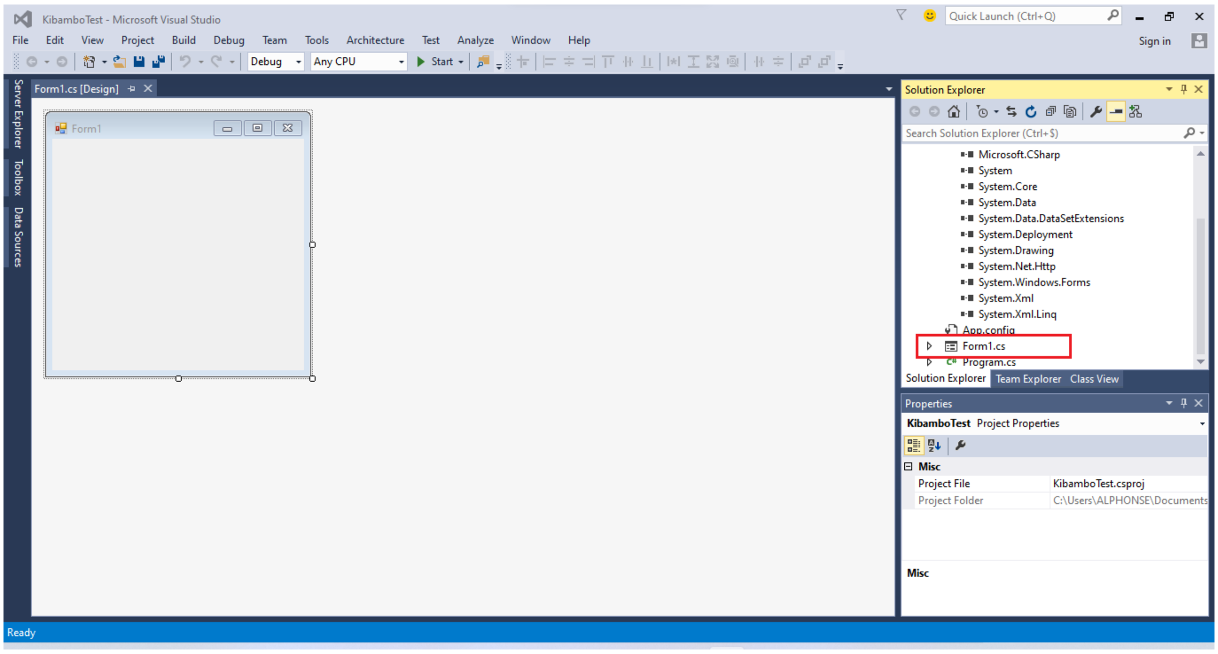
Press the « Add » button, you will have an interface similar to:



Press the « OK » button to finalize the addition of reference to the library.



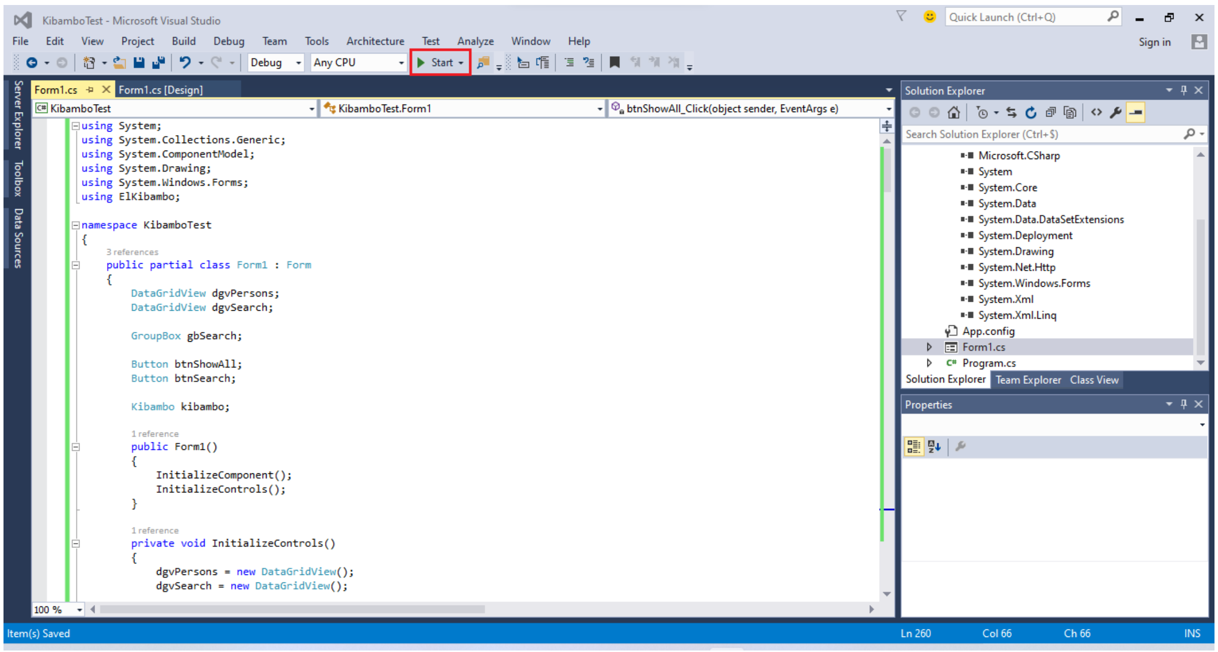
Now open the “Form1.cs” file:



Replace the code in the “Form1.cs” file with the following:

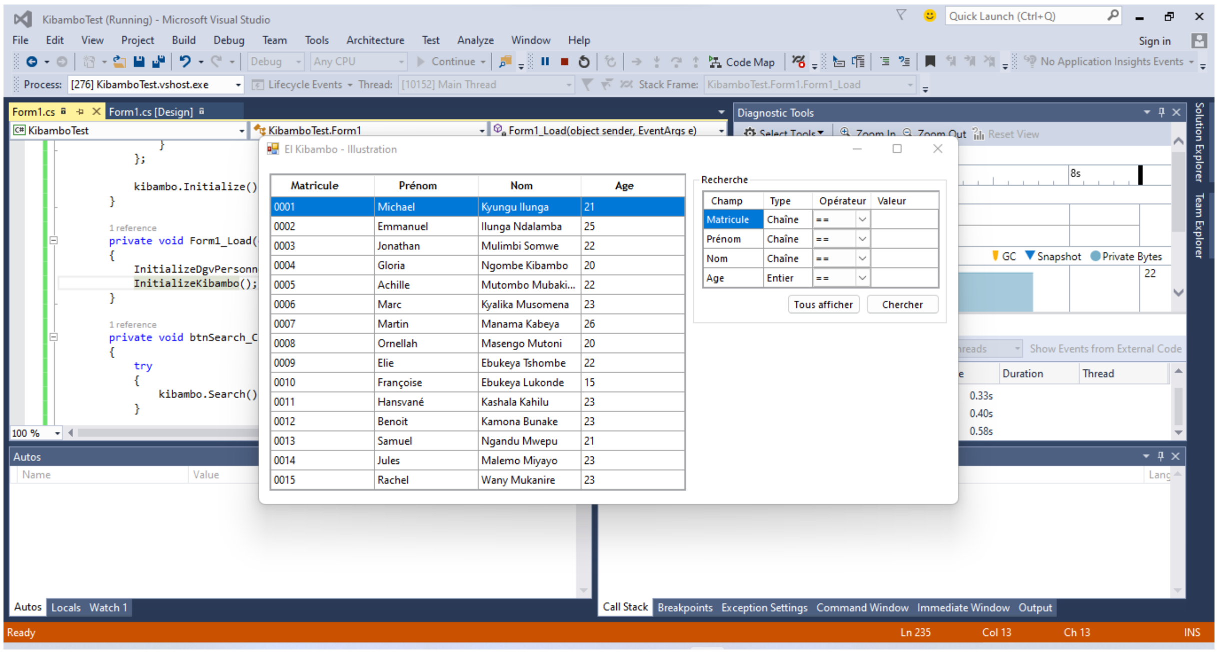
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;
using ElKibambo;

namespace KibamboTest
{
    public partial class Form1 : Form
    {
        DataGridView dgvPersons;
        DataGridView dgvSearch;

        GroupBox gbSearch;

        Button btnShowAll;
        Button btnSearch;

        Kibambo kibambo;

        public Form1()
        {
            InitializeComponent();
            InitializeControls();
        }

        private void InitializeControls()
        {
            dgvPersons = new DataGridView();
            dgvSearch = new DataGridView();

            gbSearch = new GroupBox();

            btnShowAll = new Button();
            btnSearch = new Button();

            ((ISupportInitialize)(dgvPersons)).BeginInit();
            ((ISupportInitialize)(dgvSearch)).BeginInit();

            gbSearch.SuspendLayout();
            SuspendLayout();

            // We ensure that any controls added by the user from the view
designer do not overlap
            // Delete if needed …
            Controls.Clear();

            //
            // dgvcs1
            //
            var dgvcs1 = new DataGridViewCellStyle()
            {
                Alignment = DataGridViewContentAlignment.MiddleCenter,
                BackColor = SystemColors.Control,
                Font = new Font("Trebuchet MS", 9F, FontStyle.Bold,
GraphicsUnit.Point, 0),
                ForeColor = SystemColors.WindowText,
                SelectionBackColor = SystemColors.Highlight,
                SelectionForeColor = SystemColors.HighlightText,
                WrapMode = DataGridViewTriState.True
            };

            //
            // dgvcs2
            //
            var dgvcs2 = new DataGridViewCellStyle()
            {
                Font = new Font("Segoe UI", 9F, FontStyle.Regular,
GraphicsUnit.Point, 0)
            };

            //
            // dgvPersons
            //
            dgvPersons.AllowUserToAddRows = false;
            dgvPersons.AllowUserToDeleteRows = false;
            dgvPersons.AllowUserToResizeColumns = false;
            dgvPersons.AllowUserToResizeRows = false;
            dgvPersons.Anchor = ((AnchorStyles.Top | AnchorStyles.Bottom) |
AnchorStyles.Left) | AnchorStyles.Right;
            dgvPersons.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
            dgvPersons.BackgroundColor = Color.White;
            dgvPersons.BorderStyle = BorderStyle.Fixed3D;
            dgvPersons.ColumnHeadersDefaultCellStyle = dgvcs1;
            dgvPersons.ColumnHeadersHeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.AutoSize;
            dgvPersons.EditMode =
DataGridViewEditMode.EditProgrammatically;
            dgvPersons.Location = new Point(12, 12);
            dgvPersons.MultiSelect = false;
            dgvPersons.RowHeadersVisible = false;
            dgvPersons.RowsDefaultCellStyle = dgvcs2;
            dgvPersons.ScrollBars = ScrollBars.Vertical;
            dgvPersons.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
            dgvPersons.Size = new Size(470, 358);
            dgvPersons.TabIndex = 0;

            //
            // dgvSearch
            //
            dgvSearch.Anchor = (AnchorStyles.Top | AnchorStyles.Left) |
AnchorStyles.Right;
            dgvSearch.BackgroundColor = Color.White;
            dgvSearch.BorderStyle = BorderStyle.Fixed3D;
            dgvSearch.ColumnHeadersHeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.AutoSize;
            dgvSearch.Location = new Point(10, 19);
            dgvSearch.Size = new Size(268, 111);
            dgvSearch.TabIndex = 1;

            //
            // btnShowAll
            //
            btnShowAll.Anchor = AnchorStyles.Top | AnchorStyles.Right;
            btnShowAll.Location = new Point(106, 136);
            btnShowAll.Size = new Size(83, 23);
            btnShowAll.TabIndex = 3;
            btnShowAll.Text = "Show all";
            btnShowAll.UseVisualStyleBackColor = true;
            btnShowAll.Click += btnShowAll_Click;

            //
            // btnSearch
            //
            btnSearch.Anchor = AnchorStyles.Top | AnchorStyles.Right;
            btnSearch.Location = new Point(195, 136);
            btnSearch.Size = new Size(83, 23);
            btnSearch.TabIndex = 2;
            btnSearch.Text = "Search";
            btnSearch.UseVisualStyleBackColor = true;
            btnSearch.Click += btnSearch_Click;

            //
            // gbSearch
            //
            gbSearch.Anchor = AnchorStyles.Top | AnchorStyles.Right;
            gbSearch.Controls.Add(btnShowAll);
            gbSearch.Controls.Add(btnSearch);
            gbSearch.Controls.Add(dgvSearch);
            gbSearch.Font = new Font("Segoe UI", 8.25F, FontStyle.Regular,
GraphicsUnit.Point, 0);
            gbSearch.Location = new Point(490, 12);
            gbSearch.Size = new Size(286, 170);
            gbSearch.TabIndex = 2;
            gbSearch.TabStop = false;
            gbSearch.Text = "Search";

            //
            // Form1
            //
            AutoScaleDimensions = new SizeF(6F, 13F);
            AutoScaleMode = AutoScaleMode.Font;
            BackColor = Color.White;
            ClientSize = new Size(789, 385);
            Controls.Add(gbSearch);
            Controls.Add(dgvPersons);
            Text = "El Kibambo - Illustration";
            Load += new EventHandler(Form1_Load);
            StartPosition = FormStartPosition.CenterScreen;

            ((ISupportInitialize)(dgvPersons)).EndInit();
            ((ISupportInitialize)(dgvSearch)).EndInit();

            gbSearch.ResumeLayout(false);
            ResumeLayout(false);
        }

        private void InitializeDgvPersonnes()
        {
            // Adding columns
            dgvPersons.Columns.Add("registration_number", " Registration
number");
            dgvPersons.Columns.Add("first_name", " First name");
            dgvPersons.Columns.Add("last_name", "Last name");
            dgvPersons.Columns.Add("age", "Age");
            // "Person" objects to display in the datagrid
            var persons = new List<Person>
            {
                new Person("0001", "Michael", "Kyungu Ilunga", 21),
                new Person("0002", "Emmanuel", "Ilunga Ndalamba", 25),
                new Person("0003", "Jonathan", "Mulimbi Somwe", 22),
                new Person("0004", "Gloria", "Ngombe Kibambo", 20),
                new Person("0005", "Achille", "Mutombo Mubakilay", 22),
                new Person("0006", "Marc", "Kyalika Musomena", 23),
                new Person("0007", "Martin", "Manama Kabeya", 26),
                new Person("0008", "Ornellah", "Masengo Mutoni", 20),
                new Person("0009", "Elie", "Ebukeya Tshombe", 22),
                new Person("0010", "Françoise", "Ebukeya Lukonde", 15),
                new Person("0011", "Hansvané", "Kashala Kahilu", 23),
                new Person("0012", "Benoit", "Kamona Bunake", 23),
                new Person("0013", "Samuel", "Ngandu Mwepu", 21),
                new Person("0014", "Jules", "Malemo Miyayo", 23),
                new Person("0015", "Rachel", "Wany Mukanire", 23)
            };

            // Adding data to the data grid
            foreach (Person person in persons)
            {
                dgvPersons.Rows.Add(
                    person.RegistrationNumber,
                    person.FirstName,
                    person.LastName,
                    person.Age
                );
            }
        }

        private void InitializeKibambo()
        {
            kibambo = new Kibambo(dgvPersons, dgvSearch, true)
            {
                Columns = new KibamboColumn[]
                {
                    new KibamboColumn("registration_number", " Registration
number")
                    {
                        Operators = new[] { "==", "!=" }
                    },

                    new KibamboColumn("age", "Age",
KibamboColumnType.Numeric, "Integer", new string[] { "==", "<", ">", "<=",
">=", "!=" })
                },

                SearchHandler = delegate (DataGridViewRow row)
                {
                    // We choose to make invisible the lines that have not
satisfied the search criteria.
                    // But you can still do other things, for example
change the color, etc.
                    row.Visible = false;
                },

                CancelSearchHandler = delegate (DataGridViewRow row)
                {
                    // As we have hidden the rows that did not satisfy the
search criteria during the search, it seems logical that we can re-display
them when we want to cancel the search made.
                    // So we have to reset the line to its initial state
before the search.
                    row.Visible = true;
                }
            };

            kibambo.Initialize();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            InitializeDgvPersonnes();
            InitializeKibambo();
        }

        private void btnSearch_Click(object sender, EventArgs e)
        {
            try
            {
                kibambo.Search();
            }

            catch (KibamboException ex)
            {
                switch (ex.ErrorCode)
                {
                    case KibamboErrorCode.ValueMustBeNumeric:
                        MessageBox.Show("You must enter a numeric type
value for the numeric type column!", " Input error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                        break;

                    default:
                        MessageBox.Show("An error occurred while searching.
The error message is " + ex.Message + " !", " Error while searching",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                        break;
                }
            }
        }

        private void btnShowAll_Click(object sender, EventArgs e)
        {
            kibambo.CancelSearch();
        }

        public class Person
        {
            public string RegistrationNumber { get; set; }
            public string FirstName { get; set; }
            public string LastName { get; set; }
            public int Age { get; set; }

            public Person(string registrationNumber, string firstName,
string lastName, int age)
            {
                RegistrationNumber = registrationNumber;
                FirstName = firstName;
                LastName = lastName;
                Age = age;
            }
        }
    }
}
```

Run the application by pressing the "Start" button or using the F5 function key:



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;
using ElKibambo;

namespace KibamboTest
{
    3 references
    public partial class Form1 : Form
    {
        DataGridView dgvPersons;
        DataGridView dgvSearch;

        GroupBox gbSearch;

        Button btnShowAll;
        Button btnSearch;

        Kibambo kibambo;

        1 reference
        public Form1()
        {
            InitializeComponent();
            InitializeControls();
        }

        1 reference
        private void InitializeControls()
        {
            dgvPersons = new DataGridView();
            dgvSearch = new DataGridView();
```

Result:



```
        }
    };

    kibambo.Initialize()
}

1 reference
private void Form1_Load(
{
    InitializeDgvPersonn
    InitializeKibambo();
}

1 reference
private void btnSearch_C
{
    try
    {
        kibambo.Search()
    }
```

El Kibambo - Illustration

Matricule	Prénom	Nom	Age
0001	Michael	Kyungu Ilunga	21
0002	Emmanuel	Ilunga Ndalamba	25
0003	Jonathan	Mulimbi Somwe	22
0004	Gloria	Ngombe Kibambo	20
0005	Achille	Mutombo Mubaki...	22
0006	Marc	Kyalika Musomena	23
0007	Martin	Manama Kabeya	26
0008	Ornellah	Masengo Mutoni	20
0009	Elie	Ebukeya Tshombe	22
0010	Françoise	Ebukeya Lukonde	15
0011	Hansvané	Kashala Kahilu	23
0012	Benoit	Kamona Bunake	23
0013	Samuel	Ngandu Mwepu	21
0014	Jules	Malemo Miyayo	23
0015	Rachel	Wany Mukanire	23

Recherche

Champ	Type	Opérateur	Valeur
Matricule	Chaîne	==	
Prénom	Chaîne	==	
Nom	Chaîne	==	
Age	Entier	==	

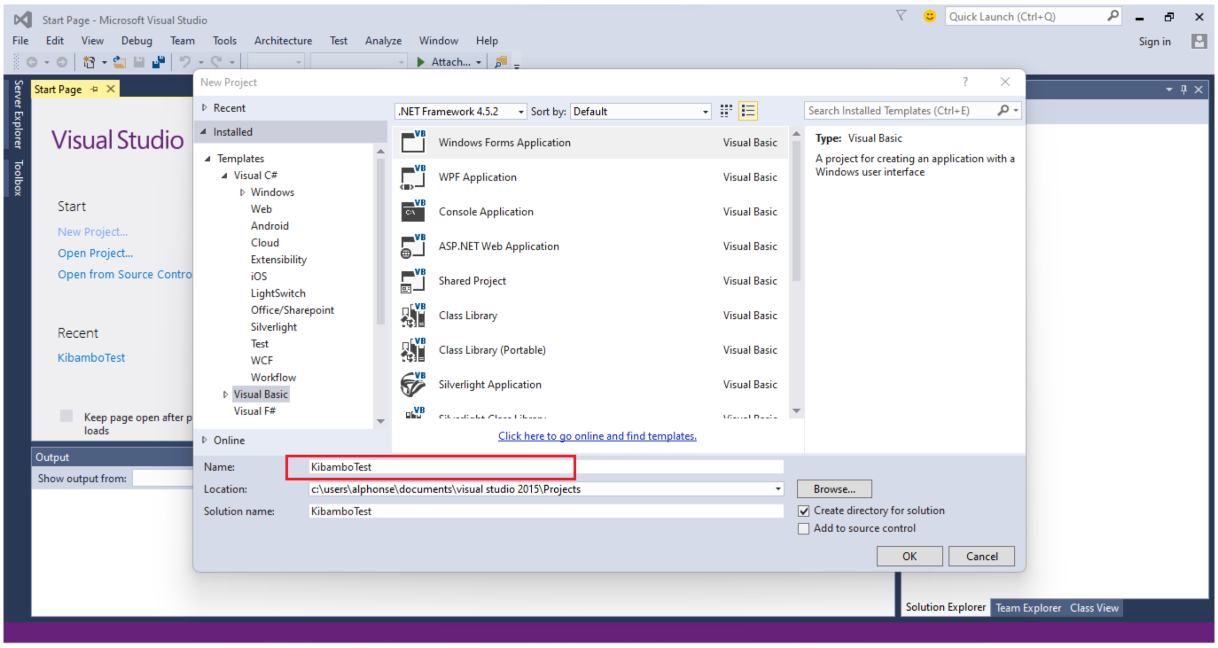
We could use a concise syntax thanks to lambda expressions when instantiating the `Kibambo` object:

```
kibambo = new Kibambo(dgvPersons, dgvSearch, true)
{
    Columns = new KibamboColumn[]
    {
        new KibamboColumn("registration_number", "Registration number")
        {
            Operators = new[] { "==", "!=" }
        },

        new KibamboColumn("age", "Age", KibamboColumnType.Numeric,
"Integer", new string[] { "==", "<", ">", "<=", ">=", "!=" })
    },

    SearchHandler = (row) => row.Visible = false,
    CancelSearchHandler = (row) => row.Visible = true
};
```

Implementation - VB .NET

In this section, we will create a C# Windows Forms program demonstrating the implementation of the El Kibambo library. We will use as IDE « Microsoft Visual Studio Enterprise 2015 ».

Let's create a new C# Windows Forms project that we name « KibamboTest ».


Start Page - Microsoft Visual Studio
File Edit View Debug Team Tools Architecture Test Analyze Window Help
New Project
Recent
Installed
Templates
Visual C#
Windows
Web
Android
Cloud
Extensibility
iOS
LightSwitch
Office/Sharepoint
Silverlight
Test
WCF
Workflow
Visual Basic
Visual F#
Online
Windows Forms Application Visual Basic
WPF Application Visual Basic
Console Application Visual Basic
ASP.NET Web Application Visual Basic
Shared Project Visual Basic
Class Library Visual Basic
Class Library (Portable) Visual Basic
Silverlight Application Visual Basic
Click here to go online and find templates.
Browse...
OK
Cancel
Solution Explorer Team Explorer Class View

.NET Framework 4.5.2 Sort by: Default

Type: Visual Basic

A project for creating an application with a Windows user interface

Name: KibamboTest

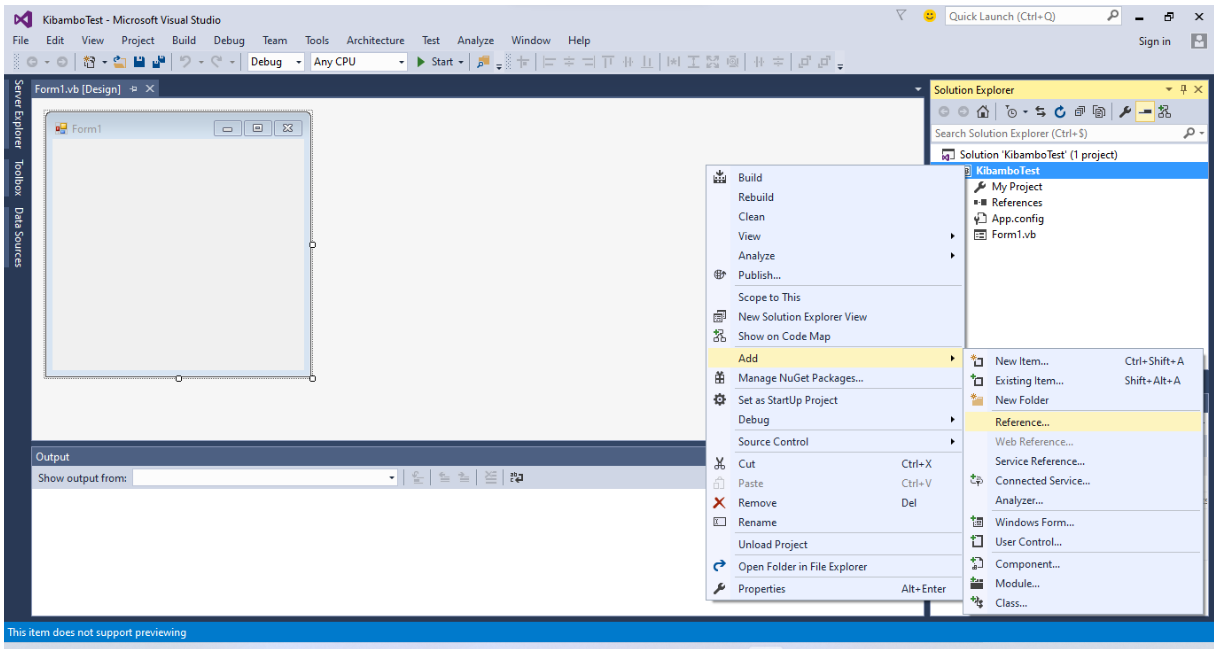
Location: c:\users\alphonse\documents\visual studio 2015\Projects

Solution name: KibamboTest

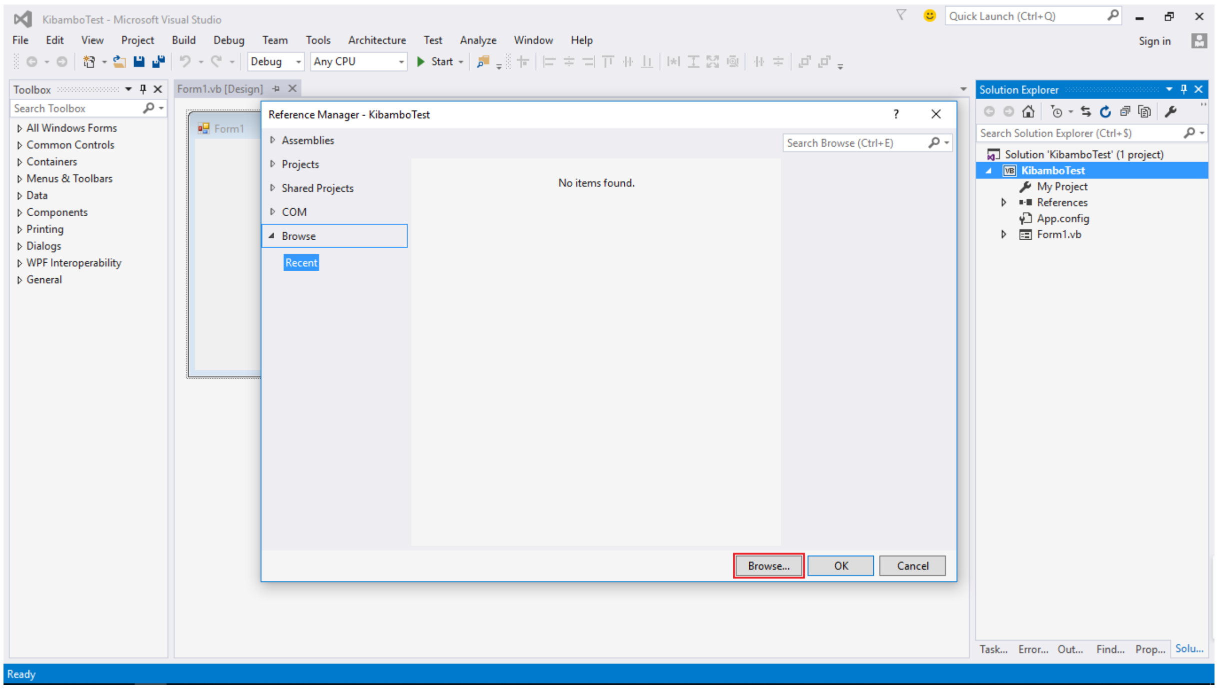
☑ Create directory for solution

☐ Add to source control

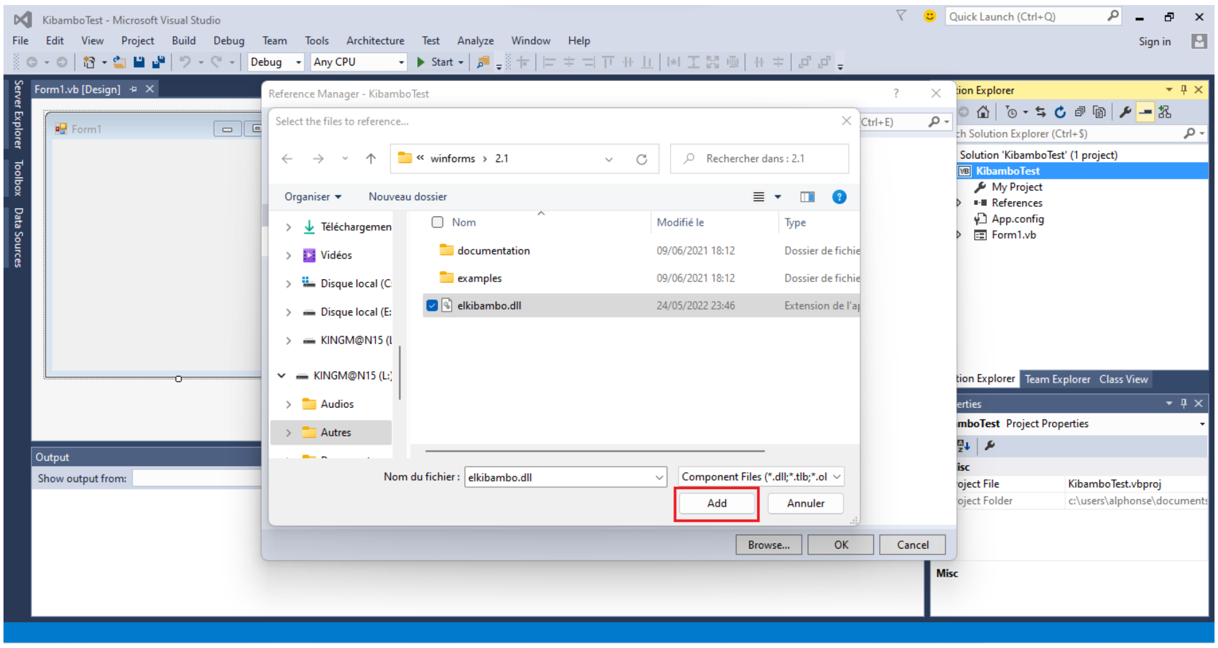
Now let's add the reference to the El Kibambo library. To do this, right-click on the name of the project, choose the “Add” option, then “Reference...”.



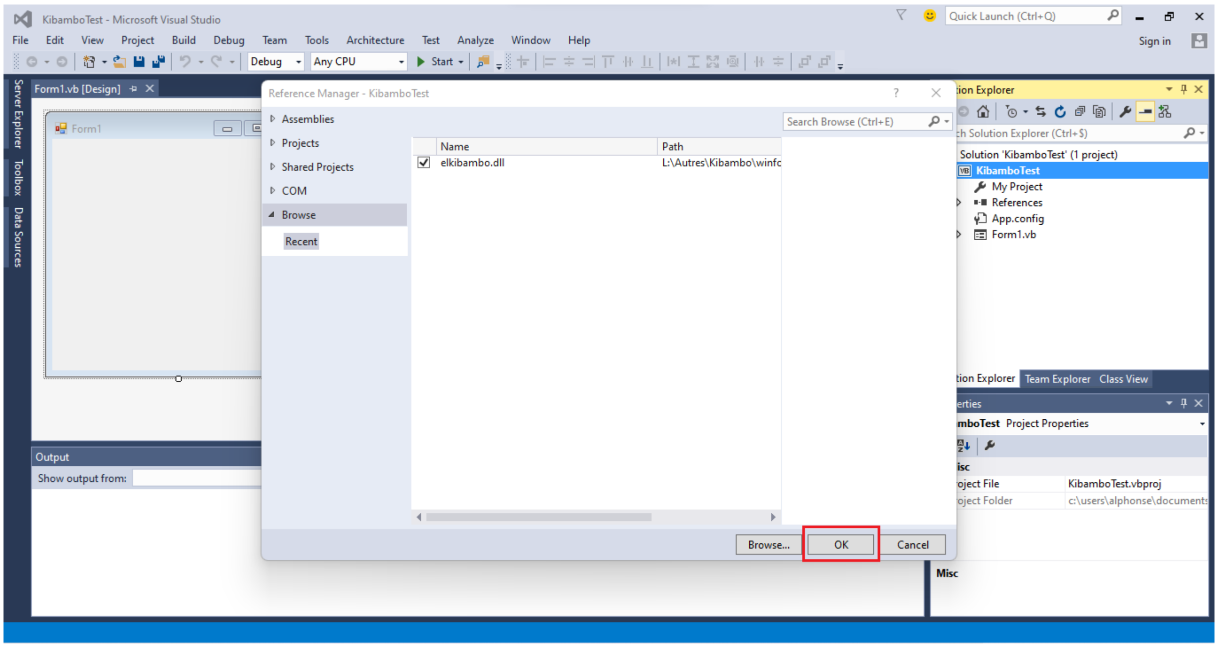
By clicking on the “Reference...” option, Visual Studio will show you an interface similar to:



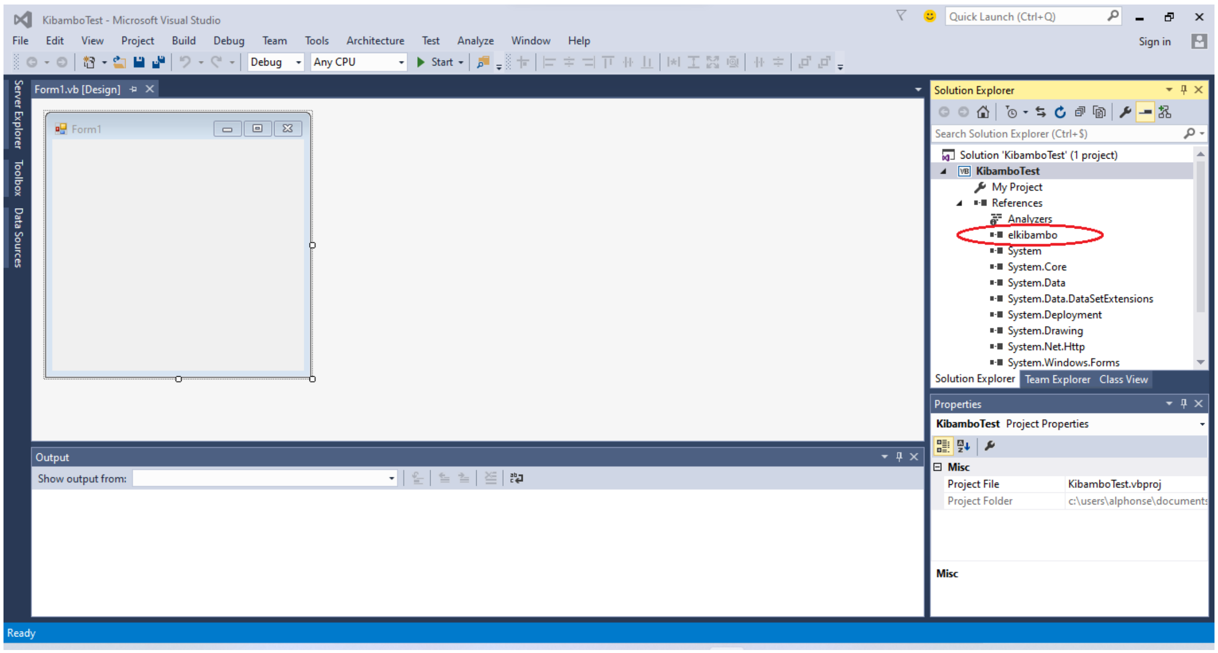
Press the "Browse..." button, this will display the file explorer so you can choose the dll file from the library.


Select the files to reference...
winforms > 2.1
documentation
examples
elkibambo.dll
Nom du fichier : elkibambo.dll
Component Files (*.dll;*.tlb;*.ol
Add
Annuler

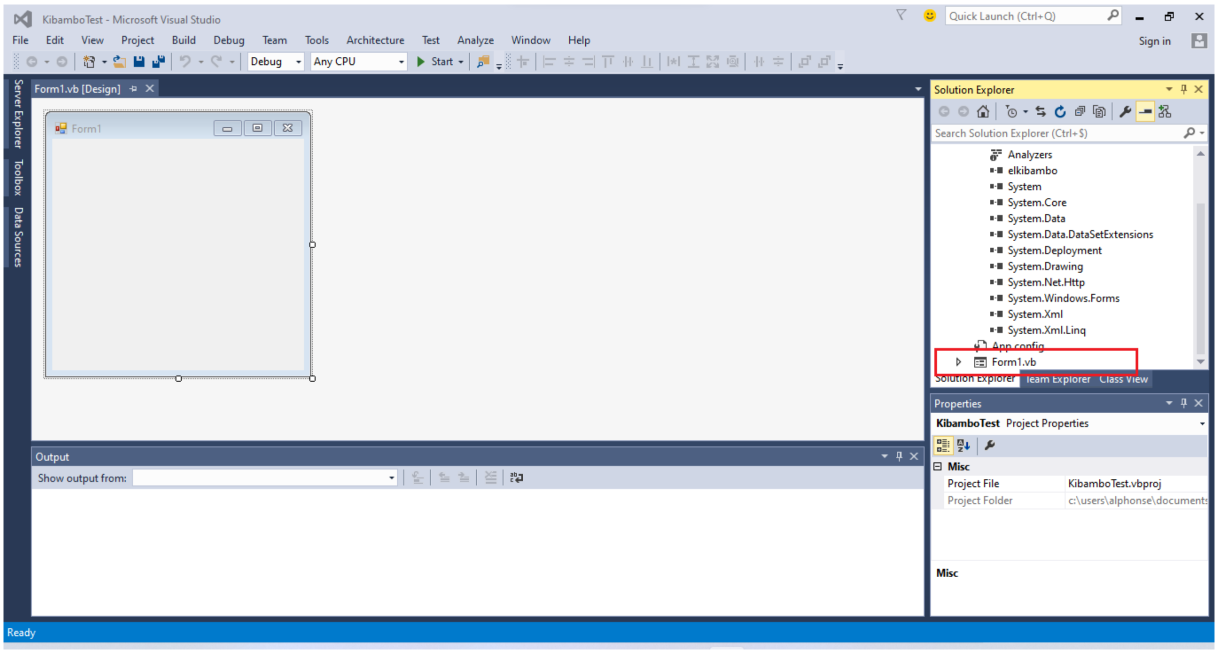
Press the "Add" button, you will have the following interface:



Press the "OK" button to finalize the addition of reference to the library.



Now open the “Form1.vb” file:

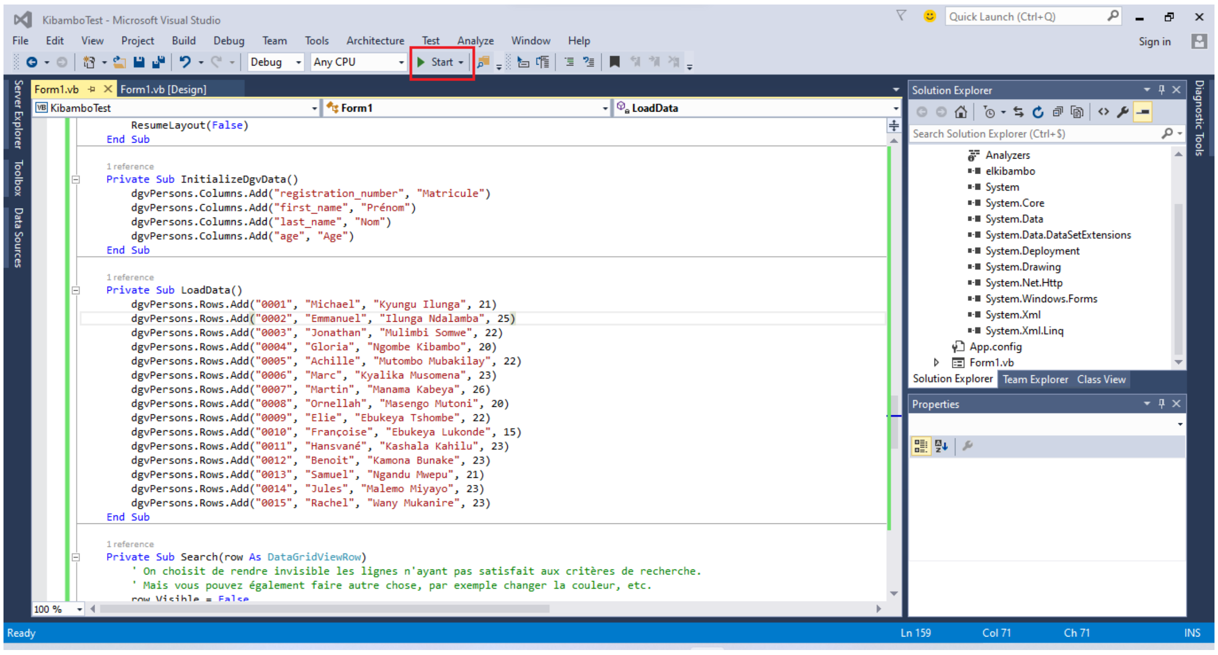


Replace the code in the “Form1.vb” file with the following:

```
Imports ElKibambo
Imports System.ComponentModel

Public Class Form1
    Dim dgvPersons As DataGridView
    Dim dgvSearch As DataGridView

    Dim gbSearch As GroupBox

    Dim btnShowAll As Button
    Dim btnSearch As Button

    Dim kibambo As Kibambo

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        InitializeControls()

        InitializeDgvData()
        LoadData()
        InitializeKibambo()
    End Sub

    Private Sub InitializeControls()
        Dim dgvcs1 As New DataGridViewCellStyle
        Dim dgvcs2 As New DataGridViewCellStyle

        dgvPersons = New DataGridView()
        dgvSearch = New DataGridView()
        gbSearch = New GroupBox()

        btnShowAll = New Button()
        btnSearch = New Button()

        CType(dgvPersons, ISupportInitialize).BeginInit()
        CType(dgvSearch, ISupportInitialize).BeginInit()

        gbSearch.SuspendLayout()
        SuspendLayout()

        ' We ensure that any controls added by the user from the view
designer do not overlap
        ' Delete if needed...
        Controls.Clear()

        '
        ' dgvcs1
        '
        dgvcs1.Alignment = DataGridViewContentAlignment.MiddleCenter
        dgvcs1.BackColor = SystemColors.Control
        dgvcs1.Font = New Font("Trebuchet MS", 9.0!, FontStyle.Bold,
GraphicsUnit.Point, CType(0, Byte))
        dgvcs1.ForeColor = SystemColors.WindowText
        dgvcs1.SelectionBackColor = SystemColors.Highlight
        dgvcs1.SelectionForeColor = SystemColors.HighlightText
        dgvcs1.WrapMode = DataGridViewTriState.True

        '
        ' dgvcs2
        '
        dgvcs2.Font = New Font("Segoe UI", 9.0!, FontStyle.Regular,
GraphicsUnit.Point, CType(0, Byte))

        '
        ' dgvPersons
        '
        dgvPersons.AllowUserToAddRows = False
        dgvPersons.AllowUserToDeleteRows = False
        dgvPersons.AllowUserToResizeColumns = False
        dgvPersons.AllowUserToResizeRows = False
        dgvPersons.Anchor = CType((((AnchorStyles.Top Or
AnchorStyles.Bottom) _
            Or AnchorStyles.Left) _
            Or AnchorStyles.Right), AnchorStyles)
        dgvPersons.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill
        dgvPersons.BackgroundColor = Color.White
        dgvPersons.BorderStyle = BorderStyle.Fixed3D
        dgvPersons.ColumnHeadersDefaultCellStyle = dgvcs1
        dgvPersons.ColumnHeadersHeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.AutoSize
        dgvPersons.EditMode = DataGridViewEditMode.EditProgrammatically
        dgvPersons.Location = New Point(12, 12)
        dgvPersons.MultiSelect = False
        dgvPersons.RowHeadersVisible = False
        dgvPersons.RowsDefaultCellStyle = dgvcs2
        dgvPersons.ScrollBars = ScrollBars.Vertical
        dgvPersons.SelectionMode = DataGridViewSelectionMode.FullRowSelect
        dgvPersons.Size = New Size(433, 336)
        dgvPersons.TabIndex = 0

        '
        ' dgvSearch
        '
        dgvSearch.Anchor = CType(((AnchorStyles.Top Or AnchorStyles.Left) _
            Or AnchorStyles.Right), AnchorStyles)
        dgvSearch.BackgroundColor = Color.White
        dgvSearch.BorderStyle = BorderStyle.Fixed3D
        dgvSearch.ColumnHeadersHeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.AutoSize
        dgvSearch.Location = New Point(10, 19)
        dgvSearch.Size = New Size(268, 111)
        dgvSearch.TabIndex = 1

        '
        ' btnShowAll
        '
        btnShowAll.Anchor = CType((AnchorStyles.Top Or AnchorStyles.Right),
AnchorStyles)
        btnShowAll.Location = New Point(106, 136)
        btnShowAll.Size = New Size(83, 23)
        btnShowAll.TabIndex = 3
        btnShowAll.Text = "Show all"
        btnShowAll.UseVisualStyleBackColor = True
        AddHandler btnShowAll.Click, AddressOf btnShowAll_Click

        '
        ' btnSearch
        '
        btnSearch.Anchor = CType((AnchorStyles.Top Or AnchorStyles.Right),
AnchorStyles)
        btnSearch.Location = New Point(195, 136)
        btnSearch.Size = New Size(83, 23)
        btnSearch.TabIndex = 2
        btnSearch.Text = "Search"
        btnSearch.UseVisualStyleBackColor = True
        AddHandler btnSearch.Click, AddressOf btnSearch_Click

        '
        ' gbSearch
        '
        gbSearch.Anchor = CType((AnchorStyles.Top Or AnchorStyles.Right),
AnchorStyles)
        gbSearch.Controls.Add(btnShowAll)
        gbSearch.Controls.Add(btnSearch)
        gbSearch.Controls.Add(dgvSearch)
        gbSearch.Font = New Font("Segoe UI", 8.25!, FontStyle.Regular,
GraphicsUnit.Point, CType(0, Byte))
        gbSearch.Location = New Point(451, 12)
        gbSearch.Size = New Size(286, 170)
        gbSearch.TabIndex = 2
        gbSearch.TabStop = False
        gbSearch.Text = "Search"

        '
        ' Form1
        '
        AutoScaleDimensions = New SizeF(6.0!, 13.0!)
        AutoScaleMode = AutoScaleMode.Font
        BackColor = Color.White
        ClientSize = New Size(750, 360)
        Controls.Add(gbSearch)
        Controls.Add(dgvPersons)
        Text = "El Kibambo - Illustration"
        StartPosition = FormStartPosition.CenterScreen

        CType(dgvPersons, ISupportInitialize).EndInit()
        CType(dgvSearch, ISupportInitialize).EndInit()

        gbSearch.ResumeLayout(False)
        ResumeLayout(False)
    End Sub

    Private Sub InitializeDgvData()
        dgvPersons.Columns.Add("registration_number", "Registration
number")
        dgvPersons.Columns.Add("first_name", "First name")
        dgvPersons.Columns.Add("last_name", "Last name")
        dgvPersons.Columns.Add("age", "Age")
    End Sub

    Private Sub LoadData()
        dgvPersons.Rows.Add("0001", "Michael", "Kyungu Ilunga", 21)
        dgvPersons.Rows.Add("0002", "Emmanuel", "Ilunga Ndalamba", 25)
        dgvPersons.Rows.Add("0003", "Jonathan", "Mulimbi Somwe", 22)
        dgvPersons.Rows.Add("0004", "Gloria", "Ngombe Kibambo", 20)
        dgvPersons.Rows.Add("0005", "Achille", "Mutombo Mubakilay", 22)
        dgvPersons.Rows.Add("0006", "Marc", "Kyalika Musomena", 23)
        dgvPersons.Rows.Add("0007", "Martin", "Manama Kabeya", 26)
        dgvPersons.Rows.Add("0008", "Ornellah", "Masengo Mutoni", 20)
        dgvPersons.Rows.Add("0009", "Elie", "Ebukeya Tshombe", 22)
        dgvPersons.Rows.Add("0010", "Françoise", "Ebukeya Lukonde", 15)
        dgvPersons.Rows.Add("0011", "Hansvané", "Kashala Kahilu", 23)
        dgvPersons.Rows.Add("0012", "Benoit", "Kamona Bunake", 23)
        dgvPersons.Rows.Add("0013", "Samuel", "Ngandu Mwepu", 21)
        dgvPersons.Rows.Add("0014", "Jules", "Malemo Miyayo", 23)
        dgvPersons.Rows.Add("0015", "Rachel", "Wany Mukanire", 23)
    End Sub

    Private Sub Search(row As DataGridViewRow)
        ' We choose to make invisible the lines that have not satisfied the
search criteria.
        ' But you can also do other things, for example change the color,
etc.
        row.Visible = False
    End Sub

    Private Sub ShowAll(row As DataGridViewRow)
        ' As the lines have been hidden during the search, it seems logical
that we can redisplay them when we decide to cancel a search made.
        ' It is therefore necessary to restore the line to its initial
state before the search.
        row.Visible = True
    End Sub

    Private Sub InitializeKibambo()
        Dim kcm As New KibamboColumn("registration_number", "Registration
number")
        kcm.Operators = {"==", "!="}

        Dim kca As New KibamboColumn("age", "Age",
KibamboColumnType.Numeric, "Entier", {"==", "<", ">", "<=", ">=", "!="})

        kibambo = New Kibambo(
            dgvPersons,
            dgvSearch,
            New KibamboColumn() {
                kcm,
                kca
            },
            AddressOf Search,
            AddressOf ShowAll,
            True
        )

        kibambo.Initialize()
    End Sub

    Private Sub btnSearch_Click(sender As Object, e As EventArgs)
```

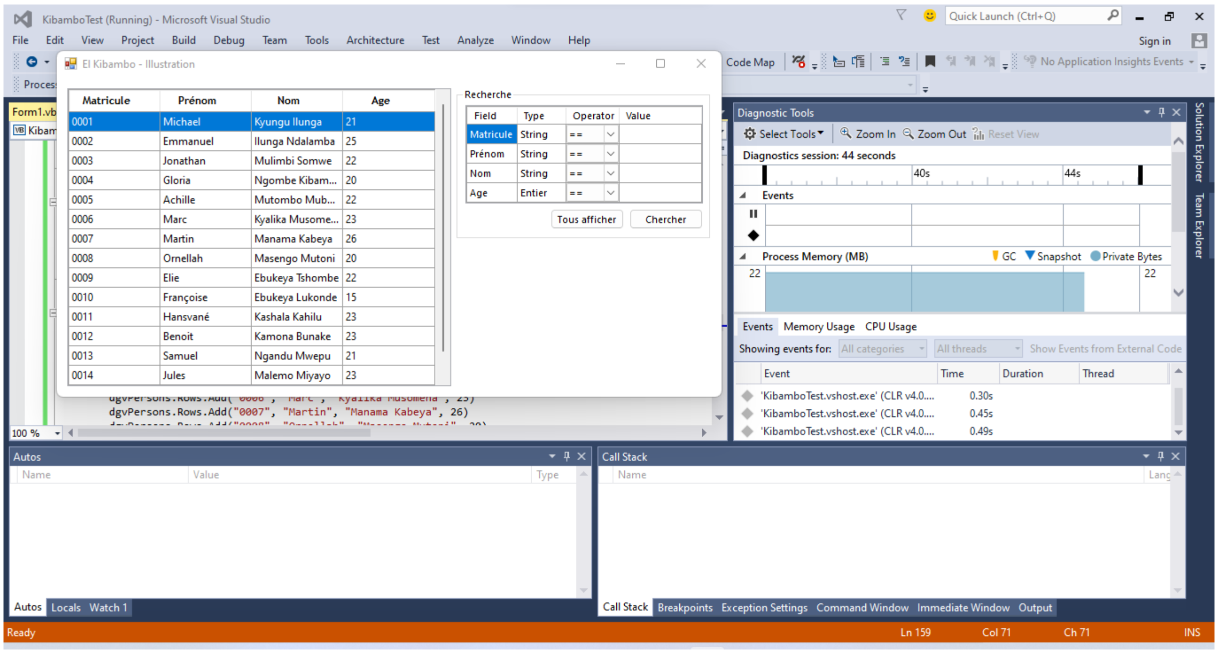
```
        Try
            kibambo.Search()
        Catch ex As KibamboException
            Select Case ex.ErrorCode
                Case KibamboErrorCode.ValueMustBeNumeric
                    MessageBox.Show("You must enter a numeric type value
for the numeric type column!", " Input error", MessageBoxButtons.OK,
MessageBoxIcon.Error)

                Case Else
                    MessageBox.Show("An error occurred while searching. The
error message is " + ex.Message + " !", " Input error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Select
        End Try
    End Sub

    Private Sub btnShowAll_Click(sender As Object, e As EventArgs)
        kibambo.CancelSearch()
    End Sub
End Class
```

Run the application by pressing the "Start" button or using the "F5" key:



```
        ResumeLayout(False)
    End Sub

    1 reference
    Private Sub InitializeDgvData()
        dgvPersons.Columns.Add("registration_number", "Matricule")
        dgvPersons.Columns.Add("first_name", "Prénom")
        dgvPersons.Columns.Add("last_name", "Nom")
        dgvPersons.Columns.Add("age", "Age")
    End Sub

    1 reference
    Private Sub LoadData()
        dgvPersons.Rows.Add("0001", "Michael", "Kyungu Ilunga", 21)
        dgvPersons.Rows.Add("0002", "Emmanuel", "Ilunga Ndalamba", 25)
        dgvPersons.Rows.Add("0003", "Jonathan", "Mulimbi Somwe", 22)
        dgvPersons.Rows.Add("0004", "Gloria", "Ngombe Kibambo", 20)
        dgvPersons.Rows.Add("0005", "Achille", "Mutombo Mubakilay", 22)
        dgvPersons.Rows.Add("0006", "Marc", "Kyalika Musomena", 23)
        dgvPersons.Rows.Add("0007", "Martin", "Manama Kabeya", 26)
        dgvPersons.Rows.Add("0008", "Ornellah", "Masengo Mutoni", 20)
        dgvPersons.Rows.Add("0009", "Elie", "Ebukeya Tshombe", 22)
        dgvPersons.Rows.Add("0010", "Françoise", "Ebukeya Lukonde", 15)
        dgvPersons.Rows.Add("0011", "Hansvané", "Kashala Kahilu", 23)
        dgvPersons.Rows.Add("0012", "Benoit", "Kamona Bunake", 23)
        dgvPersons.Rows.Add("0013", "Samuel", "Ngandu Mwepu", 21)
        dgvPersons.Rows.Add("0014", "Jules", "Malemo Miyayo", 23)
        dgvPersons.Rows.Add("0015", "Rachel", "Wany Mukanire", 23)
    End Sub

    1 reference
    Private Sub Search(row As DataGridViewRow)
        ' On choisit de rendre invisible les lignes n'ayant pas satisfait aux critères de recherche.
        ' Mais vous pouvez également faire autre chose, par exemple changer la couleur, etc.
        row.Visible = False
```

Result :


KibamboTest (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help
El Kibambo - Illustration
Recherche
Tous afficher
Chercher
Diagnostic Tools
Diagnostics session: 44 seconds
Events
Process Memory (MB)
Autos
Call Stack
Ready
Ln 159
Col 71
Ch 71
INS

Matricule	Prénom	Nom	Age
0001	Michael	Kyungu Ilunga	21
0002	Emmanuel	Ilunga Ndalamba	25
0003	Jonathan	Mulimbi Somwe	22
0004	Gloria	Ngombe Kibam...	20
0005	Achille	Mutombo Mub...	22
0006	Marc	Kyalika Musome...	23
0007	Martin	Manama Kabeya	26
0008	Ornellah	Masengo Mutoni	20
0009	Elie	Ebukeya Tshombe	22
0010	Françoise	Ebukeya Lukonde	15
0011	Hansvané	Kashala Kahilu	23
0012	Benoit	Kamona Bunake	23
0013	Samuel	Ngandu Mwepu	21
0014	Jules	Malemo Miyayo	23

Field	Type	Operator	Value
Matricule	String	==	
Prénom	String	==	
Nom	String	==	
Age	Entier	==	

```
dgvPersons.Rows.Add("0007", "Martin", "Manama Kabeya", 26)
```

Historic

The El Kibambo library was born from an observation made by Michael KYUNGU ILUNGA, then a second graduate student at the Higher Pedagogical Institute of Lubumbashi. His finding was the way search functionality was implemented in Windows Forms programs. He saw the way his colleagues and promotion students (also higher) managed to implement the search functionalities, and their way of doing it did not give the end user the possibility of carrying out multi-column searches on data grids in .NET Windows Forms programs with the VB.NET programming language, but also C#.

By the way, let's imagine a data grid containing data on people, for example "registration number", "last name", "first name". At the ISP/L'SHI, the search functionalities only allowed a search on the column serving as an identifier, such as the “registration number” column in the example above. For others, you have the possibility to search on any column, nevertheless on one and only one column. So in the example above, it's up to you, do you want to search by "registration number"? By "last name"? Or by "first name"? But never on more than one column at a time, and search tests were limited to an outright tie.

For Michael KYUNGU ILUNGA, the idea was to allow the end user to perform multicolumn searches, and whose type of comparison is not limited to equality.

During a practical work aimed at creating a complete management program, practical work of the VB.NET course then given by the assistant Deogratias KITENGE KALUME, Michael KYUNGU ILUNGA decided to take action but the result was not praise enough.

He then decided to call on two of his colleagues to be able to take up the challenge together: Benoit KAMONA BUNAKE and Samuel NGANDU MWEPU. For two days, Michael KYUNGU ILUNGA and Benoit KAMONA BUNAKE (Samuel NGANDU MWEPU did not respond favorably) worked on a project then called "Advanced Research".

It was a Windows Forms program written in C# that would allow multi-column search and with several comparison tests on data from relational databases (with the MySQL database management system). At startup, it asks you to choose a table among those of a database, database having been hard-coded in the code. You choose a table, it displays a grid containing all the data of the chosen table and another grid allowing you to carry out the search operations. To be able to perform the search, the program generated an SQL query to perfect the search. In this SQL query, a simple "WHERE" clause was specified for the columns in which the user entered a value but also the chosen comparison operator. This heavy reliance on SQL was the biggest challenge.

Thus, Michael KYUNGU ILUNGA, based on this program, worked on a .NET library project that could be used without necessarily resorting to SQL, a project he then named “El Search”.

The first versions were never released in alpha version, they were just beta versions limited to personal uses of Michael KYUNGU ILUNGA, and one of his close classmates, Hans KASHALA KAHILU. Nevertheless, version 2.1 is the first alpha version of the El Kibambo library.

The library was officially announced on January 5, 2021 under the name "El Search", the library underwent several test phases and was renamed "El Kibambo" in mid-2021.

The word "Kibambo" comes from Gloria NGOMBE KIBAMBO, a leading colleague of Michael KYUNGU ILUNGA. Why this choice of name? Well, for a thousand and one reasons [...] =)