



## Chapter 1: Introduction

### 1.1 About the book

Welcome to *Getting Started with AvisMap Engine*. This book is intended to help you get started understanding and grasping the development approaches and procedures of AvisMap Engine controls, which are components of the GIS platform. This book has 7 chapters. The first one will present a brief introduction about this book. Chapters 2 to 6 discuss how to build a simple application about map browsing and querying, and as well as we will show you how to load AvisMap Engine controls under five different development languages, the most widely used ones at present. The user can read and focus on the familiar or interested parts of the book.

### 1.2 How to read this book

- ✧ In this introduction to AvisMap Engine, you will learn to use AvisMap Engine and other program development languages to build a GIS application with AvisMap Engine. Along the way you will learn how to:
  - ✧ Load AvisMap Engine controls into a project
  - ✧ Use AvisWorkspace controls to open a AvisMap data source
  - ✧ Add and display data on a map with multiple layers
  - ✧ Control panning and zooming
  - ✧ Create a control toolbar
  - ✧ Perform spatial and logical queries
  - ✧
- ✧ Through this book you will also understand the relationship between:
  - ✧ Workspace and datasource
  - ✧ Datasource and dataset
  - ✧ Dataset and layers

### 1.3 Personal motivation

This book is mainly for the developers who have some basic development experience, but is a novice of AvisMap Engine. Users can select different development languages: Visual Basic 6.0, Delphi, Visual C++, Visual Studio C#.NET, or Visual Basic.NET, to implement some basic GIS functions in his own application by following the sample code presented in this book.

### 1.4 Sample Data

The sample data in this demo program, used to illustrate some basic operations and reversible queries, are world maps (world.sdb), which includes graticule (Grid)



## AvisMap GIS Technologies

---

and world map (World). You need to add the two layers on AvisMap to display, and then you can query on the world map which is a Spatial DataBase file World.sdb and the attribute table World.sdd. Please find these two sample data in **C:\Program Files\AvisMap\Avismap Engine 5.0\Sample Data\World\**



## Chapter 2: Loading AvisMap Engine into Visual Basic

You can find the full source code of this example in `C:\Program Files\AvisMap\Avismap Engine 5.0\Sample Projects\Tutorial\Visual Basic\`

### 2.1 Creating a new project

1. Create a working directory such as `C:\MyProject`.
2. Copy data files `World.sdb` and `World.sdd` to working directory (`C:\MyProject`).
3. Start Microsoft Visual Basic.
4. Create a new project under the working directory (`C:\MyProject`).

### 2.2 Loading AvisMap Engine controls

1. Add AvisMap Engine 5.0 control to ToolBox:

Use shortcut key `Ctrl+T` or right-click on ToolBox, then press "Components...", the Components dialog box is displayed, Figure 2-1

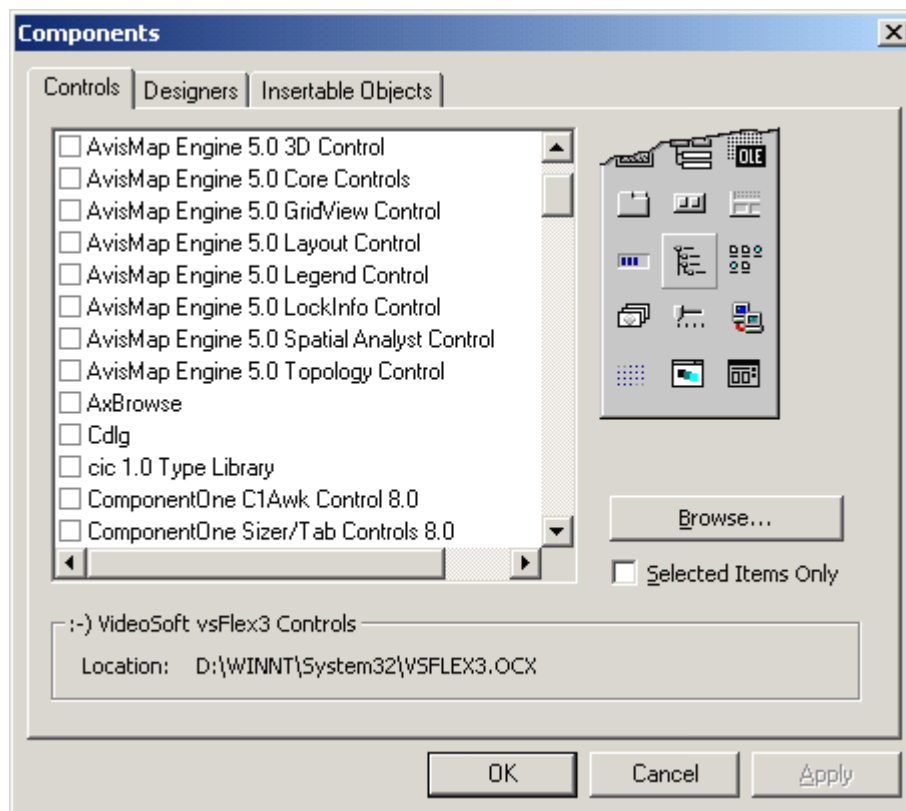


Figure 2-1 Components dialog box



## AvisMap GIS Technologies

In the Components dialog box, select "AvisMap Engine 5.0 Core Controls" (by selecting the checkbox before it) and press OK. Then two controls will appear in the Visual Basic toolbox, Figure 2-2

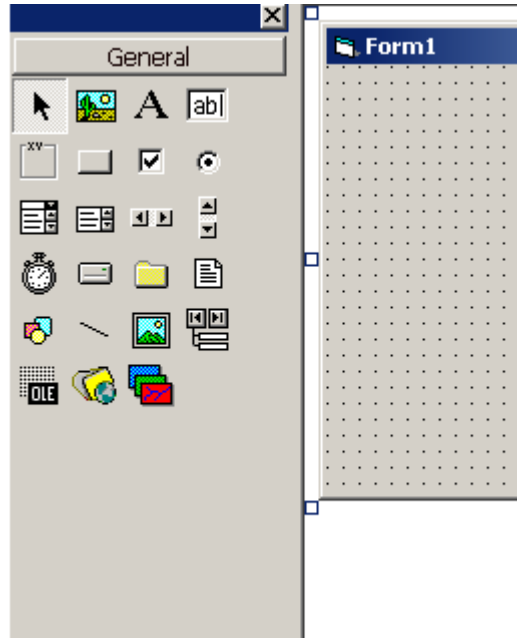


Figure 2-2 Toolbox

2. Add the two Controls to Form1

### 2.3 Opening a map and adding layers

1. Add the following code to the window form:

```
Private Sub Form_Load() 'Connect AvisWorkspace and AvisMap
    AvisMap1.Connect (AvisWorkspacel.Object)
    Dim strAlias As String ' Alias of datasource
    Dim nEngineType As seEngineType 'Data engine type
    Dim strDataSourceName As String ' The absolute path name of datasource
    Dim objDataSource As soDataSource 'Point to opening datasource
    Dim bReadOnly As Boolean 'Whether the data of datasource can read only
    Dim objLayer As soLayer 'Layer object variable, referring to the open
layer
    Dim bAddToHead As Boolean 'whether adding to head or not
    Dim i As Integer 'The cycle variable

    strAlias = "MyDataSource" 'Alias can be named arbitrarily in principle,
but you'd better advise that the main name same to datasource file
    nEngineType = sceSDBPlus 'AvisMap can support various typef-here is SDB+
    strDataSourceName = "c:\MyProject\world.sdb"
    bReadOnly = False ' Not set read only
    'Open the datasource
```



# AvisMap GIS Technologies

```
Set objDataSource = AvisWorkspacel.OpenDataSource(strDataSourceName,
strAlias, nEngineType, bReadOnly)

If objDataSource Is Nothing Then
    MsgBox "Failed to open the datasource£;", vbInformation
Else
    For i = 1 To objDataSource.Datasets.Count
        'Add all layers from datasource to AvisMap
        bAddToHead = True
        Set objLayer =
AvisMap1.Layers.AddDataset(objDataSource.Datasets.Item(i), bAddToHead)
    Next
End If

'Refresh map form
AvisMap1.Refresh
'Modify the selection object style
AvisMap1.selection.Style.PenColor = RGB(231, 77, 0)
AvisMap1.selection.Style.PenWidth = 1
AvisMap1.selection.Style.PenStyle = 1
AvisMap1.selection.Style.BrushStyle = 5
AvisMap1.selection.Style.BrushColor = RGB(115, 69, 140)
AvisMap1.selection.Style.BrushBackColor = RGB(239, 150, 255)
AvisMap1.selection.Style.BrushOpaqueRate = 50
'Release memory
Set objDataSource = Nothing
Set objLayer = Nothing
End Sub

Private Sub Form_Unload(Cancel As Integer)
    AvisMap1.Disconnect
    AvisWorkspacel.Close
End Sub
```

2. To run above code and then the result will be displayed as follows, Figure 2-3

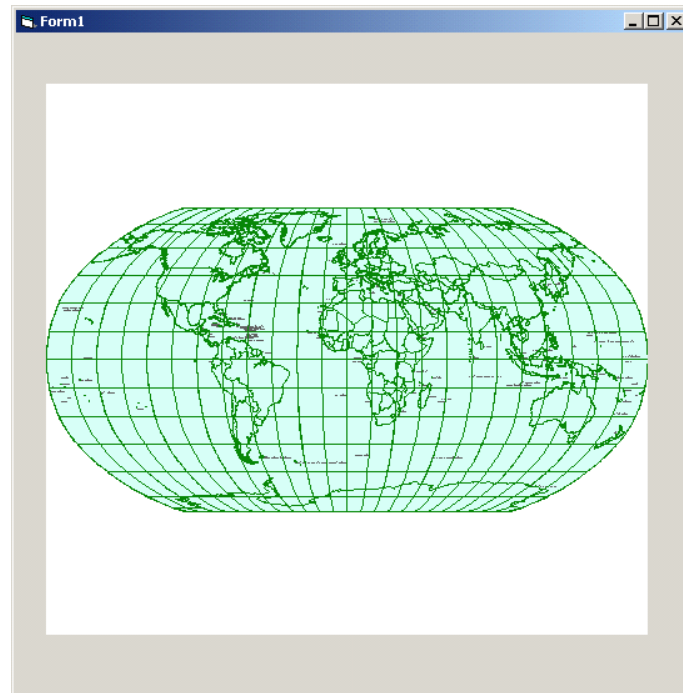


Figure 2-3 The result interface

## 2.4 Browsing the map

It is very convenient to perform map operations with AvisMap Engine, such as zoom in, zoom out, zoom freely, pan map, show the full extent of map, draw point or line, etc. Here, we list some map operations as example in this program .

First, add five command buttons to form1, and then set their properties as follows (Set the properties of others by default.), Table 2-1

Table 2-1 Button properties

<b>Name</b>	<b>Caption</b>
cmdPan	Pan
cmdZoomIn	ZoomIn
cmdZoomOut	ZoomOut
cmdZoomFree	ZoomFreely
cmdViewEntire	FullExtent

Then, add following code to each Click event to implement corresponding function.

```
Private Sub Form_Unload(Cancel As Integer)
    AvisMap1.Disconnect
    AvisWorkspace1.Close
End Sub

Private Sub cmdPan_Click()
```



## AvisMap GIS Technologies

```
'pan
AvisMap1.Action = scaPan
End Sub

Private Sub cmdViewEntire_Click()
'view entire
AvisMap1.ViewEntire
End Sub

Private Sub cmdZoomFree_Click()
'zoom freely
AvisMap1.Action = scaZoomFree
End Sub

Private Sub cmdZoomIn_Click()
'zoomin
AvisMap1.Action = scaZoomIn
End Sub

Private Sub cmdZoomOut_Click()
'zoomout
AvisMap1.Action = scaZoomOut
End Sub
```

The following picture shows the result of map zoom in operation, Figure 2-4

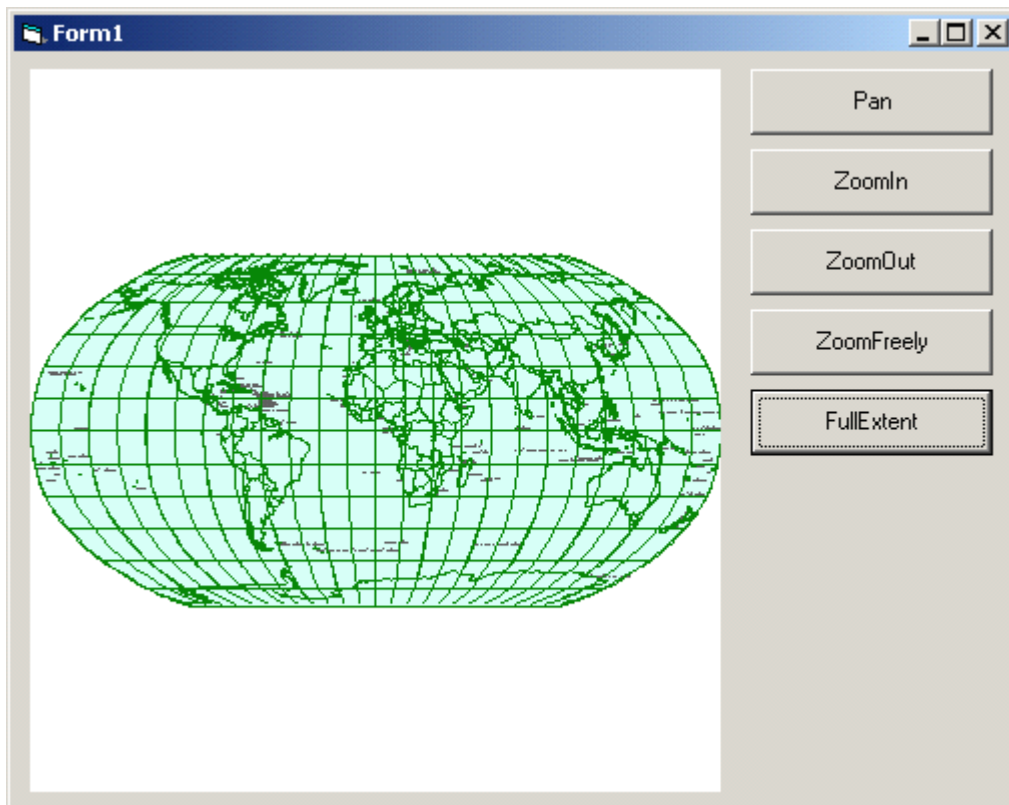


Figure 2-4 The result of map zoom in

## 2.5 Querying Properties

First, add a command button to form and then set its property as the following table:

Table 2-2 Button property

Name	Caption
cmdSelect	Identity

Second, add below code in each Click event to implement related functions:

```
Private Sub cmdSelect_Click()  
    AvisMap1.Action = scaSelect  
End Sub
```

Finally, add the following code to the GeometrySelected event of AvisMap1:

```
Private Sub AvisMap1_GeometrySelected(ByVal nSelectedGeometryCount As  
Long)  
    Dim objRecordSet As soRecordset 'Property object  
    Dim i As Integer 'Cycle variable  
    Dim strName(40) As String 'Storage property name
```



## AvisMap GIS Technologies

```
Dim strValue(40) As String ' Storage property value
Dim strMessage As String 'Show all infomation

'Pick up property information of selected object
Set objRecordSet = AvisMap1.selection.ToRecordset(False)
objRecordSet.MoveFirst 'Move to the first record
For i = 1 To objRecordSet.FieldCount
    strName(i - 1) = objRecordSet.GetFieldInfo(i).Name 'Get the
property name
    strValue(i - 1) = objRecordSet.GetFieldValue(i) 'Get the
property value
Next

strMessage = ""
For i = 1 To objRecordSet.FieldCount
    strMessage = strMessage & strName(i - 1) & ": " & strValue(i -
1) & Space(5) & vbCrLf
Next
MsgBox strMessage
Set objRecordSet = Nothing 'Release memory

End Sub
```

The following picture shows the result when querying properties from the selected object in the map, Figure 2-5

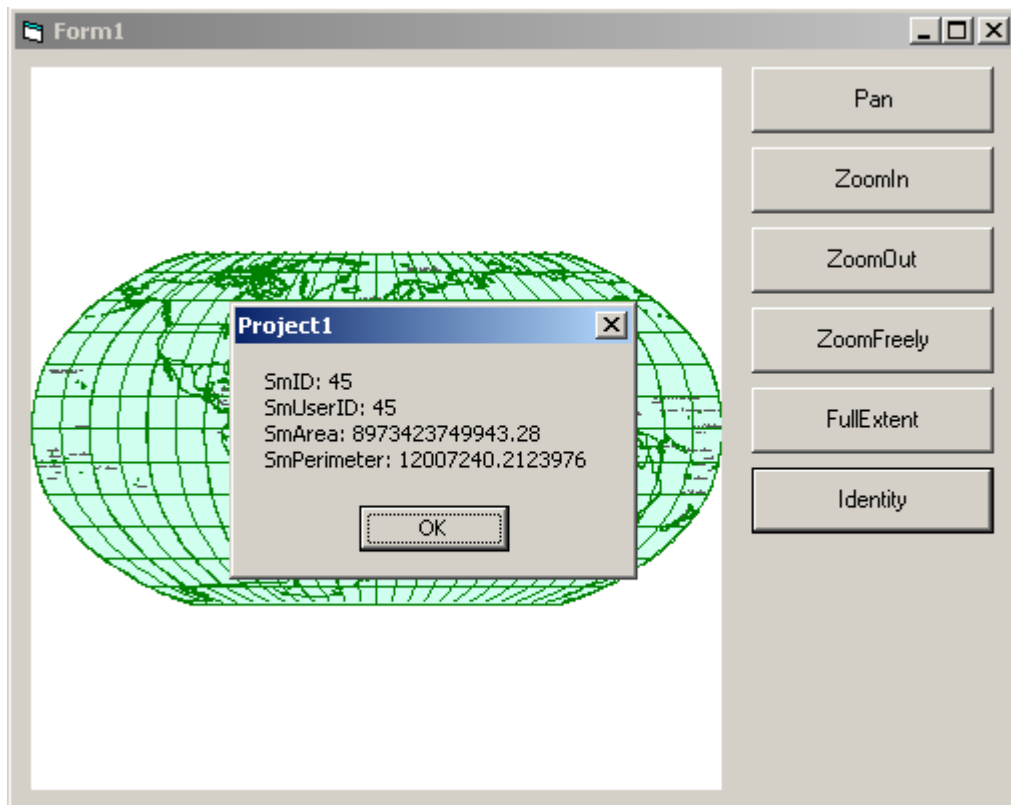


Figure 2-5  
The result of identifying map

## 2.6 Querying Map

1. Add a command button and a textbox to form, and then set their properties as following:

Table 2-3 Controls' properties

Control	Name	Caption
Button	cmdQueryMap	QueryMap
TextBox	txtExpression	---

2. Add the following code to Click event of button:

```
Private Sub cmdQueryMap_Click()  
    Dim objDtVector As soDatasetVector 'Vector dataset variable  
    Dim objRecordset As soRecordset 'Attribute dataset variable  
    Dim objSelection As soSelection 'Selection variable  
    'Get the vector dataset World_contries in this datasource on  
    which the query would be performed
```



## AvisMap GIS Technologies

```
Set objDtVector =
AvisWorkspacel.Datasources.Item("MyDataSource").Datasets("World_count
ries")

If objDtVector Is Nothing Then
    MsgBox "Error to open dataset!", vbInformation
    Exit Sub
End If

'Query attribute data from dataset (The method of query only
apply to the class object of soDatasetVector)
Set objRecordset = objDtVector.Query(txtExpression.Text, True)

If objRecordset Is Nothing Then
    Exit Sub
Else
    'Add the data which have been queried to selection
    Set objSelection = AvisMap1.selection
    objSelection.FromRecordset objRecordset
    'Refresh the map
    AvisMap1.Refresh
End If

Set objDtVector = Nothing
Set objRecordset = Nothing
Set objSelection = Nothing
End Sub
```

3. Run the program and type a SQL expression, e.g. `smid>50`, in the Query text box, and Click 'QueryMap', then the results will be displayed as follows, Figure 2-6

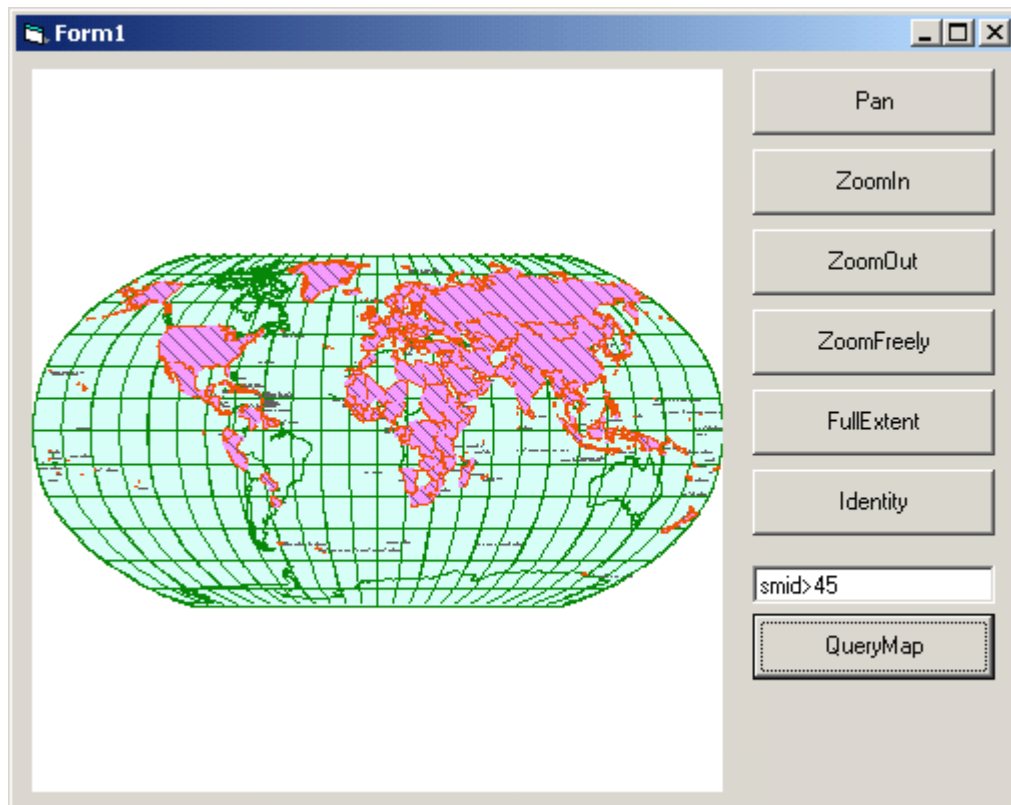


Figure 2-6 The result of query map