MAPWINGIS REFERENCE MANUAL

A function guide for the free MapWindow GIS ActiveX map component.

MAPWINDOW OPEN SOURCE TEAM
WWW.MAPWINDOW.ORG

JULY 23, 2007

Prepared for print publication by:

Daniel P. Ames, Ph.D., P.E
dpames@gmail.com
Geospatial Software Lab
Idaho State University
1784 Science Center Drive.
Idaho Falls, Idaho 83402
USA

Assembled into a single document by:

Dinesh Grover
P-Line Technologies
P-41, Dhanraj Park, Wakad
Pune 411057
India

Originally written by multiple student programmers at:

Utah Water Research Lab, Utah State University, Logan, Utah, USA
Department of Geosciences, Idaho State University, Pocatello, Idaho 83409, USA

Available for purchase in hard copy through:

www.lulu.com

1 MapWindow GIS Open Source Project

You are reading the “MAPWINGIS REFERENCE MANUAL: A function guide for the free MapWindow GIS ActiveX map component”. This manual presents all of the objects and enumerations contained within the MapWinGIS ActiveX component and is intended to be used as a reference guide. Additional step by step instructions and code samples can be found on the MapWindow.org web site and in the book: Getting Started with the MapWinGIS ActiveX Control. The remainder of this section introduces the MapWindow GIS Open Source Project, of which the MapWinGIS ActiveX Control is a part.

1.1 Introduction

MapWindow is an open source geographic information system (GIS) and an application programming interface (API) distributed under the Mozilla Public License (MPL), built upon the Microsoft Dot Net Framework 2.0. The project is maintained by an active group of international developers who regularly release updates and bug fixes through the MapWindow.org web site.

In 2005, the United States Environmental Protection Agency adopted MapWindow as the core GIS platform for its BASINS watershed analysis and modeling software which is used by environmental professionals at all levels of U.S. government and internationally. This was followed by adoption at the United Nations University and many other governmental, non-profit, and commercial entities throughout the world. Indeed, from November 2006 through July 2007 the software was downloaded over 60,000 times and has over 7,000 members on the opt-in mailing list. There are also currently 630 subscribers to source code commit notification mailing lists, and 43 regular source code committers.

The MapWindow components and end user application support manipulation, analysis and viewing of geospatial data in many standard GIS data formats. Hence, MapWindow is a mapping tool, a GIS modeling system, and a GIS API, in a redistributable open source form that is appealing for its simplicity of use and for running on Microsoft Windows. This paper presents the project history, current status, key elements and future plans.

1.2 Project Background

In 1998, researchers at Utah State University in Logan, Utah, USA were using MapObjects LT 1.0 from Environmental Systems Research Institute (ESRI) as a GIS component in several research projects. However, restrictive redistribution licenses and the inability to edit vector or attribute data files or display and interact with raster data became significant barriers to use of this component.

Indeed, dynamic and interactive functions that require low level access to spatial data were not well-supported in freely distributable Windows-based GIS programming components at that time. An alternative approach for these projects was to move to a proprietary desktop GIS system (e.g. ArcView 3.1), writing the needed tools as extensions. However, many applications in environmental science and engineering are more suited to a standalone environment due to the need to distribute the tool to individuals who do not own a desktop GIS or are not GIS experts.

The core MapWinGIS component was developed to meet this need for an alternative to existing proprietary GIS components. MapWinGIS is an ActiveX control that gives full ability to display, query, manipulate, and use spatial data. Because of the engineering-oriented nature of the projects that necessitated its development, MapWinGIS was particularly optimized for rapid display of data and is intended to be a fully functional model interface, not simply a map viewer. Development focuses included high-speed image and raster display, limiting the amount of re-drawing, and including APIs for low-level access to spatial data.

The MapWindow GIS desktop application wraps all of the MapWinGIS ActiveX functionality, providing end users with a common legend, toolbar and a consistent interface that can be extended by adding plug-ins or a customized configuration file. In short, MapWinGIS ActiveX was built to add GIS functionality to custom end user applications and MapWindow GIS desktop was built to help users deploy GIS-enabled applications assembled from pre-built plug-ins and datasets.

1.3 Project Components

1.3.1 MapWindow GIS Application

The MapWindow application is a lightweight spatial data viewer with an intentionally slim default interface intended to simplify its use and improve adaptability to specific uses. The main GUI and functionality can be extended through plug-ins and scripts which add capabilities as needed. Also the default application itself can be customized to change the look, feel and even its title-bar name at runtime.

By default, the MapWindow layout includes a map, a legend, and a preview-map. Built-in toolbar buttons allow one to manage project files (collections of data layers), to print, and to navigate the map. Figure 1 shows the MapWindow interface with the Shapefile Editor plug-in enabled and with the title bar customized for a specific project. Both open source and commercial plug-ins have been developed by users worldwide for a variety of applications; many of these are available for download from the MapWindow.org web site.

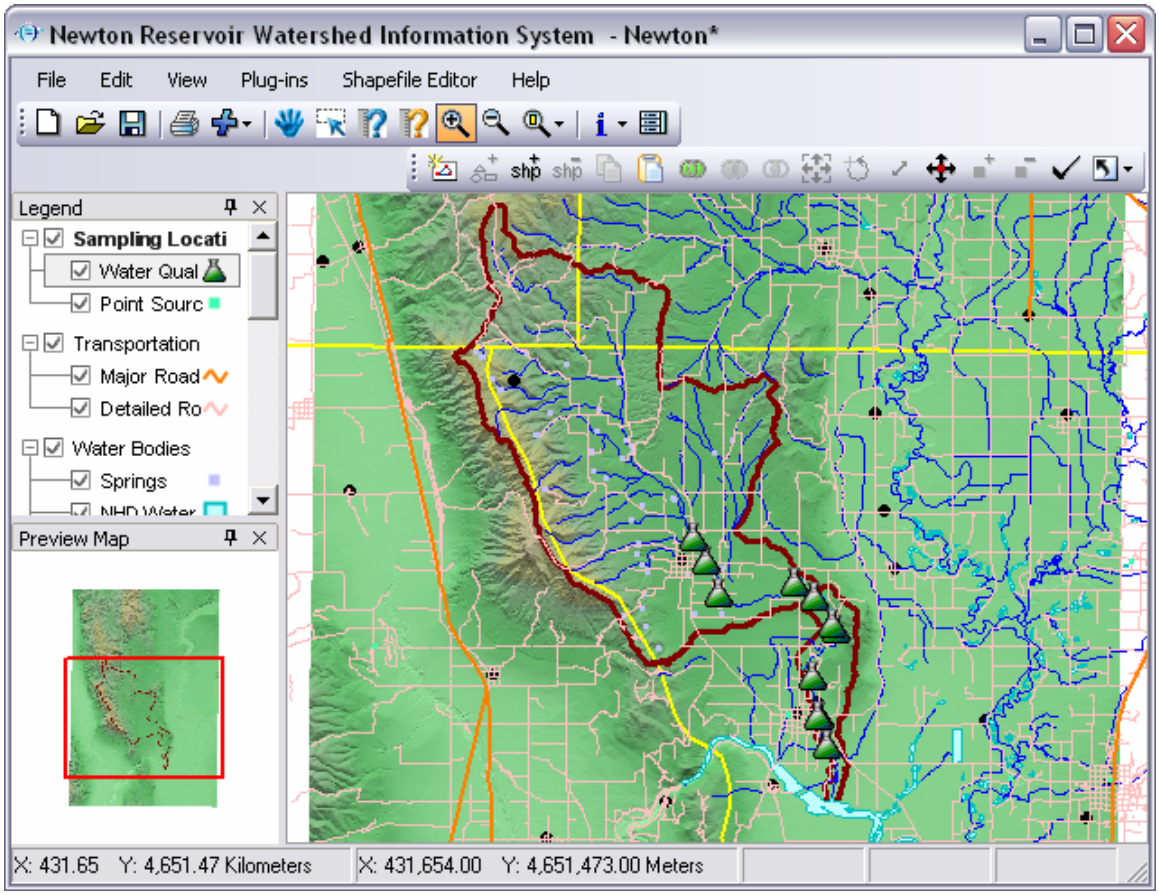


Figure 1: Basic MapWindow interface with Shapefile Editor enabled.

In addition to building plug-ins, one can customize the look and feel of the MapWindow application by changing the included "default.mwcfg" XML configuration file. Changes to this file can affect title bar, splash screen, about box, plug-in loading and window layout settings, among others.

An XML project file allows users to save a list of loaded data layers and session settings. The "Newton Reservoir Information System" project in Figure 1 is an example project included in the MapWindow installation package. All of the data loaded into MapWindow and all settings (e.g. symbology, loaded plug-ins, and project projection), are saved to the project file.

1.3.2 MapWindow Plug-ins

MapWindow has an extensible architecture that allows developers to write plug-ins to add functionality using Dot Net compatible languages such as VB.Net or C#. This is done by implementing a specific plug-in interface in a custom DLL file which is placed in the application directory and automatically detected at runtime. Plug-ins can also be written and compiled directly in the MapWindow plug-in editor—eliminating the need for an external development environment.

The MapWindow plug-in interface provides extensive capabilities allowing the developer to use as little or as much as needed to accomplish a particular goal. Default plug-ins packaged with MapWindow include tools for editing shapefiles and attributes, identifying features and performing common geoprocessing tasks.

Plug-in developers retain all copyright and licensing rights for their work and hence can redistribute their products as proprietary, open source, or otherwise as needed.

1.3.3 *MapWindow ActiveX Control*

The core MapWindow component is an ActiveX control, "MapWinGIS.ocx" that can be used in Visual Basic or any language that supports ActiveX (e.g. C#, Microsoft Access, Microsoft Excel). MapWinGIS has been optimized for fast image and raster display, limiting the amount of redrawing, and including APIs for low-level access to spatial data. A simple Visual Basic program using the control and just a few lines of code is shown in Figure 2.

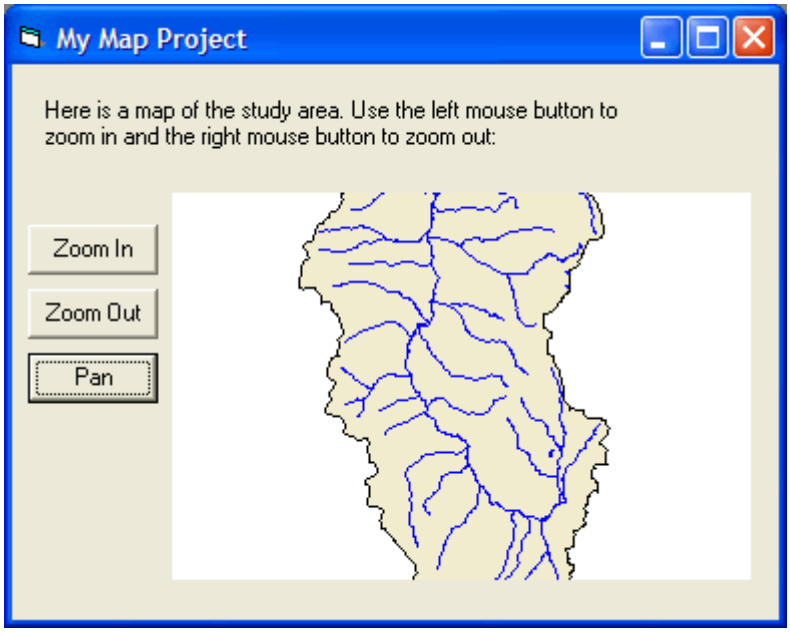


Figure 2: A Simple Mapping Application

1.4 Summary

The MapWindow GIS project is a dynamic and active effort to build and distribute open source GIS tools that integrate tightly with the Microsoft Windows operating system. Supported by a large number of commercial and non-commercial funding sources, volunteer developers, and student research activities, this project is expected to grow in the foreseeable future and continue to be a model of a successful open source for geoinformatics software development and use.
The core MapWindow component is an ActiveX control, "MapWinGIS.ocx" that can be used in Visual Basic or any language that supports ActiveX (e.g. C#, Microsoft Access, Microsoft Excel). MapWinGIS has been optimized for fast image and raster display, limiting the amount of redrawing, and including APIs for low-level access to spatial data.

The MapWinGIS ActiveX control was created with Microsoft Visual C++ and is compatible with all programming languages that can handle OLE. If you have success/failure stories to report with non-Microsoft products, we'd be happy to hear of them.

2 Getting the Latest Version of the Free MapWinGIS ActiveX

As an active open source development effort, the MapWindow GIS project and its MapWinGIS ActiveX control are regularly updated an improved. Because of this, you should always consider acquiring the latest version of the component before beginning any new development project. There are five ways you can get a copy of the MapWinGIS ActiveX control:

1) A CD can be purchased from the www.MapWindow.org web site for a nominal fee. You can find the installer by inserting the CD into your hard drive and following the link on the auto start index page that should automatically load. This will be the latest version of the component at the time you order the CD.

2) Alternatively, you can download the MapWinGIS installer directly from the MapWindow GIS project home page: www.MapWindow.org. Follow the download link and select the “MapWinGIS ActiveX Control” installer option.

3) The MapWinGIS component is also included in the installation package for the MapWindow GIS version 4 desktop application as it is used heavily by that product. The latest installer for the MapWindow GIS desktop application can be found on the MapWindow project web site, and a version of that installer is also included on the CD with this book.

4) Another option is to download the current pre-release build of the MapWinGIS ActiveX control directly from the MapWindow GIS project web site. If you take this approach, you will be on the bleeding edge, using a version of the component that has not been released for public consumption, but has any and all of the very latest features and bug fixes (and maybe new bugs!). The code repository where this version resides can be browsed at http://svn.mapwindow.org/svnroot/MapWindow4Dev/Bin/. You are looking for the file called "MapWinGIS.ocx" in this folder (or equivalent folder with later version number).

5) Finally, if you are skilled with C++, you can download the current source code to the MapWinGIS ActiveX control directly from the Subversion code repository on the MapWindow GIS project web site. Instructions for doing this are given at http://www.MapWindow.org/svn.php. Downloading the source code and recompiling the component are topics beyond the scope of this book, but you are welcome to take this approach if it fits your needs. If you are comfortable working with the C++ source code to the component, then you might want to consider joining the MapWindow GIS team http://www.MapWindow.org/team.php and contributing your own enhancements, improvements, and other great ideas to the project!

3 Objects

Below is a list of all MapWinGIS objects. Sample code snippets are provided for most of the functions and subroutines. More samples are available on the MapWindow.org web site under the discussion forums.

3.1 ESRIGridManager

An ESRI grid manager object provides functions which facilitate using ESRI grids with MapWinGIS. The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.1.1 Functions

3.1.1.1 CanUseESRIGrids

Gets whether or not MapWinGIS can open ESRI grids.
Note: Since ESRI grids are proprietary, MapWinGIS can only open them if another ESRI product has already been installed which provides the files needed to manipulate ESRI grids.

VB.NET Usage

Function CanUseESRIGrids() As Boolean
Parameters

ReturnValue	A boolean value representing whether or not MapWinGIS can open ESRI grids.

Sample Code

```
Private Sub CanUseESRIGrids()
    Dim ESRIgridmanager As New MapWinGIS.ESRIGridManager()
    'Check if MapWinGIS can open ESRI grids
    If ESRIgridmanager.CanUseESRIGrids Then
        'If ESRI grids can be opened, tell the user in a message box
        MsgBox("MapWinGIS can open ESRI grids on your system.")
    End If
End Sub
```

3.1.1.2 DeleteESRIGrids

Deletes the specified ESRI grid.

VB.NET Usage

Function DeleteESRIGrids(Filename As String) As Boolean

Parameters

Filename	The filename of the ESRI grid to be deleted.
ReturnValue	A boolean value representing the success or failure of deleting the specified ESRI grid.

Sample Code

```
Private Sub DeleteESRIGrid()
    Dim ESRIgridmanager As New MapWinGIS.ESRIGridManager()
    Dim success As Boolean
    'Delete the ESRI grid
```

```
    success = ESRIgridmanager.DeleteESRIGrids("C:\test.grd")
End Sub
```

3.1.1.3 IsESRIGrid

Gets whether or not the specified file is an ESRI grid.

VB.NET Usage

Function IsESRIGrid(Filename As String) As Boolean

Parameters

Filename	The filename of the grid to be checked.
ReturnValue	A boolean value representing whether or not the specified file is an ESRI grid.

Sample Code

```
Private Sub IsESRIGrid()
    Dim ESRIgridmanager As New MapWinGIS.ESRIGridManager()
    Dim filename As String
    'Set the file to be checked
    filename = "C:\test.grd"
    'Check if the specified file is an ESRI grid
    If ESRIgridmanager.IsESRIGrid(filename) Then
        'If the specified file is an ESRI grid, display a message to the user
        MsgBox("The file " + filename + " is an ESRI grid.")
    End If
End Sub
```

3.1.2 Properties

3.1.2.1 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.1.2.2 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
    'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
```

```
    Implements MapWinGIS.ICallback
      '...
#Region "ICallback Members"
    Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
            'Display the error message in a label
            LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
            'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
            'Display the message in a label
        Label2.Text = Message
    End Sub
#End Region
            '...
```

3.1.2.3 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
    Private Sub LastErrorCode()
        Dim errorCode As Integer
        'Get the last error in the map
        errorCode = Map1.LastErrorCode
        'Display message box giving error message for the last error in the map
        MsgBox(Map1.get_ErrorMsg(errorCode))
    End Sub
```

3.2 Extents

3.2.1 Subs

3.2.1.1 GetBounds

Gets the bounds of the extents object.

VB.NET Usage

Sub GetBounds(ByRef xMin As Double, ByRef yMin As Double, ByRef zMin As Double, ByRef xMax As Double, ByRef yMax As Double, ByRef zMax As Double)

Parameters

xMin	Reference parameter. Returns the minimum x value for the extents object.
yMin	Reference parameter. Returns the minimum y value for the extents object.
zMin	Reference parameter. Returns the minimum z value for the extents object.
xMin	Reference parameter. Returns the maximum x value for the extents object.
yMin	Reference parameter. Returns the maximum y value for the extents object.
zMin	Reference parameter. Returns the maximum z value for the extents object.

Sample Code

```
    Private Sub ExtentsGetBounds()
        Dim extents As New MapWinGIS.Extents()
        Dim xMin As Double, yMin As Double, zMin As Double, xMax As Double, yMax As Double, zMax As Double
        'Get the bounds of the extents object
        extents.GetBounds(xMin, yMin, zMin, xMax, yMax, zMax)
    End Sub
```

3.2.1.2 GetMeasureBounds

Gets the minimum and maximum measure bounds for the extents object. Note: Measure bounds only apply to shapefiles containing measure data.

VB.NET Usage

Sub GetMeasureBounds(ByRef mMin As Double, ByRef mMax As Double)

Parameters

mMin	Reference paramenter. Returns the minimum measure bound for the extents object.
mMax	Reference paramenter. Returns the minimum measure bound for the extents object.

Sample Code

```
Private Sub ExtentsMeasureBounds()
    Dim extents As New MapWinGIS.Extents()
    Dim mMin As Double, mMax As Double
    'Get the minimum and maximum measure bounds for the extents object
    extents.GetMeasureBounds(mMin, mMax)
End Sub
```

3.2.1.3 SetBounds

Sets the bounds for the extents object.

VB.NET Usage

Sub SetBounds(xMin As Double, yMin As Double, zMin As Double, xMax As Double, yMax As Double, zMax As Double)

Parameters

xMin	The new minimum x value for the bounds of the extents object.
yMin	The new minimum y value for the bounds of the extents object.
zMin	The new minimum z value for the bounds of the extents object.
xMax	The new maximum x value for the bounds of the extents object.
yMax	The new maximum y value for the bounds of the extents object.
zMax	The new maximum z value for the bounds of the extents object.

Sample Code

```
Private Sub ExtentsSetBounds()
    Dim extents As New MapWinGIS.Extents()
    Dim xMin As Double, yMin As Double, zMin As Double, xMax As Double, yMax As Double, zMax As Double
    'Set the minimum and maximum values for the x, y, and z bounds values
    xMin = 100
    yMin = 100
    zMin = 0
    xMax = 500
    yMax = 500
    zMax = 0
    'Set the bounds of the extents object
    extents.SetBounds(xMin, yMin, zMin, xMax, yMax, zMax)
End Sub
```

3.2.1.4 SetMeasureBounds

Sets the measure bounds of the extents object. Note: Measure bounds only apply to shapefiles containing measure data.

VB.NET Usage

Sub SetMeasureBounds(mMin As Double, mMax As Double)

Parameters

mMin	The new minimum measure bound for the extents object.
mMax	The new maximum measure bound for the extents object.

Sample Code

```
Private Sub ExtentsSetMeasureBounds()
    Dim extents As New MapWinGIS.Extents()
    Dim mMin As Double, mMax As Double
    'Set the minumum and maximum measure bounds values to be used
    mMin = 50
    mMax = 100
    'Get the minimum and maximum measure bounds for the extents object
    extents.SetMeasureBounds(mMin, mMax)
End Sub
```

3.2.2 Properties

3.2.2.1 mMax

The maximum measure bound in the exents object. Note: Measure bounds only apply to shapefiles containing measure data.

VB.NET Usage

ReadOnly Property mMax() As Double

Parameters

ReturnValue	The maximum measure bound for the extents object.

Sample Code

```
Private Sub Extents_mMax()
    Dim extents As New MapWinGIS.Extents()
    Dim mMax As Double
    'Get the maximum measure bound for the extents object
    mMax = extents.mMax
End Sub
```

3.2.2.2 mMin

Gets the minimum measure bound for the extents object. Note: Measure bounds only apply to shapefiles containing measure data.

VB.NET Usage

ReadOnly Property mMin() As Double

Parameters

ReturnValue	The minimum measure bound for the extents object.

Sample Code

```
Private Sub Extents_mMin()
    Dim extents As New MapWinGIS.Extents()
    Dim mMin As Double
    'Get the minimum measure bound for the extents object
    mMin = extents.mMin
End Sub
```

3.2.2.3 xMax

The maximum x bound for the extents object.

VB.NET Usage

ReadOnly Property xMax() As Double

Parameters

ReturnValue	The maximum x bound for the extents object.

Sample Code

```
Private Sub Extents_xMax()
    Dim extents As New MapWinGIS.Extents()
```

```
    Dim xMax As Double
    'Get the maximum x bound for the extents object
    xMax = extents.xMax
End Sub
```

3.2.2.4 xMin

Gets the minimum x bound for the extents object.

VB.NET Usage

ReadOnly Property xMin() As Double

Parameters

ReturnValue	The minimum x bound for the extents object.

Sample Code

```
Private Sub Extents_xMin()
    Dim extents As New MapWinGIS.Extents()
    Dim xMin As Double
    'Get the minimum x bound for the extents object
    xMin = extents.xMin
End Sub
```

3.2.2.5 yMax

The maximum y bound for the extents object.

VB.NET Usage

ReadOnly Property yMax() As Double

Parameters

ReturnValue	The maximum y bound for the extents object.

Sample Code

```
Private Sub Extents_yMax()
    Dim extents As New MapWinGIS.Extents()
    Dim yMax As Double
    'Get the maximum y bound for the extents object
    yMax = extents.yMax
End Sub
```

3.2.2.6 yMin

Gets the minimum y bound for the extents object.

VB.NET Usage

ReadOnly Property yMin() As Double

Parameters

ReturnValue	The minimum y bound for the extents object.

Sample Code

```
Private Sub Extents_yMin()
    Dim extents As New MapWinGIS.Extents()
    Dim yMin As Double
    'Get the minimum y bound for the extents object
    yMin = extents.yMin
End Sub
```

3.2.2.7 zMax

The maximum z bound for the extents object.

VB.NET Usage

ReadOnly Property zMax() As Double

Parameters

ReturnValue	The maximum z bound for the extents object.

Sample Code

```
Private Sub Extents_zMax()
    Dim extents As New MapWinGIS.Extents()
    Dim zMax As Double
    'Get the maximum z bound for the extents object
    zMax = extents.zMax
End Sub
```

3.2.2.8 zMin

Gets the minimum z bound for the extents object.

VB.NET Usage

ReadOnly Property zMin() As Double

Parameters

ReturnValue	The minumum z bound for the extents object.

Sample Code

```
Private Sub Extents_zMin()
    Dim extents As New MapWinGIS.Extents()
    Dim zMin As Double
    'Get the minimum z bound for the extents object
    zMin = extents.zMin
End Sub
```

3.3 Field

3.3.1 Properties

3.3.1.1 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.3.1.2 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
       '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
              'Display the error message in a label
             LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
             'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
             'Display the message in a label
         Label2.Text = Message
     End Sub
 #End Region
              '...
```

3.3.1.3 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
    Private Sub MapKey()
        Dim k As String
        'Get the map's key
        k = Map1.Key
        'Check if the map's key is "Map1"
        If k = "Map1" Then
            'Set the map's key to "My Map1"
            Map1.Key = "My Map1"
        Else
            'Set the map's key to "Map1"
            Map1.Key = "Map1"
        End If
    End Sub
```

3.3.1.4 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
    Private Sub LastErrorCode()
        Dim errorCode As Integer
        'Get the last error in the map
        errorCode = Map1.LastErrorCode
        'Display message box giving error message for the last error in the map
        MsgBox(Map1.get_ErrorMsg(errorCode))
```

```
End Sub
```

3.3.1.5 Name

Gets or sets the name of the field.

VB.NET Usage

Property Name() As String

Parameters

ReturnValue	The name of the field.

Sample Code

```
Private Sub FieldName()
    Dim field As New MapWinGIS.Field()
    Dim name As String
    'Set the name to use
    name = "test field"
    'Set the name for the field
    field.Name = name
    'Get the name for the field
    name = field.Name
End Sub
```

3.3.1.6 Precision

Gets or sets the precision of the field. Note: Precision only applies to fields of fieldtype double.

VB.NET Usage

Property Precision() As Integer

Parameters

ReturnValue	The precision of the field.

Sample Code

```
Private Sub FieldPrecision()
    Dim field As New MapWinGIS.Field()
    Dim precision As Integer
    'Set the precision value
    precision = 2
    'Set the precision of the field
    field.Precision = precision
    'Get the precision of the field
    precision = field.Precision
End Sub
```

3.3.1.7 Type

Gets or sets the fieldtype of the field.

VB.NET Usage

Property Type() As MapWinGIS.FieldType

Parameters

ReturnValue	The fieldtype of the field.

Sample Code

```
Private Sub FieldType()
    Dim field As New MapWinGIS.Field()
    Dim fieldtype As New MapWinGIS.FieldType()
    'Set the fieldtype for the field
    field.Type = MapWinGIS.FieldType.DOUBLE_FIELD
    'Get the fieldtype for the field
```

```
    fieldtype = field.Type
  End Sub
```

3.3.1.8 Width

Gets or sets the width of the field.

VB.NET Usage

Property Width() As Integer

Parameters

ReturnValue	The width of the field.

Sample Code

```
  Private Sub FieldWidth()
    Dim field As New MapWinGIS.Field()
    Dim width As Integer
    'Set the width value to use
    width = 20
    'Set the width of the field
    field.Width = width
    'Get the width of the field
    width = field.Width
  End Sub
```

3.4 Grid

The grid object is used to represent a grid which can be added to the map.
The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.4.1 Functions

3.4.1.1 Clear

Clears all data in the grid, setting the value of all cells in the grid to the specified clear value.

VB.NET Usage

Function Clear(ClearValue As Object) As Boolean

Parameters

ClearValue	The value to set all of the grid's cells to.
ReturnValue	A boolean value representing the success or failure of clearing the grid.

Sample Code

```
  Private Sub ClearGrid()
    Dim grid As New MapWinGIS.Grid()
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Clear the grid, using 15 as the value to assign to all grid cells
    grid.Clear(15)
    'Save the cleared grid to disk
    grid.Save("C:\cleargrid.asc")
    'Close the grid
    grid.Close()
  End Sub
```

3.4.1.2 Close

Closes the grid.

VB.NET Usage

Function Close() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of closing the grid.

Sample Code

```
Private Sub CloseGrid()
    Dim grid As New MapWinGIS.Grid()
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Close the grid
    grid.Close()
End Sub
```

3.4.1.3 CreateNew

Creates a new grid.

VB.NET Usage

Function CreateNew(Filename As String, Header As MapWinGIS.GridHeader, DataType As MapWinGIS.GridDataType, InitialValue As Object, Optional InRam As Boolean, Optional FileType As MapWinGIS.GridFileType, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

Filename	The filename for the new grid.
Header	The header defining the attributes of the new grid.
DataType	The data type of the new grid.
InitialValue	The initial value for each cell of the new grid.
InRam	Optional. A boolean value representing the grid being stored in memory(RAM) when True, and the grid being stored on disk when False.
FileType	Optional. The grid file type.
cBack	Optional. The ICallback object that will receive the progress and error events during the creation of the new grid.
ReturnValue	A boolean value representing the success or failure of the creation of the new grid.

Sample Code

```
Private Sub CreateNewGrid()
    Dim row As Integer, col As Integer
    Dim grid As New MapWinGIS.Grid()
    Dim newgrid As New MapWinGIS.Grid()
    'Create the new grid using the old grid's header and data type
    newgrid.CreateNew("", grid.Header, grid.DataType, 0, True, MapWinGIS.GridFileType.Ascii, Me)
    'Populate the new grid with the other grid's values
    For row = 0 To newgrid.Header.NumberRows - 1
        For col = 0 To newgrid.Header.NumberCols - 1
            newgrid.Value(col, row) = grid.Value(col, row)
        Next
    Next
    'Save the new grid
    newgrid.Save("C:\output.asc")
    'Close the grid
    grid.Close()
    'Close the new grid
    newgrid.Close()
End Sub
```

3.4.1.4 Open

Opens a grid.

VB.NET Usage

Function Open(Filename As String, Optional DataType As MapWinGIS.GridDataType, Optional InRam As Boolean, Optional FileType As MapWinGIS.GridFileType, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

Filename	The filename of the grid to be opened.
DataType	Optional. The data type of the grid to be opened.
InRam	Optional. A boolean value representing whether the grid will be stored in RAM or on disk.
FileType	Optional. The file type of the grid. The default file type is "Use Extension".
cBack	Optional. The ICallback object that will receive the progress and error events during the creation of the new grid.
ReturnValue	A boolean value that represents the success or failure of opening the grid.

Sample Code

```
Private Sub OpenGrid()
    Dim grid As New MapWinGIS.Grid()
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Close the grid
    grid.Close()
End Sub
```

3.4.1.5 Save

Saves the grid.

VB.NET Usage

Function Save(Optional Filename As String, Optional GridFileType As MapWinGIS.GridFileType, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

Filename	Optional. The filename the grid will be saved under. If no filename is specified the filename in the grid's Filename property is used.
GridFileType	Optional. The file type to save the grid as. If no type is specified, the type stored in the grid object is used.
cBack	Optional. The ICallback object that will receive the progress and error events during the creation of the new grid.
ReturnValue	A boolean value representing the success or failure of saving the grid.

Sample Code

```
Private Sub SaveGrid()
    Dim grid As New MapWinGIS.Grid()
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Save the grid
    grid.Save("C:\grid2.asc")
    'Close the grid
    grid.Close()
End Sub
```

3.4.2 Subs

3.4.2.1 CellToProj

Uses a cell's column and row position to find the center of the cell in projected map coordinates.

VB.NET Usage

Sub CellToProj(Column As Integer, Row As Integer, ByRef x As Double, ByRef y As Double)

Parameters

Column	The column of the cell to find the center in projected map coordinates.
Row	The row of the cell to find the center in projected map coordinates.
x	Reference parameter. Returns the x projected map coordinate of the center of the specified cell.

y	Reference parameter. Returns the yprojected map coordinate of the center of the specified cell.

Sample Code

```
Private Sub CellToProj()
    Dim grid As New MapWinGIS.Grid()
    Dim x As Double, y As Double
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Get the center of the cell in projected map coordinates stored in x and y
    grid.CellToProj(5, 5, x, y)
    'Display center of the cell in message box
    MsgBox("The center of cell col 5, row 5: x = " + Str(x) + " y = " + Str(y))
    'Close the grid
    grid.Close()
End Sub
```

3.4.2.2 ProjToCell

Converts a point in projected map coordinates to a cell (column, row) in the grid. Note: If the point lies outside the bounds of the grid, a column and row are returned which are outside the boundaries of the grid. For example, if the point lies to the left or lies below the grid boundaries, a negative column or row will be returned. Similarly, if the point lies above or to the right of the grid boundaries, a column or row which is greater than the number of columns or rows will be returned.

VB.NET Usage

Sub ProjToCell(x As Double, y As Double, ByRef Column As Integer, ByRef Row As Integer)

Parameters

x	The x projected map coordinate for which the corresponding cell in the grid is required.
y	The y projected map coordinate for which the corresponding cell in the grid is required.
Column	Reference parameter. The column the specified point lies within. Note: This value may not be within the valid bounds of the grid.
Row	Reference parameter. The row the specified point lies within. Note: This value may not be within the valid bounds of the grid.

Sample Code

```
Private Sub ProjToCell()
    Dim grid As New MapWinGIS.Grid()
    Dim x As Double, y As Double
    Dim col As Integer, row As Integer
    'Set the point in projected map coordinates
    x = 130000
    y = 135000
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Get the center of the cell in projected map coordinates stored in x and y
    grid.ProjToCell(x, y, col, row)
    'Display column and row of the specified point in a message box
    MsgBox("The column and row of the point: col = " + Str(col) + " row = " + Str(row))
    'Close the grid
    grid.Close()
End Sub
```

3.4.2.3 Properties

3.4.2.4 CdlgFilter

Returns the common dialog filter containing all supported file extensions in string format.

VB.NET Usage

ReadOnly Property CdlgFilter() As String

Parameters

ReturnValue	The filter containing all file extensions supported by MapWinGIS.

Sample Code

```
Private Sub CdlgFilter()
    Dim grid As New MapWinGIS.Grid()
```

```
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Display the supported file formats in a message box
    MsgBox(grid.CdlgFilter)
    'Close the grid
    grid.Close()
End Sub
```

3.4.2.5 DataType

Returns the data type of the values stored in the grid.

VB.NET Usage

ReadOnly Property DataType() As MapWinGIS.GridDataType

Parameters

ReturnType	The data type of the values stored in the grid.

Sample Code

```
Private Sub DataType()
    Dim grid As New MapWinGIS.Grid()
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Display the file type of the grid
    MsgBox(grid.DataType.ToString)
    'Close the grid
    grid.Close()
End Sub
```

3.4.2.6 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.4.2.7 Filename

The filename associated with the object.

VB.NET Usage

ReadOnly Property Filename() As String

Parameters

ReturnValue	The filename associated with the object.

Sample Code

```
Private Sub Filename()
    Dim grid As New MapWinGIS.Grid()
    Dim filename As String
    'Open a grid from disk
    grid.Open("C:\grid.asc")
```

```
        'Get the filename of the grid
        filename = grid.Filename
    End Sub
```

3.4.2.8 GetRow

Syntax

```
bool GetRow(int row, ref float vals)
```

Summary

This is a faster way to read the array values that are of a specific size. The row is the integer row to read from the grid object. The vals variable is actually the first element of the array of floats that you want to be populated with the values from the grid. Since arrays are stored sequentially in memory, passing the first element allows the prediction of where the other values must go. It is very important that you always dimension the array as being of type float, and always make sure that you dimension it from 0 to numCols - 1.

Parameters

row	The Integer value of the row to retrieve values for.
vals	Reference to the first element of the array of floats that will hold the row of values.

Returns

Boolean false if there was an error, true otherwise.

Visual Basic Net 2005 Example Implementation Code

```
    'Requires reference to MapWinGIS
    'mwSourceGrid is already instantiated and opened from an existing grid file
    'SourceGrid is dimensioned as Dim SourceGrid(MaxCol, MaxRow) as Float
    Public Function CopySource(ByVal mwSourceGrid As MapWinGIS.Grid) As Single(,)
        Dim row, col As Integer
        Dim vals() As Single
        m_mrow = mwSourceGrid.Header.NumberRows - 1
        m_mcol = mwSourceGrid.Header.NumberCols - 1
        Dim SourceGrid(m_mcol, m_mrow) As Single
        For row = 0 To m_mrow
            ReDim vals(m_mcol)
            mwSourceGrid.GetRow(row, vals(0))
            For col = 0 To m_mcol
                SourceGrid(col, row) = vals(col)
            Next
        Next
        Return SourceGrid
    End Function
```

3.4.2.9 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
        '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
                'Display the error message in a label
               LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
               'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
               'Display the message in a label
         Label2.Text = Message
     End Sub
 #End Region
               '...
```

3.4.2.10 Header
Returns the header of the grid.

VB.NET Usage

ReadOnly Property Header() As MapWinGIS.GridHeader

Parameters

ReturnValue	The header for the grid.

Sample Code
```
Private Sub Header()
    Dim row As Integer, col As Integer
    Dim grid As New MapWinGIS.Grid()
    Dim newgrid As New MapWinGIS.Grid()
    'Create the new grid using the old grid's header and data type
    newgrid.CreateNew("", grid.Header, grid.DataType, 0, True, MapWinGIS.GridFileType.Ascii, Me)
    'Close the grid
    grid.Close()
End Sub
```

3.4.2.11 InRam
Returns whether the grid is loaded in to RAM memory or not.

VB.NET Usage

ReadOnly Property InRam() As Boolean

Parameters

ReturnValue	A boolean value representing whether the grid is loaded into RAM or not.

Sample Code
```
Private Sub InRam()
    Dim grid As New MapWinGIS.Grid()
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Check if the grid is loaded into RAM
    If grid.InRam Then
        MsgBox("The grid is loaded into RAM.")
    Else
        MsgBox("The grid is not loaded into RAM.")
    End If
    'Close the grid
    grid.Close()
End Sub
```

3.4.2.12 Key
The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code
```
Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
    If k = "Map1" Then
        'Set the map's key to "My Map1"
        Map1.Key = "My Map1"
    Else
        'Set the map's key to "Map1"
```

```
        Map1.Key = "Map1"
    End If
End Sub
```

3.4.2.13 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.4.2.14 Maximum

Returns the maximum value stored in the grid.

VB.NET Usage

ReadOnly Property Maximum() As Object

Parameters

ReturnValue	The maximum value stored in the grid.

Sample Code

```
Private Sub Maximum()
    Dim grid As New MapWinGIS.Grid()
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Display the maximum value stored in the grid
    MsgBox(Str(grid.Maximum))
    'Close the grid
    grid.Close()
End Sub
```

3.4.2.15 Minimum

Returns the minimum value stored in the grid.

VB.NET Usage

ReadOnly Property Minimum() As Object

Parameters

ReturnValue	The minimum value stored in the grid.

Sample Code

```
Private Sub Minimum()
    Dim grid As New MapWinGIS.Grid()
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Display the minimum value stored in the grid
    MsgBox(Str(grid.Minimum))
    'Close the grid
    grid.Close()
End Sub
```

3.4.2.16 PutRow

Syntax

bool PutRow(int row, ref float vals)

Summary

This is a faster way to write the array values that are of a specific size. The row is the integer row to read from the grid object. The vals variable is actually the first element of the array of floats that you want to be populated with the values from the grid. Since arrays are stored sequentially in memory, passing the first element allows the prediction of where the other values must go. It is very important that you always dimension the array as being of type float, and always make sure that you dimension it from 0 to numCols - 1.

Parameters

row	The Integer value of the row to retrieve values for.
vals	Reference to the first element of the array of floats that will hold the row of values.

Returns

Boolean false if there was an error, true otherwise.

Visual Basic Net 2005 Example Implementation Code

```
'Requires reference to MapWinGIS
'mwSourceGrid is already instantiated and opened from an existing grid file
'SourceGrid is dimensioned as Dim SourceGrid(MaxCol, MaxRow) as Float
Public Function CopySource(ByVal mwSourceGrid As MapWinGIS.Grid) As Single(,)
    Dim row, col As Integer
    Dim vals() As Single
    m_mrow = mwSourceGrid.Header.NumberRows - 1
    m_mcol = mwSourceGrid.Header.NumberCols - 1
    Dim SourceGrid(m_mcol, m_mrow) As Single
    For row = 0 To m_mrow
        ReDim vals(m_mcol)
        mwSourceGrid.GetRow(row, vals(0))
        For col = 0 To m_mcol
            SourceGrid(col, row) = vals(col)
        Next
    Next
    Return SourceGrid
End Function
```

3.4.2.17 Value

Returns the value stored in the grid at the specified cell.

VB.NET Usage

Property Value(Column As Integer, Row As Integer) As Object

Parameters

Column	The column representing the cell for which the value is required.
Row	The row representing the cell for which the value is required.
ReturnValue	The value stored in the grid in the specified cell.

Sample Code

```
Private Sub GridValue()
    Dim grid As New MapWinGIS.Grid()
    'Open a grid from disk
    grid.Open("C:\grid.asc")
    'Display the value of column 5, row 5 in the grid
    MsgBox(Str(grid.Value(5, 5)))
    'Close the grid
    grid.Close()
End Sub
```

3.5 GridColorBreak

A grid color break object defines how a specified region of a grid will be colored using the grid color scheme containing the grid color break.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.5.1 Properties

3.5.1.1 Caption
Gets or sets the caption to display for the color break.

VB.NET Usage

Property Caption() As String

Parameters

ReturnValue	The caption to be displayed for the color break.

Sample Code

```
Private Sub ColorBreakCaption()
    Dim break As New MapWinGIS.GridColorBreak()
    Dim caption As String
    'Set the string to be used as the caption
    caption = "Color Break 1"
    'Set the caption for the color break
    break.Caption = caption
    'Get the caption for the color break
    caption = break.Caption
End Subb
```

3.5.1.2 ColoringType
Gets or sets the way a break is colored.

VB.NET Usage

Property ColoringType() As MapWinGIS.ColoringType

Parameters

ReturnValue	The coloring type for the GridColorBreak.

Sample Code

```
Private Sub ColoringType()
    Dim break As New MapWinGIS.GridColorBreak()
    'Set the grid color break coloring type to gradient
    break.ColoringType = MapWinGIS.ColoringType.Gradient
    'Display the grid color break coloring scheme in a message box
    MsgBox(break.ColoringType.ToString)
End Sub
```

3.5.1.3 ErrorMsg
Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.5.1.4 GlobalCallback
The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
   Inherits System.Windows.Forms.Form
      'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
   Implements MapWinGIS.ICallback
     '...
 #Region "ICallback Members"
    Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
            'Display the error message in a label
           LabelError.Text = ErrorMsg
   End Sub
   Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
           'Display the progress in a label
      Label1.Text = "Progress: " + Str(Percent) + "%"
           'Display the message in a label
       Label2.Text = Message
    End Sub
 #End Region
            '...
```

3.5.1.5 GradientModel

Gets or sets the model used to color the break.

VB.NET Usage

Property GradientModel() As MapWinGIS.GradientModel

Parameters

ReturnValue	The gradient model used to color the break.

Sample Code

```
   Private Sub GradientModel()
      Dim break As New MapWinGIS.GridColorBreak()
      'Set the gradient model to use with the color break
      break.GradientModel = MapWinGIS.GradientModel.Linear
      'Display the gradient model used by the color break
      MsgBox(break.GradientModel.ToString)
   End Sub
```

3.5.1.6 HighColor

Gets or sets the color representing the highest value in the color break.

VB.NET Usage

Property HighColor() As System.UInt32

Parameters

ReturnValue	The color representing the highest value in the color break. This is a UInt32 representation of an RGB value.

Sample Code

```
   Private Sub Highcolor()
      Dim break As New MapWinGIS.GridColorBreak()
      Dim col As System.UInt32
      'Set the color to represent the highest value in the color break
      break.HighColor = System.Convert.ToUInt32(RGB(0, 200, 0))
      'Get the color representing the highest value in the color break
      col = break.HighColor
   End Sub
```

3.5.1.7 HighValue
Gets or sets the highest value in the color break.

VB.NET Usage

Property HighValue() As Double

Parameters

ReturnValue	The highest value in the color break.

Sample Code
```
Private Sub HighValue()
    Dim break As New MapWinGIS.GridColorBreak()
    Dim high As Double
    'Get the high value in the color break
    high = break.HighValue
    'Set the high value in the color break
    break.HighValue = 1000
End Sub
```

3.5.1.8 Key
The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code
```
Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
    If k = "Map1" Then
        'Set the map's key to "My Map1"
        Map1.Key = "My Map1"
    Else
        'Set the map's key to "Map1"
        Map1.Key = "Map1"
    End If
End Sub
```

3.5.1.9 LastErrorCode
Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code
```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.5.1.10 LowColor

Gets or sets the color representing the lowest value in the color break.

VB.NET Usage

Property LowColor() System.UInt32

Parameters

ReturnValue	The color representing the lowest value in the color break. This is a UInt32 representation of an RGB value.

Sample Code

```
Private Sub Lowcolor()
    Dim break As New MapWinGIS.GridColorBreak()
    Dim col As System.UInt32
    'Set the color to represent the lowest value in the color break
    break.LowColor = System.Convert.ToUInt32(RGB(0, 200, 0))
    'Get the color representing the lowest value in the color break
    col = break.LowColor
End Sub
```

3.5.1.11 LowValue

Gets or sets the lowest value in the color break.

VB.NET Usage

Property LowValue() As Double

Parameters

ReturnValue	The lowest value in the color break.

Sample Code

```
Private Sub LowValue()
    Dim break As New MapWinGIS.GridColorBreak()
    Dim low As Double
    'Get the low value in the color break
    low = break.LowValue
    'Set the low value in the color break
    break.LowValue = 1000
End Sub
```

3.6 GridColorScheme

A grid color scheme defines how a grid will be colored. A grid color scheme may either contain grid color breaks created by the user, or may use a predefined color scheme.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.6.1 Functions

3.6.1.1 GetLightSource

Gets the direction the light source is pointing for the color scheme in the form of a vector which originates from the light source. This only applies to hillshaded breaks.

VB.NET Usage

Function GetLightSource() As MapWinGIS.Vector

Parameters

ReturnValue	The direction the light source is pointing represented by a vector.

Sample Code

```
Private Sub GetLightSource()
    Dim scheme As MapWinGIS.GridColorScheme
    Dim vector As MapWinGIS.Vector
```

```
    'Get the vector representing the direction from the light source
    vector = scheme.GetLightSource
  End Sub
```

3.6.2 Subs

3.6.2.1 Clear

Clears all color breaks in the color scheme.

VB.NET Usage

Sub Clear()

Parameters

None	

Sample Code

```
  Private Sub ClearBreaks()
    Dim scheme As MapWinGIS.GridColorScheme
    Dim break As New MapWinGIS.GridColorBreak()
    'Add color break to color scheme
    scheme.InsertBreak(break)
    'Clear all color breaks in color scheme
    scheme.Clear()
  End Sub
```

3.6.2.2 DeleteBreak

Deletes the specified color break from the color scheme.

VB.NET Usage

Sub DeleteBreak(Index As Integer)

Parameters

Index	The index of the color break to be removed.

Sample Code

```
  Private Sub DeleteBreak()
    Dim scheme As MapWinGIS.GridColorScheme
    Dim break As New MapWinGIS.GridColorBreak()
    Dim index As Integer
    'Add color break to color scheme
    scheme.InsertBreak(break)
    'Set the index of the color break to be deleted
    index = 0
    'Delete the first color break from the color scheme
    scheme.DeleteBreak(index)
  End Sub
```

3.6.2.3 InsertBreak

Inserts a color break into the color scheme.

VB.NET Usage

Sub InsertBreak(BrkInfo As MapWinGIS.GridColorBreak)

Parameters

BrkInfo	The color break to be inserted into the color scheme.

Sample Code

```
  Private Sub InsertBreak()
    Dim grid As New MapWinGIS.Grid()
    Dim colorscheme As New MapWinGIS.GridColorScheme()
    Dim break As New MapWinGIS.GridColorBreak()
    'Set the color break high value to represent the grid's maximum value
    break.HighValue = grid.Maximum
```

```
    'Set the color break low value to represent the grid's minimum value
    break.LowValue = grid.Minimum
    'Set the color break high color to red
    break.HighColor = Convert.ToUInt32(RGB(255, 0, 0))
    'Set the color break low color to blue
    break.LowColor = Convert.ToUInt32(RGB(0, 0, 255))
    'Insert color break into color scheme
    colorscheme.InsertBreak(break)
End Sub
```

3.6.2.4 SetLightSource

Sets the direction of the light source.

VB.NET Usage

Sub SetLightSource(Azimuth As Double, Elevation As Double)

Parameters

Azimuth	The azimuth heading of the light source. Also known as the compass heading.
Elevation	The elevation of the light source.

Sample Code

```
Private Sub SetLightSource()
    Dim scheme As New MapWinGIS.GridColorScheme()
    Dim heading As Double, elevation As Double
    'Set the heading to be 010
    heading = 10
    'Set the elevation to be 3000 meters
    elevation = 3000
    'Set the light source using the heading and elevation from above
    scheme.SetLightSource(heading, elevation)
End Sub
```

3.6.2.5 UsePredefined

Loads the values into the color scheme from pre-defined color scheme.

VB.NET Usage

Sub UsePredefined(LowValue As Double, HighValue As Double, Optional Preset As MapWinGIS.PredefinedColorScheme)

Parameters

LowValue	The lowest value in the grid.
HighValue	The highes value in the grid.
Preset	Optional. The pre-defined color scheme to be used. The default is MapWinGIS.PredefinedColorScheme.SummerMountains.

Sample Code

```
Private Sub PredefinedScheme()
    Dim grid As New MapWinGIS.Grid()
    Dim scheme As New MapWinGIS.GridColorScheme()
    'Set the color scheme to be a predefined color scheme SummerMountains
    scheme.UsePredefined(grid.Minimum, grid.Maximum, MapWinGIS.PredefinedColorScheme.SummerMountains)
End Sub
```

3.6.3 Properties

3.6.3.1 AmbientIntensity

Gets or sets the intensity of the ambient lighting for the color scheme. Only applies when creating hillshaded breaks.

VB.NET Usage

Property AmbientIntensity() As Double

Parameters

ReturnValue	The intensity of the ambient light in the color scheme.

Sample Code

```
Private Sub AmbientIntensity()
    Dim scheme As New MapWinGIS.GridColorScheme()
    Dim intensity As Double
    'Set the intensity to be used for the color scheme
    intensity = 50
    'Set the ambient intensity of the light for the color scheme
    scheme.AmbientIntensity = intensity
    'Get the ambient intensity of the light for the color scheme
    intensity = scheme.AmbientIntensity
End Sub
```

3.6.3.2 Break

Gets a break from the color scheme using the specified index.

VB.NET Usage

ReadOnly Property Break(Index As Integer) As MapWinGIS.GridColorBreak

Parameters

Index	The index of the color break to be returned.
ReturnValue	The color break specified by the index.

Sample Code

```
Private Sub GetBreak()
    Dim scheme As New MapWinGIS.GridColorScheme()
    Dim break As New MapWinGIS.GridColorBreak()
    'Get the break indexed by 0 in the color scheme
    break = scheme.Break(0)
End Sub
```

3.6.3.3 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.6.3.4 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
       '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
              'Display the error message in a label
             LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
             'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
             'Display the message in a label
         Label2.Text = Message
     End Sub
 #End Region
              '...
```

3.6.3.5 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
    Private Sub MapKey()
        Dim k As String
        'Get the map's key
        k = Map1.Key
        'Check if the map's key is "Map1"
        If k = "Map1" Then
            'Set the map's key to "My Map1"
            Map1.Key = "My Map1"
        Else
            'Set the map's key to "Map1"
            Map1.Key = "Map1"
        End If
    End Sub
```

3.6.3.6 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
    Private Sub LastErrorCode()
        Dim errorCode As Integer
        'Get the last error in the map
        errorCode = Map1.LastErrorCode
        'Display message box giving error message for the last error in the map
        MsgBox(Map1.get_ErrorMsg(errorCode))
    End Sub
```

3.6.3.7 LightSourceAzimuth

Gets the azimuth, or compass heading of the light source. Only applies to hillshaded breaks.

VB.NET Usage

ReadOnly Property LightSourceAzimuth() As Double

Parameters

ReturnValue	The azimuth, or compass heading of the light source.

Sample Code

```
Private Sub LightSourceAzimuth()
    Dim scheme As New MapWinGIS.GridColorScheme()
    Dim heading As Double
    'Get the azimuth, or compass heading of the light source
    heading = scheme.LightSourceAzimuth
End Sub
```

3.6.3.8 LightSourceElevation

Gets the elevation of the light source. Only applies to hillshaded breaks.

VB.NET Usage

ReadOnly Property LightSourceElevation() As Double

Parameters

ReturnValue	The elevation of the light source for the color scheme.

Sample Code

```
Private Sub LightSourceElevation()
    Dim scheme As New MapWinGIS.GridColorScheme()
    Dim elevation As Double
    'Get the elevation of the light source for the color scheme
    elevation = scheme.LightSourceElevation
End Sub
```

3.6.3.9 LightSourceIntensity

Gets or sets the intensity of the light source for the color scheme. Only applies to hillshaded breaks.

VB.NET Usage

Property LightSourceIntensity() As Double

Parameters

ReturnValue	The intensity of the light source for the color scheme.

Sample Code

```
Private Sub LightSourceIntensity()
    Dim scheme As New MapWinGIS.GridColorScheme()
    Dim intensity As Double
    'Set the intensity value
    intensity = 50
    'Set the intensity of the light source for the color scheme
    scheme.LightSourceIntensity = intensity
    'Get the intensity of the light source for the color scheme
    intensity = scheme.LightSourceIntensity
End Sub
```

3.6.3.10 NoDataColor

The color to use for the color scheme when drawing grid cells with a no-data value.

VB.NET Usage

Property NoDataColor() System.UInt32

Parameters

ReturnValue	The color for the color scheme to use when drawing a grid cell with a no-data value. This is a UInt32 representation of an RGB value.

Sample Code

```
Private Sub NoDataColor()
    Dim scheme As New MapWinGIS.GridColorScheme()
    Dim col As System.UInt32
    'Set the color for the color scheme to use when drawing a grid cell with a no-data value
    scheme.NoDataColor = System.Convert.ToUInt32(RGB(0, 0, 0))
    'Get the color used by the color scheme for drawing a grid cell with a no-data value
    col = scheme.NoDataColor
End Sub
```

3.6.3.11 NumBreaks

Gets the number of color breaks in the color scheme.

VB.NET Usage

ReadOnly Property NumBreaks() As Integer

Parameters

ReturnValue	The number of color breaks in the color scheme.

Sample Code

```
Private Sub NumBreaks()
    Dim scheme As New MapWinGIS.GridColorScheme()
    Dim count As Integer
    'Get the number of color breaks in the color scheme
    count = scheme.NumBreaks
End Sub
```

3.7 GridHeader

A grid header object is used to hold the information contained in a grid's header

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.7.1 Properties

3.7.1.1 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
```

```
End Sub
```

3.7.1.2 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
   Inherits System.Windows.Forms.Form
      'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
   Implements MapWinGIS.ICallback
     '...
 #Region "ICallback Members"
    Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
            'Display the error message in a label
           LabelError.Text = ErrorMsg
   End Sub
   Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
           'Display the progress in a label
      Label1.Text = "Progress: " + Str(Percent) + "%"
           'Display the message in a label
       Label2.Text = Message
    End Sub
 #End Region
            '...
```

3.7.1.3 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
   Private Sub MapKey()
      Dim k As String
      'Get the map's key
      k = Map1.Key
      'Check if the map's key is "Map1"
      If k = "Map1" Then
         'Set the map's key to "My Map1"
         Map1.Key = "My Map1"
      Else
         'Set the map's key to "Map1"
         Map1.Key = "Map1"
      End If
   End Sub
```

3.7.1.4 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.7.1.5 NodataValue

The value representing cells in the grid with no-data or missing data values.

VB.NET Usage

Property NodataValue() As Object

Parameters

ReturnValue	The value representing cells in the grid with no-data values.

Sample Code

```
Private Sub NodataValue()
    Dim header As New MapWinGIS.GridHeader()
    Dim nodata As Double
    'Set the nodata value to be used
    nodata = -9999
    'Set the no data value for the grid header
    header.NodataValue = nodata
    'Get the no data value for the grid header
    nodata = header.NodataValue
End Sub
```

3.7.1.6 Notes

Gets or sets notes or any additional information about the grid stored in the grid header.

VB.NET Usage

Property Notes() As String

Parameters

ReturnValue	The notes or additional information associated with the grid header.

Sample Code

```
Private Sub HeaderNotes()
    Dim header As New MapWinGIS.GridHeader()
    Dim notes As String
    'Set the notes to be used
    notes = "This is a test grid header."
    'Set the notes for the grid header
    header.Notes = notes
    'Get the notes for the grid header
    notes = header.Notes
End Sub
```

3.7.1.7 NumberCols

Gets or sets the number of columns for the grid represented by the grid header.

VB.NET Usage

Property NumberCols() Integer

Parameters

ReturnValue	The number of columns in the grid represented by the grid header.

Sample Code

```
Private Sub NumberCols()
    Dim header As New MapWinGIS.GridHeader()
    Dim columns As Integer
    'Set the number of columns to be used
    columns = 300
    'Store the number of columns for the grid in the grid header
    header.NumberCols = columns
    'Get the number of columns for the grid from the grid header
    columns = header.NumberCols
End Sub
```

3.7.1.8 NumberRows

Gets or sets the number of rows in the grid represented by the grid header.

VB.NET Usage

Property NumberRows() Integer

Parameters

ReturnValue	The number of rows in the grid represented by the grid header.

Sample Code

```
Private Sub NumberRows()
    Dim header As New MapWinGIS.GridHeader()
    Dim rows As Integer
    'Set the number of rows to be used
    rows = 300
    'Store the number of rows for the grid in the grid header
    header.NumberRows = rows
    'Get the number of rows for the grid from the grid header
    rows = header.NumberRows
End Sub
```

3.7.1.9 Projection

Gets or sets information about the projection of the grid represented by the grid header. Setting this value has no effect on the actual grid data. The projection should be expressed as a PROJ4 string.

VB.NET Usage

Property Projection() As String

Parameters

ReturnValue	The PROJ4 string containing invormation about the projection of the grid represented by the grid header.

Sample Code

```
Private Sub Projection()
    Dim header As New MapWinGIS.GridHeader()
    Dim projection As String
    'Set the projection information to be used
    projection = "+proj=utm +zone=12 +ellps=GRS80 +datum=NAD83 +units=m +no_defs"
    'Set the projection information for the grid header
    header.Projection = projection
    'Get the projection information for the grid header
    projection = header.Projection
End Sub
```

3.7.1.10 XllCenter

Gets or sets the x coordinate of the center of the lower-left cell in the grid represented by the grid header.

VB.NET Usage

Property XllCenter() As Double

Parameters

ReturnValue	The x coordinate of the center of the lower-left cell in the grid represented by the grid header.

Sample Code

```
Private Sub XLowerLeftCenter()
    Dim header As New MapWinGIS.GridHeader()
    Dim x As Double
    'Set the x coordinate to use
    x = 33000
    'Set the x lower left center coordinate for the grid header
    header.XllCenter = x
    'Get the x lower left center coordinate for the grid header
    x = header.XllCenter
End Sub
```

3.7.1.11 YllCenter

Gets or sets the y coordinate of the center of the lower-left cell in the grid represented by the grid header.

VB.NET Usage

Property YllCenter() As Double

Parameters

ReturnValue	The y coordinate of the center of the lower-left cell in the grid represented by the grid header..

Sample Code

```
Private Sub YLowerLeftCenter()
    Dim header As New MapWinGIS.GridHeader()
    Dim y As Double
    'Set the y coordinate to use
    y = 33000
    'Set the y lower left center coordinate for the grid header
    header.YllCenter = y
    'Get the y lower left center coordinate for the grid header
    y = header.YllCenter
End Sub
```

3.7.1.12 dX

Gets or sets the width of the grid cell in physical units for the grid represented by the grid header.

VB.NET Usage

Property dX() As Double

Parameters

ReturnValue	The width of a grid cell in physical units for the grid represented by the grid header.

Sample Code

```
Private Sub GriddX()
    Dim header As New MapWinGIS.GridHeader()
    Dim width As Double
    'Set the width value to be used
    width = 30
    'Set the width of a grid cell in the grid header
    header.dX = width
    'Get the width of a grid cell from the grid header
    width = header.dX
End Sub
```

3.7.1.13 dY

Gets or sets the height of a cell in the grid represented by the grid header.

VB.NET Usage

Property dY() As Double

Parameters

ReturnValue	The height of the grid cells in the grid represented by the grid header.

Sample Code

```
Private Sub GriddY()
    Dim header As New MapWinGIS.GridHeader()
    Dim height As Double
    'Set the height value to be used
    height = 30
    'Set the height of a grid cell in the grid header
    header.dY = height
    'Get the height of a grid cell from the grid header
    height = header.dY
End Sub
```

3.8 Image

An image object is used to represent an image which may be added to the map.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.8.1 Functions

3.8.1.1 Clear

Clears the image using the specified canvas color as the new color for every pixel in the image.

VB.NET Usage

Function Clear(CanvasColor As System.UInt32, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

CanvasColor	The color to be used to as the new color of every pixel in the image. This is a UInt32 representation of an RGB value.
cBack	Optional. The ICallback object that will receive the progress and error events while the image is cleared.
ReturnValue	A boolean value representing the success or failure of clearing the image.

Sample Code

```
Private Sub ClearImage()
    Dim image As New MapWinGIS.Image
    Dim col As System.UInt32
    Dim success As Boolean
    'Set the color to use
    col = System.Convert.ToUInt32(RGB(255, 0, 0))
    'Clear the image using red as the canvas color
    success = image.Clear(col, Me)
    'Display failure message if the image was not successfully cleared
    If Not success Then
        MsgBox("The image was not successfully cleared.")
    End If
End Sub
```

3.8.1.2 Close

Closes the image.

VB.NET Usage

Function Close() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of closing the image.

Sample Code

```
Private Sub CloseImage()
    Dim image As New MapWinGIS.Image
    Dim success As Boolean
    'Close the image
    success = image.Close
    'Display failure message if the image was not successfully closed
    If Not success Then
        MsgBox("The image was not successfully closed.")
    End If
End Sub
```

3.8.1.3 CreateNew

Creates a new image with specified width and height.

VB.NET Usage

Function CreateNew(NewWidth As Integer, NewHeight As Integer) As Boolean

Parameters

NewWidth	The width of the new image in pixels.
NewHeight	The height of the new image in pixels.
ReturnValue	A boolean value representing the success or failure of creating the new image.

Sample Code

```
Private Sub CreateImage()
    Dim image As New MapWinGIS.Image
    Dim success As Boolean
    'Create a new image with dimensions of 200 x 200 pixels
    success = image.CreateNew(200, 200)
    'Display failure message if the image was not successfully created
    If Not success Then
        MsgBox("The image was not successfully created.")
    End If
End Sub
```

3.8.1.4 GetImageBitsDC

Gets the data from the image and puts it into the selected bitmap in the specified device context. This function requires the width and height of the selected bitmap and the image to match. Note: Use of this function requires advanced knowledge in windows graphics concepts and is intended for advanced users only.

VB.NET Usage

Function GetImageBitsDC(hDC As Integer) As Boolean

Parameters

hDC	The handle to the device context.
ReturnValue	A boolean value representing the success or failure of getting the device context handle for the image.

Sample Code

```
'These are API functions needed to get a device context handle and select a bitmap for the device context
Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Integer) As Integer
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Integer, ByVal hObject As Integer) As Integer
'...
Private Sub GetImageBitsDC()
    Dim hBmpPtr As IntPtr
    Dim hBitmap As Integer, hndl As Integer, hDC As Integer
    Dim image As New MapWinGIS.Image()
    Dim success As Boolean
    Dim i As Integer, j As Integer, r As Integer, g As Integer, b As Integer
```

```
    Dim bitmap As System.Drawing.Bitmap
    Dim result As Long
    'Create a new Bitmap for the original image
    bitmap = New System.Drawing.Bitmap("C:\Test.bmp")
    'Get the hBitmap handle to the modified image
    hBmpPtr = bitmap.GetHbitmap()
    'Get an integer from the intPtr hBitmap
    hBitmap = hBmpPtr.ToInt32
    'Create a compatible device context handle
    hDC = CreateCompatibleDC(0)
    'Select the bitmap for the specified device context
    result = SelectObject(hDC, hBitmap)
    'Create a new MapWinGIS.Image using the width and height of the bitmap
    success = image.Open("C:\Test.bmp")
    'Get the image's data and put it into the selected bitmap for the specified device context
    success = image.GetImageBitsDC(hDC)
    'Add the image to the map
    hndl = Map1.AddLayer(image, True)
End Sub
```

3.8.1.5 GetRow

Gets a row of pixels from the image.

VB.NET Usage

Function GetRow(Row As Integer, ByRef Vals() As Integer) As Boolean

Parameters

Row	The row of pixels you want to get from the image.
Vals()	Reference parameter. An array which will return the pixels in the specified row of the image.
ReturnValue	A boolean value representing the success or failure of getting the row of pixels from the image.

Sample Code

```
Private Sub GetRow()
    Dim image As New MapWinGIS.Image(), image2 As New MapWinGIS.Image()
    Dim row As Integer, i As Integer, j As Integer
    Dim pixels(400) As Integer
    Dim success As Boolean
    'Open image from file
    image.Open("C:\test.bmp")
    'Create a new image to copy rows of pixels to
    image2.CreateNew(400, 500)
    'Set the row to get the pixels from
    row = 0
    'Get 400 rows x 400 columns of pixels from image and copy them into image2
    For i = 1 To 400
        'Get the pixels from the specified row in the image
        success = image.GetRow(row, pixels(0))
        'Get 400 pixel values from the row retrieved from image
        For j = 1 To 400
            'Copy the current pixel into image2
            image2.Value(i, j) = pixels(j)
        Next
        'Advance to the next row in the image
        row = row + 1
    Next
    'Add the new image to the map
    Map1.AddLayer(image2, True)
End Sub
```

3.8.1.6 Open

Opens an image from file.

VB.NET Usage

Function Open(ImageFileName As String, Optional FileType As MapWinGIS.ImageType, Optional InRam As

Boolean, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

ImageFileName	The filename of the image to be opened.
FileType	Optional. The type of image being opened. The default is "USE_FILE_EXTENSION".
InRam	Optional. A boolean value representing whether the image is stored in RAM memory while open. The default is True.
cBack	Optional. The ICallback object which will receive progress and error events while the image is being opened.
ReturnValue	A boolean value representing the success or failure of opening the image.

Sample Code

```
Private Sub OpenImage()
    Dim image As New MapWinGIS.Image()
    Dim success As Boolean
    'Open image from file
    success = image.Open("C:\test.bmp")
    'If the image wasn't opened successfully, display an error message
    If Not success Then
        MsgBox("There was an error opening the image.")
    End If
End Sub
```

3.8.1.7 Save

Saves the image to file.

VB.NET Usage

Function Save(ImageFileName As String, Optional WriteWorldFile As Boolean, Optional FileType As MapWinGIS.ImageType, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

ImageFileName	The filename to use to save the image.
WriteWorldFile	Optional. A boolean value representing whether a world file is also written. The default is false.
FileType	Optional. The filetype to use to save the image. Default is "USE_FILE_EXTENSION".
cBack	Optional. The ICallback object which will receive the progress and error messages while the image is being saved.
ReturnValue	A boolean value representing the success or failure of saving the image.

Sample Code

```
Private Sub SaveImage()
    Dim image As New MapWinGIS.Image()
    Dim success As Boolean
    Dim filename As String
    'Set the filename to be used to save the image
    filename = "C:\test.bmp"
    'Save the image as a bitmap
    success = image.Save(filename, False, MapWinGIS.ImageType.BITMAP_FILE, Me)
    'If the save wasn't successful display an error message
    If Not success Then
        MsgBox("There were errors saving the image.")
    End If
End Sub
```

3.8.1.8 SetImageBitsDC

This function uses a bitmap selected in the specified device context handle as the data to copy into the image for which the function is called. This function requires the image to be the same width and height as the bitmap selected in the device context. Note: Use of this function requires advanced knowledge in windows graphics concepts and is intended for advanced users only.

VB.NET Usage

Function SetImageBitsDC(hDC As Integer) As Boolean

Parameters

hDC	The device context handle of the device context for which the selected bitmap is to be used to copy the bits into the image used to call the function.
ReturnValue	A boolean value representing success or failure of setting the image's bits using the bitmap selected in the specified device context.

Sample Code

```
'These are API functions needed to get a device context handle and select a bitmap for the device context
Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Integer) As Integer
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Integer, ByVal hObject As Integer) As Integer
'...
Private Sub SetImageBitsDC()
    Dim hBmpPtr As IntPtr
    Dim hBitmap As Integer, hndl As Integer, hDC As Integer
    Dim image As New MapWinGIS.Image()
    Dim success As Boolean
    Dim i As Integer, j As Integer, r As Integer, g As Integer, b As Integer
    Dim bitmap As System.Drawing.Bitmap
    Dim newbitmap As System.Drawing.Bitmap
    Dim color As System.Drawing.Color, newcolor As System.Drawing.Color
    Dim result As Long
    'Create a new Bitmap for the original image
    bitmap = New System.Drawing.Bitmap("C:\Test.bmp")
    'Create a new Bitmap for the modified image
    newbitmap = New System.Drawing.Bitmap(bitmap.Width, bitmap.Height,
Drawing.Imaging.PixelFormat.Format24bppRgb)
    'For each of the bits in the bitmap, invert the colors for the bit
    For i = 1 To bitmap.Height - 1
        For j = 1 To bitmap.Width - 1
            color = bitmap.GetPixel(j, i)
            r = 255 - color.R
            g = 255 - color.G
            b = 255 - color.B
            newcolor = newcolor.FromArgb(r, g, b)
            newbitmap.SetPixel(j, i, newcolor)
        Next
    Next
    'Get the hBitmap handle to the modified image
    hBmpPtr = newbitmap.GetHbitmap()
    'Get an integer from the intPtr hBitmap
    hBitmap = hBmpPtr.ToInt32
    'Create a compatible device context handle
    hDC = CreateCompatibleDC(0)
    'Select the bitmap for the specified device context
    result = SelectObject(hDC, hBitmap)
    'Create a new MapWinGIS.Image using the width and height of the bitmap
    success = image.CreateNew(newbitmap.Width, newbitmap.Height)
    'Set the image's data to be the bitmap selected in the specified device context
    success = image.SetImageBitsDC(hDC)
    'Add the image to the map
    hndl = Map1.AddLayer(image, True)
End Sub
```

3.8.2 Properties

3.8.2.1 CdlgFilter

Returns the common dialog filter containing all supported file extensions in string format.

VB.NET Usage

ReadOnly Property CdlgFilter() As String

Parameters

ReturnValue	The filter containing all file extensions supported by MapWinGIS.

Sample Code

```
Private Sub CdlgFilter()
    Dim img As New MapWinGIS.Image()
```

```
    'Display the supported file formats in a message box
    MsgBox(img.CdlgFilter)
  End Sub
```

3.8.2.2 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
  Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
  End Sub
```

3.8.2.3 Filename

The filename associated with the object.

VB.NET Usage

ReadOnly Property Filename() As String

Parameters

ReturnValue	The filename associated with the object.

Sample Code

```
  Private Sub Filename()
    Dim img As New MapWinGIS.Image()
    Dim filename As String
    'Open an image from disk
    img.Open("C:\image.bmp")
    'Get the filename of the image
    filename = img.Filename
  End Sub
```

3.8.2.4 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
  Inherits System.Windows.Forms.Form
    'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
  Implements MapWinGIS.ICallback
    '...
 #Region "ICallback Members"
   Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
        'Display the error message in a label
        LabelError.Text = ErrorMsg
```

```
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
            'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
            'Display the message in a label
        Label2.Text = Message
    End Sub
#End Region
            '...
```

3.8.2.5 Height

Gets the height of the image in pixels.

VB.NET Usage

ReadOnly Property Height() As Integer

Parameters

ReturnValue	The height of the image in pixels.

Sample Code

```
    Private Sub ImageHeight()
        Dim image As New MapWinGIS.Image()
        Dim height As Integer
        'Get the height of the image in pixels
        height = image.Height
    End Sub
```

3.8.2.6 ImageType

Gets the image object's image type.

VB.NET Usage

ReadOnly Property ImageType() MapWinGIS.ImageType

Parameters

ReturnValue	The image type of the image object.

Sample Code

```
    Private Sub ImageType()
        Dim image As New MapWinGIS.Image()
        Dim type As MapWinGIS.ImageType
        'Get the image type of the image
        type = image.ImageType
    End Sub
```

3.8.2.7 IsInRam

Returns true when the image is loaded in RAM memory, and returns false when the image is not loaded in RAM memory.

VB.NET Usage

ReadOnly Property IsInRam() As Boolean

Parameters

ReturnValue	A boolean value representing whether the image is stored in RAM memory or not.

Sample Code

```
    Private Sub ImageIsInRam()
        Dim image As New MapWinGIS.Image()
        Dim inRam As Boolean
        'Get the status of IsInRam
        inRam = image.IsInRam
        'If the image is in RAM, display a message to the user
        If inRam Then
            MsgBox("The image is stored in RAM memory.")
```

```
    End If
  End Sub
```

3.8.2.8 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
  Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
    If k = "Map1" Then
      'Set the map's key to "My Map1"
      Map1.Key = "My Map1"
    Else
      'Set the map's key to "Map1"
      Map1.Key = "Map1"
    End If
  End Sub
```

3.8.2.9 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
  Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
  End Sub
```

3.8.2.10 Picture

Gets or sets the picture object.

VB.NET Usage

Property Picture() As stdole.IPictureDisp

Parameters

ReturnValue	The picture for the image object.

Sample Code

```
  Private Sub Picture()
    Dim image As New MapWinGIS.Image()
    Dim p As stdole.IPictureDisp
    Dim c As New ImageConverter() 'This is a class defined below
    'Set the picture for the image object
    image.Picture = c.ImageToPicture(ImageList1.Images(0))
```

```
        'Get the picture from the image object
        p = image.Picture
    End Sub
'...
'This code was adapted from .NET 247 forum contribution by Ray Frankulin
'http://www.dotnet247.com/247reference/msgs/20/103409.aspx
'This class helps convert from an Image to an IPictureDisp in VB.NET
Public Class ImageConverter
    Inherits System.Windows.Forms.AxHost
    Public Sub New()
        MyBase.New("00000000-9999-0000-9999-000000000001")
    End Sub
    Public Function ImageToPicture(ByRef Data As Image) As Object
        Return MyBase.GetIPictureDispFromPicture(Data)
    End Function
End Class
```

3.8.2.11 TransparencyColor

Gets or sets the color which is used as the transparency color for the image.

VB.NET Usage

Property TransparencyColor() As System.UInt32

Parameters

ReturnValue	The transparency color for the image. This is a UInt32 representation of an RGB value.

Sample Code

```
    Private Sub TransparencyColor()
        Dim image As New MapWinGIS.Image()
        Dim col As UInt32
        'Set white to be used for the transparency color
        col = System.Convert.ToUInt32(RGB(255, 255, 255))
        'Set the transparency color for the image
        image.TransparencyColor = col
        'Get the transparency color for the image
        col = image.TransparencyColor
    End Sub
```

3.8.2.12 UseTransparencyColor

Gets or sets whether or not the transparency color is being used when the image is displayed.

VB.NET Usage

Property UseTransparencyColor() As Boolean

Parameters

ReturnValue	A boolean value representing whether the transparency color is being used when the image is displayed.

Sample Code

```
    Private Sub UseTransparencyColor()
        Dim image As New MapWinGIS.Image()
        'Check if the transparency color is being used when the image is displayed
        If image.UseTransparencyColor Then
            'Set the image not to use the transparency color
            image.UseTransparencyColor = False
        Else
            'Set the image to use the transparency color
            image.UseTransparencyColor = True
        End If
    End Sub
```

3.8.2.13 Value

Gets or sets the value of a pixel in the image.

VB.NET Usage

Property Value(Row As Integer, Col As Integer) As Integer

Parameters

Row	The row in the image for which the value is required.
Col	The column in the image for which the value is required.
ReturnValue	The value of the specified pixel in the image.

Sample Code

```
Private Sub ImagePixelValue()
    Dim image As New MapWinGIS.Image()
    Dim row As Integer, col As Integer, value As Integer
    'Set the value of the desired row
    row = 10
    'Set the value of the desired column
    col = 10
    'Set the value to assign to the specified row and column in the image
    value = 250
    'Set the value of the specified row and column in the image to the specified value
    image.Value(row, col) = value
    'Get the value of the specified row and column in the image
    value = image.Value(row, col)
End Sub
```

3.8.2.14 Width

Gets the width of the image in pixels.

VB.NET Usage

ReadOnly Property Width() As Integer

Parameters

ReturnValue	The width of the image in pixels.

Sample Code

```
Private Sub ImageWidth()
    Dim image As New MapWinGIS.Image()
    Dim width As Integer
    'Get the width of the image in pixels
    width = image.Width
End Sub
```

3.8.2.15 XllCenter

Gets or sets the x coordinate of the center of the lower-left pixel in the image.

VB.NET Usage

Property XllCenter() Double

Parameters

ReturnValue	The x coordinate of the center of the lower-left pixel in the image.

Sample Code

```
Private Sub ImageXllCenter()
    Dim image As New MapWinGIS.Image()
    Dim x As Double
    'Set the value to use for the x coordinate
    x = 100
    'Set the x coordinate of the lower-left pixel in the image
    image.XllCenter = x
    'Get the x coordinate of the lower-left pixel in the image
    x = image.XllCenter
End Sub
```

3.8.2.16 YllCenter
Gets or sets the y coordinate of the center of the lower-left pixel in the image.

VB.NET Usage

Property YllCenter() As Double

Parameters

ReturnValue	The y coordinate of the center of the lower-left pixel in the image.

Sample Code

```
Private Sub ImageYllCenter()
    Dim image As New MapWinGIS.Image()
    Dim y As Double
    'Set the value to use for the y coordinate
    y = 100
    'Set the y coordinate of the lower-left pixel in the image
    image.YllCenter = y
    'Get the y coordinate of the lower-left pixel in the image
    y = image.YllCenter
End Sub
```

3.9 Map

The map object represents a map which displays grids, images, and shapefiles. This is the cornerstone of the MapWinGIS ActiveX control.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.9.1 Functions

3.9.1.1 AddLayer
Adds a layer to the map.

VB.NET Usage

Sub AddLayer(Object As Object, Visible As Boolean) As Integer

Parameters

Object	The object (Image or Shapefile) to add to the map.
Visible	Sets whether the layer is visible after being added to the map.
Return Value	Returns the integer handle for the layer added to the map.

Sample Code

```
Private Sub AddLayerToMap()
    'Add Shapefile to Map
    Dim hnd As Integer
    Dim sf As MapWinGIS.Shapefile
    'Open a shapefile
    sf.Open("C:\CITIES.SHP", Me)
    'Add shapefile to map, saving layer handle
    hnd = Map1.AddLayer(sf, True)
    '****************************************
    'Add Grid to Map
    Dim grid As MapWinGIS.Grid
    'Open a grid
    grid.Open("C:\GRID.ASC", MapWinGIS.GridDataType.UnknownDataType, True,
MapWinGIS.GridFileType.UseExtension, Me)
    'create the grid coloring scheme
    Dim scheme As New MapWinGIS.GridColorScheme()
    scheme.NoDataColor = System.Convert.ToUInt32(RGB(0, 0, 0))
```

```
    scheme.UsePredefined(grid.Minimum, grid.Maximum, MapWinGIS.PredefinedColorScheme.SummerMountains)
    'convert the grid to an image
    Dim image As MapWinGIS.Image
    Dim util As New MapWinGIS.Utils()
    image = util.GridToImage(grid, scheme, Me)
    'add the image to the map
    Map1.AddLayer(image, True)
End Sub
```

3.9.1.2 ApplyLegendColors

Applies the coloring scheme to a layer (the layer handle is specified in the coloring scheme object).

VB.NET Usage

Function ApplyLegendColors(Legend As Object) As Boolean

Parameters

Legend	The coloring scheme to apply.
Return Value	Boolean value indicating the successful application when true, unsuccessful application when false.

Sample Code

```
Private Sub LegendColors()
    'open a shape file
    Dim sf As New MapWinGIS.Shapefile()
    sf.Open("C:\CITIES.SHP", Me)
    Dim handle As Long, i As Long, fieldIndex As Long
    Dim colorScheme As New MapWinGIS.ShapefileColorScheme()
    Dim break As MapWinGIS.ShapefileColorBreak
    'add the shapefile to the map
    handle = Map1.AddLayer(sf, True)
    'get the field index to color by
    fieldIndex = 0
    'find the max and min values for that field
    Dim min As Double, max As Double
    min = getMinValueForField(sf, fieldIndex)
    max = getMaxValueForField(sf, fieldIndex)
    With colorScheme
        .LayerHandle = handle
        'the field value to color by
        colorScheme.FieldIndex = fieldIndex
        'create a new break object
        break = New MapWinGIS.ShapefileColorBreak()
        break.Caption = "test"
        break.StartColor = System.Convert.ToUInt32(RGB(255, 0, 0))
        break.EndColor = System.Convert.ToUInt32(RGB(0, 0, 255))
        break.StartValue = min
        break.EndValue = max
        'add the break
        colorScheme.Add(break)
    End With
    'apply the coloring scheme for this layer
    Map1.ApplyLegendColors(colorScheme)
End Sub
```

3.9.1.3 GetColorScheme

Returns the color scheme for the specified layer.

VB.NET Usage

Function GetColorScheme(LayerHandle As Integer) As Object

Parameters

LayerHandle	The layer handle of the layer for which the coloring scheme is required
Return Value	The color scheme for the specified layer.

Sample Code

```
Private Sub ColorScheme()
    Dim hnd As Integer
    Dim clscheme As MapWinGIS.ShapefileColorScheme
    'Get handle for layer 0
    hnd = Map1.get_LayerHandle(0)
    'Get color scheme for layer 0
    clscheme = Map1.GetColorScheme(hnd)
End Sub
```

3.9.1.4 MoveLayer

Moves a layer in the map from the initial position to a target position.

VB.NET Usage

Function MoveLayer(InitialPosition As Integer, TargetPosition As Integer) As Boolean

Parameters

InitialPosition	The initial position of the layer to be moved.
TargetPosition	The final position of the layer being moved.

Sample Code

```
Private Sub MoveLayer()
    Dim hnd As Integer
    'Get handle of layer 0
    hnd = Map1.get_LayerHandle(0)
    'Move layer from position 0 to position 1
    Map1.MoveLayer(Map1.get_LayerPosition(hnd), 1)
End Sub
```

3.9.1.5 MoveLayerBottom

Moves the specified layer below all other layers.

VB.NET Usage

Function MoveLayerBottom(InitialPosition As Integer) As Boolean

Parameters

InitialPosition	The position of the layer to move to the bottom.
Return Value	Boolean value representing success when true, and failure when false.

Sample Code

```
Private Sub MoveLayerBottom()
    Dim hnd As Integer
    'Get handle of layer 0
    hnd = Map1.get_LayerHandle(0)
    'Move layer from position 0 below all other layers
    Map1.MoveLayerBottom(Map1.get_LayerPosition(hnd))
End Sub
```

3.9.1.6 MoveLayerDown

Moves the specified layer down one layer in the map.

VB.NET Usage

Function MoveLayerDown(InitialPosition As Integer) As Boolean

Parameters

InitialPosition	The position of the layer to be moved.
ReturnValue	Boolean value representing succes when true, and failure when false.

Sample Code

```
Private Sub MoveLayerDown()
    Dim hnd As Integer
```

```
    'Get handle of layer 0
    hnd = Map1.get_LayerHandle(0)
    'Move layer down one layer
    Map1.MoveLayerDown(Map1.get_LayerPosition(hnd))
End Sub
```

3.9.1.7 MoveLayerTop

Moves the specified layer to the top of all other layers.

VB.NET Usage

Function MoveLayerTop(InitialPosition As Integer) As Boolean

Parameters

InitialPosition	The position of the layer to be moved.
ReturnValue	Boolean value representing success when true, failure when false.

Sample Code

```
Private Sub MoveLayerTop()
    Dim hnd As Integer
    'Get handle of layer 0
    hnd = Map1.get_LayerHandle(0)
    'Move layer above all other layers
    Map1.MoveLayerTop(Map1.get_LayerPosition(hnd))
End Sub
```

3.9.1.8 MoveLayerUp

Moves the specified layer up one layer in the map.

VB.NET Usage

Function MoveLayerUp(InitialPosition As Integer) As Boolean

Parameters

InitialPosition	The initial position of the layer to be moved.
ReturnValue	Boolean value representing success when true, failure when false.

Sample Code

```
Private Sub MoveLayerUp()
    Dim hnd As Integer
    'Get handle of layer 0
    hnd = Map1.get_LayerHandle(0)
    'Move layer up one layer
    Map1.MoveLayerUp(Map1.get_LayerPosition(hnd))
End Sub
```

3.9.1.9 NewDrawing

Creates a new drawing layer on the map returning its handle.

VB.NET Usage

Function NewDrawing(Projection As MapWinGIS.tkDrawReferenceList) As Integer

Parameters

Projection	Sets the coordinate system to use for the new drawing layer to be created. (ScreenReferenced uses pixels in screen coordinates. SpatiallyReferenced uses projected map units.)
ReturnValue	The handle for the new drawing layer in the map.

Sample Code

```
Private Sub NewDrawing(ByVal DrawSpatiallyReferenced As Boolean)
    Dim hndl As Integer
    Dim xProjCenter As Double, yProjCenter As Double
    Dim xScreenCenter As Double, yScreenCenter As Double
```

```
    Dim extents As MapWinGIS.Extents
    'Get map extents
    extents = Map1.Extents
    'find the center of the view
    xProjCenter = extents.xMin + ((extents.xMax - extents.xMin) / 2)
    yProjCenter = extents.yMin + ((extents.yMax - extents.yMin) / 2)
    'create a new drawing surface
    If (DrawSpatiallyReferenced) Then
        'Use spatially referenced coordinates for new drawing layer and save handle
        hndl = Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlSpatiallyReferencedList)
        'Draw circle on last created drawing surface
        Map1.DrawCircle(xProjCenter, yProjCenter, 100, System.Convert.ToUInt32(RGB(255, 0, 0)), True)
    Else
        'Get center of the screen in pixel units from projected map coordinates
        Map1.ProjToPixel(xProjCenter, yProjCenter, xScreenCenter, yScreenCenter)
        'Use screen referenced coordinates for new drawing layer and save handle
        hndl = Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlScreenReferencedList)
        'Draw circle on last created drawing surface
        Map1.DrawCircle(xScreenCenter, yScreenCenter, 100, System.Convert.ToUInt32(RGB(255, 0, 0)), True)
    End If
End Sub
```

3.9.1.10 SetImageLayerColorScheme

Sets a coloring scheme to be associated with an image layer. The coloring scheme is for reference purposes only and has no effect the coloring or display of the image.

VB.NET Usage

Function SetImageLayerColorScheme(LayerHandle As Integer, ColorScheme As Object) As Boolean

Parameters

LayerHandle	The handle of the image layer to attach the coloring scheme to.
ColorScheme	The coloring scheme to attatch to the specified image layer.
ReturnValue	Boolean value representing success when true, failure when false.

Sample Code

```
Private Sub ImageLayerColoringScheme()
    Dim hndl As Integer
    Dim scheme As New MapWinGIS.GridColorScheme()
    Dim break As New MapWinGIS.GridColorBreak()
    'Create color break
    break.Caption = "forest"
    break.HighColor = System.Convert.ToUInt32(RGB(0, 255, 0))
    break.LowColor = System.Convert.ToUInt32(RGB(0, 255, 0))
    break.LowValue = 0
    break.HighValue = 0
    'Add color break to coloring scheme
    scheme.InsertBreak(break)
    'Get handle for layer 0
    hndl = Map1.get_LayerHandle(0)
    'set the coloring scheme for this layer
    Map1.SetImageLayerColorScheme(hndl, scheme)
End Sub
```

3.9.1.11 SnapShot

Takes snap shot of the contents of the map within the bounds of the specified rectangle, returning an image of the results.

VB.NET Usage

Function SnapShot(BoundBox As Object) As Object

Parameters

BoundBox	The bounds (rectangle) to get the snapshot from.
ReturnValue	An image of the contents of the map displayed inside the bounds of the specified rectangle.

Sample Code

```
Private Sub SnapShot()
    Dim image As New MapWinGIS.Image()
    Dim extents As MapWinGIS.Extents
    'Set extents to be the extents of the map
    extents = CType(Map1.Extents, MapWinGIS.Extents)
    'Take a picture of what is being displayed in map1 and store it in image
    image = Map1.SnapShot(extents)
End Sub
```

3.9.1.12 ZoomToPrev

Zooms the map view to the previous extents if there are previous extents in the extents history.

VB.NET Usage

Function ZoomToPrev() As Integer

Parameters

ReturnValue	Returns the number of extents left in the extents history after zooming to previous extents.

Sample Code

```
Private Sub ZoomPrevious()
    Dim history As Integer
    'Set map view to previous zoom extents, storing the number of previous extents remaining
    history = Map1.ZoomToPrev
End Sub
```

3.9.2 Subs

3.9.2.1 AddLabel

Adds a label to the map.

VB.NET Usage

Sub AddLabel(LayerHandle As Integer, Text As String, Color As System.UInt32, x As Double, y As Double, hJustification As MapWinGIS.tkHJustification)

Parameters

LayerHandle	The handle of the layer where the label will be added to.
Text	The text to be used for the label.
Color	The color of the added label. This is a UInt32 representation of an RGB color.
x	The x coordinate in projected map units which determines where the label will be added on the map.
y	The y coordinate in projected map units which determines where the label will be added on the map
hJustification	Specifies whether to justify the label's text right, left, or center.

Sample Code

```
Private Sub AddLabel()
    Dim hndl As Integer, field As Integer, i As Integer
    Dim sf As MapWinGIS.Shapefile
    Dim text As String
    Dim x As Double, y As Double
    Dim col As UInt32
    'Get handle for layer 0 which must contain a shapefile
    hndl = Map1.get_LayerHandle(0)
    'Get the shapefile contained in layer 0
    sf = Map1.get_GetObject(hndl)
    'Set shapefile field to use when labeling layer as field 0
    field = 0
    'Set the color for the labels to be black
    col = System.Convert.ToUInt32(RGB(0, 0, 0))
    'Label every shape in the shapefile
    For i = 0 To sf.NumShapes - 1
```

```
        'Set the text for this shape
        text = sf.CellValue(field, i)
        'Set the x and y coordinates for this label to be the min x and y coordinates of this shape
        x = sf.Shape(i).Extents.xMin
        y = sf.Shape(i).Extents.yMin
        'Add the label to the layer by the shape centering the text
        Map1.AddLabel(hndl, text, col, x, y, MapWinGIS.tkHJustification.hjCenter)
    Next
End Sub
```

3.9.2.2 AddLabelEx

Adds an extended label to the map, allowing for rotated labels.

VB.NET Usage

Sub AddLabelEx(LayerHandle As Integer, Text As String, Color As System.UInt32, x As Double, y As Double, hJustification As MapWinGIS.tkHJustification, Rotation As Double)

Parameters

LayerHandle	The handle of the layer where the label will be added to.
Text	The text to be used for the label.
Color	The color of the added label. This is a UInt32 representation of an RGB color.
x	The x coordinate in projected map units which determines where the label will be added on the map.
y	The y coordinate in projected map units which determines where the label will be added on the map
hJustification	Specifies whether to justify the label's text right, left, or center.
Rotation	The number of degrees to rotate the label. Positive angles rotate the text counter-clockwise, and negative angles rotate the text clockwise.

Sample Code

```
Private Sub AddLabel()
    Dim hndl As Integer, field As Integer, i As Integer
    Dim sf As MapWinGIS.Shapefile
    Dim text As String
    Dim x As Double, y As Double
    Dim col As UInt32
    'Get handle for layer 0 which must contain a shapefile
    hndl = Map1.get_LayerHandle(0)
    'Get the shapefile contained in layer 0
    sf = Map1.get_GetObject(hndl)
    'Set shapefile field to use when labeling layer as field 0
    field = 0
    'Set the color for the labels to be black
    col = System.Convert.ToUInt32(RGB(0, 0, 0))
    'Label every shape in the shapefile
    For i = 0 To sf.NumShapes - 1
        'Set the text for this shape
        text = sf.CellValue(field, i)
        'Set the x and y coordinates for this label to be the min x and y coordinates of this shape
        x = sf.Shape(i).Extents.xMin
        y = sf.Shape(i).Extents.yMin
        'Add the label to the layer by the shape centering the text and rotating it 45 degrees
        Map1.AddLabelEx(hndl, text, col, x, y, MapWinGIS.tkHJustification.hjCenter, 45)
    Next
End Sub
```

3.9.2.3 ClearDrawing

Clears all drawings on the drawing layer specified.

VB.NET Usage

Sub ClearDrawing(DrawHandle As Integer)

Parameters

DrawHandle	Drawing handle of the drawing layer for which all drawings are to be cleared.

Sample Code

```
Private Sub ClearDrawing()
    Dim draw_hndl As Integer
    'Create new drawing layer on map
    draw_hndl = Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlScreenReferencedList)
    'Draw filled red circle at x = 100, y = 100, with radius 50
    Map1.DrawCircle(100, 100, 50, System.Convert.ToUInt32(RGB(255, 0, 0)), True)
    'Allow user to see circle before it is cleared
    MsgBox ("Circle drawn")
    'Clear drawing we just made above
    Map1.ClearDrawing(draw_hndl)
End Sub
```

3.9.2.4 ClearDrawings

Clears all drawings on all drawing layers. This method is slower than using ClearDrawing on a specific layer.

VB.NET Usage

Sub ClearDrawings()

Parameters

None

Sample Code

```
Private Sub ClearAllDrawings()
    'Clear drawings in all drawing layers
    Map1.ClearDrawings()
End Sub
```

3.9.2.5 ClearLabels

Clears all labels that have been added to the specified layer.

VB.NET Usage

Sub ClearLabels(LayerHandle As Integer)

Parameters

LayerHandle	Layer handle of the layer to clear labels from.

Sample Code

```
Private Sub ClearAllLabels()
    Dim hndl As Integer
    'Get handle for layer 0
    hndl = Map1.get_LayerHandle(0)
    'Clear all labels in layer 0
    Map1.ClearLabels(hndl)
End Sub
```

3.9.2.6 DrawCircle

Draws a circle on the last drawing layer created by NewDrawing.

VB.NET Usage

Sub DrawCircle(x As Double, y As Double, pixelRadius As Double, Color As System.UInt32, fill As Boolean)

Parameters

x	Center x coordinate for the circle to be drawn.
y	Center y coordinate for the circle to be drawn.
pixelRadius	Radius in pixels of the circle to be drawn.

Color	Color of the circle to be drawn. This is a UInt32 representation of an RGB color.
fill	Boolean value which determines whether the circle will be drawn with a fill or not.

Sample Code

```
Private Sub DrawCircle()
    Dim draw_hndl As Integer
    'Create new drawing layer on map
    draw_hndl = Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlScreenReferencedList)
    'Draw filled red circle at x = 100, y = 100, with radius 50
    Map1.DrawCircle(100, 100, 50, System.Convert.ToUInt32(RGB(255, 0, 0)), True)
End Sub
```

3.9.2.7 DrawLine

Draws a line on the last drawing layer created using NewDrawing.

VB.NET Usage

Sub DrawLine(x1 As Double, y1 As Double, x2 As Double, y2 As Double, pixelWidth As Integer, Color As System.UInt32)

Parameters

x1	X coordinate of the first point used to draw the line.
y1	Y coordinate of the first point used to draw the line.
x2	X coordinate of the second point used to draw the line.
y2	Y coordinate of the second point used to draw the line.
pixelWidth	Width of the line in pixels.
color	Color to draw the line with. This is a UInt32 representation of an RGB value.

Sample Code

```
Private Sub DrawLine()
    Dim draw_hndl As Integer, width As Integer
    Dim x1 As Double, y1 As Double, x2 As Double, y2 As Double
    'Set point 1 as x = 100, y = 100
    x1 = 100
    y1 = 100
    'Set point2 as x = 500, y = 500
    x2 = 500
    y2 = 500
    'Set pixel width for the line as 1
    width = 1
    'Create new drawing layer on map
    draw_hndl = Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlScreenReferencedList)
    'Draw green line from point 1 to point 2
    Map1.DrawLine(x1, y1, x2, y2, width, System.Convert.ToUInt32(RGB(0, 255, 0)))
End Sub
```

3.9.2.8 DrawPoint

Draws a point on the last drawing layer created by NewDrawing.
See also NewDrawing

VB.NET Usage

Sub DrawPoint(x As Double, y As Double, pixelSize As Integer, color As System.UInt32)

Parameters

x	The x coordinate of the point to draw.
y	The x coordinate of the point to draw.
pixelSize	The size in pixels of the point to be drawn.
color	The color of the point to be drawn. This is a UInt32 representation of an RGB color.

Sample Code

```
Private Sub DrawPoint()
    Dim draw_hndl As Integer, size As Integer
    Dim x1 As Double, y1 As Double
```

```
    'Set point as x = 100, y = 100
    x1 = 100
    y1 = 100
    'Set pixel size for point
    size = 5
    'Create new drawing layer on map
    draw_hndl = Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlScreenReferencedList)
    'Draw black point on map
    Map1.DrawPoint(x1, y1, size, System.Convert.ToUInt32(RGB(0, 0, 0)))
End Sub
```

3.9.2.9 DrawPolygon

Draws a polygon to the last drawing layer created using NewDrawing.

VB.NET Usage

Sub DrawPolygon(xPts() As Double, yPts() As Double, numPoints As Integer, color As System.UInt32, fill As Boolean)

Parameters

xPts	An array containing x-coordinates for each point in the polygon.
yPts	An array containing y-coordiniates for each point in the polygon.
numPoints	The number of points in the polygon.
color	The color to use when drawing the polygon. This is a UInt32 representation of an RGB color.
fill	A boolean value representing whether the polygon is drawn with a fill or not.

Sample Code

```
Private Sub DrawPolygon()
    Dim draw_hndl As Integer, num_points As Integer
    Dim x1(6) As Double, y1(6) As Double
    'Set x coordinates for the 6 points of the polygon
    x1(0) = 300
    x1(1) = 200
    x1(2) = 300
    x1(3) = 100
    x1(4) = 100
    x1(5) = 300
    'Set y coordinates for the 6 points in the polygon
    y1(0) = 300
    y1(1) = 200
    y1(2) = 100
    y1(3) = 100
    y1(4) = 300
    y1(5) = 300
    'Set number of points used to draw polygon as 6
    num_points = 6
    'Create new drawing layer on map
    draw_hndl = Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlScreenReferencedList)
    'Draw red polygon on map using the 6 points defined in the x1 and y1 arrays
    Map1.DrawPolygon(x1, y1, num_points, System.Convert.ToUInt32(RGB(255, 0, 0)), False)
End Sub
```

3.9.2.10 LayerFont

Sets the font to use when drawing labels for a layer.

VB.NET Usage

Sub LayerFont(LayerHandle As Integer, FontName As String, FontSize As Integer)

Parameters

LayerHandle	Layer handle of the layer for which the label fonts are to be set.
FontName	Name of the font to use for the labels. (Ex: "Arial")
FontSize	Size of the font to use for the labels.

Sample Code

```
Private Sub SetLayerFont()
    Dim hndl As Integer
    'Get handle for layer 0
    hndl = Map1.get_LayerHandle(0)
    'Set the font for layer 0 to Arial, with font size 18
    Map1.LayerFont(hndl, "Arial", 18)
End Sub
```

3.9.2.11 LockWindow

Locks the window so that any changes will not be displayed until it is unlocked. This is very useful if you are making a large number of changes at once and don't want the map to be redrawn between each change.

VB.NET Usage

Sub LockWindow(LockMode As MapWinGIS.tkLockMode)

Parameters

LockMode	Lock or unlock the map.

Sample Code

```
Private Sub LockWindow()
    'Lock the map's window while we make multiple changes
    Map1.LockWindow(MapWinGIS.tkLockMode.lmLock)
    'Draw a bunch of circles, lines, and polygons
    DrawCircles()
    DrawLines()
    DrawPolygons()
    'Unlock the map's window to show the changes we've made
    Map1.LockWindow(MapWinGIS.tkLockMode.lmUnlock)
End Sub
```

3.9.2.12 PixelToProj

Converts pixel coordinates to projected map coordinates.

VB.NET Usage

Sub PixelToProj(pixelX As Double, pixelY As Double, ByRef projX As Double, ByRef projY As Double)

Parameters

pixelX	The x pixel coordinate to be converted into the projected x map coordinate.
pixelY	The y pixel coordinate to be converted into the projected y map coordinate.
projX	The projected x map coordinate is returned through this reference parameter.
projY	The projected y map coordinate is returned through this reference parameter.

Sample Code

```
Private Sub Map1_MouseUpEvent(ByVal sender As System.Object, ByVal e As
AxMapWinGIS._DMapEvents_MouseUpEvent) Handles Map1.MouseUpEvent
    'This will draw a point where you click on the map if you have a
    'spatially referenced shapefile or grid displayed on the map
    Dim x As Double, y As Double
    'Convert pixel coordinates of mouse up event e.x and e.y to projected coordinates returned by x and y
    Map1.PixelToProj(e.x, e.y, x, y)
    'Create a new drawing layer in map1
    Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlSpatiallyReferencedList)
    'Draw a red point where the mouse up event occured in map1
    Map1.DrawPoint(x, y, 5, System.Convert.ToUInt32(RGB(255, 0, 0)))
End Sub
```

3.9.2.13 ProjToPixel

Converts projected map coordinates into screen pixel units.

VB.NET Usage

Sub ProjToPixel(projX As Double, projY As Double, ByRef pixelX As Double, ByRef pixelY As Double)

Parameters

projX	The projected x map coordinate to be converted into the x pixel coordinate.
projY	The projected y map coordinate to be converted into the y pixel coordinate.
pixelX	The pixel x coordinate is returned through this reference parameter.
pixelY	The pixel y coordinate is returned through this reference parameter.

Sample Code

```
Private Sub DrawCenterCircle()
    Dim draw_hndl As Integer
    Dim xProjCenter As Double, yProjCenter As Double
    Dim xScreenCenter As Double, yScreenCenter As Double
    Dim extents As MapWinGIS.Extents
    'Get the extents of Map1
    extents = Map1.Extents
    'Calculate the center x coordinate of the map
    xProjCenter = extents.xMin + ((extents.xMax - extents.xMin) / 2)
    'Calculate the center y coordinate of the map
    yProjCenter = extents.yMin + ((extents.yMax - extents.yMin) / 2)
    'Convert from the projected map coordinates of the center of the map to pixel coordinates
    Map1.ProjToPixel(xProjCenter, yProjCenter, xScreenCenter, yScreenCenter)
    'Create a new drawing layer
    draw_hndl = Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlScreenReferencedList)
    'Draw a circle centered around the center of the map's view
    Map1.DrawCircle(xScreenCenter, yScreenCenter, 50, System.Convert.ToUInt32(RGB(255, 0, 0)), False)
End Sub
```

3.9.2.14 Redraw

Redraws all layers in the map if the map is not locked.

VB.NET Usage

Sub Redraw()

Parameters

None	

Sample Code

```
Private Sub ReDrawMap()
    'Check if the map is locked
    If Not Map1.IsLocked Then
        'If the map is unlocked, then redraw all of the map's layers
        Map1.Redraw()
    End If
End Sub
```

3.9.2.15 RemoveAllLayers

Removes all layers from the map.

VB.NET Usage

Sub RemoveAllLayers()

Parameters

None	

Sample Code

```
Private Sub RemoveAllLayers()
    'Remove all layers from the map
    Map1.RemoveAllLayers()
End Sub
```

3.9.2.16 RemoveLayer

Removes the specified layer from the map.

VB.NET Usage

Sub RemoveLayer(LayerHandle As Integer)

Parameters

LayerHandle	The handle of the layer to be removed from the map.

Sample Code

```
Private Sub RemoveLayer()
    Dim hndl As Integer
    'Get layer handle for layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Remove layer at position 0 from the map
    Map1.RemoveLayer(hndl)
End Sub
```

3.9.2.17 Resize

Resizes the map to the given width and height.

VB.NET Usage

Property Size() As System.Drawing.Size

Parameters

ReturnValue	The size of the map returned as a System.Drawing.Size object. (Width and height can be accessed by using mySize.Width or mySize.Height.)

Sample Code

```
Private Sub ResizeMap()
    Dim width As Integer, height As Integer
    Dim newSize As Size
    'Set the width and height to be used to resize map in pixels
    width = 200
    height = 200
    'Set newSize with new height and width values
    newSize.Width = width
    newSize.Height = height
    'Resize the map with the new width and height of newSize
    Map1.Size = newSize
End Sub
```

3.9.2.18 ShowToolTip

Displays a tootip under the cursor.

VB.NET Usage

Sub ShowToolTip(Text As String, Milliseconds As Integer)

Parameters

Text	The message to display in the tooltip.
Milliseconds	The length of time to display the tooltip message measured in milliseconds.

Sample Code

```
Private Sub Map1_MouseMoveEvent(ByVal sender As Object, ByVal e As
AxMapWinGIS._DMapEvents_MouseMoveEvent) Handles Map1.MouseMoveEvent
    Dim hndl As Integer
    Dim projX As Double, projY As Double
```

```
    Dim sf As MapWinGIS.Shapefile
    'Get handle for layer at position 0 in map
    hndl = Map1.get_LayerHandle(0)
    'Get shapefile in layer at position 0 in map
    sf = Map1.get_GetObject(hndl)
    'Convert mouse move x and y into projected map coordinates
    Map1.PixelToProj(e.x, e.y, projX, projY)
    'See if the mouse has moved over shape 0 in the shapefile
    If sf.PointInShape(0, projX, projY) Then
        'Show the tooltip for 2 seconds
        Map1.ShowToolTip("Shape 0", 2000)
    End If
End Sub
```

3.9.2.19 UpdateImage

Updates the display of the specified image object.

VB.NET Usage

Sub UpdateImage(LayerHandle As Integer)

Parameters

LayerHandle	The handle of the image layer to be updated.

Sample Code

```
Private Sub UpdateImage()
    Dim hndl As Integer
    'Get handle for image layer at position 0 in map
    hndl = Map1.get_LayerHandle(0)
    'Update the map's display of the specified image
    Map1.UpdateImage(hndl)
End Sub
```

3.9.2.20 ZoomIn

Zooms the display in by the given factor.

VB.NET Usage

Sub ZoomIn(Percent As Double)

Parameters

Percent	The factor to zoom in by.

Sample Code

```
Private Sub ZoomIn()
    'Zoom in map view by 0.3 percent
    Map1.ZoomIn(0.3)
End Sub
```

3.9.2.21 ZoomOut

Zooms the display out by the specified factor.

VB.NET Usage

Sub ZoomOut(Percent As Double)

Parameters

Percent	The factor to zoom out by.

Sample Code

```
Private Sub ZoomOut()
    'Zoom out map view by 0.3 percent
    Map1.ZoomOut(0.3)
End Sub
```

3.9.2.22 ZoomToLayer
Zooms the map display to the specified layer.

VB.NET Usage

Sub ZoomToLayer(LayerHandle As Integer)

Parameters

LayerHandle	The handle of the layer to zoom to.

Sample Code

```
Private Sub ZoomToLayer()
    Dim hndl As Integer
    'Get handle for layer at position 0 in map
    hndl = Map1.get_LayerHandle(0)
    'Zoom to the layer at position 0 in the map
    Map1.ZoomToLayer(hndl)
End Sub
```

3.9.2.23 ZoomToMaxExtents
Zooms the map to the maximum extents of all loaded layers. (Note: Layers which are not visible are still used to compute maximum extents.)

VB.NET Usage

Sub ZoomToMaxExtents()

Parameters

None

Sample Code

```
Private Sub ZoomToMaxExtents()
    'Zoom map view to the maximum extents of all layers(whether they are visible or not)
    Map1.ZoomToMaxExtents()
End Sub
```

3.9.2.24 ZoomToMaxVisibleExtents
Zooms the map to the maximum extents of all loaded visible layers. (Note: Layers which are not visible are not used to compute maximum extents.)

VB.NET Usage

Sub ZoomToMaxVisibleExtents()

Parameters

None

Sample Code

```
Private Sub ZoomToMaxExtents()
    'Zoom map view to the maximum extents of all visible layers
    Map1.ZoomToMaxVisibleExtents()
End Sub
```

3.9.2.25 ZoomToShape
Zooms the map display to the specified shape in the shapefile contained by the specified layer.

VB.NET Usage

Sub ZoomToShape(LayerHandle As Integer, Shape As Integer)

Parameters

LayerHandle	The handle of the layer containing the shape to zoom to.

Shape	The index of the shape to zoom to.

Sample Code

```
Private Sub ZoomToShape()
    Dim hndl As Integer
    'Get handle for layer at position 0 in map
    hndl = Map1.get_LayerHandle(0)
    'Zoom to shape 0 in the shapefile contained in the specified layer
    Map1.ZoomToShape(hndl, 0)
End Sub
```

3.9.3 Properties

3.9.3.1 BackColor

Gets or sets the background color of the map. (When using VB.NET the color is represented as a System.Drawing.Color. When using VB 6 the color can be an OLE_COLOR or an integer representation of an RGB value.)

VB.NET Usage

Property BackColor() As System.Drawing.Color

Parameters

ReturnValue	The back color of the map returned as a System.Drawing.Color object.

Sample Code

```
Private Sub myBackColor()
    Dim col As System.Drawing.Color
    'Get the backcolor of the map
    col = Map1.BackColor
    'Set the backcolor of the map
    Map1.BackColor = System.Drawing.Color.Beige
End Sub
```

3.9.3.2 CursorMode

Gets or sets the cursor mode for the map. The cursor mode determines how the map handles mouse click events on the map. The only mode not handled by the map is cmNone.

VB.NET Usage

Property CursorMode() As MapWinGIS.tkCursorMode

Parameters

ReturnValue	The current cursor mode for the map as a MapWinGIS.tkCursorMode object.

Sample Code

```
Private Sub ToolBar1_ButtonClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ToolBarButtonClickEventArgs) Handles ToolBar1.ButtonClick
    Dim tag As String
    'Get the tag for the button that was clicked on the toolbar
    tag = e.Button.Tag
    'Select the right cursor mode to switch to
    Select Case (tag)
        Case "Arrow"
            'Set cursor mode to cmNone where clicks on map have no effect on the map view
            Map1.CursorMode = MapWinGIS.tkCursorMode.cmNone
        Case "ZoomIn"
            'Set cursor mode to cmZoomIn where left clicks on map zoom in map view
            Map1.CursorMode = MapWinGIS.tkCursorMode.cmZoomIn
        Case "ZoomOut"
            'Set cursor mode to cmZoomOut where left clicks on map zoom out map view
            Map1.CursorMode = MapWinGIS.tkCursorMode.cmZoomOut
        Case "Pan"
            'Set cursor mode to cmPan where holding down the left mouse button and dragging
            ' pans contents of the map changing what is shown in the map view
            Map1.CursorMode = MapWinGIS.tkCursorMode.cmPan
        Case "Select"
```

```
        'Set cursor mode to cmSelection where clicking and dragging selects items in the map view
        Map1.CursorMode = MapWinGIS.tkCursorMode.cmSelection
    End Select
End Sub
```

3.9.3.3 DoubleBuffer

Gets or sets whether or not the map uses double buffering. Double buffering uses a little more memory, but allows the view to be smoother with less flickering.

VB.NET Usage

Property DoubleBuffer() As Boolean

Parameters

ReturnValue	A boolean value representing whether the map is using double buffering or not.

Sample Code

```
Private Sub ToggleDoubleBuffering()
    'See if the map is using double buffering
    If Map1.DoubleBuffer() Then
        'Display message telling user that double buffering is being turned off
        MsgBox("Double buffering is on, turning it off.")
        'Set the map not to use double buffering
        Map1.DoubleBuffer = False
    Else
        'Display message telling user that double buffering is being turned on
        MsgBox("Double buffering is off, turning it on.")
        'Set the map to use double buffering
        Map1.DoubleBuffer = True
    End If
End Sub
```

3.9.3.4 DrawingKey

The drawing key may be used by the programmer to store any data desired in string format for any drawing layer specified by the drawing handle.

VB.NET Usage

Property set_DrawingKey(DrawHandle As Integer, param0 As String)

Parameters

DrawHandle	The drawing handle for the drawing layer for which the drawing key is to be set.
param0	The drawing key string for the specified drawing layer.

Property get_DrawingKey(DrawHandle As Integer) As String

Parameters

DrawHandle	The drawing handle for the drawing layer for which the drawing key is required.
ReturnValue	The drawing key string for the specified drawing layer.

Sample Code

```
Private Sub DrawingKey()
    Dim draw_hndl As Integer
    Dim draw_key As String
    'Create new drawing layer on map
    draw_hndl = Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlScreenReferencedList)
    'Draw filled red circle at x = 100, y = 100, with radius 50
    Map1.DrawCircle(100, 100, 50, System.Convert.ToUInt32(RGB(255, 0, 0)), True)
    'Set the drawing key for the drawing layer
    Map1.set_DrawingKey(draw_hndl, "Red Circle")
    'Get the drawing key for the drawing layer
    draw_key = Map1.get_DrawingKey(draw_hndl)
End Sub
```

3.9.3.5 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.9.3.6 ExtentHistory

Gets or sets the number of extents to cache in the extents history.

VB.NET Usage

Property ExtentHistory() As Integer

Parameters

ReturnValue	The number of extents to be stored in cache history.

Sample Code

```
Private Sub ExtentHistory()
    Dim extentHistory As Integer
    'Get extent history
    extentHistory = Map1.ExtentHistory
    If extentHistory = 5 Then
        'Set extent history to 10
        Map1.ExtentHistory = 10
    Else
        'Set extent history to 5
        Map1.ExtentHistory = 5
    End If
End Sub
```

3.9.3.7 ExtentPad

Gets or sets the percentage of the view used to pad the extents of a layer when zooming to a layer or maximum extents. Padding makes it so that there is a small border around the layer when you zoom to it.

VB.NET Usage

Property ExtentPad() As Double

Parameters

ReturnValue	The amount of padding around the extents when zooming to a layer or to maximum extents.

Sample Code

```
Private Sub ExtentPad()
    Dim padding As Double
    'Get extent padding for zooming
    padding = Map1.ExtentPad
    If Map1.ExtentPad = 5 Then
        'Set extent padding for zooming
        Map1.ExtentPad = 10
    Else
        'Set extent padding for zooming
        Map1.ExtentPad = 5
```

```
    End If
End Sub
```

3.9.3.8 Extents

Gets or sets the extents of the map using an Extents object. If the given extents do not fit the aspect ratio of the map, the map will fit the given extents as well as possible.

VB.NET Usage

Property Extents() As Object

Parameters

ReturnValue	The extents of the map as represented by an Extents object.

Sample Code

```
Private Sub ExtentsExample()
    Dim image As New MapWinGIS.Image()
    Dim extents As MapWinGIS.Extents
    Dim hndl As Integer
    'Set extents to the current extents of the map
    extents = CType(Map1.Extents, MapWinGIS.Extents)
    'Set the image to a snapshot of what is currently displayed in the map
    image = Map1.SnapShot(extents)
    'Remove all layers in the map
    Map1.RemoveAllLayers()
    'Add the snapshot taken above to the map as a new layer
    hndl = Map1.AddLayer(image, True)
End Sub
```

3.9.3.9 GetLayerStandardViewWidth

Gets the standard view width used to scale the labels on the layer.

VB.NET Usage

Property GetLayerStandardViewWidth(layerHandle As Integer, ByRef width As Double)

Parameters

layerHandle	The handle of the layer for which the standard view width is required.
width	Reference parameter. The standard view width for the specified layer is returned through this parameter.

Sample Code

```
Public Sub GetStandardViewWidth()
    Dim viewWidth As Double
    'Get the standard view width used to scale the labels for layer 0
    Map1.GetLayerStandardViewWidth(0, viewWidth)
End Sub
```

3.9.3.10 GetObject

Returns the layer object with the given handle. The object could be a Shapefile, Grid, or Image object.

VB.NET Usage

Property get_GetObject(LayerHandle As Integer) As Object

Parameters

LayerHandle	The handle of the layer to be retrieved.

ReturnValue	A Shapefile, Grid, or Image object.

Sample Code

```
Private Sub GetObject()
    Dim hndl As Integer
    Dim obj As Object
    Dim sh As MapWinGIS.Shapefile
    Dim image As MapWinGIS.Image
    'Get handle for layer at position 0 in map
    hndl = Map1.get_LayerHandle(0)
    'Get the object in the specified layer
    obj = Map1.get_GetObject(hndl)
    'Display the type of object in a message box
    MsgBox(System.Convert.ToString(obj.GetType()))
    'Check if the object in the layer is a Shapefile
    If System.Convert.ToString(obj.GetType()) = "MapWinGIS.ShapefileClass" Then
        'If the object in the layer is a Shapefile, store it in our shapefile variable
        sh = Map1.get_GetObject(hndl)
    Else
        'If the object in the layer is not a Shapefile, store it in our image variable
        image = Map1.get_GetObject(hndl)
    End If
End Sub
```

3.9.3.11 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
        '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
             'Display the error message in a label
            LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
            'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
            'Display the message in a label
        Label2.Text = Message
    End Sub
 #End Region
            '...
```

3.9.3.12 GridFileName

Gets or sets the filename of the Grid object associated with an Image object loaded into the map.

VB.NET Usage

Property set_GridFilename(LayerHandle As Integer, param0 As String)

Parameters

LayerHandle	The handle of the layer for which the grid filename is to be set.
param0	The grid filename to be associated with the specified layer.

Property get_GridFilename(LayerHandle As Integer) As String

Parameters

LayerHandle	The handle of the layer for which the grid filename is required.
ReturnValue	The grid filename associated with the specified layer.

Sample Code

```
Private Sub GridFilename()
    Dim hndl As Integer
    Dim gfname As String
    'Get handle for layer at position 0 in map
    hndl = Map1.get_LayerHandle(0)
    'Get the gridfilename
    gfname = Map1.get_GridFileName(hndl)
    'Set the gridfilename
    Map1.set_GridFileName(hndl, "newGridFileName")
End Sub
```

3.9.3.13 ImageLayerPercentTransparent

******NOT IMPLEMENTED AT THIS TIME******

Gets or sets the percentage of transparency of an Image layer.

VB.NET Usage

Property set_ImageLayerPercentTransparent(LayerHandle As Integer, param0 As Single)

Parameters

LayerHandle	The layer handle of the Image layer for which the transparency is to be set.
param0	The percentage of transparency for the specified image layer.

Property get_ImageLayerPercentTransparent(LayerHandle As Integer) As Single

Parameters

LayerHandle	The layer handle of the Image layer for which the transparency is required.
ReturnValue	The percentage of transparency for the specified image layer.

Sample Code

```
Private Sub ImageLayerTransparent()
    'This assumes you have added an image layer in the map at position 0
    Dim image As New MapWinGIS.Image()
    Dim hndl As Integer
    Dim trans As Single
    'Get handle for layer at position 0 in map
    hndl = Map1.get_LayerHandle(0)
    'Get the image in the layer at position 0
    image = Map1.get_GetObject(hndl)
    'Get the image layer transparency for the specified layer
    trans = Map1.get_ImageLayerPercentTransparent(hndl)
    'See if the current transparency is 100 percent
    If trans = 100 Then
        'If transparency is currently 100 percent, set it to 50 percent
        Map1.set_ImageLayerPercentTransparent(hndl, 50)
    Else
        'If transparency is not currently 100 percent, set it to 100 percent
        Map1.set_ImageLayerPercentTransparent(hndl, 100)
    End If
End Sub
```

3.9.3.14 IsLocked

Checks to see if the map is currently locked or not.

VB.NET Usage

Property IsLocked() As MapWinGIS.tkLockMode

Parameters

ReturnValue	The current lock mode of the map. (Lock modes: lmLock, lmUnlock.)

Sample Code

```
Private Sub MapLockCheck()
    'Check to see if the map is currently locked
    If Map1.IsLocked = MapWinGIS.tkLockMode.lmLock Then
        'If the map lock mode is lmLock, display message box telling user map is locked
        MsgBox("The map is locked.")
    Else
        'If the map lock mode is lmUnlock, display message box telling user map is unlocked
        MsgBox("The map is unlocked.")
    End If
End Sub
```

3.9.3.15 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
    If k = "Map1" Then
        'Set the map's key to "My Map1"
        Map1.Key = "My Map1"
    Else
        'Set the map's key to "Map1"
        Map1.Key = "Map1"
    End If
End Sub
```

3.9.3.16 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.9.3.17 LayerHandle

Gets the handle of the layer at the given position in the map. Returns -1 if there is no layer at the specified position.

VB.NET Usage

ReadOnly Property get_LayerHandle(LayerPosition As Integer) As Integer

Parameters

LayerPosition	The position of the layer for which the layer handle is required.
ReturnValue	The layer handle for the layer at the specified position.

Sample Code

```
Private Sub LayerHandle()
    Dim hndl As Integer
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Use layer handle to set specified layer as visible in map
    Map1.set_LayerVisible(hndl, True)
End Sub
```

3.9.3.18 LayerKey

Gets or sets a string associated with a layer in the map. This string can be used to store any string data desired by the developer.

VB.NET Usage

Property set_LayerKey(LayerHandle As Integer, param0 As String)

Parameters

LayerHandle	The handle for the layer for which the layer key is to be set.
param0	The layer key for the specified layer in string format.

Property get_LayerKey(LayerHandle As Integer) As String

Parameters

LayerHandle	The handle for the layer for which the layer key is required.
ReturnValue	The layer key for the specified layer in string format.

Sample Code

```
Private Sub LayerKey()
    Dim hndl As Integer
    Dim lkey As String
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Get the layer key for the specified layer
    lkey = Map1.get_LayerKey(hndl)
    'Set the layer key for the specified layer
    Map1.set_LayerKey(hndl, "New Layer Key")
End Sub
```

3.9.3.19 LayerLabelsOffset

Gets and sets the offset for the labels on the layer. The offset is the distance in pixels from the label point to the text.

VB.NET Usage

Property set_LayerLabelsOffset(LayerHandle As Integer, param0 As Integer)

Parameters

LayerHandle	The handle of the layer for which the offset is to be set.
param0	The offset for the labels on the specified layer.

Property get_LayerLabelsOffset(LayerHandle As Integer) As Integer

Parameters

LayerHandle	The handle of the layer for which the offset is to be set.
ReturnValue	The offset for the labels on the specified layer.

Sample Code

```
Public Sub LayerLabelsOffset()
    Dim hndl As Integer
    Dim offset As Integer
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Get the label offset for the layer
    offset = Map1.get_LayerLabelsOffset(hndl)
```

```
    'Set the label offset for the layer
    Map1.set_LayerLabelsOffset(hndl, offset)
End Sub
```

3.9.3.20 LayerLabelsScale

Gets or sets whether to scale the labels on the layer.

VB.NET Usage

Property set_LayerLabelsScale(LayerHandle As Integer, param0 As Boolean)

Parameters

LayerHandle	The handle of the layer for which the scaling of labels is to be set.
param0	A boolean value representing whether or not to scale the labels on the specified layer.

Property get_LayerLabelsScale(LayerHandle As Integer) As Boolean

Parameters

LayerHandle	The handle of the layer for which the scaling of labels is to be set.
ReturnValue	A boolean value representing whether or not to scale the labels on the specified layer.

Sample Code

```
Public Sub LayerLabelsScale()
    Dim hndl As Integer
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Check if the labels are scaled for the specified layer
    If Map1.get_LayerLabelsScale(hndl) Then
        'Set the labels on the layer not to be scaled
        Map1.set_LayerLabelsScale(hndl, False)
    Else
        'Set the labels on the layer to be scaled
        Map1.set_LayerLabelsScale(hndl, True)
    End If
End Sub
```

3.9.3.21 LayerLabelsShadow

Gets or sets whether to use shadows for the labels on the layer.

VB.NET Usage

Property set_LayerLabelsShadow(LayerHandle As Integer, param0 As Boolean)

Parameters

LayerHandle	The handle of the layer for which the use of label shadows is to be set.
param0	A boolean value representing whether or not label shadows will be used for the specified layer.

Property get_LayerLabelsShadow(LayerHandle As Integer) As Boolean **Parameters**

LayerHandle	The handle of the layer for which the use of label shadows is to be set.
ReturnValue	A boolean value representing whether or not label shadows are being used for the specified layer.

Sample Code

```
Public Sub LayerLabelsShadow()
    Dim hndl As Integer
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Check whether the labels have shadows for the layer
    If Map1.get_LayerLabelsShadow(hndl) Then
        'Set the layer's labels not to use shadows
        Map1.set_LayerLabelsShadow(hndl, False)
    Else
        'Set the layer's labels to use shadows
        Map1.set_LayerLabelsShadow(hndl, True)
    End If
End Sub
```

3.9.3.22 LayerLabelsShadowColor

Gets and sets the shadow color for the labels on the layer.

VB.NET Usage

Property set_LayerLabelsShadowColor(LayerHandle As Integer, param0 As System.UInt32)

Parameters

LayerHandle	The layer handle of the layer for which the shadow color is to be set.
param0	The color to use as the shadow color of the specified layer. This is a System.UInt32 representation of an RGB color.

Property get_LayerLabelsShadowColor(LayerHandle As Integer) As System.Drawing.Color **Parameters**

LayerHandle	The layer handle of the layer for which the shadow color is to be set.
ReturnValue	The color to use as the shadow color of the specified layer. This is a System.UInt32 representation of an RGB color.

Sample Code

```
Public Sub LayerLabelsShadowColor()
    Dim hndl As Integer
    Dim color As System.Drawing.Color
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Get the label shadow color for the layer
    color = Map1.get_LayerLabelsShadowColor(hndl)
    'Set the label shadow color for the layer
    Map1.set_LayerLabelsShadowColor(hndl, System.Convert.ToUInt32(RGB(color.R, color.G, color.B)))
End Sub
```

3.9.3.23 LayerLabelsVisible

Gets or sets label visibility for the specified layer.

VB.NET Usage

Property set_LayerLabelsVisible(LayerHandle As Integer, param0 As Boolean)

Parameters

LayerHandle	The layer handle for the layer for which its label's visibility are to be set.
param0	A boolean value which determines whether the specified layer's labels are visible or not.

Property get_LayerLabelsVisible(LayerHandle As Integer) As Boolean
Parameters

LayerHandle	The layer handle for the layer for which its label's visibility is being checked.
ReturnValue	A boolean value which determines whether the specified layer's labels are visible or not.

Sample Code

```
Private Sub LayerLabelsVisible()
    Dim hndl As Integer
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Check if the specified layer's labels are visible
    If Map1.get_LayerLabelsVisible(hndl) Then
        'Set the specified layer's labels to be visible
        Map1.set_LayerLabelsVisible(hndl, True)
    Else
        'Set the specified layer's labels to be hidden
        Map1.set_LayerLabelsVisible(hndl, False)
    End If
End Sub
```

3.9.3.24 LayerName

Gets or sets the name of the specified layer.

VB.NET Usage

Property set_LayerName(LayerHandle As Integer, param0 As String)

Parameters

LayerHandle	The handle of the layer for which the name is to be set.
param0	The layer name for the specified layer.

Property get_LayerName(LayerHandle As Integer) As String

Parameters

LayerHandle	The handle of the layer for which the name is required.
ReturnValue	The layer name for the specified layer.

Sample Code

```
Private Sub LayerName()
    Dim hndl As Integer
    Dim lname As String
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Get the layer name for the specified layer
    lname = Map1.get_LayerName(hndl)
    'Set the layer name for the specified layer
    Map1.set_LayerName(hndl, "New Layer Name")
End Sub
```

3.9.3.25 LayerPosition

Gets the position of the specified layer in the map.

VB.NET Usage

ReadOnly Property get_LayerPosition(LayerHandle As Integer) As Integer

Parameters

LayerHandle	The layer handle of the layer for which the layer position is required.
ReturnValue	The layer position of the specified layer in the map.

Sample Code

```
Private Sub LayerPosition()
    Dim hndl As Integer
    Dim lpos As Integer
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Get the layer position for the specified layer
    lpos = Map1.get_LayerPosition(hndl)
End Sub
```

3.9.3.26 LayerVisible

Gets or sets the visibility of the specified layer.

VB.NET Usage

Property set_LayerVisible(LayerHandle As Integer, param0 As Boolean)

Parameters

LayerHandle	The handle of the layer for which the visibility is being set.
param0	A boolean value representing whether the layer is to be visible or not.

Property get_LayerVisible(LayerHandle As Integer) As Boolean

Parameters

LayerHandle	The handle of the layer for which the visibility is being checked.
ReturnValue	A boolean value representing whether the layer is to be visible or not.

Sample Code

```
Private Sub LayerVisible()
```

```
    Dim hndl As Integer
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Check if the specified layer is visible or not
    If Map1.get_LayerVisible(hndl) Then
        'Set specified layer to be hidden in the map
        Map1.set_LayerVisible(hndl, False)
    Else
        'Set specified layer to be visible in the map
        Map1.set_LayerVisible(hndl, True)
    End If
End Sub
```

3.9.3.27 MapCursor

Gets or sets the cursor used with the map. When using crsrUserDefined be sure to set a UDCursorHandle.

VB.NET Usage

Property MapCursor() As MapWinGIS.tkCursor

Parameters

ReturnValue	The current cursor used by the map.

Sample Code

```
Private Sub MapCursor()
    Dim mc As MapWinGIS.tkCursor
    'Get the current map cursor
    mc = Map1.MapCursor()
    'Set the current map cursor
    Map1.MapCursor = MapWinGIS.tkCursor.crsrCross
End Sub
```

3.9.3.28 MapState

Gets or sets the MapState string which stores all information needed to restore a view. This includes layer information and coloring schemes.

VB.NET Usage

Property MapState() As String

Parameters

ReturnValue	The MapState string which allows a view to be restored.

Sample Code

```
Private Sub MapState()
    Dim mapState As String
    'Get the current MapState and save it
    mapState = Map1.MapState
    'Remove all layers from the map
    Map1.RemoveAllLayers()
    'Redraw the view of the map to show the layers have been removed
    Map1.Redraw()
    'Display message box to allow user to see the change to the map view
    MsgBox("Map cleared")
    'Set the MapState to the saved MapState before all layers were removed
    Map1.MapState = mapState
    'Redraw the view of the map to show the map has been restored
    Map1.Redraw()
    'Display message box to indicate the map has been restored to user
    MsgBox("Map restored")
End Sub
```

3.9.3.29 NumLayers

Gets the number of layers loaded in the map.

VB.NET Usage

ReadOnly Property NumLayers() As Integer

Parameters

ReturnValue	The number of layers currently in the map.

Sample Code

```
Private Sub NumLayers()
    Dim num As Integer
    'Get the number of layers currently in the map
    num = Map1.NumLayers
End Sub
```

3.9.3.30 SendMouseDown

Gets or sets whether the map sends mouse down events.

VB.NET Usage

Property SendMouseDown() As Boolean

Parameters

ReturnValue	A boolean value representing whether the map sends mouse down events.

Sample Code

```
Private Sub SendMouseDown()
    'Check if the map is sending the mouse down event
    If Map1.SendMouseDown Then
        'Set the map not to send the mouse down event
        Map1.SendMouseDown = False
    Else
        'Set the map to send the mouse down event
        Map1.SendMouseDown = True
    End If
End Sub
```

3.9.3.31 SendMouseMove

Gets or sets whether the map sends mouse move events.

VB.NET Usage

Property SendMouseMove() As Boolean
]

Parameters

ReturnValue	A boolean value representing whether the map sends mouse move events.

Sample Code

```
Private Sub SendMouseMove()
    'Check if the map is sending the mouse move event
    If Map1.SendMouseMove Then
        'Set the map not to send the mouse move event
        Map1.SendMouseMove = False
    Else
        'Set the map to send the mouse move event
        Map1.SendMouseMove = True
    End If
End Sub
```

3.9.3.32 SendMouseUp

Gets or sets whether the map sends mouse up events.

VB.NET Usage

Property SendMouseUp() As Boolean

Parameters

ReturnValue	A boolean value representing whether the map sends mouse up events.

Sample Code

```
Private Sub SendMouseUp()
    'Check if the map is sending the mouse up event
    If Map1.SendMouseUp Then
        'Set the map not to send the mouse up event
        Map1.SendMouseUp = False
    Else
        'Set the map to send the mouse up event
        Map1.SendMouseUp = True
    End If
End Sub
```

3.9.3.33 SendSelectBoxDrag

Gets or sets whether the map sends the SelectBoxDrag event.

VB.NET Usage

Property SendSelectBoxDrag() As Boolean

Parameters

ReturnValue	A boolean value representing whether the map sends the select box drag event or not.

Sample Code

```
Private Sub SendSelectBoxDrag()
    'Check if the map is sending the select box drag event
    If Map1.SendSelectBoxDrag Then
        'Set the map not to send the select box drag event
        Map1.SendSelectBoxDrag = False
    Else
        'Set the map to send the select box drag event
        Map1.SendSelectBoxDrag = True
    End If
End Sub
```

3.9.3.34 SendSelectBoxFinal

Gets or sets whether the map sends the SelectBoxFinal event.

VB.NET Usage

Property SendSelecBoxFinal() As Boolean

Parameters

ReturnValue	A boolean value representing whether the map sends the SelectBoxFinal event or not.

Sample Code

```
Private Sub SendSelectBoxFinal()
    'Check if the map is sending the select box final event
    If Map1.SendSelectBoxFinal Then
        'Set the map not to send the select box final event
        Map1.SendSelectBoxFinal = False
    Else
        'Set the map to send the select box final event
        Map1.SendSelectBoxFinal = True
    End If
End Sub
```

3.9.3.35 SerialNumber

Serial number functionality is deprecated as MapWindow is now Open Source.
This property may be left unset.

3.9.3.36 SetLayerStandardViewWidth

Sets the standard view width used to scale the labels on the layer.

VB.NET Usage

Property SetLayerStandardViewWidth(layerHandle As Integer, width As Double)

Parameters

layerHandle	The handle of the layer for which the standard view width is to be set.
width	The new standard view width for the specified layer.

Sample Code

```
Public Sub SetStandardViewWidth()
    Dim extents As MapWinGIS.Extents
    Dim width As Double
    'Get the current extents of the map
    extents = Map1.Extents
    'Calculate the width of the current map extents
    width = extents.xMax - extents.xMin
    'Set the standard view width for layer 0 to be the width of the current extents
    Map1.SetLayerStandardViewWidth(0, width)
End Sub
```

3.9.3.37 ShapeDrawFill

Gets or sets whether the specified shape is drawn with a fill. Only works on polygon shapefiles.

VB.NET Usage

Property set_ShapeDrawFill(LayerHandle As Integer, Shape As Integer, param0 as Boolean)

Parameters

LayerHandle	Handle of the layer containing the shape for which the fill is to be set.
Shape	Handle of the shape for which the fill is to be set.
param0	Sets whether the shape is drawn with a fill or not.

Property get_ShapeDrawFill(LayerHandle As Integer, Shape As Integer) As Boolean

Parameters

LayerHandle	Handle of the layer containing the shape which is tested to see if it is being drawn with a fill or not.
Shape	Handle of the shape to test if it is being drawn with a fill or not.
Return Value	Gets whether the shape is being drawn with a fill or not.

Sample Code

```
Private Sub DrawFill()
    Dim hnd As Integer
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Check if shape 0 in the layer is being drawn with a fill
    If (Map1.get_ShapeDrawFill(hnd,0)) Then
        'Set shape 0 not to be drawn with a fill
        Map1.set_ShapeDrawFill(hnd, 0, False)
    Else
        'Set shape 0 to be drawn with fill
        Map1.set_ShapeDrawFill(hnd, 0, True)
    End If
End Sub
```

3.9.3.38 ShapeDrawLine

Gets or sets whether the lines for the specified shape are drawn.

VB.NET Usage

Property set_ShapeDrawLine(LayerHandle As Integer, Shape As Integer, param0 as Boolean)

Parameters

LayerHandle	Handle of the layer containing the shape for which the lines are to be set.

Shape	Handle of the shape for which the lines are to be set.
param0	Sets whether the shape is drawn with lines or not.

Property get_ShapeDrawLine(LayerHandle As Integer) As Boolean

Parameters

LayerHandle	Handle of the layer to test if the specified shape is being drawn with a lines or not.
Shape	Handle of the shape to test if it is being drawn with lines or not.
Return Value	Gets whether the layer is being drawn with lines or not.

Sample Code

```
Private Sub DrawLine()
    Dim hnd As Integer
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Check if shape 0 in the layer is being drawn with lines
    If (Map1.get_ShapeDrawLine(hnd, 0)) Then
        'Set shape 0 not to be drawn with lines
        Map1.set_ShapeDrawLine(hnd, 0, False)
    Else
        'Set shape 0 to be drawn with lines
        Map1.set_ShapeDrawLine(hnd, 0, True)
    End If
End Sub
```

3.9.3.39 ShapeDrawPoint

Gets or sets whether the points/vertices in specified shape are drawn.

VB.NET Usage

Property set_ShapeDrawPoint(LayerHandle As Integer, Shape As Integer, param0 as Boolean)

Parameters

LayerHandle	Handle of the layer containing the shape for which the points are to be set.
Shape	Handle of the shape for which the points are to be set.
param0	Sets whether the shape is drawn with points or not.

Property get_ShapeDrawPoint(LayerHandle As Integer, Shape As Integer) As Boolean

Parameters

LayerHandle	Handle of the layer to test if it is being drawn with a points or not.
Shape	Handle of the shape to test if it is being drawn with points or not.
Return Value	Gets whether the shape is being drawn with points or not.

Sample Code

```
Private Sub DrawPoint()
    Dim hnd As Integer
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Check if shape 0 in the layer is being drawn with points
    If (Map1.get_ShapeDrawPoint(hnd, 0)) Then
        'Set shape 0 not to be drawn with points
        Map1.set_ShapeDrawPoint(hnd, 0, False)
    Else
        'Set shape 0 to be drawn with points
        Map1.set_ShapeDrawPoint(hnd, 0, True)
    End If
End Sub
```

3.9.3.40 ShapeFillColor

Gets or sets the fill color for the specified shape. Only works on polygon shapefiles.

VB.NET Usage

Property set_ShapeFillColor(LayerHandle As Integer, Shape As Integer, param0 as System.UInt32)

Parameters

LayerHandle	Handle of the layer containing the shape for which the fill color is to be set.
Shape	Handle for the shape for which the fill color is to be set.
param0	Fill color for the shape in the polygon shapefile. This is a System.UInt32 representation of an RGB color.

Property get_ShapeFillColor(LayerHandle As Int32, Shape As Integer) As System.Drawing.Color **Parameters**

LayerHandle	Handle of the layer containing the shape for which the fill color is required.
Shape	Handle for the shape for which the fill color is required.
Return Value	Fill color for the shape in the polygon shapefile.

Sample Code

```
Private Function FillColor()
    Dim hnd As Integer
    Dim col As System.UInt32
    Dim col2 As System.Drawing.Color
    Dim R As Integer = 122
    Dim G As Integer = 255
    Dim B As Integer = 109
    'get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'convert an RGB color to a UInt32
    col = System.Convert.ToUInt32(RGB(R, G, B))
    'set the fill color for the layer
    Map1.set_ShapeLayerFillColor(hnd, col)
    'get the fill color for a layer 0
    col2 = Map1.get_ShapeLayerFillColor(hnd)
End Function
```

3.9.3.41 ShapeFillStipple

Gets or sets the fill stipple for the specified shape. Only works on polygon shapefiles.
See also [MapWinGIS:tkFillStipple tkFillStipple]

VB.NET Usage

Property set_ShapeFillStipple(LayerHandle As Integer, Shape As Integer, param0 as MapWinGIS.tkFillStipple)

Parameters

LayerHandle	Handle of the layer containing the shape for which the fill stipple is to be set.
Shape	Handle of the shape for which the fill stipple is to be set.
param0	Sets fill stipple for the specified shape.

Property get_ShapeFillStipple(LayerHandle As Integer, Shape As Integer) As MapWinGIS.tkFillStipple
Parameters

LayerHandle	Handle of the layer contianing the shape for which the fill stipple is required.
Shape	Handle of the shape for which the fill stipple is required.
Return Value	Gets the fill stipple for the specified shape.

Sample Code

```
Private Sub FillStipple()
    Dim hnd As Integer
    Dim stp As MapWinGIS.tkFillStipple
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the fill stipple of shape 0 in the layer
    stp = Map1.get_ShapeFillStipple(hnd, 0)
    'Set the fill stipple of shape 0 in the layer to polka dot
    Map1.set_ShapeFillStipple(hnd, 0, MapWinGIS.tkFillStipple.fsPolkaDot)
End Sub
```

3.9.3.42 ShapeFillTransparency

Gets or sets the percentage of fill trancparency for the specified layer. Only works on polygon shapefiles.

VB.NET Usage

Property set_ShapeFillTransparency(LayerHandle As Integer, Shape As Integer, param0 as Single)

Parameters

LayerHandle	Handle of the layer containing the shape for which the fill transparency is to be set.
Shape	Handle of the shape for which the fill transparency is to be set.
param0	Sets the percentage of fill transparency for the specified shape.

Property get_ShapeFillTransparency(LayerHandle As Integer, Shape As Integer) As Single

Parameters

LayerHandle	Handle of the layer containing the shape for which the fill transparency is required..
Shape	Handle of the shape for which the fill transparency is required.
Return Value	Gets the percentage of fill transparency for the specified shape.

Sample Code

```
Private Sub FillTransparency()
    Dim hnd As Integer
    Dim percent As Short
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the fill transparency percentage of shape 0 in the layer
    percent = Map1.get_ShapeFillTransparency(hnd, 0)
    percent = 50
    'Set the fill transparency percentage of shape 0 in the layer
    Map1.set_ShapeFillTransparency(hnd, 0, percent)
End Sub
```

3.9.3.43 ShapeLayerDrawFill

Gets or sets whether the specified layer is drawn with a fill. Only works on polygon shapefiles.

VB.NET Usage

Property set_ShapeLayerDrawFill(LayerHandle As Integer, param0 as Boolean)

Parameters

LayerHandle	Handle of the layer for which the fill is to be set.
param0	Sets whether the layer is drawn with a fill or not.

Property get_ShapeLayerDrawFill(LayerHandle As Integer) As Boolean

Parameters

LayerHandle	Handle of the layer to test if it is being drawn with a fill or not.
Return Value	Gets whether the layer is being drawn with a fill or not.

Sample Code

```
Private Sub DrawFill()
    Dim hnd As Integer
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Check if the layer is being drawn with a fill
    If (Map1.get_ShapeLayerDrawFill(hnd)) Then
        'Set the layer not to be drawn with a fill
        Map1.set_ShapeLayerDrawFill(hnd, False)
    Else
        'Set the layer to be drawn with fill
        Map1.set_ShapeLayerDrawFill(hnd, True)
    End If
End Sub
```

3.9.3.44 ShapeLayerDrawLine

Gets or sets whether the lines for the shapefile in specified layer are drawn.

VB.NET Usage

Property set_ShapeLayerDrawLine(LayerHandle As Integer, param0 as Boolean)

Parameters

LayerHandle	Handle of the layer for which the lines are to be set.
param0	Sets whether the layer is drawn with lines or not.

Property get_ShapeLayerDrawLine(LayerHandle As Integer) As Boolean

Parameters

LayerHandle	Handle of the layer to test if it is being drawn with a lines or not.
Return Value	Gets whether the layer is being drawn with lines or not.

Sample Code

```
Private Sub DrawLine()
    Dim hnd As Integer
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Check if the layer is being drawn with lines
    If (Map1.get_ShapeLayerDrawLine(hnd)) Then
        'Set the layer not to be drawn with lines
        Map1.set_ShapeLayerDrawLine(hnd, False)
    Else
        'Set the layer to be drawn with lines
        Map1.set_ShapeLayerDrawLine(hnd, True)
    End If
End Sub
```

3.9.3.45 ShapeLayerDrawPoint

Gets or sets whether the points/vertices for the shapefile in specified layer are drawn.

VB.NET Usage

Property set_ShapeLayerDrawPoint(LayerHandle As Integer, param0 as Boolean)

Parameters

LayerHandle	Handle of the layer for which the points are to be set.
param0	Sets whether the layer is drawn with points or not.

Property get_ShapeLayerDrawPoint(LayerHandle As Integer) As Boolean

Parameters

LayerHandle	Handle of the layer to test if it is being drawn with a points or not.
Return Value	Gets whether the layer is being drawn with points or not.

Sample Code

```
Private Sub DrawPoint()
    Dim hnd As Integer
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Check if the layer is being drawn with points
    If (Map1.get_ShapeLayerDrawPoint(hnd)) Then
        'Set the layer not to be drawn with points
        Map1.set_ShapeLayerDrawPoint(hnd, False)
    Else
        'Set the layer to be drawn with points
        Map1.set_ShapeLayerDrawPoint(hnd, True)
    End If
End Sub
```

3.9.3.46 ShapeLayerFillColor

Gets or sets the fill color for the specified layer. Only works on polygon shapefiles.

VB.NET Usage

Property set_ShapeLayerFillColor(LayerHandle As Integer, param0 as System.UInt32)

Parameters

LayerHandle	Handle of the layer for which the fill color is to be set.
param0	Fill color for the polygon shapefile. This is a System.UInt32 representation of an RGB color.

Property get_ShapeLayerFillColor(LayerHandle As Integer) As System.Drawing.Color Parameters

LayerHandle	Handle of the layer for which the fill color is required.
Return Value	Fill color for the polygon shapefile.

Sample Code

```
Private Function FillColor()
    Dim hnd As Integer
    Dim col As System.UInt32
    Dim col2 As System.Drawing.Color
    Dim R As Integer = 122
    Dim G As Integer = 255
    Dim B As Integer = 109
    'get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'convert an RGB color to a UInt32
    col = System.Convert.ToUInt32(RGB(R, G, B))
    'set the fill color for the shapes in layer 0
    Map1.set_ShapeLayerFillColor(hnd, col)
    'get the fill color for the shapes in layer 0
    col2 = Map1.get_ShapeLayerFillColor(hnd)
End Function
```

3.9.3.47 ShapeLayerFillStipple

Gets or sets the fill stipple for the specified layer. Only works on polygon shapefiles.
See also tkFillStipple

VB.NET Usage

Property set_ShapeLayerFillStipple(LayerHandle As Integer, param0 as MapWinGIS.tkFillStipple)

Parameters

LayerHandle	Handle of the layer for which the fill stipple is to be set.
param0	Sets fill stipple for the specified layer.

Property get_ShapeLayerFillStipple(LayerHandle As Integer) As MapWinGIS.tkFillStipple Parameters

LayerHandle	Handle of the layer for which the fill stipple is required.
Return Value	Gets the fill stipple for the specified layer.

Sample Code

```
Private Sub FillStipple()
    Dim hnd As Integer
    Dim stp As MapWinGIS.tkFillStipple
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the fill stipple of layer 0 in Map1
    stp = Map1.get_ShapeLayerFillStipple(hnd)
    'Set the fill stipple of layer 0 in Map1 to polka dot
    Map1.set_ShapeLayerFillStipple(hnd, MapWinGIS.tkFillStipple.fsPolkaDot)
End Sub
```

3.9.3.48 ShapeLayerFillTransparency

Gets or sets the percentage of fill transparency for the specified layer. Only works on polygon shapefiles.

VB.NET Usage

Property set_ShapeLayerFillTransparency(LayerHandle As Integer, param0 as Single)

Parameters

LayerHandle	Handle of the layer for which the fill transparency is to be set.
param0	Sets the percentage of fill transparency for the specified layer.

Property get_ShapeLayerFillTransparency(LayerHandle As Integer) As Single

Parameters

LayerHandle	Handle of the layer to get percentage of fill transparency.
Return Value	Gets the percentage of fill transparency for the specified layer.

Sample Code

```
Private Sub FillTransparency()
    Dim hnd As Integer
    Dim percent As Single
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the fill transparency percentage of layer 0 in Map1
    percent = Map1.get_ShapeLayerFillTransparency(hnd)
    percent = 50
    'Set the fill transparency percentage of layer 0 in Map1
    Map1.set_ShapeLayerFillTransparency(hnd, percent)
End Sub
```

3.9.3.49 ShapeLayerLineColor

Gets or sets the line color for the specified layer. Only works on shapefiles.

VB.NET Usage

Property set_ShapeLayerLineColor(LayerHandle As Integer, param0 as System.UInt32)

Parameters

LayerHandle	Handle of the layer for which the line color is to be set.
param0	Line color for the polygon shapefile. This is a System.UInt32 representation of an RGB color.

Property get_ShapeLayerLineColor(LayerHandle As Integer) As System.Drawing.Color

Parameters

LayerHandle	Handle of the layer for which the line color is required.
Return Value	Line color for the polygon shapefile.

Sample Code

```
Private Sub LineColor()
    Dim hnd As Integer
    Dim col As System.UInt32
    Dim col2 As System.Drawing.Color
    Dim R As Integer = 122
    Dim G As Integer = 255
    Dim B As Integer = 109
    'get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'convert an RGB color to a UInt32
    col = System.Convert.ToUInt32(RGB(R, G, B))
    'set the line color for the layer
    Map1.set_ShapeLayerLineColor(hnd, col)
    'get the line color for a layer 0
    col2 = Map1.get_ShapeLayerLineColor(hnd)
End Sub
```

3.9.3.50 ShapeLayerLineStipple

Gets or sets the line stipple for the specified layer. Only works on shapefiles.
See also tkLineStipple

VB.NET Usage

Property set_ShapeLayerLineStipple(LayerHandle As Integer, param0 as MapWinGIS.tkLineStipple)

Parameters

LayerHandle	Handle of the layer for which the line stipple is to be set.
param0	Line stipple for the shapefile.

Property get_ShapeLayerLineStipple(LayerHandle As Integer) As MapWinGIS.tkLineStipple
Parameters

LayerHandle	Handle of the layer for which the line stipple is required.
Return Value	Line stipple for the shapefile.

Sample Code

```
Private Sub LineStipple()
    Dim hnd As Integer
    Dim t As MapWinGIS.tkPointType
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the point type for layer 0 in Map1
    t = Map1.get_ShapeLayerLineStipple(hnd)
    'Set the point type for layer 0 in Map1 to circle
    Map1.set_ShapeLayerLineStipple(hnd, MapWinGIS.tkPointType.ptCircle)
End Sub
```

3.9.3.51 ShapeLayerLineWidth

Gets or sets the line width for the specified layer. Only works on shapefiles.
Suggested values for line width: 1 - 5

VB.NET Usage

Property set_ShapeLayerLineWidth(LayerHandle As Integer, param0 as Single)

Parameters

LayerHandle	Handle of the layer for which the line width is to be set.
param0	Line width for the shapefile.

Property get_ShapeLayerLineWidth(LayerHandle As Integer) As Single
Parameters

LayerHandle	Handle of the layer for which the line width is required.
Return Value	Line width for the shapefile.

Sample Code

```
Private Sub LineWidth()
    Dim hnd As Integer
    Dim width As Single
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the line width for layer 0 in Map1
    width = Map1.get_ShapeLayerLineWidth(hnd)
    'Set the line width for layer 0 in Map1
    Map1.set_ShapeLayerLineWidth(hnd, 3)
End Sub
```

3.9.3.52 ShapeLayerPointColor

Gets or sets the point color for the specified layer. Only works on shapefiles.

VB.NET Usage

Property set_ShapeLayerPointColor(LayerHandle As Integer, param0 as System.UInt32)

Parameters

LayerHandle	Handle of the layer for which the point color is to be set.
param0	Point color for the polygon shapefile. This is a System.UInt32 representation of an RGB color.

Property get_ShapeLayerPointColor(LayerHandle As Integer) As System.Drawing.Color

Parameters

LayerHandle	Handle of the layer for which the point color is required.
Return Value	Point color for the polygon shapefile.

Sample Code

```
Private Function PointColor()
    Dim hnd As Integer
    Dim col As System.UInt32
    Dim col2 As System.Drawing.Color
    Dim R As Integer = 122
    Dim G As Integer = 255
    Dim B As Integer = 109
    'get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'convert an RGB color to a UInt32
    col = System.Convert.ToUInt32(RGB(R, G, B))
    'set the point color for the layer
    Map1.set_ShapeLayerPointColor(hnd, col)
    'get the point color for a layer 0
    col2 = Map1.get_ShapeLayerPointColor(hnd)
End Function
```

3.9.3.53 ShapeLayerPointSize

Gets or sets the line point/vertex size for the specified layer. Only works on shapefiles.

VB.NET Usage

Property set_ShapeLayerPointSize(LayerHandle As Integer, param0 as Single)

Parameters

LayerHandle	Handle of the layer for which the point/vertex size is to be set.
param0	Point/vertex size for the shapefile.

Property get_ShapeLayerPointSize(LayerHandle As Integer) As Single

Parameters

LayerHandle	Handle of the layer for which the point/vertex size is required.
Return Value	Point/vertex size for the shapefile.

Sample Code

```
Private Sub PointSize()
    Dim hnd As Integer
    Dim size As Single
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the point/vertex size for layer 0 in Map1
    size = Map1.get_ShapeLayerPointSize(hnd)
    'Set the point/vertex size for layer 0 in Map1
    Map1.set_ShapeLayerPointSize(hnd, 3)
End Sub
```

3.9.3.54 ShapeLayerPointType

Gets or sets the line point type for the specified layer. Only works on shapefiles.
See also tkPointType

VB.NET Usage

Property set_ShapeLayerPointType(LayerHandle As Integer, param0 as MapWinGIS.tkPointType)

Parameters

LayerHandle	Handle of the layer for which the point type is to be set.
param0	Point type for the shapefile.

Property get_ShapeLayerPointType(LayerHandle As Integer) As MapWinGIS.tkPointType
Parameters

LayerHandle	Handle of the layer for which the point type is required.
Return Value	Point type for the shapefile.

Sample Code

```
Private Sub PointType()
    Dim hnd As Integer
    Dim t As MapWinGIS.tkPointType
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the point type for layer 0 in Map1
    t = Map1.get_ShapeLayerPointType(hnd)
    'Set the point type for layer 0 in Map1 to circle
    Map1.set_ShapeLayerPointType(hnd, MapWinGIS.tkPointType.ptCircle)
End Sub
```

3.9.3.55 ShapeLineColor

Gets or sets the line color for the specified shape. Only works on shapefiles.

VB.NET Usage

Property set_ShapeLineColor(LayerHandle As Integer, Shape As Integer, param0 as System.UInt32)

Parameters

LayerHandle	Handle of the layer containing the shape for which the line color is to be set.
Shape	Handle of the shape for which the line color is to be set.
param0	Line color for the shape in the polygon shapefile. This is a System.UInt32 representation of an RGB color.

Property get_ShapeLineColor(LayerHandle As Integer, Shape As Integer) As System.Drawing.Color
Parameters

LayerHandle	Handle of the layer containing the shape for which the line color is required.
Shape	Handle of the shape for which the line color is required.
Return Value	Line color for the shape in the polygon shapefile.

Sample Code

```
Private Sub LineColor()
    Dim hnd As Integer
    Dim col As System.UInt32
    Dim col2 As System.Drawing.Color
    Dim R As Integer = 122
    Dim G As Integer = 255
    Dim B As Integer = 109
    'get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'convert an RGB color to a UInt32
    col = System.Convert.ToUInt32(RGB(R, G, B))
    'set the line color for shape 0 in the layer
    Map1.set_ShapeLineColor(hnd, 0, col)
    'get the line color for shape 0 in the layer
    col2 = Map1.get_ShapeLineColor(hnd, 0)
End Sub
```

3.9.3.56 ShapeLineStipple

Gets or sets the line stipple for the specified shape. Only works on shapefiles.
See also tkLineStipple

VB.NET Usage

Property set_ShapeLineStipple(LayerHandle As Integer, Shape As Integer, param0 as MapWinGIS.tkLineStipple)

Parameters

LayerHandle	Handle of the layer containing the shape for which the line stipple is to be set.
Shape	Handle of the shape for which the line stipple is to be set.
param0	Line stipple for the shape in the shapefile.

Property get_ShapeLineStipple(LayerHandle As Integer, Shape As Integer) As MapWinGIS.tkLineStipple

Parameters

LayerHandle	Handle of the layer containing the shape for which the line stipple is required.
Shape	Handle of the shape for which the line stipple is required.
Return Value	Line stipple for the shape in the shapefile.

Sample Code

```
Private Sub LineStipple()
    Dim hnd As Integer
    Dim t As MapWinGIS.tkPointType
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the point type for shape 0 in the layer
    t = Map1.get_ShapeLineStipple(hnd, 0)
    'Set the point type for shape 0 in the layer to circle
    Map1.set_ShapeLineStipple(hnd, 0, MapWinGIS.tkPointType.ptCircle)
End Sub
```

3.9.3.57 ShapeLineWidth

Gets or sets the line width for the specified shape. Only works on shapefiles.
Suggested values for line width: 1 - 5

VB.NET Usage

Property set_ShapeLineWidth(LayerHandle As Integer, Shape As Integer, param0 as Single)

Parameters

LayerHandle	Handle of the layer containing the shape for which the line width is to be set.
Shape	Handle of the shape for which the line width is to be set.
param0	Line width for the shape in the shapefile.

Property get_ShapeLineWidth(LayerHandle As Integer, Shape As Integer) As Single

Parameters

LayerHandle	Handle of the layer for which the line width is required.
Shape	Handle of the shape for which the line width is required.
Return Value	Line width for the shape in the shapefile.

Sample Code

```
Private Sub LineWidth()
    Dim hnd As Integer
    Dim width As Single
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the line width for shape 0 in the layer
    width = Map1.get_ShapeLineWidth(hnd, 0)
    'Set the line width for shape 0 in the layer to 3
    Map1.set_ShapeLineWidth(hnd, 0, 3)
End Sub
```

3.9.3.58 ShapePointColor

Gets or sets the point color for the specified shape. Only works on shapefiles.

VB.NET Usage

Property set_ShapePointColor(LayerHandle As Integer, Shape As Integer, param0 as System.UInt32)

Parameters

LayerHandle	Handle of the layer containing the shape for which the point color is to be set.
Shape	Handle of the shape for which the point color is to be set.
param0	Point color for the shape in the polygon shapefile. This is a System.UInt32 representation of an RGB color.

Property get_ShapePointColor(LayerHandle As Integer, Shape As Integer) As System.Drawing.Color

Parameters

LayerHandle	Handle of the layer containing the shape for which the point color is required.
Shape	Handle of the shape for which the point color is required.
Return Value	Point color for the shape in the polygon shapefile.

Sample Code

```
Private Function PointColor()
    Dim hnd As Integer
    Dim col As System.UInt32
    Dim col2 As System.Drawing.Color
    Dim R As Integer = 122
    Dim G As Integer = 255
    Dim B As Integer = 109
    'get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'convert an RGB color to a UInt32
    col = System.Convert.ToUInt32(RGB(R, G, B))
    'set the point color for shape 0 in the layer
    Map1.set_ShapePointColor(hnd, 0, col)
    'get the point color for shape 0 in the layer
    col2 = Map1.get_ShapePointColor(hnd, 0)
End Function
```

3.9.3.59 ShapePointImageListID

Allows you specify an image from the image list so that one point shapefile can have multiple icons.

VB.NET Usage

Property set_ShapePointImageListID(LayerHandle As Integer, shape As Integer, param0 As Integer)

Parameters

LayerHandle	The layer handle of the layer containing the list to specify an image from.
shape	The specific shape to be assigned a value in the layer.
param0	The index of the image in the imagelist that you want to assign to the specified shape

Property get_ShapePointImageListID(LayerHandle As Integer, shape as Integer) As Integer

Parameters

LayerHandle	The layer handle of the layer for which the image list is defined.
shape	The specific shape in the layer that you wish to determine the image index for
ReturnValue	The index in the image list of user defined images.

Sample Code

```
' sf is a global shapefile variable
' layer is a global integer variable
 <summary>
 This function will plot different images on the map for the same shapefile by
 establishing an image list using AxMap1.set_UDPointImageListAdd and then
 setting the shapes to use the list using a point type called
 MapWinGIS.tkPointType.ptImageList and then specifying which image to use with
 AxMap1.set_ShapePointImageList
 </summary>
Private Sub cmdAddPoints_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
cmdAddPoints.Click
```

```
        Dim result As Boolean
        Dim shp, type, I As Integer
        Dim Images(3) As MapWinGIS.Image
        Dim PointShapeFile As String

        'Load a point shapefile into memory
        PointShapefile = "C:\Documents and Settings\Ted\My Documents\Fundamentals\Week
3\Lab3\DataVisualization\LIGHTS.SHP"
        sf = New MapWinGIS.Shapefile
        result = sf.Open(PointShapeFile)
        If result = False Then
            MessageBox.Show(sf.ErrorMsg(sf.LastErrorCode))
        End If
        layer = AxMap1.AddLayer(sf, True)
        AxMap1.ZoomToLayer(layer)


        ' Create an array of MapWinGIS Image objects to load the images from files
        For I = 0 To 3
            Images(I) = New MapWinGIS.Image
        Next
        Images(0).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\North.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        Images(1).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\East.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        Images(2).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\South.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        Images(3).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\West.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        For I = 0 To 3
            AxMap1.set_UDPointImageListAdd(layer, Images(I))
        Next

        ' Assign images from image list to points
        For shp = 0 To sf.NumShapes - 1
            AxMap1.set_ShapePointType(layer, shp, MapWinGIS.tkPointType.ptImageList)
            type = shp Mod 4 'simply assign each of the four images over and over again
            AxMap1.set_ShapePointImageListID(layer, shp, type)
        Next

        '  Later, if you needed to determine which images were in the list:
        For I = 0 To AxMap1.get_UDPointImageListCount(layer) - 1
            Images(I) = AxMap1.get_UDPointImageListItem(layer, I)
        Next
    End Sub
```

3.9.3.60 ShapePointSize

Gets or sets the line point/vertex size for the specified shape. Only works on shapefiles.

VB.NET Usage

Property set_ShapePointSize(LayerHandle As Integer, Shape As Integer, param0 as Single)

Parameters

LayerHandle	Handle of the layer containing the shape for which the point/vertex size is to be set.
Shape	Handle of the shape for which the point/vertex size is to be set.
param0	Point/vertex size for the shape in the shapefile.

'***Property get_ShapePointSize(LayerHandle As Integer, Shape As Integer) As Single***'

Parameters

LayerHandle	Handle of the layer containing the shape for which the point/vertex size is required.
Shape	Handle of the shape for which the point/vertex size is required.
Return Value	Point/vertex size for the shape in the shapefile.

Sample Code

```
    Private Sub PointSize()
        Dim hnd As Integer
        Dim size As Single
        'Get the handle for the layer at position 0
        hnd = Map1.get_LayerHandle(0)
```

```
    'Get the point/vertex size for image 0 in the layer
    size = Map1.get_ShapePointSize(hnd, 0)
    'Set the point/vertex size for image 0 in the layer to 3
    Map1.set_ShapePointSize(hnd, 0, 3)
End Sub
```

3.9.3.61 ShapePointType

Gets or sets the line point type for the specified shape. Only works on shapefiles.
See also tkPointType

VB.NET Usage

Property set_ShapePointType(LayerHandle As Integer, Shape As Integer, param0 As MapWinGIS.tkPointType)

Parameters

LayerHandle	Handle of the layer containing the shape for which the point type is to be set.
Shape	Handle of the shape for which the point type is to be set.
param0	Point type for the shape in the shapefile.

'***Property get_ShapePointType(LayerHandle As Integer, Shape As Integer) As MapWinGIS.tkPointType***'

Parameters

LayerHandle	Handle of the layer containing the shape for which the point type is required.
Shape	Handle of the shape for which the point type is required.
Return Value	Point type for the shape in the shapefile.

Sample Code

```
Private Sub PointType()
    Dim hnd As Integer
    Dim type As MapWinGIS.tkPointType
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'Get the point type for shape 0 in the layer
    type = Map1.get_ShapePointType(hnd, 0)
    'Set the point type for shape 0 in the layer to circle
    Map1.set_ShapePointType(hnd, 0, MapWinGIS.tkPointType.ptCircle))
End Sub
```

3.9.3.62 ShapeVisible

Gets or sets the visibility of the specified shape.

VB.NET Usage

Property set_ShapeVisible(LayerHandle As Integer, Shape As Integer, param0 As Boolean)

Parameters

LayerHandle	Handle of the layer containing the shape for which the visibility is to be set.
Shape	Handle of the shape for which the visibility is to be set.
param0	Boolean value representing whether the shape is to be visible or not.

'***Property get_ShapeVisible(LayerHandle As Integer, Shape As Integer) As Boolean***'

Parameters

LayerHandle	Handle of the layer containing the shape for which the visibility is required.
Shape	Handle of the shape for which the visibility is required.
Return Value	Boolean value representing whether the the shape is visible or not.

Sample Code

```
Private Sub ShapeVisible()
    Dim hnd As Integer
    'Get the handle for the layer at position 0
    hnd = Map1.get_LayerHandle(0)
    'See if shape 0 in the layer is visible
    If Map1.get_ShapeVisible(hnd, 0) Then
        'Set shape 0 in the layer as not visible
        Map1.set_ShapeVisible(hnd, 0, False)
    Else
        'Set shape 0 in the layer as visible
```

```
        Map1.set_ShapeVisible(hnd, 0, True)
    End If
End Sub
```

3.9.3.63 UDCursorHandle

******NOT IMPLEMENTED AT THIS TIME******

Gets or sets the user defined cursor handle. The handle is a windows cursor handle.

VB.NET Usage
Property UDCursorHandle() As Integer

Parameters

ReturnValue	The user defined cursor handle for the map.

Sample Code

```
Private Sub MapCursor()
    Dim cur As New Cursor(Me.GetType, "Cursor.cur")
    'Set the map cursor mode as cmNone
    Map1.CursorMode = MapWinGIS.tkCursorMode.cmNone
    'Set the map cursor as a user defined cursor
    Map1.MapCursor = MapWinGIS.tkCursor.crsrUserDefined
    'Set the user defined cursor handle to the current
    Map1.UDCursorHandle = cur.Handle.ToInt32
End Sub
```

3.9.3.64 UDFillStipple

******NOT IMPLEMENTED AT THIS TIME******

Gets or sets one user defined stipple row. The user defined fill stipple contains 32 rows of 32 bits. The stipple is created by setting patterns in the bits contained in each row.

VB.NET Usage Property set_UDFillStipple(LayerHandle As Integer, StippleRow As Integer, param0 As Integer)

Parameters

LayerHandle	The handle of the layer for which the fill stipple is to be set.
StippleRow	The row in the custom stipple for which the pattern is to be set.
param0	The custom fill stipple as an integer value where the stipple is defined by arranging the 32 bits in the desired pattern.

Property get_UDFillStipple(LayerHandle As Integer, StippleRow As Integer) As Integer

Parameters

LayerHandle	The handle of the layer for which the fill stipple is required.
StippleRow	The row in the custom stipple for which the pattern is to be returned.
ReturnValue	The custom fill stipple as an integer value where the stipple is defined by arranging the 32 bits in the desired pattern.

Sample Code

```
Private Sub UDFillStipple()
    Dim hndl As Integer, cust_stipple As Integer, i As Integer
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Set the custom fill stipple
    cust_stipple = 2147483647
    'Set the fill stipple for the shapes in the specified layer to be custom fill stipple
    Map1.set_ShapeLayerFillStipple(hndl, MapWinGIS.tkFillStipple.fsCustom)
    'Set the user defined fill stipple for each row
    For i = 1 To 32
        Map1.set_UDFillStipple(hndl, i, cust_stipple)
    Next
    'Get the user defined fill stipple for row 0 of the fill stipple
    cust_stipple = Map1.get_UDFillStipple(hndl, 0)
End Sub
```

3.9.3.65 UDLineStipple

This gets or sets the user defined line stipple for the specified layer.

The user defined line stipple is represented by an Integer (VB.NET) or a Long (VB 6). The first digit represents the stipple multiplier. Each of the following digits alternate between representing pixels drawn and pixels skipped in the pattern. The number of pixels to be drawn or skipped are determined by multiplying the digit representing that segment of the stipple by the stipple multiplier.

Example: 23456
This custom line stipple would draw six pixels(2*3), skip eight pixels(2*4), draw ten pixels(2*5), skip twelve pixels(2*6), with the pattern repeating from the beginning after that. The largest valid line stipple is 2147483647. Any value greater than this will result in overflow. A line stipple value less than 111 will result in a solid line.

VB.NET Usage

Property set_UDLineStipple(LayerHandle As Integer, param0 As Integer)

Parameters

LayerHandle	The handle of the layer for which the user defined shape line stipple is to be set.
param0	The user defined line stipple.

Property get_UDLineStipple(LayerHandle As Integer) As Integer

Parameters

LayerHandle	The handle of the layer for which the user defined shape line stipple is to be set.
ReturnValue	The user defined line stipple.

Sample Code

```
Private Sub UDLineStipple()
    Dim hndl As Integer, cust_stipple As Integer
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Set the custom line stipple
    cust_stipple = 2147483647
    'Set the shape line stipple to custom for the specified layer
    Map1.set_ShapeLayerLineStipple(hndl, MapWinGIS.tkLineStipple.lsCustom)
    'Set the user defined shape line stipple for the specified layer
    Map1.set_UDLineStipple(hndl, cust_stipple)
    'Get the user defined shape line stipple for the specified layer
    cust_stipple = Map1.get_UDLineStipple(hndl)
End Sub
```

3.9.3.66 UDPointImageListAdd

Allows you to build an image list so that one point shapefile can have multiple icons.

VB.NET Usage

Property set_UDImageListAdd(LayerHandle As Integer, NewValue As Object) As Integer

Parameters

LayerHandle	The layer handle of the layer for which the the list is being generated.
NewValue	The new image object to be added to the list being used for the specified layer.

Property get_UDPointImageListItem(LayerHandle As Integer, imageIndex as Integer) As Object

Parameters

LayerHandle	The layer handle of the layer for which the image list is defined.
imageIndex	The Integer index for the image to retrieve
ReturnValue	The image object which is used as the point image for the specified layer.

Sample Code

```
' sf is a global shapefile variable
' layer is a global integer variable
 <summary>
This function will plot different images on the map for the same shapefile by
establishing an image list using AxMap1.set_UDPointImageListAdd and then
setting the shapes to use the list using a point type called
MapWinGIS.tkPointType.ptImageList and then specifying which image to use with
AxMap1.set_ShapePointImageList
```

```
    </summary>
    Private Sub cmdAddPoints_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
cmdAddPoints.Click
        Dim result As Boolean
        Dim shp, type, I As Integer
        Dim Images(3) As MapWinGIS.Image
        Dim PointShapeFile As String

        'Load a point shapefile into memory
        PointShapefile = "C:\Documents and Settings\Ted\My Documents\Fundamentals\Week
3\Lab3\DataVisualization\LIGHTS.SHP"
        sf = New MapWinGIS.Shapefile
        result = sf.Open(PointShapeFile)
        If result = False Then
            MessageBox.Show(sf.ErrorMsg(sf.LastErrorCode))
        End If
        layer = AxMap1.AddLayer(sf, True)
        AxMap1.ZoomToLayer(layer)


        ' Create an array of MapWinGIS Image objects to load the images from files
        For I = 0 To 3
            Images(I) = New MapWinGIS.Image
        Next
        Images(0).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\North.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        Images(1).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\East.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        Images(2).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\South.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        Images(3).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\West.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        For I = 0 To 3
            AxMap1.set_UDPointImageListAdd(layer, Images(I))
        Next

        ' Assign images from image list to points
        For shp = 0 To sf.NumShapes - 1
            AxMap1.set_ShapePointType(layer, shp, MapWinGIS.tkPointType.ptImageList)
            type = shp Mod 4 'simply assign each of the four images over and over again
            AxMap1.set_ShapePointImageListID(layer, shp, type)
        Next

        '  Later, if you needed to determine which images were in the list:
        For I = 0 To AxMap1.get_UDPointImageListCount(layer) - 1
            Images(I) = AxMap1.get_UDPointImageListItem(layer, I)
        Next
    End Sub
```

3.9.3.67 UDPointImageListCount

Gets the number of images currently stored in the image list for custom points images.

VB.NET Usage

Property get_UDPointImageListCount(LayerHandle As Integer) As Integer

Parameters

LayerHandle	The layer handle of the layer for which the image list is defined.
ReturnValue	The Count of images currently stored in the image index of custom point types for that layer. Remember to subtract 1 from this count when indexing the 0 based image list.

Sample Code

```
    ' sf is a global shapefile variable
    ' layer is a global integer variable
    <summary>
    This function will plot different images on the map for the same shapefile by
    establishing an image list using AxMap1.set_UDPointImageListAdd and then
    setting the shapes to use the list using a point type called
    MapWinGIS.tkPointType.ptImageList and then specifying which image to use with
    AxMap1.set_ShapePointImageList
    </summary>
```

```
    Private Sub cmdAddPoints_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
cmdAddPoints.Click
        Dim result As Boolean
        Dim shp, type, I As Integer
        Dim Images(3) As MapWinGIS.Image
        Dim PointShapeFile As String

        'Load a point shapefile into memory
        PointShapefile = "C:\Documents and Settings\Ted\My Documents\Fundamentals\Week
3\Lab3\DataVisualization\LIGHTS.SHP"
        sf = New MapWinGIS.Shapefile
        result = sf.Open(PointShapeFile)
        If result = False Then
            MessageBox.Show(sf.ErrorMsg(sf.LastErrorCode))
        End If
        layer = AxMap1.AddLayer(sf, True)
        AxMap1.ZoomToLayer(layer)


        ' Create an array of MapWinGIS Image objects to load the images from files
        For I = 0 To 3
            Images(I) = New MapWinGIS.Image
        Next
        Images(0).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\North.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        Images(1).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\East.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        Images(2).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\South.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        Images(3).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\West.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
        For I = 0 To 3
            AxMap1.set_UDPointImageListAdd(layer, Images(I))
        Next

        ' Assign images from image list to points
        For shp = 0 To sf.NumShapes - 1
            AxMap1.set_ShapePointType(layer, shp, MapWinGIS.tkPointType.ptImageList)
            type = shp Mod 4 'simply assign each of the four images over and over again
            AxMap1.set_ShapePointImageListID(layer, shp, type)
        Next

        '  Later, if you needed to determine which images were in the list:
        For I = 0 To AxMap1.get_UDPointImageListCount(layer) - 1
            Images(I) = AxMap1.get_UDPointImageListItem(layer, I)
        Next
    End Sub
```

3.9.3.68 UDPointImageListItem

Allows you to build an image list so that one point shapefile can have multiple icons.

VB.NET Usage

Property set_UDImageListAdd(LayerHandle As Integer, NewValue As Object) As Integer

Parameters

LayerHandle	The layer handle of the layer for which the the list is being generated.
NewValue	The new image object to be added to the list being used for the specified layer.

Property get_UDPointImageListItem(LayerHandle As Integer, imageIndex as Integer) As Object

Parameters

LayerHandle	The layer handle of the layer for which the image list is defined.
imageIndex	The Integer index for the image to retrieve
ReturnValue	The image object which is used as the point image for the specified layer.

Sample Code

```
    ' sf is a global shapefile variable
    ' layer is a global integer variable
     <summary>
    This function will plot different images on the map for the same shapefile by
```

```
    establishing an image list using AxMap1.set_UDPointImageListAdd and then
    setting the shapes to use the list using a point type called
    MapWinGIS.tkPointType.ptImageList and then specifying which image to use with
    AxMap1.set_ShapePointImageList
    </summary>
   Private Sub cmdAddPoints_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
cmdAddPoints.Click
       Dim result As Boolean
       Dim shp, type, I As Integer
       Dim Images(3) As MapWinGIS.Image
       Dim PointShapeFile As String

       'Load a point shapefile into memory
       PointShapefile = "C:\Documents and Settings\Ted\My Documents\Fundamentals\Week
3\Lab3\DataVisualization\LIGHTS.SHP"
       sf = New MapWinGIS.Shapefile
       result = sf.Open(PointShapeFile)
       If result = False Then
          MessageBox.Show(sf.ErrorMsg(sf.LastErrorCode))
       End If
       layer = AxMap1.AddLayer(sf, True)
       AxMap1.ZoomToLayer(layer)


       ' Create an array of MapWinGIS Image objects to load the images from files
       For I = 0 To 3
          Images(I) = New MapWinGIS.Image
       Next
       Images(0).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\North.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
       Images(1).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\East.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
       Images(2).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\South.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
       Images(3).Open("C:\Dev\Unpublished\taudem\Tester\Graphics\West.bmp",
MapWinGIS.ImageType.USE_FILE_EXTENSION)
       For I = 0 To 3
          AxMap1.set_UDPointImageListAdd(layer, Images(I))
       Next

       ' Assign images from image list to points
       For shp = 0 To sf.NumShapes - 1
          AxMap1.set_ShapePointType(layer, shp, MapWinGIS.tkPointType.ptImageList)
          type = shp Mod 4 'simply assign each of the four images over and over again
          AxMap1.set_ShapePointImageListID(layer, shp, type)
       Next

       '  Later, if you needed to determine which images were in the list:
       For I = 0 To AxMap1.get_UDPointImageListCount(layer) - 1
          Images(I) = AxMap1.get_UDPointImageListItem(layer, I)
       Next
   End Sub
```

3.9.3.69 UDPointType

Gets or sets the image used when drawing points on the specified layer.

VB.NET Usage

Property set_UDPointType(LayerHandle As Integer, param0 As Object)

Parameters

LayerHandle	The layer handle of the layer for which the point type is to be set.
param0	The image object to be used as the point image for the specified layer.

Property get_UDPointType(LayerHandle As Integer) As Object

Parameters

LayerHandle	The layer handle of the layer for which the point type is required.
ReturnValue	The image object which is used as the point image for the specified layer.

Sample Code

```
Private Sub UDPointType()
    'This function assumes you have added a point shapefile to the map
    ' at position 0
    Dim hndl As Integer
    Dim image As New MapWinGIS.Image()
    'Get the layer handle for the layer at position 0
    hndl = Map1.get_LayerHandle(0)
    'Open the image to use for the user defined points
    image.Open("C:\PointImage.bmp")
    'Use transparency color
    image.UseTransparencyColor = True
    'Set the transparencey color to white
    image.TransparencyColor = System.Convert.ToUInt32(RGB(255, 255, 255))
    'Set the user defined point type to the image opened earlier
    Map1.set_UDPointType(hndl, image)
    'Set the point size of the layer to 1
    Map1.set_ShapeLayerPointSize(hndl, 1)
    'Set the point type for the layer to user defined
    Map1.set_ShapeLayerPointType(hndl, MapWinGIS.tkPointType.ptUserDefined)
End Sub
```

3.9.3.70 ZoomPercent

Gets or sets the factor by which to zoom the view of the map in or out.

VB.NET Usage

Property ZoomPercent() As Double

Parameters

ReturnValue	The factor used to determine how much the view of the map changes when the user zooms in or out.

Sample Code

```
Private Sub ZoomIn()
    Dim zoompercent As Double
    'Set the zoom percent to 0.5 percent
    zoompercent = 0.5
    'Set the map's zoom percent
    Map1.ZoomPercent = zoompercent
    'Zoom in map view by the map's zoom percent
    Map1.ZoomIn(Map1.ZoomPercent)
    'Get the map's zoom percent
    zoompercent = Map1.ZoomPercent
End Sub
```

3.9.3.71 Events

3.9.3.72 ExtentsChanged

This event is fired when the extents of the map change.

VB.NET Usage

Event ExtentsChanged(Sender As Object, e As System.EventArgs)

Parameters

Sender	The sender of the event.
e	There are no event arguments accessible through this parameter for this event.

Sample Code

```
Private Sub Map1_ExtentsChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Map1.ExtentsChanged
    Dim extents As MapWinGIS.Extents
    Dim xmin As Double, ymin As Double, zmin As Double, xmax As Double, ymax As Double, zmax As Double
    'Get the new map extents
```

```
    extents = Map1.Extents
    'Get the minimum and maximum values of x and y that define the map extents
    extents.GetBounds(xmin, ymin, zmin, xmax, ymax, zmax)
    'Display a message box that tells the user the size of the extents for the map view
    MsgBox("New extents. Width:" + Str(xmax - xmin) + " Height:" + Str(ymax - ymin))
End Sub
```

3.9.3.73 FileDropped

This event is fired when a user drags a file and drops it on the map.

VB.NET Usage

Event FileDropped(Sender As Object, e As AxMapWinGIS._DMapEvents_FileDroppedEvent)

Parameters

Sender	The sender of the event.
e.filename	The filename of the file dropped on the map.

Sample Code

```
Private Sub Map1_FileDropped(ByVal sender As Object, ByVal e As AxMapWinGIS._DMapEvents_FileDroppedEvent)
Handles Map1.FileDropped
    'If a shapefile is dropped on the map then load it
    AddShapefile(e.filename)
End Sub
```

3.9.3.74 MapState

Event MapState(LayerHandle As Int32)

This event occurs when the MapState has been applied to a specific layer. The event will be called once per layer until all layers in the MapState string are loaded.

Parameters

LayerHandle	Handle of the layer.

3.9.3.75 MouseDown

This event is fired when a user has pressed a mouse button while the cursor is inside the map control. The map property SendMouseDown must be set to True for this event to be fired.
See also SendMouseDown

VB.NET Usage

Event MouseDownEvent(Sender As Object, e As AxMapWinGIS._DMapEvents_MouseDownEvent)

Parameters

Sender	The sender of the event.
e.button	The button pressed on the mouse to create this event.
e.shift	The shift/ctrl modifiers pressed during the creation of this event.
e.x	The x coordinate of this event in pixel coordinates.
e.y	The y coordinate of this event in pixel coordinates.

Sample Code

```
Private Sub Map1_MouseDownEvent(ByVal sender As Object, ByVal e As
AxMapWinGIS._DMapEvents_MouseDownEvent) Handles Map1.MouseDownEvent
    'This will draw a point where you click on the map if you have a
    'spatially referenced shapefile or grid displayed on the map
    Dim x As Double, y As Double
    'Convert pixel coordinates of mouse up event e.x and e.y to projected coordinates returned by x and y
    Map1.PixelToProj(e.x, e.y, x, y)
    'Create a new drawing layer in map1
    Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlSpatiallyReferencedList)
    'Draw a red point where the mouse up event occured in map1
    Map1.DrawPoint(x, y, 5, System.Convert.ToUInt32(RGB(255, 0, 0)))
End Sub
```

3.9.3.76 MouseMove

This event is fired when the mouse is moved while the cursor is inside the map control. The map property SendMoveMouse must be set to True for this event to be fired.
See also SendMouseMove

VB.NET Usage

Event MouseMoveEvent(Sender As Object, e As AxMapWinGIS._DMapEvents_MouseMoveEvent)

Parameters

Sender	The sender of the event.
e.button	The button pressed on the mouse to create this event.
e.shift	The shift/ctrl modifiers pressed during the creation of this event.
e.x	The x coordinate of this event in pixel coordinates.
e.y	The y coordinate of this event in pixel coordinates.

Sample Code

```
Private Sub Map1_MouseMoveEvent(ByVal sender As Object, ByVal e As
AxMapWinGIS._DMapEvents_MouseMoveEvent) Handles Map1.MouseMoveEvent
    Dim hndl As Integer
    Dim projX As Double, projY As Double
    Dim sf As MapWinGIS.Shapefile
    'Get handle for layer at position 0 in map
    hndl = Map1.get_LayerHandle(0)
    'Get shapefile in layer at position 0 in map
    sf = Map1.get_GetObject(hndl)
    'Convert mouse move x and y into projected map coordinates
    Map1.PixelToProj(e.x, e.y, projX, projY)
    'See if the mouse has moved over shape 0 in the shapefile
    If sf.PointInShape(0, projX, projY) Then
        'Show the tooltip for 2 seconds
        Map1.ShowToolTip("Shape 0", 2000)
    End If
End Sub
```

3.9.3.77 MouseUp

This event is fired when the mouse button is released while the cursor is in the map control. The map property SendMouseUp must be set to True for this event to be fired.
See also SendMouseUp

VB.NET Usage

Event MouseUpEvent(Sender As Object, e As AxMapWinGIS._DMapEvents_MouseUpEvent)

Parameters

Sender	The sender of the event.
e.button	The button pressed on the mouse to create this event.
e.shift	The shift/ctrl modifiers pressed during the creation of this event.
e.x	The x coordinate of this event in pixel coordinates.
e.y	The y coordinate of this event in pixel coordinates.

Sample Code

```
Private Sub Map1_MouseUpEvent(ByVal sender As Object, ByVal e As AxMapWinGIS._DMapEvents_MouseUpEvent)
Handles Map1.MouseUpEvent
    'This will draw a point where you click on the map if you have a
    'spatially referenced shapefile or grid displayed on the map
    Dim x As Double, y As Double
    'Convert pixel coordinates of mouse up event e.x and e.y to projected coordinates returned by x and y
    Map1.PixelToProj(e.x, e.y, x, y)
    'Create a new drawing layer in map1
    Map1.NewDrawing(MapWinGIS.tkDrawReferenceList.dlSpatiallyReferencedList)
    'Draw a red point where the mouse up event occured in map1
    Map1.DrawPoint(x, y, 5, System.Convert.ToUInt32(RGB(255, 0, 0)))
End Sub
```

3.9.3.78 SelectBoxDrag

This event is fired while the user is dragging a selection box in the map control. The map property SendSelectBoxDrag must be set to True for this event to be fired.
See also SendSelectBoxDrag

VB.NET Usage

Event SelectBoxDrag(Sender As Object, e As AxMapWinGIS._DMapEvents_SelectBoxDragEvent)

Parameters

Sender	The sender of the event.
e.bottom	The bottom boundary of the selection box in pixel coordinates.
e.left	The left boundary of the selection box in pixel coordinates.
e.right	The right boundary of the selection box in pixel coordinates.
e.top	The top boundary of the selection box in pixel coordinates.

Sample Code

```
  Private Sub Map1_SelectBoxDrag(ByVal sender As Object, ByVal e As
AxMapWinGIS._DMapEvents_SelectBoxDragEvent) Handles Map1.SelectBoxDrag
    Dim sf As MapWinGIS.Shapefile
    Dim myExtents As New MapWinGIS.Extents()
    Dim selectedShapes() As Integer
    Dim i As Integer, hndl As Integer
    Dim pxMin As Double, pxMax As Double, pyMin As Double, pyMax As Double, pzMin As Double, pzMax As Double
    Dim col As System.Drawing.Color
    'Check if the map is in selection mode
    If Map1.CursorMode = MapWinGIS.tkCursorMode.cmSelection Then
      'Get the handle of the layer at position 0
      hndl = Map1.get_LayerHandle(0)
      'Get the shapefile in the specified layer
      sf = Map1.get_GetObject(hndl)
      'Convert the boundaries of the selection box from pixel units to projected map coordinates
      Map1.PixelToProj(e.left, e.bottom, pxMin, pyMin)
      Map1.PixelToProj(e.right, e.top, pxMax, pyMax)
      'Set the extents object to be used to find shapes that have been selected in the shapefile
      myExtents.SetBounds(pxMin, pyMin, 0, pxMax, pyMax, 0)
      'Check if there are any shapes with in the shapefile that intersect with the selection box
      If sf.SelectShapes(myExtents, 0, MapWinGIS.SelectMode.INTERSECTION, selectedShapes) Then
        'Get the System.Drawing.Color which is being used as the fill color for the shapes in the layer
        col = Map1.get_ShapeLayerFillColor(hndl)
        'Set all shapes in the shapefile back to their original color
        Map1.set_ShapeLayerFillColor(hndl, System.Convert.ToUInt32(RGB(col.R, col.G, col.B)))
        'For each of the selected shapes in the shapefile, color them differently than their original fill color
        For i = 0 To UBound(selectedShapes)
          Map1.set_ShapeFillColor(hndl, selectedShapes(i), System.Convert.ToUInt32(RGB(100, 100, 0)))
        Next
      End If
    End If
  End Sub
```

3.9.3.79 SelectBoxFinal

This event is fired when the user finishes dragging a selection box in the map control. The map property SendSelectBoxFinal must be set to True for this event to be fired.
See also SendSelectBoxFinal

VB.NET Usage

Event SelectBoxFinal(Sender As Object, e As AxMapWinGIS._DMapEvents_SelectBoxFinalEvent)

Parameters

Sender	The sender of the event.
e.bottom	The bottom boundary of the selection box in pixel coordinates.
e.left	The left boundary of the selection box in pixel coordinates.

e.right	The right boundary of the selection box in pixel coordinates.
e.top	The top boundary of the selection box in pixel coordinates.

Sample Code

```
  Private Sub Map1_SelectBoxFinal(ByVal sender As Object, ByVal e As
AxMapWinGIS._DMapEvents_SelectBoxFinalEvent) Handles Map1.SelectBoxFinal
    Dim sf As MapWinGIS.Shapefile
    Dim myExtents As New MapWinGIS.Extents()
    Dim selectedShapes() As Integer
    Dim i As Integer, hndl As Integer
    Dim pxMin As Double, pxMax As Double, pyMin As Double, pyMax As Double, pzMin As Double, pzMax As Double
    Dim col As System.Drawing.Color
    'Check if the map is in selection mode
    If Map1.CursorMode = MapWinGIS.tkCursorMode.cmSelection Then
      'Get the handle of the layer at position 0
      hndl = Map1.get_LayerHandle(0)
      'Get the shapefile in the specified layer
      sf = Map1.get_GetObject(hndl)
      'Convert the boundaries of the selection box from pixel units to projected map coordinates
      Map1.PixelToProj(e.left, e.bottom, pxMin, pyMin)
      Map1.PixelToProj(e.right, e.top, pxMax, pyMax)
      'Set the extents object to be used to find shapes that have been selected in the shapefile
      myExtents.SetBounds(pxMin, pyMin, 0, pxMax, pyMax, 0)
      'Check if there are any shapes with in the shapefile that intersect with the selection box
      If sf.SelectShapes(myExtents, 0, MapWinGIS.SelectMode.INTERSECTION, selectedShapes) Then
        'Get the System.Drawing.Color which is being used as the fill color for the shapes in the layer
        col = Map1.get_ShapeLayerFillColor(hndl)
        'Set all shapes in the shapefile back to their original color
        Map1.set_ShapeLayerFillColor(hndl, System.Convert.ToUInt32(RGB(col.R, col.G, col.B)))
        'For each of the selected shapes in the shapefile, color them differently than their original fill color
        For i = 0 To UBound(selectedShapes)
          Map1.set_ShapeFillColor(hndl, selectedShapes(i), System.Convert.ToUInt32(RGB(100, 100, 0)))
        Next
      End If
    End If
  End Sub
```

3.10 Point

A point object represents a point with x, y, Z, and M values. Shapes created by adding point objects to the shape.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.10.1 Properties

3.10.1.1 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
  Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
  End Sub
```

3.10.1.2 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
      '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
              'Display the error message in a label
             LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
             'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
             'Display the message in a label
         Label2.Text = Message
     End Sub
 #End Region
              '...
```

3.10.1.3 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
    Private Sub MapKey()
        Dim k As String
        'Get the map's key
        k = Map1.Key
        'Check if the map's key is "Map1"
        If k = "Map1" Then
            'Set the map's key to "My Map1"
            Map1.Key = "My Map1"
        Else
            'Set the map's key to "Map1"
            Map1.Key = "Map1"
        End If
    End Sub
```

3.10.1.4 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.10.1.5 M

Gets or sets the measure value of this point. Measures only apply to shapefiles with measure data.

VB.NET Usage

Property M() As Double

Parameters

ReturnValue	The measure value of the point.

Sample Code

```
Private Sub PointM()
    Dim point As New MapWinGIS.Point()
    Dim measure As Double
    'Set the measure value
    measure = 3000
    'Set the measure value for the point
    point.M = measure
    'Get the measure value for the point
    measure = point.M
End Sub
```

3.10.1.6 Z

Gets or sets the Z value of this point.

VB.NET Usage

Property Z() As Double

Parameters

ReturnValue	The z value of the point.

Sample Code

```
Private Sub PointZ()
    Dim point As New MapWinGIS.Point()
    Dim zvalue As Double
    'Set the z value value
    zvalue = 3000
    'Set the z value value for the point
    point.Z = zvalue
    'Get the z value value for the point
    zvalue = point.Z
End Sub
```

3.10.1.7 x

Gets or sets the x value of the point.

VB.NET Usage

Property x() As Double

Parameters

ReturnValue	The x value of the point.

Sample Code

```
Private Sub Pointx()
    Dim point As New MapWinGIS.Point()
    Dim xvalue As Double
    'Set the x value value
    xvalue = 3000
    'Set the x value value for the point
    point.x = xvalue
    'Get the x value value for the point
    xvalue = point.x
End Sub
```

3.10.1.8 Y

Gets or sets the y value of the point.

VB.NET Usage

Property y() As Double

Parameters

ReturnValue	The y value of the point.

Sample Code

```
Private Sub Pointy()
    Dim point As New MapWinGIS.Point()
    Dim yvalue As Double
    'Set the y value value
    yvalue = 3000
    'Set the y value value for the point
    point.y = yvalue
    'Get the y value value for the point
    yvalue = point.y
End Sub
```

3.11 Shape

A shape object represents a geometric shape which can be added to a shapefile which is displayed in the map.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.11.1 Functions

3.11.1.1 Create

Creates a new shape of the specified type.

VB.NET Usage

Function Create(ShpType As MapWinGIS.ShpfileType) As Boolean

Parameters

ShpType	The type of the shape to be created.
ReturnValue	A boolean value representing the success or failure of creating the new shape.

Sample Code

```
Private Sub CreateShape()
    Dim shape As New MapWinGIS.Shape()
    Dim success As Boolean
    'Create a new polygon shape
    success = shape.Create(MapWinGIS.ShpfileType.SHP_POLYGON)
End Sub
```

3.11.1.2 CreateFromString

Initializes the Shape object and fills it with the geometry defined by the input string. The input string should be in the serialized string format as produced by the function SerializeToString.

VB.NET Usage

Function CreateFromString(Serialized As String) As Boolean

Parameters

Serialized	The serialized string to load.
ReturnValue	A boolean value representing the success or failure of loading the shape.

Visual Basic Net Example Code

```
Private Sub CopyShapes()
    Dim text As String = vbNullString
    Dim ShapeStrings() As String
    Dim s As MapWinGIS.Shape
    Dim shp As Integer
    Dim sf As New MapWinGIS.Shapefile
    Dim typ As MapWinGIS.ShpfileType
    Dim indx As Integer
    Dim res As Boolean
    Dim start As Integer
    res = sf.Open("C:\FLORIDA\CountyBounds\SolidBnds.shp")
    If res = False Then
        MessageBox.Show(sf.ErrorMsg(sf.LastErrorCode))
        Exit Sub
    End If
    '-----------Encode an entire shapefile to a string
    For shp = 0 To sf.NumShapes - 1
        s = sf.Shape(shp)
        If shp = 0 Then
            text = s.SerializeToString()
        Else
            text += "," & s.SerializeToString()
        End If
    Next
    sf.Close()
    '------------Read a string for an entire shapefile
    ShapeStrings = text.Split(",")
    'Use the first shape to figure out the shapefile type
    s = New MapWinGIS.Shape
    start = 0
    'The first shape might not be readable, so get the first one that is
    While s.CreateFromString(ShapeStrings(start)) = False
        start += 1
        If start > ShapeStrings.GetUpperBound(0) Then
            MessageBox.Show("No readable shapes could be found in the strings.")
            Exit Sub
        End If
    End While
    typ = s.ShapeType
    If System.IO.File.Exists("C:\Test.shp") Then System.IO.File.Delete("C:\Test.shp")
    If System.IO.File.Exists("C:\Test.shx") Then System.IO.File.Delete("C:\Test.shx")
    If System.IO.File.Exists("C:\Test.dbf") Then System.IO.File.Delete("C:\Test.dbf")
    res = sf.CreateNew("C:\Test.shp", typ)
    If res = False Then
        MessageBox.Show(sf.ErrorMsg(sf.LastErrorCode))
        Exit Sub
    End If
    sf.StartEditingShapes()
    sf.StartEditingTable()
    Dim fld As Integer
    Dim field As New MapWinGIS.Field()
    field.Type = MapWinGIS.FieldType.INTEGER_FIELD
    field.Name = "Index"
    res = sf.EditInsertField(field, fld)
    If res = False Then If (ShowError(sf.ErrorMsg(sf.LastErrorCode), 0)) = True Then Exit Sub
    res = sf.EditInsertShape(s, indx)
    If res = False Then If (ShowError(sf.ErrorMsg(sf.LastErrorCode), shp)) = True Then Exit Sub
    sf.EditCellValue(fld, indx, start)
```

```
    'Note, this code is for illustration purposes and does not include handling for fields which may be necessary
    start += 1
    For shp = start To ShapeStrings.GetUpperBound(0)
        s = New MapWinGIS.Shape
        res = s.CreateFromString(ShapeStrings(shp))
        If res = False Then
            'We will warn on an individual failure, but not halt execution
            'Of course our new shapefile will not have the correct number of shapes in it
            'if we continue here.
            If (ShowError(s.ErrorMsg(s.LastErrorCode), shp)) = True Then Exit Sub
        Else
            res = sf.EditInsertShape(s, indx)
            If res = False Then If (ShowError(sf.ErrorMsg(sf.LastErrorCode), shp)) = True Then Exit Sub
            sf.EditCellValue(fld, indx, shp)
        End If
    Next
    sf.StopEditingTable()
    sf.StopEditingShapes() 'This should effectively save the shapefile
    sf.Close()
End Sub
Function ShowError(ByVal Message As String, ByVal Shape As Integer) As Boolean
    'This function shows the error, and returns true if they decide to abort
    Return (MessageBox.Show("The following error occured on shape " & Shape & "." & vbNewLine & _
      Message, "Serializing Error", MessageBoxButtons.YesNo, MessageBoxIcon.Error) = _
      Windows.Forms.DialogResult.No)
End Function

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    CopyShapes()
    MessageBox.Show("Done.")
End Sub
```

3.11.1.3 DeletePart

Deletes a part from a shape.

VB.NET Usage

Function DeletePart(PartIndex As Integer) As Boolean

Parameters

PartIndex	The index of the part to be deleted.
ReturnValue	A boolean value representing the success or failure of deleting the part.

Sample Code

```
Private Sub DeletePart()
    Dim shape As New MapWinGIS.Shape()
    Dim success As Boolean
    'Delete the part of index 0 in the shape
    success = shape.DeletePart(0)
End Sub
```

3.11.1.4 DeletePoint

Deletes a point in the shape.

VB.NET Usage

Function DeletePoint(PointIndex As Integer) As Boolean

Parameters

PointIndex	The index of the point in the shape to be deleted.
ReturnValue	A boolean value representing the success or failure of deleting the point in the shape.

Sample Code

```
Private Sub DeletePoint()
    Dim shape As New MapWinGIS.Shape()
    Dim success As Boolean
    'Delete the point indexed by 0 in the shape
```

```
    success = shape.DeletePoint(0)
End Sub
```

3.11.1.5 InsertPart

Inserts a part into the shape. Parts are used to create polygons with holes. Parts with points ordered in a clockwise direction are filled. Parts with points ordered in a counter-clockwise direction are cut out. Only clockwise parts should be used to define the outer-most regions of a shape.

VB.NET Usage

Function InsertPart(PointIndex As Integer, ByRef PartIndex As Integer) As Boolean

Parameters

PointIndex	The index of the first point in the part to be inserted.
PartIndex	The part index desired. This value may be modified if it is not possible to use the desired part index.
ReturnValue	A boolean value representing the success or failure of inserting the part into the shape.

Sample Code

```
Private Sub InsertPart()
    Dim shape As New MapWinGIS.Shape()
    Dim point(5) As MapWinGIS.Point, point2(5) As MapWinGIS.Point
    Dim partindex As Integer, pointindex As Integer, i As Integer
    Dim success As Boolean
    'Create two arrays of point objects
    For i = 0 To 4
        point(i) = New MapWinGIS.Point()
        point2(i) = New MapWinGIS.Point()
    Next
    'Create a new polygon shape object
    success = shape.Create(MapWinGIS.ShpfileType.SHP_POLYGON)
    'Set the x and y coordinates for the first part's points
    '  Note: These points are arranged in a clockwise order.
    '        As a result, these points specify the part of the shape
    '        that will be filled.
    point(0).x = 100
    point(0).y = 100
    point(1).x = 100
    point(1).y = 200
    point(2).x = 200
    point(2).y = 200
    point(3).x = 200
    point(3).y = 100
    point(4).x = 100
    point(4).y = 100
    'Insert the first part into the shape with points starting at point index 0
    success = shape.InsertPart(0, partindex)
    'Increment the part index
    partindex = partindex + 1
    'Insert each point in the point array into the shape in the first part
    For i = 0 To 4
        success = shape.InsertPoint(point(i), pointindex)
        'Increment the point index
        pointindex = pointindex + 1
    Next
    'Set the x and y coordinates for the second part's points
    '  Note: These points are arranged in a counter-clockwise order.
    '        As a result, these points specify the part to
    '        be cut out of the shape.
    point2(0).x = 120
    point2(0).y = 120
    point2(1).x = 150
    point2(1).y = 120
    point2(2).x = 150
    point2(2).y = 150
    point2(3).x = 120
    point2(3).y = 150
```

```
    point2(4).x = 120
    point2(4).y = 120
    'Insert the second part using the points from the next point index and on
    success = shape.InsertPart(pointindex, partindex)
    'Increment the part index
    partindex = partindex + 1
    'Insert each point in the point2 array into the shape in the second part
    For i = 0 To 4
       success = shape.InsertPoint(point2(i), pointindex)
       'Increment the pointindex
       pointindex = pointindex + 1
    Next
  End Sub
```

3.11.1.6 InsertPoint

Inserts the specified point object into the shape using the desired point index if possible.

VB.NET Usage

Function InsertPoint(NewPoint As MapWinGIS.Point, ByRef PointIndex As Integer) As Boolean

Parameters

NewPoint	The point object to be inserted into the shape.
PointIndex	Refrence parameter. The index where the point will be placed if possible. If the desired index cannot be used, the actual index will be returned.
ReturnValue	A boolean value representing the success or failure of inserting the point into the shape.

Sample Code

```
  Private Sub InsertPoint()
    Dim shape As New MapWinGIS.Shape()
    Dim point As New MapWinGIS.Point()
    Dim pointindex As Integer
    Dim success As Boolean
    'Set the values for the point to be inserted
    point.x = 100
    point.y = 100
    'Set the desired point index for the point to be inserted
    pointindex = 0
    'Insert the point into the shape
    success = shape.InsertPoint(point, pointindex)
  End Sub
```

3.11.1.7 SerializeToString

Provides a string representing the shape's geometry. The string will be in the serialized string format and can be reloaded with CreateFromString.

VB.NET Usage

Function SerializeToString() As String

Parameters

(none)	
ReturnValue	A string representing the shape geometry.

Visual Basic Net Example Code

```
  Private Sub CopyShapes()
    Dim text As String = vbNullString
    Dim ShapeStrings() As String
    Dim s As MapWinGIS.Shape
    Dim shp As Integer
    Dim sf As New MapWinGIS.Shapefile
    Dim typ As MapWinGIS.ShpfileType
    Dim indx As Integer
    Dim res As Boolean
    Dim start As Integer
    res = sf.Open("C:\FLORIDA\CountyBounds\SolidBnds.shp")
```

```
    If res = False Then
        MessageBox.Show(sf.ErrorMsg(sf.LastErrorCode))
        Exit Sub
    End If
    '-----------Encode an entire shapefile to a string
    For shp = 0 To sf.NumShapes - 1
        s = sf.Shape(shp)
        If shp = 0 Then
            text = s.SerializeToString()
        Else
            text += "," & s.SerializeToString()
        End If
    Next
    sf.Close()
    '------------Read a string for an entire shapefile
    ShapeStrings = text.Split(",")
    'Use the first shape to figure out the shapefile type
    s = New MapWinGIS.Shape
    start = 0
    'The first shape might not be readable, so get the first one that is
    While s.CreateFromString(ShapeStrings(start)) = False
        start += 1
        If start > ShapeStrings.GetUpperBound(0) Then
            MessageBox.Show("No readable shapes could be found in the strings.")
            Exit Sub
        End If
    End While
    typ = s.ShapeType
    If System.IO.File.Exists("C:\Test.shp") Then System.IO.File.Delete("C:\Test.shp")
    If System.IO.File.Exists("C:\Test.shx") Then System.IO.File.Delete("C:\Test.shx")
    If System.IO.File.Exists("C:\Test.dbf") Then System.IO.File.Delete("C:\Test.dbf")
    res = sf.CreateNew("C:\Test.shp", typ)
    If res = False Then
        MessageBox.Show(sf.ErrorMsg(sf.LastErrorCode))
        Exit Sub
    End If
    sf.StartEditingShapes()
    sf.StartEditingTable()
    Dim fld As Integer
    Dim field As New MapWinGIS.Field()
    field.Type = MapWinGIS.FieldType.INTEGER_FIELD
    field.Name = "Index"
    res = sf.EditInsertField(field, fld)
    If res = False Then If (ShowError(sf.ErrorMsg(sf.LastErrorCode), 0)) = True Then Exit Sub
    res = sf.EditInsertShape(s, indx)
    If res = False Then If (ShowError(sf.ErrorMsg(sf.LastErrorCode), shp)) = True Then Exit Sub
    sf.EditCellValue(fld, indx, start)
    'Note, this code is for illustration purposes and does not include handling for fields which may be necessary
    start += 1
    For shp = start To ShapeStrings.GetUpperBound(0)
        s = New MapWinGIS.Shape
        res = s.CreateFromString(ShapeStrings(shp))
        If res = False Then
            'We will warn on an individual failure, but not halt execution
            'Of course our new shapefile will not have the correct number of shapes in it
            'if we continue here.
            If (ShowError(s.ErrorMsg(s.LastErrorCode), shp)) = True Then Exit Sub
        Else
            res = sf.EditInsertShape(s, indx)
            If res = False Then If (ShowError(sf.ErrorMsg(sf.LastErrorCode), shp)) = True Then Exit Sub
            sf.EditCellValue(fld, indx, shp)
        End If
    Next
    sf.StopEditingTable()
    sf.StopEditingShapes() 'This should effectively save the shapefile
    sf.Close()
End Sub
Function ShowError(ByVal Message As String, ByVal Shape As Integer) As Boolean
    'This function shows the error, and returns true if they decide to abort
    Return (MessageBox.Show("The following error occured on shape " & Shape & "." & vbNewLine & _
        Message, "Serializing Error", MessageBoxButtons.YesNo, MessageBoxIcon.Error) = _
```

```
        Windows.Forms.DialogResult.No)
    End Function

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        CopyShapes()
        MessageBox.Show("Done.")
    End Sub
```

3.11.2 Properties

3.11.2.1 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.11.2.2 Extents

Gets the extents of the shape.

VB.NET Usage

ReadOnly Property Extents() As MapWinGIS.Extents

Parameters

ReturnValue	The extents of the shape.

Sample Code

```
Private Sub ShapeExtents()
    Dim shape As New MapWinGIS.Shape()
    Dim extents As New MapWinGIS.Extents()
    'Get the extents of the shape
    extents = shape.Extents
End Sub
```

3.11.2.3 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
        '...
 #Region "ICallback Members"
```

```
    Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
            'Display the error message in a label
            LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
            'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
            'Display the message in a label
        Label2.Text = Message
    End Sub
#End Region
            '...
```

3.11.2.4 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
    Private Sub MapKey()
        Dim k As String
        'Get the map's key
        k = Map1.Key
        'Check if the map's key is "Map1"
        If k = "Map1" Then
            'Set the map's key to "My Map1"
            Map1.Key = "My Map1"
        Else
            'Set the map's key to "Map1"
            Map1.Key = "Map1"
        End If
    End Sub
```

3.11.2.5 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
    Private Sub LastErrorCode()
        Dim errorCode As Integer
        'Get the last error in the map
        errorCode = Map1.LastErrorCode
        'Display message box giving error message for the last error in the map
        MsgBox(Map1.get_ErrorMsg(errorCode))
    End Sub
```

3.11.2.6 NumParts

Gets the number of parts contained in the shape. A polygon shape may have several parts. An example of this would be a doughnut shape. The outermost perimeter would be one part, and the hole cut out of the center making up the innermost perimeter would be the second part. The outermost perimeter would be specified by a series of points arranged in clockwise order, meaning that this part will be filled. The innermost perimeter would be specified by a series of points arranged in counter-clockwise order, meaning that this part will not be filled.

VB.NET Usage

ReadOnly Property NumParts() As Integer

Parameters

ReturnValue	The number of parts contained in the shape.

Sample Code

```
Private Sub ShapeNumParts()
    Dim shape As New MapWinGIS.Shape()
    Dim count As Integer
    'Get the number of parts contained in the shape
    count = shape.NumParts
End Sub
```

3.11.2.7 Part

Gets or sets the first point index in the specified part.

VB.NET Usage

Property Part(PartIndex As Integer) As Integer

Parameters

PartIndex	The index of the part for which the first point index is required.
ReturnValue	The index of the first point in the specified part.

Sample Code

```
Private Sub PartIndex()
    Dim shape As New MapWinGIS.Shape()
    Dim pointindex As Integer
    'Set the pointindex value
    pointindex = 0
    'Set the first point in part 0
    shape.Part(0) = pointindex
    'Get the first point in part 0
    pointindex = shape.Part(0)
End Sub
```

3.11.2.8 Point

Gets or sets the specified point in the shape.

VB.NET Usage

Property Point(PointIndex As Integer) As MapWinGIS.Point

Parameters

PointIndex	The index of the point which is to be accessed in the point.
ReturnValue	The specified point in the shape.

Sample Code

```
Private Sub ShapePoint()
    Dim shape As New MapWinGIS.Shape()
    Dim point As New MapWinGIS.Point()
    'Set the point x and y values
    point.x = 100
    point.y = 200
    'Set the point at index 0 in the shape
    shape.Point(0) = point
    'Get the point at index 1 in the shape
    point = shape.Point(1)
End Sub
```

3.11.2.9 ShapeType

Gets or sets the type of the shape. Note: The shape's type must match the type of the shapefile the shape is to be added to except for shapes of type SHP_NULLSHAPE.
See also ShpfileType

VB.NET Usage

Property ShapeType() As MapWinGIS.ShpfileType

Parameters

ReturnValue	The shapefile type of the shape. This shapefile type must match the type of any shapefile the shape is added to.

Sample Code

```
Private Sub ShapeType()
    Dim shape As New MapWinGIS.Shape()
    Dim shapetype As New MapWinGIS.ShpfileType()
    'Set the type of the shape
    shape.ShapeType = MapWinGIS.ShpfileType.SHP_POLYGON
    'Get the type of the shape
    shapetype = shape.ShapeType
End Sub
```

3.11.2.10 numPoints

Gets the number of points contained in the shape.

VB.NET Usage

ReadOnly Property numPoints() As Integer

Parameters

ReturnValue	The number of points in the shape.

Sample Code

```
Private Sub ShapeNumPoints()
    Dim shape As New MapWinGIS.Shape()
    Dim count As Integer
    'Get the number of points in the shape
    count = shape.numPoints
End Sub
```

3.12 ShapeNetwork

A shape network is created from the shapes in a polyline shapefile which allows you to traverse the shapefile simulating water flowing in a watershed to a single outlet. To create a shape network from a polyline shapefile, first choose an outlet shape from the shapes in the shapefile. Next, choose an outlet point within the specified outlet shape. Finally, you need to select a tolerance value. The tolerance is used to determine whether to include a shape in the shape network if it is not connected to the network. Using this starting criteria, a shape network can be created from the polyline shapefile.

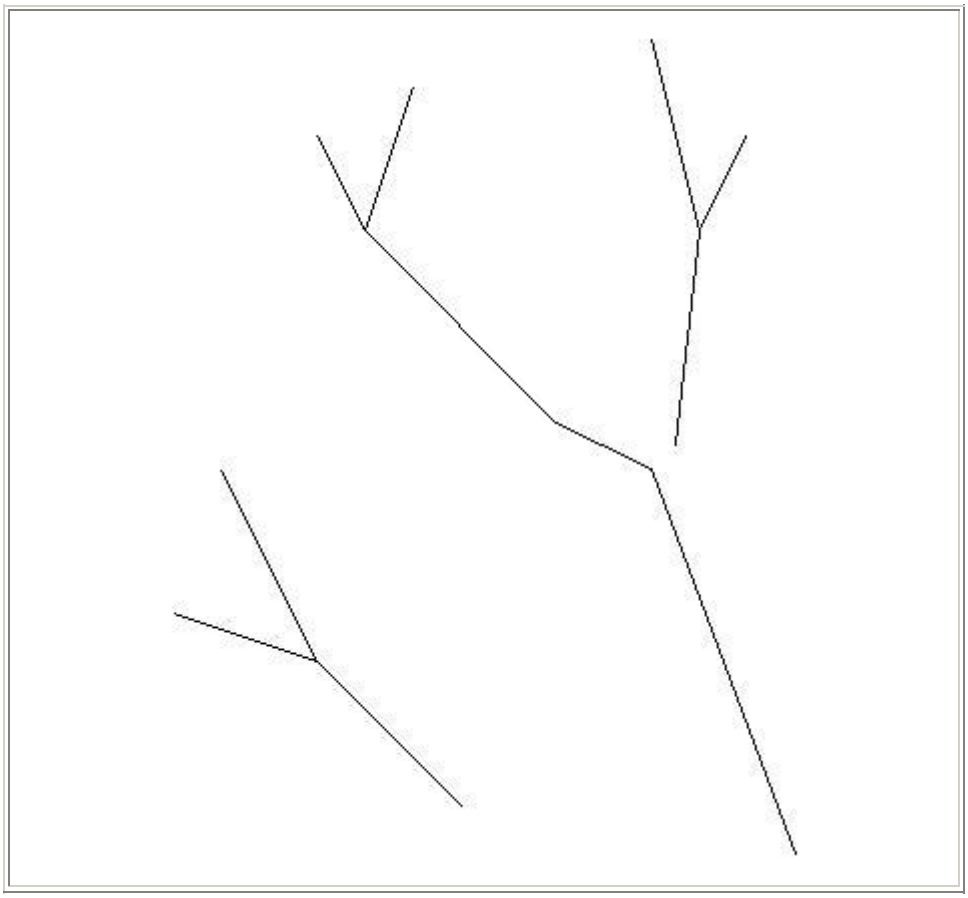
The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.12.1 Examples

3.12.1.1 ShapeNetwork Example

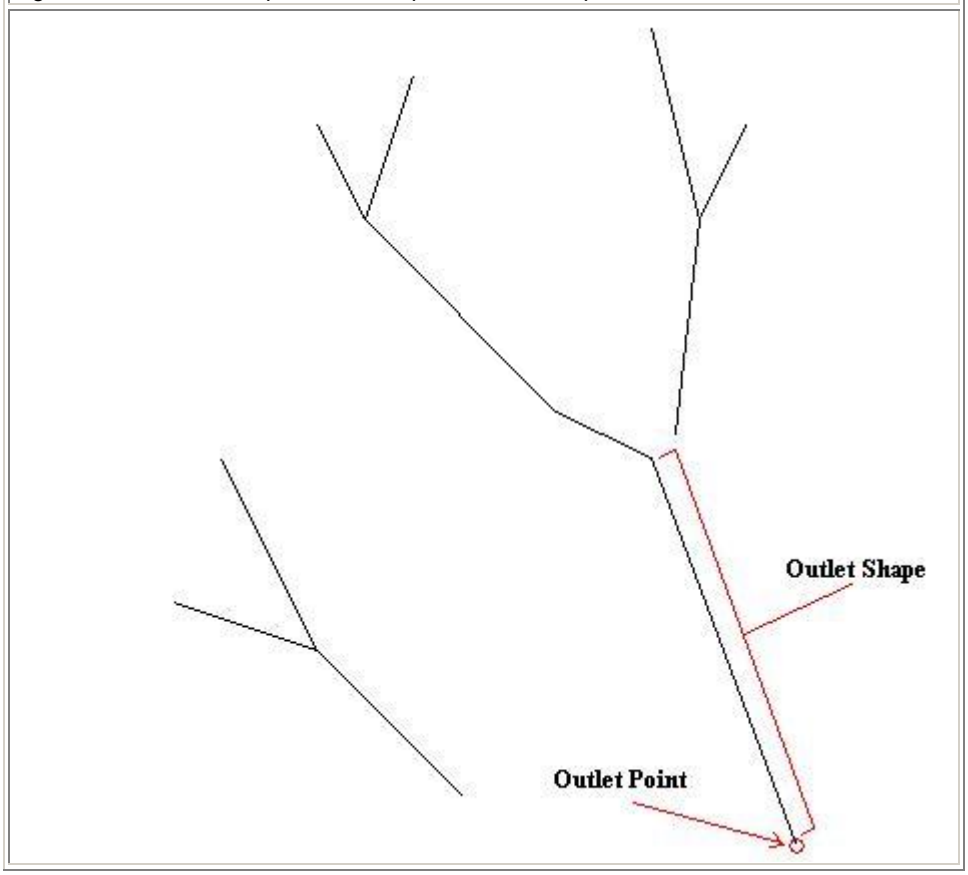
Here is a simple example of a shape network:

Figure 1: The shapefile for the shape network:



This shapefile can be used to create a shape network because it is a polyline shapefile.
Before the shape network can be created, we must choose the outlet shape and outlet point for the shape network.

Figure 2: The outlet shape and outlet point for the shape network:



In this shapefile, the outlet shape and outlet point have been selected as shown above in Figure 2. Now we are ready to create a shape network. When a shape network is created from this shapefile, we may get different networks, depending on our chosen tolerance value. Given a tolerance value of Tolerance A (Figure 3), the upper branch of the shapefile would be included in the shape network.

Figure 3: Tolerance A
(The tolerance value is represented by the radius of the red circle.)

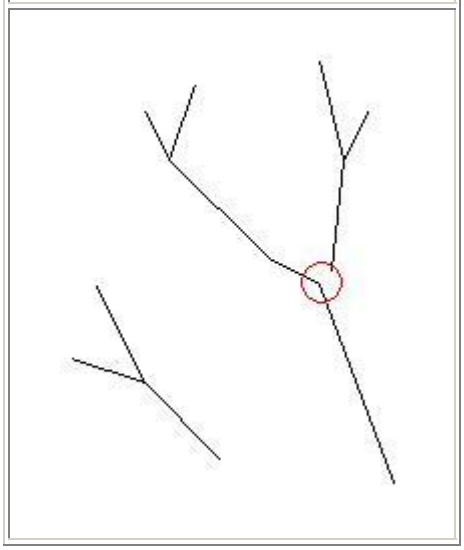
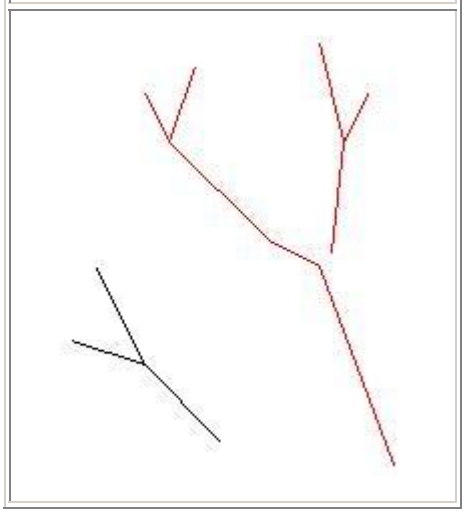


Figure 4: The shape network built using Tolerance A
(The shape network is shown in red.)



However, given a smaller tolerance value such as Tolerance B, shown in figure 5, the upper branch would be excluded from the shape network.

Figure 5: Tolerance B
(The tolerance value is represented by the radius of the red circle.)

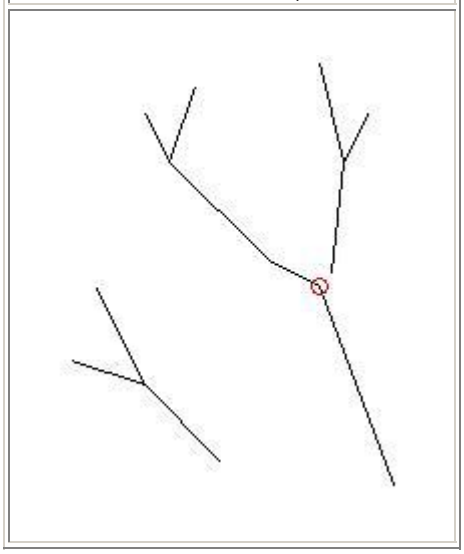
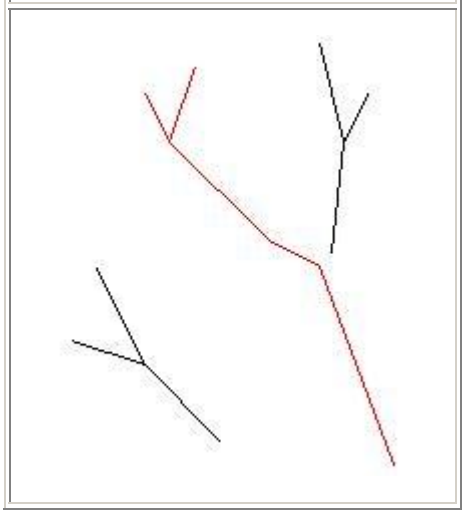


Figure 6: The shape network built using Tolerance B
(The shape network is shown in red.)



3.12.2 Functions

3.12.2.1 Build

Builds a shape network from the specified line shapefile using the shape index to determine which shape in the shapefile is to be used as the outlet shape.
See also Shapefile and AmbiguityResolution

VB.NET Usage

Function Build(Shapefile As MapWinGIS.Shapefile, ShapeIndex As Integer, FinalPointIndex As Integer, Tolerance As Double, ar As MapWinGIS.AmbiguityResolution, Optional cBack As MapWinGIS.ICallback) As Integer

Parameters

Shapefile	The shapefile to be used to create the shape network.
ShapeIndex	The index of the shape in the specified shapefile to be used as the outlet shape for the shape network.
FinalPointIndex	The point index of the point in the specified outlet shape to be used as the outlet point for the shape network.
Tolerance	The tolerance used to find disconnected segments to merge into the network.
ar	The ambiguity resolution method used to resolve any ambiguity while building the shape network. Distance to outlet is the only ambiguity resolution method implemented at this time.
cBack	Optional. The ICallback object which will receive progress and error messages during the creation of the shape network.
ReturnValue	The success or failure of building the shape network. If a non-zero integer is returned, the shape network build was successful. If a zero is returned, the shape network build was not successful.

Sample Code

```
Private Sub BuildNetwork()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim result As Integer, hndl As Integer
    'Open a line shapefile
    sf.Open("C:\test.shp")
    'Build a shape network from the shapefile using shape 0 as the outlet shape and point 0 in shape 0 as the outlet point
    result = shapenetwork.Build(sf, 0, 0, 10, MapWinGIS.AmbiguityResolution.DISTANCE_TO_OUTLET)
End Sub
```

3.12.2.2 Close

Closes the shape network.

VB.NET Usage

Function Close() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of closing the shape network.

Sample Code

```
Private Sub CloseNetwork()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim result As Integer
    'Open a line shapefile
    sf.Open("C:\test.shp")
    'Build a shape network from the shapefile using shape 0 as the outlet shape and point 0 in shape 0 as the outlet point
    result = shapenetwork.Build(sf, 0, 0, 10, MapWinGIS.AmbiguityResolution.DISTANCE_TO_OUTLET)
    'Close the shape network
    shapenetwork.Close()
End Sub
```

3.12.2.3 DeleteShape

Deletes a shape from the shape network.

VB.NET Usage

Function DeleteShape(ShapeIndex As Integer) As Boolean

Parameters

ShapeIndex	The index of the shape to be deleted from the shape network.
ReturnValue	A boolean value representing the success or failure of deleting the shape from the shape network.

Sample Code

```
Private Sub DeleteNetworkShape()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim result As Integer
```

```
    'Open a line shapefile
    sf.Open("C:\test.shp")
    'Build a shape network from the shapefile using shape 0 as the outlet shape and point 0 in shape 0 as the outlet point
    result = shapenetwork.Build(sf, 0, 0, 10, MapWinGIS.AmbiguityResolution.DISTANCE_TO_OUTLET)
    'Delete shape 0 from the shape network
    shapenetwork.DeleteShape(0)
End Sub
```

3.12.2.4 MoveDown

Moves the current shape pointer down the network by one link.

VB.NET Usage

Function MoveDown() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of moving the current shape pointer down the shape network by one link.

Sample Code

```
Private Sub ShapeNetworkMoveDown()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim result As Integer
    'Open a line shapefile
    sf.Open("C:\test.shp")
    'Build a shape network from the shapefile using shape 0 as the outlet shape and point 0 in shape 0 as the outlet point
    result = shapenetwork.Build(sf, 0, 0, 10, MapWinGIS.AmbiguityResolution.DISTANCE_TO_OUTLET)
    'Move the shape pointer down the network by one link
    shapenetwork.MoveDown()
End Sub
```

3.12.2.5 MoveTo

Moves the current shape pointer to the shape specified by the shape index.

VB.NET Usage

Function MoveTo(ShapeIndex As Integer) As Boolean

Parameters

ShapeIndex	The shape index of the shape in the network to move the current shape pointer to.
ReturnValue	A boolean value representing the success or failure of moving the current shape pointer to the specified shape.

Sample Code

```
Private Sub ShapeNetworkMoveTo()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim result As Integer, shapeindex As Integer
    'Open a line shapefile
    sf.Open("C:\test.shp")
    'Build a shape network from the shapefile using shape 0 as the outlet shape and point 0 in shape 0 as the outlet point
    result = shapenetwork.Build(sf, 0, 0, 10, MapWinGIS.AmbiguityResolution.DISTANCE_TO_OUTLET)
    'Set the index of the shape
    shapeindex = 2
    'Move the shape pointer to the shape specified by the shape index
    shapenetwork.MoveTo(shapeindex)
End Sub
```

3.12.2.6 MoveToOutlet

Moves the current shape pointer to the outlet shape in the shape network.

VB.NET Usage

Function MoveToOutlet() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of moving the current shape pointer to the outlet shape in the shape network.

Sample Code

```
Private Sub ShapeNetworkMoveToOutlet()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim result As Integer, shapeindex As Integer
    'Open a line shapefile
    sf.Open("C:\test.shp")
    'Build a shape network from the shapefile using shape 0 as the outlet shape and point 0 in shape 0 as the outlet point
    result = shapenetwork.Build(sf, 0, 0, 10, MapWinGIS.AmbiguityResolution.DISTANCE_TO_OUTLET)
    'Move the shape pointer to the outlet shape in the shape network
    shapenetwork.MoveToOutlet()
End Sub
```

3.12.2.7 MoveUp

Moves the current shape pointer up the shape network, taking the specified path.

VB.NET Usage

Function MoveUp(UpIndex As Integer) As Boolean

Parameters

UpIndex	The index of the shape in the shape network to use as the upstream path. This is necessary because a node may have more than one upstream path.
ReturnValue	A boolean value representing the success or failure of moving the current shape pointer up in the shape network.

Sample Code

```
Private Sub ShapeNetworkMoveUp()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim result As Integer, shapeindex As Integer
    'Open a line shapefile
    sf.Open("C:\test.shp")
    'Build a shape network from the shapefile using shape 0 as the outlet shape and point 0 in shape 0 as the outlet point
    result = shapenetwork.Build(sf, 0, 0, 10, MapWinGIS.AmbiguityResolution.DISTANCE_TO_OUTLET)
    'Move the shape pointer up in the shape network using shape 1 as the path
    shapenetwork.MoveUp(1)
End Sub
```

3.12.2.8 Open

Opens a shape network.
See also Shapefile

VB.NET Usage

Function Open(sf As MapWinGIS.Shapefile, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

sf	The shapefile object to open as a shape network.
cBack	Optional. The ICallback object which will receive progress and error messages while the shape network is opened.
ReturnValue	A boolean value representing the success or failure of opening the shape network.

Sample Code

```
Private Sub OpenShapeNetwork()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    'Open the shapefile
    sf.Open("C:\test.shp")
    'Open the shapefile as a shape network
```

```
    shapenetwork.Open(sf)
End Sub
```

3.12.2.9 RasterizeD8

Rasterizes the network into a D8 grid.
See also GridHeader and Grid

VB.NET Usage

Function RasterizeD8(UseNetworkBounds As Boolean, Optional Header As MapWinGIS.GridHeader, Optional Cellsize As Double, Optional cBack As MapWinGIS.ICallback) As MapWinGIS.Grid

Parameters

UseNetworkBounds	A boolean value representing whether the shape network bounds will be used as the only bounds for the D8 grid or not.
Header	Optional. A grid header to use to create the new grid.
Cellsize	Optional. The cell size to use in creating the D8 grid.
cBack	Optional. The ICallback object which will receive progress and error messages when the grid is being created.
ReturnValue	Optional. The rasterized Grid representing the shape network.

Sample Code

```
Private Sub RasterizeD8()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim grid As New MapWinGIS.Grid()
    Dim result As Integer
    'Open a line shapefile
    sf.Open("C:\test.shp")
    'Build a shape network from the shapefile using shape 0
     as the outlet shape and point 0 in shape 0 as the outlet point
    result = shapenetwork.Build(sf, 0, 0, 10, MapWinGIS.AmbiguityResolution.DISTANCE_TO_OUTLET)
    'Get a rasterized D8 grid from the shape network
    grid = shapenetwork.RasterizeD8(True)
End Sub
```

3.12.3 Properties

3.12.3.1 AmbigShapeIndex

Gets the shape index of an ambiguous shape in the shape network. The first ambiguous shape is at index 0. If there are no ambiguous shapes, AmbigShapeIndex(0) will return -1. If AmbigShapeIndex returns -1 and the index you specified isn't 0, it means there are no more ambiguous shapes in the shape network.

VB.NET Usage

ReadOnly Property AmbigShapeIndex(Index As Integer) As Integer

Parameters

Index	The index into the ambiguous shapes in the shape network.
ReturnValue	The shape index of the ambiguous shape. Returns -1 if there isn't an ambiguous shape at the specified ambiguous shape index.

Sample Code

```
Private Sub AmbiguousShape()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim shapeindex As Integer, i As Integer
    'Initialize the shapeindex and i
    shapeindex = 0
    i = 0
    'Get all ambiguous shape indexes in the shape network
    While shapeindex <> -1
        'Get the next ambiguous shapeindex in the shape network
        shapeindex = shapenetwork.AmbigShapeIndex(i)
        'Output the shape index of the ambiguous shape in the shape network
```

```
        MsgBox("Ambiguous shape index: " + Str(shapeindex))
        'Increment i
        i = i + 1
    End While
End Sub
```

3.12.3.2 CurrentShape

Gets the current shape object in the shape network.

VB.NET Usage

ReadOnly Property CurrentShape() As MapWinGIS.Shape

Parameters

ReturnValue	The current shape object in the shape network.

Sample Code

```
Private Sub CurrentShape()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim currentshape As MapWinGIS.Shape
    'Get the current shape in the shape network
    currentshape = shapenetwork.CurrentShape
End Sub
```

3.12.3.3 CurrentShapeIndex

Gets the index of the current shape in the shape network.

VB.NET Usage

ReadOnly Property CurrentShapeIndex() As Integer

Parameters

ReturnValue	The index of the current shape in the shape network.

Sample Code

```
Private Sub CurrentShapeIndex()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim shapeindex As Integer
    'Get the index of the current shape in the shape network
    shapeindex = shapenetwork.CurrentShapeIndex
End Sub
```

3.12.3.4 DistanceToOutlet

Gets the distance from the specified point in the current shape to the outlet point in the shape network.

VB.NET Usage

ReadOnly Property DistanceToOutlet(PointIndex As Integer) As Double

Parameters

PointIndex	The index of a point in the current shape.
ReturnValue	The distance from the specified point in the current shape to the outlet point in the shape network.

Sample Code

```
Private Sub DistanceToOutlet()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim distance As Double
    'Get the distance from point 0 in the current shape to the outlet point in the shape network
    distance = shapenetwork.DistanceToOutlet(0)
End Sub
```

3.12.3.5 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.12.3.6 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
      '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
              'Display the error message in a label
             LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
             'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
             'Display the message in a label
         Label2.Text = Message
     End Sub
 #End Region
              '...
```

3.12.3.7 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
```

```
    If k = "Map1" Then
        'Set the map's key to "My Map1"
        Map1.Key = "My Map1"
    Else
        'Set the map's key to "Map1"
        Map1.Key = "Map1"
    End If
End Sub
```

3.12.3.8 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.12.3.9 NetworkSize

Gets the number of shapes in the shape network.

VB.NET Usage

ReadOnly Property NetworkSize() As Integer

Parameters

ReturnValue	The number of shapes in the shape network.

Sample Code

```
Private Sub NetworkSize()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim shapecount As Integer
    'Get the number of shapes in the shape network
    shapecount = shapenetwork.NetworkSize
End Sub
```

3.12.3.10 NumDirectUps

Gets the number of upstream shapes in the network before the next fork.

VB.NET Usage

ReadOnly Property NumDirectUps() As Integer

Parameters

ReturnValue	The number of upstream shapes in the network before the next fork.

Sample Code

```
Private Sub NumDirectUps()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim directupcount As Integer
    'Get the number of upstream shapes before the next fork
    directupcount = shapenetwork.NumDirectUps
End Sub
```

3.12.3.11 ParentIndex

Gets the index of the parent of the current shape. If the current shape is the outlet shape in the shape network, -1 will be returned.

VB.NET Usage

Property ParentIndex() Integer

Parameters

ReturnValue	The index of the parent of the current shape in the shape network. If the current shape is the outlet shape in the shape network, -1 will be returned.

Sample Code

```
Private Sub ParentIndex()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim pindex As Integer
    'Get the parent index of the current shape in the shape network
    pindex = shapenetwork.ParentIndex
End Sub
```

3.12.3.12 Shapefile

Gets the shapefile used to create the network.
See also Shapefile

VB.NET Usage

ReadOnly Property Shapefile() As MapWinGIS.Shapefile

Parameters

ReturnValue	The shapefile used to create the shape network.

Sample Code

```
Private Sub ShapeNetworkShapefile()
    Dim shapenetwork As New MapWinGIS.ShapeNetwork()
    Dim sf As New MapWinGIS.Shapefile()
    'Get the shape file used to create the shape network
    sf = shapenetwork.Shapefile
End Sub
```

3.13 Shapefile

A shapefile object is an object containing shapes which define how it is to be displayed in the map.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.13.1 Functions

3.13.1.1 BeginPointInShapefile

Loads all points from all shapes in the shapefile into memory to allow better performance when calling the shapefile function PointInShapefile. If you are making extensive use of the PointInShapefile function you should notice improved performance in your application when you call BeginPointInShapefile before calling PointInShapefile.
See also PointInShapefile and EndPointInShapefile

VB.NET Usage

Function BeginPointInShapefile() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of loading all points in all shapes in the shapefile.

Sample Code

```
Private Sub BeginPtInSf()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapeindex As Integer
    'Load all points in all shapes in the shapefile into memory
    sf.BeginPointInShapefile()
    'See if a point is in the shapefile, returning the index of the shape or -1 if point doesn't lie in shapefile
    shapeindex = sf.PointInShapefile(100, 200)
End Sub
```

3.13.1.2 Close

Closes the shapefile.

VB.NET Usage

Function Close() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of closing the shapefile.

Sample Code

```
Private Sub CloseShapefile()
    Dim sf As New MapWinGIS.Shapefile()
    'Open the shapefile
    sf.Open("C:\test.shp")
    'Close the shapefile
    sf.Close()
End Sub
```

3.13.1.3 CreateNew

Creates a new shapefile with the specified filename and type. After a shapefile is created, the attribute table and shapefile are automatically in editing mode. At least one field is required in the table to be a valid shapefile.
See also ShpfileType

VB.NET Usage

Function CreateNew(ShapefileName As String, ShapefileType As MapWinGIS.ShpfileType) As Boolean

Parameters

ShapefileName	The filename to use for the new shapefile.
ShapefileType	The type of shapefile to be created.
ReturnValue	A boolean value representing the success or failure of creating the shapefile.

Sample Code

```
Private Sub NewShapefile()
    Dim sf As New MapWinGIS.Shapefile()
    Dim success As Boolean
    'Create a new polygon shapefile
    success = sf.CreateNew("test.shp", MapWinGIS.ShpfileType.SHP_POLYGON)
End Sub
```

3.13.1.4 EditCellValue

Modifies a value in the shapefile attribute table.

VB.NET Usage

Function EditCellValue(FieldIndex As Integer, ShapeIndex As Integer, NewVal As Object) As Boolean

Parameters

FieldIndex	The index of the field in the table to be modified.
ShapeIndex	The index of the shape for which the attribute value is to be modified.
NewVal	The new value to store in the table.
ReturnValue	A boolean value representing the success or failure of editing the cell value.

Sample Code

```
Private Sub EditCellValue()
    Dim sf As New MapWinGIS.Shapefile()
    Dim data As Double
    'Set the value of the data value
    data = 100
    'Enable editing of the Table in the .dbf file
    sf.StartEditingTable()
    'Change the value of the shapefile's attribute table for field 0, shape 0 to 100
    sf.EditCellValue(0, 0, data)
    'Halt editing of the Table in the .dbf file and commit changes
    sf.StopEditingTable(true)
End Sub
```

3.13.1.5 EditClear

Clears all shapes and attribute data from the shapefile. Note: Both the shapefile and attribute table must be in editing mode for this function to succeed..

VB.NET Usage

Function EditClear() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of clearing all shapes and attribute data.

Sample Code

```
Private Sub EditClear()
    Dim sf As New MapWinGIS.Shapefile()
    'Switch the shapefile into editing mode
    sf.StartEditingShapes()
    'Switch the shapefile attribute table into editing mode
    sf.StartEditingTable()
    'Clear all shapes and attribute data in the shapefile
    sf.EditClear()
End Sub
```

3.13.1.6 EditDeleteField

Deletes a field from the attribute table. Note: The attribute table must be in editing mode to delete fields.

VB.NET Usage

Function EditDeleteFields(FieldIndex As Integer, cBack As MapWinGIS.ICallBack) As Boolean

Parameters

FieldIndex	The index of the field to be deleted.
cBack	The ICallback object which will receive the progress and error messages while the field is being deleted.
ReturnValue	A boolean value representing the success or failure of deleting the field in the attribute table.

Sample Code

```
Private Sub EditDeleteField()
    Dim sf As New MapWinGIS.Shapefile()
    'Switch the shapefile attribute table into editing mode
    sf.StartEditingTable()
    'Delete field 0 in the shapefile attribute table
    sf.EditDeleteField(0)
End Sub
```

3.13.1.7 EditDeleteShape

Deletes a shape from the shapefile. Note: Both the shapefile and the attribute table must be in editing mode to delete a shape from the shapefile.

VB.NET Usage

Function EditDeleteShape(ShapeIndex As Integer) As Boolean

Parameters

ShapeIndex	The index of the shape to be deleted from the shapefile.
ReturnValue	A boolean value representing the success or failure of deleting the shape from the shapefile.

Sample Code

```
Private Sub EditDeleteShape()
    Dim sf As New MapWinGIS.Shapefile()
    'Switch the shapefile into editing mode
    sf.StartEditingShapes()
    'Delete shape 0 from the shapefile
    sf.EditDeleteShape(0)
End Sub
```

3.13.1.8 EditInsertField

Inserts a new field into the shapefile attribute table. Note: The shapefile attribute table must be in editing mode to insert a new field..
See also Field

VB.NET Usage

Function EditInsertField(NewField As MapWinGIS.Field, ByRef FieldIndex As Integer, cBack As MapWinGIS.ICallback) As Boolean

Parameters

NewField	The object to be inserted into the specified field.
FieldIndex	Reference parameter. The desired index for the new field. If the desired index is not valid, the actual index the object is placed in will be returned.
cBack	The ICallback object which will receive progress and error messages while the field is being inserted into the attribute table.
ReturnValue	A boolean value representing the success or failure of inserting the field into the attribute table.

Sample Code

```
Private Sub EditInsertField()
    Dim sf As New MapWinGIS.Shapefile()
    Dim field As New MapWinGIS.Field()
    Dim fieldindex As Integer
    'Set the field index
    fieldindex = 0
    'Switch the shapefile attribute table into editing mode
    sf.StartEditingTable()
    'Insert new field into the shapefile attribute table at index 0 if available
    sf.EditInsertField(field, fieldindex)
End Sub
```

3.13.1.9 EditInsertShape

Inserts a shape into the shapefile. Note: Both the shapefile and the attribute table must be in editing mode to insert a shape.
See also Shape

VB.NET Usage

Function EditInsertShape(Shape As MapWinGIS.Shape, ByRef ShapeIndex As Integer) As Boolean

Parameters

Shape	The shape object to be inserted into the shapefile.
ShapeIndex	Reference parameter. The desired shape index to place the shape into the shapefile. If this desired index is invalid or unavailable, the actual index will be returned.
ReturnValue	A boolean value representing the success or failure of inserting the shape into the shapefile.

Sample Code

```
Private Sub EditInsertShape()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shape As New MapWinGIS.Shape()
    Dim shapeindex As Integer
```

```
    'Set the shape index
    shapeindex = 0
    'Switch the shapefile into editing mode
    sf.StartEditingShapes()
    'Insert the shape into the shapefile at index 0 if available
    sf.EditInsertShape(shape, shapeindex)
End Sub
```

3.13.1.10 Open

Opens the specified shapefile.

VB.NET Usage

Function Open(ShapefileName As String, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

ShapefileName	The filename of the shapefile to be opened.
cBack	Optional. The ICallback object which will recieve the progress and error messages while the shapefile is being opened.
ReturnValue	A boolean value representing the success or failure of opening the shapefile.

Sample Code

```
Private Sub OpenShapefile()
    Dim sf As New MapWinGIS.Shapefile()
    'Open the shapefile
    sf.Open("C:\test.shp")
    'Close the shapefile
    sf.Close()
End Sub
```

3.13.1.11 PointInShape

Checks to see if a point lies within the specified shape. This function only applies to polygon shapefiles.

VB.NET Usage

Function PointInShape(ShapeIndex As Integer, x As Double, y As Double) As Boolean

Parameters

ShapeIndex	The index of the shape to be tested.
x	The x coordinate in projected map units of the point to be tested.
y	The ycoordinate in projected map units of the point to be tested.
ReturnValue	A boolean value representing whether the point lies within the specified shape or not.

Sample Code

```
Private Sub PointInShape()
    Dim sf As New MapWinGIS.Shapefile()
    Dim x As Double, y As Double
    'Set the values for x and y in projected map coordinates
    x = 3000
    y = 2500
    'Check if the point lies within shape 0 in the shapefile
    If sf.PointInShape(0, x, y) Then
        'Display message in message box
        MsgBox("The point lies inside the specified shape.")
    Else
        'Display message in message box
        MsgBox("The point does not lie inside the specified shape.")
    End If
End Sub
```

3.13.1.12 PointInShapefile

Gets the index of the first shape the specified point lies within. If there are no shapes the point lies within, -1 is returned. Note: If you are making extensive use of this function for a shapefile, you may be able to improve the performance of your

application by using BeginPointInShapefile before your call(s) to PointInShapefile, and EndPointInShapefile after you are finished calling PointInShapefile. This loads all of the points for all of the shapefile's shapes into memory to speed up calculating whether a point lies within a shape or not.
See also BeginPointInShapefile and EndPointInShapefile

VB.NET Usage

Function PointInShapefile(x As Double, y As Double) As Integer

Parameters

x	The x coordinate in projected map coordinates of the point to be tested.
y	The y coordinate in projected map coordinates of the point to be tested.
ReturnValue	The shape index of the first shape the point lies within, or -1 if there are no shapes the point lies within.

Sample Code

```
Private Sub PointInShapefile()
    Dim sf As New MapWinGIS.Shapefile()
    Dim x As Double, y As Double
    Dim shapeindex As Integer
    'Set the values for x and y in projected map coordinates
    x = 3000
    y = 2500
    'Get the shape index of any shape in the shapefile that the point lies within, if any
    shapeindex = sf.PointInShapefile(x, y)
End Sub
```

3.13.1.13 QuickExtents

Gets the extents of the specified shape. Note: Use this function to get the extents of a shape in a shapefile when speed is essential.
See also Extents

VB.NET Usage

Function QuickExtents(ShapeIndex As Integer) As MapWinGIS.Extents

Parameters

ShapeIndex	The index of the shape for which the extents are required.
ReturnValue	The Extents object representing the extents of the specified shape in the shapefile.

Sample Code

```
Private Sub QuickExtents()
    Dim sf As New MapWinGIS.Shapefile()
    Dim extents As New MapWinGIS.Extents
    'Get the extents for shape 0 in the shapefile using the fastest method available
    extents = sf.QuickExtents(0)
    'Get the extents for shape 0 in the shapeifle using the slowest method avaiable
    extents = sf.Shape(0).Extents()
End Sub
```

3.13.1.14 QuickPoint

Gets a single point from the specified shape in the shapefile. Note: Use this function to get a point from a shape in the shapefile when speed is essential.
See also Point

VB.NET Usage

Function QuickPoint(ShapeIndex As Integer, PointIndex As Integer) As MapWinGIS.Point

Parameters

ShapeIndex	The index of the shape in the shapfefile to retrieve the point from.
PointIndex	The index of the point to be retrieveed from the specified shape in the shapefile.
ReturnValue	The point object representing the retrieved point.

Sample Code

```
Private Sub QuickPoint()
    Dim sf As New MapWinGIS.Shapefile()
    Dim point As New MapWinGIS.Point()
    'Get point 0 in shape 0 in the shapefile using the fastest method available
    point = sf.QuickPoint(0, 0)
    'Get point 0 in shape 0 in the shapefile using the slowest method available
    point = sf.Shape(0).Point(0)
End Sub
```

3.13.1.15 QuickPoints

Gets all of the points in the specified shape in the shapefile. Note: Use this method to get all of the specified points from a shape in the shapefile when speed is essential.

VB.NET Usage

Function QuickPoints(ShapeIndex As Integer, ByRef numPoints As Integer) As Double

Parameters

ShapeIndex	The index of the shape for which all points are required.
numPoints	Reference parameter. The number of points in the shape will be returned through this parameter.
ReturnValue	An array of all the points in the specified shape will be returned. The points are ordered as follows: (x1, y1, x2, y2, ... , xn-1, yn-1, xn, yn) n = numPoints.

Sample Code

```
Private Sub QuickPoints()
    Dim sf As New MapWinGIS.Shapefile()
    Dim point As New MapWinGIS.Point()
    Dim x() As Double, y() As Double, points() As Double
    Dim i As Integer, numpoints As Integer
    'Get the x and y coordinates of all points in shape 0 using the fastest method available
    points = sf.QuickPoints(0, numpoints)
    'Get the x and y coordinates of all points in shape 0 using the slowest method available
    For i = 1 To sf.Shape(0).numPoints
        'Get the current point in shape 0
        point = sf.Shape(0).Point(i)
        'Get the x and y coordinates of the current point
        x(i) = point.x
        y(i) = point.y
    Next
End Sub
```

3.13.1.16 SaveAs

Saves the shapefile using the specfied filename.

VB.NET Usage

Function SaveAs(ShapefileName As String, cBack As MapWinGIS.ICallback) As Boolean

Parameters

ShapefileName	The filename to use when saving the shapefile.
cBack	The ICallback object which will receive progress and error messages while the shapefile is being saved.
ReturnValue	A boolean value representing the success or failure of saving the shapfile.

Sample Code

```
Private Sub ShapefileSaveAs()
    Dim sf As New MapWinGIS.Shapefile()
    Dim success As Boolean
    'Save the shapefile
    success = sf.SaveAs("C:\test.shp", Me)
End Sub
```

3.13.1.17 SelectShapes

Gets all shapes in the shapefile that lie within the specified bounding box.
See also Extents and SelectMode

VB.NET Usage

Function SelectShapes(BoundBox As MapWinGIS.Extents, Optional Tolerance As Double, Optional SelectMode As MapWinGIS.SelectMode, Optional ByRef result As Object) As Boolean

Parameters

BoundBox	The bounding box used to determine which shapes in the shapefile to return as selected. The bounds must be in projected map units.
Tolerance	Optional. This is used to extend the boundaries of the specified bounding box to allow shapes not otherwise selected to be considered within the bounding box. The default value is 0.0.
SelectMode	Optional. Determines whether shapes must be completely enclosed by the bounding box or just intersected by the bounding box to be considered selected. The default is MapWinGIS.SelectMode.INTERSECTION.
result	Optional, reference parameter. An array of shape indexes of shapes in the shapefile that have been selected.
ReturnValue	A boolean value representing whether any shapes in the shapefile have been selected or not.

Sample Code

```
Private Sub ShapefileSelectShapes()
    Dim sf As New MapWinGIS.Shapefile()
    Dim extents As New MapWinGIS.Extents()
    Dim top As Integer, bottom As Integer, left As Integer, right As Integer, result() As Integer
    Dim success As Boolean
    'Set the value of the bounds to use for the bounding box in projected map coordinates
    top = 300
    bottom = 100
    left = 100
    right = 300
    'Set the extents using the bounding box coordinates
    extents.SetBounds(left, bottom, 0, right, top, 0)
    'Get the shapes that are completely contained in the bounding box in the result array
    success = sf.SelectShapes(extents, 0.0, MapWinGIS.SelectMode.INCLUSION, result)
End Sub
```

3.13.1.18 StartEditingShapes

Sets the shapefile to allow shapes to be edited. This allows existing shapes to be edited using the shapefile edit functions.
Note: To add or remove a shape from the shapefile, the attribute table must also be in editing mode.
See also StopEditingShapes

VB.NET Usage

Function StartEditingShapes(Optional StartEditTable As Boolean, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

StartEditTable	Optional. A boolean value representing whether the attribute table is to be set to editing mode to allow adding or removing shapes from the shapefile.
cBack	Optional. The ICallback object which will receive progress and error messages while the shapefile is being changed to editing mode.
ReturnValue	A boolean value representing the success or failure of changing the shapefile to editing mode.

Sample Code

```
Private Sub ShapefileEditShapes()
    Dim sf As New MapWinGIS.Shapefile()
    Dim success As Boolean
    'Set the shapefile to be in editing mode, also setting the attribute table to editing mode
    success = sf.StartEditingShapes(True, Me)
End Sub
```

3.13.1.19 StartEditingTable

Sets the shapefile to allow the attribute table to be edited. Note: The shapefile must also be put in shape editing mode to allow shapes to be added or removed from the shapefile.
See also StopEditingTable

VB.NET Usage

Function StartEditingTable(Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

cBack	Optional. The ICallback object which will receive progress and error messages while the attribute table is changed to editing mode.
ReturnValue	A boolean value representing the success or failure of changing the attribute table to editing mode.

Sample Code

```
Private Sub ShapefileEditTable()
    Dim sf As New MapWinGIS.Shapefile()
    Dim success As Boolean
    'Set the shapefile attribute table to be in editing mode
    success = sf.StartEditingTable(Me)
End Sub
```

3.13.1.20 StopEditingShapes

Sets the shapefile to prevent shapes from being edited. Note: It is recommended that the attribute table is also set to prevent editing when shapes are set to prevent editing.
See also StartEditingShapes

VB.NET Usage

Function StopEditingShapes(Optional ApplyChanges As Boolean, Optional StopEditTable As Boolean, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

ApplyChanges	Optional. A boolean value representing whether changes to shapes will be saved. The default is True.
StopEditingTable	Optional. A boolean value representing whether the attribute table is set to prevent editing or not. The default is True.
cBack	The ICallback object which will receive progress and error messages while the shapefile is set to stop editing shapes.
ReturnValue	A boolean value representing the success or failure of stopping editing shapes.

Sample Code

```
Private Sub ShapefileStopEditingShapes()
    Dim sf As New MapWinGIS.Shapefile()
    Dim success As Boolean
    'Stop editing shapes in the shapefile, saving changes to shapes, also stopping editing of the attribute table
    success = sf.StopEditingShapes(True, True, Me)
End Sub
```

3.13.1.21 StopEditingTable

Sets the shapefile to prevent the attribute table from being edited.
See also StartEditingTable

VB.NET Usage

Function StopEditingTable(Optional ApplyChanges As Boolean, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

ApplyChanges	Optional. A boolean value representing whether the changes to the attribute table are saved or not. The default is True.
cBack	Optional. The ICallback object which will receive progress and error messages while the attribute table is set to prevent editing.
ReturnValue	A boolean value representing the success or failure of setting the shapefile to prevent editing of

	the attribute table.

Sample Code

```
Private Sub ShapefileStopEditingTable()
    Dim sf As New MapWinGIS.Shapefile()
    Dim success As Boolean
    'Stop editing the shapefile attribute table, saving changes to the attribute table
    success = sf.StopEditingTable(True, Me)
End Sub
```

3.13.2 Subs

3.13.2.1 EndPointInShapefile

Unloads all points in all shapes in the shapefile from memory. These points are loaded into memory by the function BeginPointInShapefile to increase the performance of calling the function PointInShapefile for the shapefile.
See also BeginPointInShapefile and PointInShapefile

VB.NET Usage

Sub EndPointInShapefile()

Parameters

None	

Sample Code

```
Private Sub EndPtInSf()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shapeindex As Integer
    'Load all points in all shapes in the shapefile into memory
    sf.BeginPointInShapefile()
    'See if a point is in the shapefile, returning the index of the shape or -1 if point doesn't lie in shapefile
    shapeindex = sf.PointInShapefile(100, 200)
    'Unload all points loaded by BeginPointInShapefile
    sf.EndPointInShapefile()
End Sub
```

3.13.3 Properties

3.13.3.1 CdlgFilter

Returns the common dialog filter containing all supported file extensions in string format.

VB.NET Usage

ReadOnly Property CdlgFilter() As String

Parameters

ReturnValue	The filter containing all file extensions supported by MapWinGIS.

Sample Code

```
Private Sub CdlgFilter()
    Dim tin As New MapWinGIS.Tin()
    'Open a tin from disk
    tin.Open("C:\test.tin")
    'Display the supported file formats in a message box
    MsgBox(tin.CdlgFilter)
    'Close the tin
    tin.Close()
End Sub
```

3.13.3.2 CellValue

Gets a value from the specified cell in the shapefile attribute table.

VB.NET Usage

ReadOnly Property CellValue(FieldIndex As Integer, ShapeIndex As Integer) As Object

Parameters

FieldIndex	The index of the field for which the specified shape index value is required in the attribute table.
ShapeIndex	The shape index of the shape for which the value is required in the attribute table.
ReturnValue	The value of the specified cell.

Sample Code

```
Private Sub ShapefileCellValue()
    Dim sf As New MapWinGIS.Shapefile()
    Dim value As Boolean
    'Get the value for the in field 0 for shape 0 in the shapefile attribute table
    value = sf.CellValue(0, 0)
End Sub
```

3.13.3.3 EditingShapes

Gets whether or not the shapefile is in shape editing mode.

VB.NET Usage

ReadOnly Property EditingShapes() As Boolean

Parameters

ReturnValue	A boolean value representing whether the shapefile is in the mode to allow shapes to be edited.

Sample Code

```
Private Sub ShapefileEditingShapes()
    Dim sf As New MapWinGIS.Shapefile()
    'Check if the shapefile is in the mode to allow shapes to be edited
    If sf.EditingShapes Then
        'Delete point 5 in shape 0 in the shapefile
        sf.Shape(0).DeletePoint(5)
    End If
End Sub
```

3.13.3.4 EditingTable

Gets whether or not the shapefile attribute table is set to allow editing.

VB.NET Usage

ReadOnly Property EditingTable() As Boolean

Parameters

ReturnValue	A boolean value representing whether or not the shapefile attribute table is set to allow editing.

Sample Code

```
Private Sub ShapefileEditingTable()
    Dim sf As New MapWinGIS.Shapefile()
    Dim success As Boolean
    'Check if the shapefile is in the mode to allow the attribute table to be edited
    If sf.EditingTable Then
        'Set the value of the field 0, shape 0 cell to 100
        success = sf.EditCellValue(0, 0, 100)
    End If
End Sub
```

3.13.3.5 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.13.3.6 Extents

Gets the extents of the shapefile.
See also Extents

VB.NET Usage

ReadOnly Property Extents() As MapWinGIS.Extents

Parameters

ReturnValue	The extents of the shapefile.

Sample Code

```
Private Sub ShapefileExtents()
    Dim sf As New MapWinGIS.Shapefile()
    Dim extents As New MapWinGIS.Extents()
    'Get the extents of the shapefile
    extents = sf.Extents
End Sub
```

3.13.3.7 Field

Gets a field header from the shapefile's attribute table.
See also Field

VB.NET Usage

ReadOnly Property Field(FieldIndex As Integer) As MapWinGIS.Field

Parameters

FieldIndex	The field index of the field for which the header is required.
ReturnValue	The field header for the specified field in the attribute table.

Sample Code

```
Private Sub ShapefileField()
    Dim sf As New MapWinGIS.Shapefile()
    Dim field As New MapWinGIS.Field()
    'Get the field header for field 0
    field = sf.Field(0)
End Sub
```

3.13.3.8 Filename

The filename associated with the object.

VB.NET Usage

ReadOnly Property Filename() As String

Parameters

ReturnValue	The filename associated with the object.

Sample Code

```
Private Sub Filename()
    Dim shapefile As New MapWinGIS.Shapefile()
    Dim filename As String
    'Open a shapefile from disk
    shapefile.Open("C:\test.shp")
    'Get the filename of the shapefile
    filename = shapefile.Filename
End Sub
```

3.13.3.9 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
      '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
              'Display the error message in a label
             LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
             'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
             'Display the message in a label
         Label2.Text = Message
     End Sub
 #End Region
              '...
```

3.13.3.10 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
    If k = "Map1" Then
        'Set the map's key to "My Map1"
        Map1.Key = "My Map1"
    Else
        'Set the map's key to "Map1"
        Map1.Key = "Map1"
    End If
End Sub
```

3.13.3.11 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.13.3.12 NumFields

Gets the number of fields in the shapefile attribute table.

VB.NET Usage

ReadOnly Property NumFields() As Integer

Parameters

ReturnValue	The number of fields in the shapefile attribute table.

Sample Code

```
Private Sub ShapefileNumfields()
    Dim sf As New MapWinGIS.Shapefile()
    Dim count As Integer
    'Get the number of fields in the shapefile attribute table
    count = sf.NumFields
End Sub
```

3.13.3.13 NumShapes

Gets the number of shapes in the shapefile.

VB.NET Usage

ReadOnly Property NumShapes() As Integer

Parameters

ReturnValue	The number of shapes in the shapefile.

Sample Code

```
Private Sub ShapefileNumShapes()
    Dim sf As New MapWinGIS.Shapefile()
    Dim count As Integer
    'Get the number of shapes in the shapefile
    count = sf.NumShapes
End Sub
```

3.13.3.14 Shape

Gets the shape in the shapefile specified by the index of the shape. Note: Null will be returned if an invalid shape index is specified

See also Shape

VB.NET Usage

ReadOnly Property Shape(ShapeIndex As Integer) As MapWinGIS.Shape

Parameters

ShapeIndex	The index of the shape to be returned.
ReturnValue	The shape in the shapefile specified by the shape index.

Sample Code

```
Private Sub ShapefileShape()
    Dim sf As New MapWinGIS.Shapefile()
    Dim shape As New MapWinGIS.Shape()
    'Get shape 0 in the shapefile
    shape = sf.Shape(0)
End Sub
```

3.13.3.15 ShapefileType

Gets the type of the shapefile. All shapes contained in a shapefile must have the same shapefile type as the shapefile except for shapes of type SHP_NULLSHAPE.
See also ShpfileType

VB.NET Usage

ReadOnly Property ShapefileType() As MapWinGIS.ShpfileType

Parameters

ReturnValue	The type of the shapefile.

Sample Code

```
Private Sub ShapefileType()
    Dim sf As New MapWinGIS.Shapefile()
    Dim sf_type As New MapWinGIS.ShpfileType()
    'Get the shapefile type of the shapefile
    sf_type = sf.ShapefileType
End Sub
```

3.14 ShapefileColorBreak

A shapefile color break object defines how a specified region of a shapefile will be colored using the the shapefile color scheme containing the shapefile color break.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.14.1 Properties

3.14.1.1 Caption

Gets or sets the caption for the shapefile color break.

VB.NET Usage

Property Caption() As String

Parameters

ReturnValue	The caption for the shapefile color break.

Sample Code

```
Private Sub ShapefileColorBreakCaption()
    Dim colorbreak As New MapWinGIS.ShapefileColorBreak()
    Dim caption As String
    'Set the string to be used as the caption
    caption = "test color break"
    'Set the color break caption
    colorbreak.Caption = caption
    'Get the color break caption
```

```
    caption = colorbreak.Caption
End Sub
```

3.14.1.2 EndColor

Gets or sets the end color of the shapefile color break.

VB.NET Usage

Property EndColor() As System.UInt32

Parameters

ReturnValue	The end color for the shapefile color break. This is a UInt32 representation of an RGB value.

Sample Code

```
Private Sub ShapefileColorBreakEndColor()
    Dim colorbreak As New MapWinGIS.ShapefileColorBreak()
    Dim color As System.UInt32
    'Set the color for the end color
    color = System.Convert.ToUInt32(RGB(255, 0, 0))
    'Set the end color to red for the shapefile color break
    colorbreak.EndColor = color
    'Get the end color for the shapefile color break
    color = colorbreak.EndColor
End Sub
```

3.14.1.3 EndValue

Gets or sets the value representing the end of the shapefile color break.

VB.NET Usage

Property EndValue() AsObject

Parameters

ReturnValue	The value representing the end of this color break.

Sample Code

```
Private Sub ShapefileColorBreakEndValue()
    Dim colorbreak As New MapWinGIS.ShapefileColorBreak()
    Dim endvalue As Double
    'Set the value to use as the end value
    endvalue = 3000
    'Set the endvalue for the shapefile color break
    colorbreak.EndValue = endvalue
    'Get the endvalue for the shapefile color break
    endvalue = colorbreak.EndValue
End Sub
```

3.14.1.4 StartColor

Gets or sets the start color or the shapefile color break.

VB.NET Usage

Property StartColor() As System.UInt32

Parameters

ReturnValue	The start color for the shapefile color break. This is a UInt32 representation of an RGB value.

Sample Code

```
Private Sub ShapefileColorBreakStartColor()
    Dim colorbreak As New MapWinGIS.ShapefileColorBreak()
    Dim color As System.UInt32
    'Set the color for the start color
    color = System.Convert.ToUInt32(RGB(0, 0, 255))
    'Set the start color to blue for the shapefile color break
    colorbreak.StartColor = color
```

```
    'Get the start color for the shapefile color break
    color = colorbreak.StartColor
End Sub
```

3.14.1.5 StartValue

Gets or sets the value representing the start of the shapefile color break.

VB.NET Usage

Property StartValue() As Object

Parameters

ReturnValue	The value representing the start of the color break.

Sample Code

```
Private Sub ShapefileColorBreakStartValue()
    Dim colorbreak As New MapWinGIS.ShapefileColorBreak()
    Dim startvalue As Double
    'Set the value to use as the start value
    startvalue = 1000
    'Set the startvalue for the shapefile color break
    colorbreak.StartValue = startvalue
    'Get the startvalue for the shapefile color break
    startvalue = colorbreak.StartValue
End Sub
```

3.15 ShapefileColorScheme

A shapefile color scheme defines how a shapefile will be colored. A shapefile color scheme contains shapefile color breaks which define how specified regions within the shapefile are colored.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.15.1 Functions

3.15.1.1 Add

Adds a color break to the shapefile color scheme.
See also ShapefileColorBreak

VB.NET Usage

Function Add(Break As MapWinGIS.ShapefileColorBreak) As Integer

Parameters

Break	The shapefile color break to be added to the shapefile color scheme.
ReturnValue	The index of the color break just added to the color scheme.

Sample Code

```
Private Sub AddShapefileColorBreak()
    Dim colorscheme As New MapWinGIS.ShapefileColorScheme()
    Dim colorbreak As New MapWinGIS.ShapefileColorBreak()
    Dim index As Integer
    'Add a color break to the shapefile color scheme, saving the color break index
    index = colorscheme.Add(colorbreak)
End Sub
```

3.15.1.2 NumBreaks

Gets the number of color breaks in the shapefile color scheme.

VB.NET Usage

Function NumBreaks() As Integer

Parameters

ReturnValue	The number of color breaks in the shapefile color scheme.

Sample Code

```
Private Sub ShapefileColorSchemeBreakCount()
    Dim colorscheme As New MapWinGIS.ShapefileColorScheme()
    Dim count As Integer
    'Get the number of color breaks in the color scheme
    count = colorscheme.NumBreaks
End Sub
```

3.15.2 Subs

3.15.2.1 Remove

Removes the color break specified by the color break index from the shapefile color scheme.

VB.NET Usage

Sub Remove(Index As Integer)

Parameters

Index	The index of the color break to be removed from the shapefile color scheme.

Sample Code

```
Private Sub RemoveShapefileColorBreak()
    Dim colorscheme As New MapWinGIS.ShapefileColorScheme()
    Dim colorbreak As New MapWinGIS.ShapefileColorBreak()
    Dim index As Integer
    'Add a color break to the shapefile color scheme, saving the color break index
    index = colorscheme.Add(colorbreak)
    'Remove the color break just added to the shapefile color scheme
    colorscheme.Remove(index)
End Sub
```

3.15.3 Properties

3.15.3.1 ColorBreak

Gets or sets a shapefile color break in the shapefile color scheme.
See also ShapefileColorBreak

VB.NET Usage

Property ColorBreak(Index As Integer) As MapWinGIS.ShapefileColorBreak

Parameters

Index	The index of the color break to be returned or set.
ReturnValue	The color break specified by the color break index.

Sample Code

```
Private Sub ColorSchemeColorBreak()
    Dim colorscheme As New MapWinGIS.ShapefileColorScheme()
    Dim colorbreak As New MapWinGIS.ShapefileColorBreak()
    'Set color break 0 in the color scheme
    colorscheme.ColorBreak(0) = colorbreak
    'Get color break 0 in the color scheme
    colorbreak = colorscheme.ColorBreak(0)
End Sub
```

3.15.3.2 ErrorMsg
Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.15.3.3 FieldIndex
Gets or sets the field index in the attribute table the color scheme is associated with.

VB.NET Usage

Property FieldIndex() As Integer

Parameters

ReturnValue	The field index the color scheme is associated with.

Sample Code

```
Private Sub ColorSchemeFieldIndex()
    Dim colorscheme As New MapWinGIS.ShapefileColorScheme()
    Dim fieldindex As Integer
    'Set the value for the field index
    fieldindex = 0
    'Set the field index for the color scheme
    colorscheme.FieldIndex = fieldindex
    'Get the filed index for the color scheme
    fieldindex = colorscheme.FieldIndex
End Sub
```

3.15.3.4 GlobalCallback
The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
        '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
                'Display the error message in a label
               LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
```

```
            'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
            'Display the message in a label
         Label2.Text = Message
    End Sub
#End Region
            '...
```

3.15.3.5 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
    If k = "Map1" Then
        'Set the map's key to "My Map1"
        Map1.Key = "My Map1"
    Else
        'Set the map's key to "Map1"
        Map1.Key = "Map1"
    End If
End Sub
```

3.15.3.6 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.15.3.7 LayerHandle

Gets or sets the layer handle the color scheme is associated with.

VB.NET Usage

Property LayerHandle() As Integer

Parameters

ReturnValue	The layer handle of the layer the color scheme is associated with.

Sample Code

```
Private Sub ColorSchemeLayerHandle()
```

```
    Dim colorscheme As New MapWinGIS.ShapefileColorScheme()
    Dim layerhandle As Integer
    'Set the layer handle of the layer to associate the color scheme with
    layerhandle = 0
    'Set the color scheme layerhandle
    colorscheme.LayerHandle = layerhandle
    'Get the color scheme layerhandle
    layerhandle = colorscheme.LayerHandle
End Sub
```

3.16 Table

The table object is used to store information from the dbf file associated with a shapefile.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.16.1 Functions

3.16.1.1 Close

Closes the attribute table.

VB.NET Usage

Function Close() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of closing the attribute table.

Sample Code

```
Private Sub CloseTable()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Create a new table using the specified filename
    success = table.CreateNew("c:\test.dbf")
    'Close the table
    success = table.Close
End Sub
```

3.16.1.2 CreateNew

Creates a new attribute table. Note: A new table is automatically in editing mode after it is created.

VB.NET Usage

Function CreateNew(dbfFilename As String) As Boolean

Parameters

dbfFilename	The filename for the new table.
ReturnValue	A boolean value representing the success or failure of creating the new table.

Sample Code

```
Private Sub CreateTable()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Create a new table using the specified filename
    success = table.CreateNew("c:\test.dbf")
End Sub
```

3.16.1.3 EditCellValue

Sets the value of the cell. Note: The table must be set to allow editing before a cell's value may be edited.
See also StartEditingTable

VB.NET Usage

Function EditCellValue(FieldIndex As Integer, RowIndex As Integer, NewVal As Object) As Boolean

Parameters

FieldIndex	The field index of the cell to be edited.
RowIndex	The row index of the cell to be edited.
NewVal	The new value to be used to set the specified cell's value.
ReturnValue	A boolean value representing the success or failure of setting the value of the specified cell in the table.

Sample Code

```
Private Sub TableEditCellValue()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Open a table
    success = table.Open("C:\test.dbf")
    'Start editing the table
    success = table.StartEditingTable()
    'Set the value of field 0, row 0 to 100
    success = table.EditCellValue(0, 0, 100)
End Sub
```

3.16.1.4 EditClear

Deletes all rows and fields from the table. Note: The table must be set to allow editing before the rows and fields can be deleted from the table.
See also StartEditingTable

VB.NET Usage

Function EditClear() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of deleting all rows and fields from the table.

Sample Code

```
Private Sub TableClear()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Open a table
    success = table.Open("C:\test.dbf")
    'Start editing the table
    success = table.StartEditingTable()
    'Delete all rows and fields from the table
    success = table.EditClear()
End Sub
```

3.16.1.5 EditDeleteField

Deletes a field from the table. Note: The table must be set to allow editing before a field can be deleted from the table.
See also StartEditingTable

VB.NET Usage

Function EditDeleteField(FieldIndex As Integer, cBack As MapWinGIS.ICallback) As Boolean

Parameters

FieldIndex	The index of the field to be deleted from the table.
cBack	The ICallback object which will receive progress and error messages while the field is being deleted from the table.
ReturnValue	A boolean value representing the success or failure of deleting the field from the table.

Sample Code

```
Private Sub TableDeleteField()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
```

```
    'Open a table
    success = table.Open("C:\test.dbf")
    'Start editing the table
    success = table.StartEditingTable()
    'Delete field 0 from the table
    success = table.EditDeleteField(0)
End Sub
```

3.16.1.6 EditDeleteRow

Deletes a row from the table. Note: The table must be set to allow editing before a row can be deleted from the table.
See also StartEditingTable

VB.NET Usage

Function EditDeleteRow(RowIndex As Integer) As Boolean

Parameters

RowIndex	The index of the row to be deleted from the table.
ReturnValue	A boolean value representing the success or failure of deleting the specified row from the table.

Sample Code

```
Private Sub TableDeleteRow()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Open a table
    success = table.Open("C:\test.dbf")
    'Start editing the table
    success = table.StartEditingTable()
    'Delete row 0 from the table
    success = table.EditDeleteRow(0)
End Sub
```

3.16.1.7 EditInsertField

Inserts a new field into the table. Note: The table must be set to allow editing before a field can be inserted into the table.
See also Field and StartEditingTable

VB.NET Usage

Function EditInsertField(Field As MapWinGIS.Field, ByRef FieldIndex As Integer, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

Field	The new field to be inserted into the table.
FieldIndex	Reference parameter. The desired index to be used for the new field being inserted into the table. If the desired index is invalid or unavailable, the actual index used for the new field will be returned.
cBack	The ICallback object which will receive progress and error messages while the new field is being inserted into the table.
ReturnValue	A boolean value representing the success or failure of inserting the new field into the table.

Sample Code

```
Private Sub TableInsertField()
    Dim table As New MapWinGIS.Table()
    Dim field As New MapWinGIS.Field()
    Dim fieldindex As Integer
    Dim success As Boolean
    'Set the desired index for the new field in the table
    fieldindex = 2
    'Open a table
    success = table.Open("C:\test.dbf")
    'Start editing the table
    success = table.StartEditingTable()
    'Insert a field into the table at index 2
    success = table.EditInsertField(field, fieldindex, Me)
End Sub
```

3.16.1.8 EditInsertRow

Inserts a new row into the table. Note: The table must be set to allow editing before a row can be inserted into the table.
See also StartEditingTable

VB.NET Usage

Function EditInsertRow(ByRef RowIndex As Integer) As Boolean

Parameters

RowIndex	Reference parameter. The desired index to use when inserting the new row into the table. If the desired index is invalid or unavailable, the actual index used to insert the new row will be returned.
ReturnValue	A boolean value representing the success or failure of inserting the new row into the table.

Sample Code

```
Private Sub TableInsertRow()
    Dim table As New MapWinGIS.Table()
    Dim rowindex As Integer
    Dim success As Boolean
    'Set the desired index for the new row in the table
    rowindex = 2
    'Open a table
    success = table.Open("C:\test.dbf")
    'Start editing the table
    success = table.StartEditingTable()
    'Insert a row into the table at index 2
    success = table.EditInsertRow(rowindex)
End Sub
```

3.16.1.9 EditReplaceField

Replaces the specified field in the table with the new field. Note: The table must be set to allow editing before a field can be replaced in the table.
See also Field and StartEditingTable

VB.NET Usage

Function EditReplaceField(FieldIndex As Integer, NewField As MapWinGIS.Field, cBack As MapWinGIS.ICallback) As Boolean

Parameters

FieldIndex	The index of the field to be replaced.
NewField	The field to be used to replace the specified field in the table.
cBack	The ICallback object which will receive progress and error messages while the specified field is being replaced by the new field.
ReturnValue	A boolean value representing the success or failure of replacing the specified field.

Sample Code

```
Private Sub TableReplaceField()
    Dim table As New MapWinGIS.Table()
    Dim field As New MapWinGIS.Field()
    Dim success As Boolean
    'Open a table
    success = table.Open("C:\test.dbf")
    'Start editing the table
    success = table.StartEditingTable()
    'Replace field 0 with the new field
    success = table.EditReplaceField(0, field, Me)
End Sub
```

3.16.1.10 Open

Opens a .dbf table from file.

VB.NET Usage

Function Open(dbfFilename As String, cBack As MapWinGIS.ICalback) As Boolean

Parameters

dbfFilename	The filename of the table to be opened.
cBack	The ICallback object which will receive progress and error messages while the table is being opened.
ReturnValue	A boolean value representing the success or failure of opening the table.

Sample Code

```
Private Sub OpenTable()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Open a table
    success = table.Open("C:\test.dbf")
End Sub
```

3.16.1.11 SaveAs

Saves the table using the specified filename.

VB.NET Usage

Function SaveAs(dbfFilename As String, cBack As MapWinGIS.ICallback) As Boolean

Parameters

dbfFilename	The filename to be used to save the table.
cBack	The ICallback object which will receive progress and error messages while the table is being saved.
ReturnValue	A boolean value representing the success or failure of saving the table.

Sample Code

```
Private Sub SaveTableAs()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Save the table under the given filename
    success = table.SaveAs("C:\test.dbf", Me)
End Sub
```

3.16.1.12 StartEditingTable

Sets the table to allow table editing.
See also StopEditingTable

VB.NET Usage

Function StartEditingTable(Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

cBack	Optional. The ICallback object which will receive progress and error events while the table is being set to allow editing.
ReturnValue	A boolean value representing the success or failure of setting the table to allow editing.

Sample Code

```
Private Sub StartEditingTable()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Open a table
    success = table.Open("C:\test.dbf")
    'Start editing the table
    success = table.StartEditingTable()
End Sub
```

3.16.1.13 StopEditingTable

Sets the table to prevent editing.
See also StartEditingTable

VB.NET Usage

Function StopEditingTable(Optional ApplyChanges As Boolean, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

ApplyChanges	Optional. A boolean value representing whether or not to save changes to the table. The default is True, to save the changes.
cBack	Optional. The ICallback object which will receive progress and error messages while the table is being set to prevent editing.
ReturnValue	A boolean value representing the success or failure of setting the table to prevent editing.

Sample Code

```
Private Sub StopEditingTable()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Create a new table using the specified filename
    success = table.CreateNew("c:\test.dbf")
    'Stop editing the table, saving the changes
    success = table.StopEditingTable(True, Me)
End Sub
```

3.16.2 Properties

3.16.2.1 CdlgFilter

Returns the common dialog filter containing all supported file extensions in string format.

VB.NET Usage

ReadOnly Property CdlgFilter() As String

Parameters

ReturnValue	The filter containing all file extensions supported by MapWinGIS.

Sample Code

```
Private Sub CdlgFilter()
    Dim tin As New MapWinGIS.Tin()
    'Open a tin from disk
    tin.Open("C:\test.tin")
    'Display the supported file formats in a message box
    MsgBox(tin.CdlgFilter)
    'Close the tin
    tin.Close()
End Sub
```

3.16.2.2 CellValue

Gets the value of the specified cell in the table.

VB.NET Usage

ReadOnly Property CellValue(FieldIndex As Integer, RowIndex As Integer) As Object

Parameters

FieldIndex	The field index of the cell for which the value is required.
RowIndex	The row index of the cell for which the value is required.
ReturnValue	The value of the specified cell in the table.

Sample Code

```
Private Sub TableCellValue()
    Dim table As New MapWinGIS.Table()
    Dim value As Double
    'Get the value of field 0, row 0 in the table to 100
    value = table.CellValue(0, 0)
End Sub
```

3.16.2.3 EditingTable

Gets whether or not the table is in editing mode.

VB.NET Usage

ReadOnly Property EditingTable() As Boolean

Parameters

ReturnValue	A boolean value representing whether or not the table is set to allow editing.

Sample Code

```
Private Sub EditingTable()
    Dim table As New MapWinGIS.Table()
    Dim success As Boolean
    'Check if the table is in editing mode or not
    If table.EditingTable Then
        'If the table is in editing mode, set the value of field 0, row 0 to 100
        success = table.EditCellValue(0, 0, 100)
    End If
End Sub
```

3.16.2.4 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.16.2.5 Field

Gets the field object at the specified field index in the table.
See also Field

VB.NET Usage

ReadOnly Property Field(FieldIndex As Integer) As MapWinGIS.Field

Parameters

FieldIndex	The index of the field in the table to be returned.
ReturnValue	The field object specified by the field index.

Sample Code

```
Private Sub TableField()
    Dim table As New MapWinGIS.Table()
```

```
    Dim field As New MapWinGIS.Field()
    'Get field 0 in the table
    field = table.Field(0)
End Sub
```

3.16.2.6 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
      '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
              'Display the error message in a label
             LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
             'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
             'Display the message in a label
         Label2.Text = Message
     End Sub
 #End Region
              '...
```

3.16.2.7 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
    Private Sub MapKey()
        Dim k As String
        'Get the map's key
        k = Map1.Key
        'Check if the map's key is "Map1"
        If k = "Map1" Then
            'Set the map's key to "My Map1"
            Map1.Key = "My Map1"
        Else
            'Set the map's key to "Map1"
            Map1.Key = "Map1"
        End If
    End Sub
```

3.16.2.8 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.16.2.9 NumFields

Gets the number of fields in the table.

VB.NET Usage

ReadOnly Property NumFields() As Integer

Parameters

ReturnValue	The number of fields in the table.

Sample Code

```
Private Sub TableNumFields()
    Dim table As New MapWinGIS.Table()
    Dim count As Integer
    'Get the number of fields in the table
    count = table.NumFields
End Sub
```

3.16.2.10 NumRows

Gets the number of rows in the table.

VB.NET Usage

ReadOnly Property NumRows() As Integer

Parameters

ReturnValue	The number of rows in the table.

Sample Code

```
Private Sub TableNumRows()
    Dim table As New MapWinGIS.Table()
    Dim count As Integer
    'Get the number of rows in the table
    count = table.NumRows
End Sub
```

3.17 Tin

A tin object represents a tin which may be displayed in the map. A tin may be created from a grid, or opened from file.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.17.1 Functions

3.17.1.1 Close

Closes the TIN.

VB.NET Usage

Function Close() As Boolean

Parameters

ReturnValue	A boolean value representing the success or failure of closing the TIN.

Sample Code

```
Private Sub CloseTIN()
    Dim tin As New MapWinGIS.Tin()
    Dim success As Boolean
    'Open a TIN file
    success = tin.Open("C:\test.tin", Me)
    'Close a TIN file
    success = tin.Close
End Sub
```

3.17.1.2 CreateNew

Creates a new TIN object from the specified grid.
See also Grid and SplitMethod

VB.NET Usage

Function CreateNew(Grid As MapWinGIS.Grid, Deviation As Double, SplitTest As MapWinGIS.SplitMethod, STParam As Double, MeshDivisions As Integer, Optional MaximumTriangles As Integer, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

Grid	The grid to be used to create the new TIN.
Deviation	If the distance between the grid elevation and the triangle surface elevation at any given point is greater than this value, the triangle will be split at this location. This value is in projected map coordinates.
SplitTest	The method to use when splitting triangles.
STParam	Split Test Parameter. This value depends on the SplitMethod specified. It will either be the smallest inscribed radius allowed measured in projected map coordinates, or it will be the smallest angle allowed measured in degrees.
MeshDivisions	This is the number of dividers used to create an initial mesh for the algorithm. Each cell in the initial mesh is subdivided into smaller triangles depending on the deviation within the cell.
MaximumTriangles	Optional. The maximum number of triangles allowed in the TIN. The default value is 1073741824.
cBack	Optional. The ICallback object which will receive progress and error messages while the TIN is being created.
ReturnValue	A boolean value representing the success or failure of creating the new TIN.

Sample Code

```
Private Sub CreateTIN()
    Dim tin As New MapWinGIS.Tin()
    Dim grid As New MapWinGIS.Grid()
    Dim deviation As Double, stparam As Double
    Dim divisions As Integer, maxtriangles As Integer
    Dim success As Boolean
    'Set the maximum deviation between grid elevation and triangle elevation before triangle is split
    deviation = 100
    'Set the minimum allowed radius of an inscribed circle in a triangle in the TIN
    stparam = 300
    'Set the number of dividers to use to split the grid area in to an initial mesh for the TIN
    divisions = 5
    'Set the maximum number of triangles that can be created in the TIN
    maxtriangles = 1073741824
    'Open the grid to be used to create the TIN
    success = grid.Open("C:\test.bgd")
    'Create a new tin
```

```
    success = tin.CreateNew(grid, deviation, MapWinGIS.SplitMethod.InscribedRadius, stparam, divisions, maxtriangles,
Me)
  End Sub
```

3.17.1.3 Open

Opens a TIN from the specified file.

VB.NET Usage

Function Open(TinFile As String, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

TinFile	The filename of the TIN to be opened.
cBack	Optional. The ICallback object which will receive progress and error messages while the TIN is being opened.
ReturnValue	A boolean value representing the success or failure of opening the TIN.

Sample Code

```
  Private Sub OpenTIN()
    Dim tin As New MapWinGIS.Tin()
    Dim success As Boolean
    'Open a TIN file
    success = tin.Open("C:\test.tin", Me)
  End Sub
```

3.17.1.4 Save

Saves the TIN object under the specified filename.

VB.NET Usage

Function Save(TinFilename As String, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

TinFilename	The filename to be used to save the TIN.
cBack	Optional. The ICallback object which will receive progress and error messages while the TIN is saved.
ReturnValue	A boolean value representing the success or failure of saving the TIN.

Sample Code

```
  Private Sub SaveTIN()
    Dim tin As New MapWinGIS.Tin()
    Dim success As Boolean
    'Open a TIN file
    success = tin.Open("C:\test.tin", Me)
    'Save a TIN file
    success = tin.Save("C:\test2.tin", Me)
  End Sub
```

3.17.1.5 Select

Checks if there is a triangle in the TIN at the specified x and y coordinate. The index of any triangle at that location is returned through TriangleHint, and the z coordinate for the triangle surface at the specified location is returned through Z.

VB.NET Usage

Function Select(ByRef TriangleHint As Integer, x As Double, y As Double, ByRef Z As Double) As Boolean

Parameters

TriangleHint	Reference parameter. The index of the triangle to start the search with. If the triangle hint is a good guess, the search can be much faster.
x	The x projected map coordinate of the point that is being used to test if it lies within a TIN triangle.
y	The y projected map coordinate of the point that is being used to test if it lies within a TIN triangle.

Z	Reference parameter. The z projected map coordinate of the selected triangle's surface will be returned through this parameter.
ReturnValue	A boolean value representing whether a triangle in the TIN contained the specifed point.

Sample Code

```
Private Sub SelectTIN()
    Dim tin As New MapWinGIS.Tin()
    Dim trianglehint As Integer
    Dim x As Double, y As Double, z As Double
    Dim success As Boolean
    'Set the index of the triangle hint
    trianglehint = 0
    'Set the projected map coordinates of x and y
    x = 1422051.92226415
    y = 2093405.51962264
    'Open a TIN file
    success = tin.Open("C:\test.tin", Me)
    'See if there is a triangle at the specified x and y
    success = tin.Select(trianglehint, x, y, z)
    'Check to see if a triangle contained the point (x,y)
    If success Then
        'If there was triangle selected, display the triangle's index and the z value of its surface
        MsgBox(" trianglehint:" + Str(trianglehint) + " z:" + Str(z))
    Else
        'Display a failure message if there was not a triangle selected by the point (x,y)
        MsgBox("No triangle contained the point (x,y)")
    End If
End Sub
```

3.17.2 Subs

3.17.2.1 Max

Gets the maximum x, y, and z values of the TIN's extents.

VB.NET Usage

Sub Max(ByRef x As Double, ByRef y As Double, ByRef z As Double)

Parameters

x	Reference parameter. Returns the maximum x value of the TIN's extents.
y	Reference parameter. Returns the maximum y value of the TIN's extents.
Z	Reference parameter. Returns the maximum Z value of the TIN's extents.

Sample Code

```
Private Sub TINMax()
    Dim tin As New MapWinGIS.Tin()
    Dim x As Double, y As Double, Z As Double
    'Get the maximum values for x, y, and z in the TIN's extents
    tin.Max(x, y, Z)
End Sub
```

3.17.2.2 Min

Gets the minimum x, y, and z values of the TIN's extents.

VB.NET Usage

Sub Min(ByRef x As Double, ByRef y As Double, ByRef Z As Double)

Parameters

x	Reference parameter. Returns the minimum x value of the TIN's extents.
y	Reference parameter. Returns the minimum y value of the TIN's extents.
Z	Reference parameter. Returns the minimum Z value of the TIN's extents.

Sample Code

```
Private Sub TINMin()
    Dim tin As New MapWinGIS.Tin()
    Dim x As Double, y As Double, Z As Double
    'Get the minimum values for x, y, and z in the TIN's extents
    tin.Min(x, y, Z)
End Sub
```

3.17.2.3 Triangle

Gets the vertex indices of the specified triangle in the TIN.

VB.NET Usage

Sub Triangle(TriIndex As Integer, ByRef vtx1Index As Integer, ByRef vtx2Index As Integer, vtx3Index As Integer)

Parameters

TriIndex	The index of the triangle in the TIN for which the vertex indices are required.
vtx1Index	Reference parameter. The index of the first vertex in the specified triangle in the TIN.
vtx2Index	Reference parameter. The index of the second vertex in the specified triangle in the TIN.
vtx3Index	Reference parameter. The index of the third vertex in the specified triangle in the TIN.

Sample Code

```
Private Sub TIN_Triangle()
    Dim tin As New MapWinGIS.Tin()
    Dim vertex1index As Integer, vertex2index As Integer, vertex3index As Integer
    'Get the vertex indices of triangle 0 in the TIN
    tin.Triangle(0, vertex1index, vertex2index, vertex3index)
End Sub
```

3.17.2.4 TriangleNeighbors

Gets the indices of the specified triangle's neighboring triangles in the TIN.

VB.NET Usage

Sub TriangleNeighbors(TriIndex As Integer, ByRef triIndex1 As Integer, ByRef triIndex2 As Integer, ByRef triIndex3 As Integer)

Parameters

TriIndex	The index of the triangle to get the neighbors of in the TIN.
triIndex1	Reference parameter. The index of the first neighbor the specified triangle.
triIndex2	Reference parameter. The index of the second neighbor the specified triangle.
triIndex3	Reference parameter. The index of the third neighbor the specified triangle..

Sample Code

```
Private Sub TIN_TriangleNeighbors()
    Dim tin As New MapWinGIS.Tin()
    Dim neighbor1 As Integer, neighbor2 As Integer, neighbor3 As Integer
    'Get the index of each of triangle 0's neighbors in the TIN
    tin.TriangleNeighbors(0, neighbor1, neighbor2, neighbor3)
End Sub
```

3.17.2.5 Vertex

Gets the x, y, and Z projected map coordinates of the specified vertex in the TIN.

VB.NET Usage

Sub Vertex(VtxIndex As Integer, ByRef x As Double, ByRef y As Double, ByRef Z As Double)

Parameters

VtxIndex	The index of the vertex for which the x, y, and Z coordinates are required.
x	Reference parameter. Returns the x projected map coordinate of the specified vertex in the TIN.
y	Reference parameter. Returns the x projected map coordinate of the specified vertex in the TIN.

Z	Reference parameter. Returns the x projected map coordinate of the specified vertex in the TIN.

Sample Code

```
Private Sub TIN_Vertex()
    Dim tin As New MapWinGIS.Tin()
    Dim x As Double, y As Double, Z As Double
    'Get the x, y, and Z coordinates of vertex 0 in the TIN
    tin.Vertex(0, x, y, Z)
End Sub
```

3.17.3 Properties

3.17.3.1 CdlgFilter

Returns the common dialog filter containing all supported file extensions in string format.

VB.NET Usage

ReadOnly Property CdlgFilter() As String

Parameters

ReturnValue	The filter containing all file extensions supported by MapWinGIS.

Sample Code

```
Private Sub CdlgFilter()
    Dim tin As New MapWinGIS.Tin()
    'Open a tin from disk
    tin.Open("C:\test.tin")
    'Display the supported file formats in a message box
    MsgBox(tin.CdlgFilter)
    'Close the tin
    tin.Close()
End Sub
```

3.17.3.2 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.17.3.3 Filename

The filename associated with the object.

VB.NET Usage

ReadOnly Property Filename() As String

Parameters

ReturnValue	The filename associated with the object.

Sample Code

```
Private Sub Filename()
```

```
    Dim tin As New MapWinGIS.Tin()
    Dim filename As String
    'Open a tin from disk
    tin,Open("C:\test.tin")
    'Get the filename of the tin
    filename = tin.Filename
  End Sub
```

3.17.3.4 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
  Inherits System.Windows.Forms.Form
    'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
  Implements MapWinGIS.ICallback
    '...
#Region "ICallback Members"
   Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
        'Display the error message in a label
        LabelError.Text = ErrorMsg
  End Sub
  Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
        'Display the progress in a label
    Label1.Text = "Progress: " + Str(Percent) + "%"
        'Display the message in a label
     Label2.Text = Message
   End Sub
#End Region
        '...
```

3.17.3.5 IsNDTriangle

Gets whether or not a triangle is a no-data triangle.

VB.NET Usage

ReadOnly Property IsNDTriangle(TriIndex As Integer) As Boolean

Parameters

TriIndex	The index of the triangle to be tested whether or not it is a no-data triangle.
ReturnValue	A boolean value representing whether or not the triangle is a no-data triangle.

Sample Code

```
  Private Sub TIN_IsNDTriangle()
    Dim tin As New MapWinGIS.Tin()
    'Check if triangle 0 is a no-data triangle in the TIN
    If tin.IsNDTriangle(0) Then
      'Display a message if triangle 0 is a no-data triangle
      MsgBox("Triangle 0 is a no-data triangle.")
    End If
  End Sub
```

3.17.3.6 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
    If k = "Map1" Then
        'Set the map's key to "My Map1"
        Map1.Key = "My Map1"
    Else
        'Set the map's key to "Map1"
        Map1.Key = "Map1"
    End If
End Sub
```

3.17.3.7 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.17.3.8 NumTriangles

Gets the number of triangles in the TIN.

VB.NET Usage

ReadOnly Property NumTriangles() As Integer

Parameters

ReturnValue	The number of triangles in the TIN.

Sample Code

```
Private Sub TIN_NumTriangles()
    Dim tin As New MapWinGIS.Tin()
    Dim count As Integer
    'Get the number of triangles in the TIN
    count = tin.NumTriangles
End Sub
```

3.17.3.9 NumVertices

Gets the number of vertices in the TIN.

VB.NET Usage

ReadOnly Property NumVertices() As Integer

Parameters

ReturnValue	The number of vertices in the TIN.

Sample Code

```
Private Sub TIN_NumVertices()
    Dim tin As New MapWinGIS.Tin()
    Dim count As Integer
    'Get the number of vertices in the TIN
    count = tin.NumVertices
End Sub
```

3.18 Utils

A utils object provides access to a set of utility functions to perform a variety of tasks on other objects such as grids, images, points, shapes, shapefiles, tins, etc.

Please see also the MapWinX Library, where new functionality to perform these types of operations is being actively added.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.18.1 Examples

3.18.1.1 Grid Ambiguity

See also GridToShapefile

When creating a shapefile from a grid using the Utils function GridToShapefile, ambiguities in the grid may result in a shapefile made up of unpredictable shapes. Figure 1 illustrates what an ambiguous grid is.

Figure 1: An example of an ambiguous grid:

1	2
2	1

When the shapefile is being created from the grid shown in Figure 1, there will be an ambiguity because there are two different ways to group the grid values in the shapefile to be created.

Figure 2: The shapes created when the grid cells with a value of one are grouped together to form a shape:

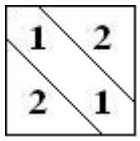
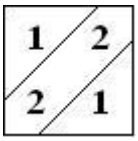


Figure 3: The shapes created when the grid cells with a value of two are grouped together to form a shape:



When the shapes are created, either the grid cells of value one could be grouped to form one diagonal shape with two triangle shapes representing the grid cells of value two (Figure 2), or the grid cells of value two could be grouped to form a different diagonal shape with two triangle shapes representing the grid cells of value one (Figure 3). A connection grid (also known as a flow grid when working with watersheds) helps resolve creating a shapefile from grids containing ambiguities. A connection grid is created by looking at each cell in the original grid and finding which of its eight neighboring cells it is most likely to connect to. If the grid is an elevation grid, then the neighboring cell with the lowest elevation less than the elevation of the current cell would be the cell water would flow to from the current cell. The connection grid cell contains an integer representing the direction of flow. This value represents the direction of the connection for the cell. A connection grid for the ambiguous grid shown earlier is shown in Figure 4. The cell with the value four is the outlet of this flow grid. The four represents a flow direction of northwest from the corresponding cell in the original grid. In the grid shown, as indicated by the key, the value five represents water flowing to the west, the value three represents water flowing north, and the value eight represents water flowing southeast.

Figure 4: A possible flow grid for the ambiguous grid shown above:

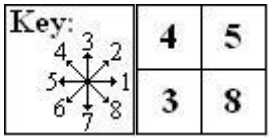
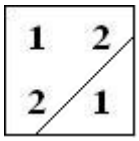


Figure 5: The shapes created from the grid in figure 1 when the flow grid in figure 4 is used to create the shapefile:



3.18.2 Functions

3.18.2.1 ClipPolygon

Modifies a polygon using the specified method.
See also PolygonOperation and Shape

VB.NET Usage

Function ClipPolygon(op As MapWinGIS.PolygonOperation, SubjectPolygon As MapWinGIS.Shape, ClipPolygon As MapWinGIS.Shape) As MapWinGIS.Shape

Parameters

op	The operation to use on the subject polygon.
SubjectPolygon	The first polygon to perform the specifed polygon operation on.
ClipPolygon	The second polygon to perform the specified polygon operation with.
ReturnValue	The polygon shape created using the specified polygon operation.

Sample Code

```
Private Sub ClipPolygon()
    Dim shape1 As New MapWinGIS.Shape(), shape2 As New MapWinGIS.Shape(), shape3 As New MapWinGIS.Shape()
```

```
    Dim utils As New MapWinGIS.Utils()
    'This assumes a function which creates the desired shape for shape1
    shape1 = CreateShape1(MapWinGIS.ShpfileType.SHP_POLYGON)
    'This assumes a function which creates the desired shape for shape1
    shape2 = CreateShape2(MapWinGIS.ShpfileType.SHP_POLYGON)
    'Create a new shape3 which will hold the result of the clip polygon operation
    shape3.Create(MapWinGIS.ShpfileType.SHP_POLYGON)
    'Create shape3 as the union of shape1 and shape2
    shape3 = utils.ClipPolygon(MapWinGIS.PolygonOperation.UNION_OPERATION, shape1, shape2)
  End Sub
```

3.18.2.2 GenerateHillShade

Generates a hillshade image for a raster data source.
This code was written by Matt Perry, perrygeo@gmail.com, published in Gdal-dev Digest, Vol 19, Issue 20
Note: Scale for Feet/Latlong use scale=370400, for Meters/LatLong use scale=111120 (based on the GRASS GIS algorithm from r.shaded.relief)

VB6 and VB.NET Usage

Function GenerateHillshade(bstrGridFilename As String, bstrShadeFilename as String, [Z as Single = 1.0], [scale as Single = 1.0], [az as single = 315.0], [alt as Single = 45.0]) As Boolean

Parameters

bstrGridFilename	The input raster data. This can be in virtually any raster data format.
bstrShadeFilename	The output image filename. This can be virtually any image format, the extension will determine the format.
Z	Z Factor. Defaults to 1.
scale	Scale Factor. Defaults to 1.
az	Azimuth. Defaults to 315.
alt	Altitude. Defaults to 45.

VB.Net Usage

```
  Private Sub GenerateAHillshade()
    Dim utils As New MapWinGIS.Utils()
    utils.GenerateHillShade("c:\input.tif", "c:\shaded.bmp", 1, 1, 315, 45)
  End Sub
```

3.18.2.3 GridInterpolateNoData

Uses valid data in a grid to replace grid cells containing no-data values with an interpolated value.

VB.NET Usage

Function GridInterpolateNoData(Grd As MapWinGIS.Grid, cBack As MapWinGIS.ICallback) As Boolean

Parameters

Grd	The grid for which no-data cells will be replaced with interpolated values.
cBack	The ICallback object which will receive progress and error messages.
ReturnValue	A boolean value representing the success or failure of replacing no-data cells in the grid with interpolated values.

Sample Code

```
  Private Sub GridInterpolateNoData()
    Dim grid As New MapWinGIS.Grid()
    Dim utils As New MapWinGIS.Utils()
    Dim success As Boolean
    'Open a grid
    grid.Open("C:\test.bgd")
    'Replace no-data cells in the grid with interpolated values
    success = utils.GridInterpolateNoData(grid, Me)
  End Sub
```

3.18.2.4 GridMerge

Merges multiple grids into a single output grid.
See also GridFileType and Grid

VB.NET Usage

Function GridMerge(Grids As Object, MergeFilename As String, Optional InRam As Boolean, Optional GrdFileType As MapWinGIS.GridFileType, Optional cBack As MapWinGIS.ICallback) As MapWinGIS.Grid

Parameters

Grids	An array of grid objects to be merged into one grid.
MergeFilename	The filename to use for the new merged grid.
InRam	Optional. A boolean value representing whether to create the merged grid in RAM or on disk.
GrdFileType	Optional. The file type of the new merged grid.
cBack	Optional. The ICallback object which will receive progress and error messages while the grids are being merged.
ReturnValue	The new merged grid.

Sample Code

```
Private Sub GridsMerge()
    Dim grids(2) As MapWinGIS.Grid
    Dim grid1 As New MapWinGIS.Grid(), grid2 As New MapWinGIS.Grid(), grid3 As New MapWinGIS.Grid()
    Dim utils As New MapWinGIS.Utils()
    'Open the first grid
    grid1.Open("C:\grid1.asc")
    'Open the second grid
    grid2.Open("C:\grid2.asc")
    'Set the first grid in the array
    grids(0) = grid1
    'Set the second grid in the array
    grids(1) = grid2
    'Merge grid1 and grid2 by passing the array containing them to the merge function , putting the new array in grid3
    grid3 = utils.GridMerge(grids, "C:\merged_grid.asc")
End Sub
```

3.18.2.5 GridReplace

Replaces all occurances of a value in the grid with a new value.
See also Grid

VB.NET Usage

Function GridReplace(Grd As MapWinGIS.Grid, OldValue As Object, NewValue As Object, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

Grd	The grid to replace the specified value in.
OldValue	The value in the grid to be replaced with the new value.
NewValue	The value to replace the old values with.
cBack	The ICallback object which will receive progress and error messages while the old value is being replaced with the new value in the grid.
ReturnValue	A boolean value representing the success or failure of replacing the old value with the new value in the grid.

Sample Code

```
Private Sub GridReplace()
    Dim grid As New MapWinGIS.Grid()
    Dim oldvalue As Double, newvalue As Double
    Dim utils As New MapWinGIS.Utils()
    Dim success As Boolean
    'Set the value of the old value to be replaced
    oldvalue = 3500
    'Set the value of the new value to be replaced
    newvalue = 3460
    'Replace the old value with the new value in the grid
    success = utils.GridReplace(grid, oldvalue, newvalue, Me)
End Sub
```

3.18.2.6 GridToGrid

Creates a new grid of the same size as the original grid, converting the values of the original grid into a different data type in the new grid.
See also Grid and GridDataType

VB.NET Usage

Function GridToGrid(Grid As MapWinGIS.Grid, OutDataType As MapWinGIS.GridDataType, Optional cBack As MapWinGIS.ICallback) MapWinGIS.Grid

Parameters

Grid	The original grid.
OutDataType	The data type to convert the original grid values to for the new grid values.
cBack	Optional. The ICallback object which will receive progress and error messages while the original grid values are converted and copied into the new grid.
ReturnValue	The new grid filled with the converted values of the original grid.

Sample Code

```
Private Sub GridToGrid()
    Dim originalgrid As New MapWinGIS.Grid(), newgrid As New MapWinGIS.Grid()
    Dim utils As New MapWinGIS.Utils()
    'Create the new grid using the values of the originalgrid by converting them to Double
    newgrid = utils.GridToGrid(originalgrid, MapWinGIS.GridDataType.DoubleDataType, Me)
End Sub
```

3.18.2.7 GridToImage

Create an image from the grid using the given grid color scheme.

VB.NET Usage

Function GridToImage(Grid As MapWinGIS.Grid, cScheme As MapWinGIS.GridColorScheme, Optional cBack As MapWinGIS.ICallback) As MapWinGIS.Image

Parameters

Grid	The grid to use to create the image.
cScheme	The grid color scheme to use to create an image from the grid.
cBack	Optional. The ICallback object which will receive progress and error messages while the image is being created from the grid.
ReturnValue	The image created from the grid.

Sample Code

```
Private Sub GridToImage()
    Dim grid As New MapWinGIS.Grid()
    Dim utils As New MapWinGIS.Utils()
    Dim image As New MapWinGIS.Image()
    Dim colorscheme As New MapWinGIS.GridColorScheme()
    'Set the grid color scheme to use when creating an image from the grid
    colorscheme.UsePredefined(grid.Minimum, grid.Maximum, MapWinGIS.PredefinedColorScheme.SummerMountains)
    'Create an image from the grid using the grid color scheme
    image = utils.GridToImage(grid, colorscheme, Me)
End Sub
```

3.18.2.8 GridToShapefile

Creates a new shapefile from the grid. Note: It is important that the flow grid is used when there are any ambiguous parts of the grid. Ambiguity Example
See also Grid and Shapefile

VB.NET Usage

Function GridToShapefile(Grid As MapWinGIS.Grid, Optional ConnectionGrid As MapWinGIS.Grid, Optional cBack

As MapWinGIS.ICallback) As MapWinGIS.Shapefile

Parameters

Grid	The grid to be converted to a shapefile.
ConnectionGrid	Optional. The flow grid that is used to resolve any ambiguity. Without this flow grid the output results are somewhat unpredictable.
cBack	Optional. The ICallback object which will receive progress and error messages while the grid is being converted to a shapefile.
ReturnValue	The new shapefile created from the grid.

Sample Code

```
Private Sub GridToShapefile()
    Dim grid As New MapWinGIS.Grid(), flowgrid As New MapWinGIS.Grid()
    Dim utils As New MapWinGIS.Utils()
    Dim sf As New MapWinGIS.Shapefile()
    'Open a grid
    grid.Open("C:\test.asc")
    'Open the flow grid
    grid.Open("C:\test_flow.asc")
    'Create a new shapefile from the grid
    sf = utils.GridToShapefile(grid, flowgrid, Me)
End Sub
```

3.18.2.9 PointInPolygon

Gets whether or not a point lies within the specified polygon shape. Note: There is a faster option available when working with shapefiles: PointInShape.
See also PointInShape and Shapefile Shape and Point

VB.NET Usage

Function PointInPolygon(Shp As MapWinGIS.Shape, TestPoint As MapWinGIS.Point) As Boolean

Parameters

Shp	The polygon shape to perform the test on.
TestPoint	The point to test whether or not it lies within the specified polygon shape.
ReturnValue	A boolean value representing whether or not the point lies within the shape.

Sample Code

```
Private Sub UtilPointInPolygon()
    Dim utils As New MapWinGIS.Utils()
    Dim point As New MapWinGIS.Point()
    Dim shape As New MapWinGIS.Shape()
    Dim success As Boolean
    'Set the x and y values of the point to be tested
    point.x = 3000
    point.y = 2500
    'Check if the point lies within the polygon shape
    success = utils.PointInPolygon(shape, point)
End Sub
```

3.18.2.10 RemoveColinearPoints

Removes colinear points from a shapefile. Note: A shapefile containing colinear points leads to the shapefile taking up unecessary space since some points in the shapefile add detail to the display of the shapes.

VB.NET Usage

Function RemoveColinearPoints(Shapes As MapWinGIS.Shapefile, LinearTolerance As Double, Optional cBack As MapWinGIS.ICallback) As Boolean

Parameters

Shapes	The shapefile to remove colinear points from.
LinearTolerance	Points will be considered colinear if the distance in between them is with in this tolerance. This distance is measured in projected map coordinates.

cBack	Optional. The ICallback object which will receive progress and error messages while colinear points are being removed.
ReturnValue	A boolean value representing the success or failure of removing colinear points from the shapefile.

Sample Code

```
Private Sub RemoveColinearPoints()
    Dim utils As New MapWinGIS.Utils()
    Dim sf As New MapWinGIS.Shapefile()
    Dim tolerance As Double
    Dim success As Boolean
    'Set the tolerance which will be used to determine the maximum distance between points considered colinear
    tolerance = 5
    'Remove all colinear points from the shapefile
    success = utils.RemoveColinearPoints(sf, tolerance, Me)
End Sub
```

3.18.2.11 ShapeMerge

Merges two shapes in a shapefile to create a new shape.
See also Shapefile and Shape

VB.NET Usage

Function ShapeMerge(Shapes As MapWinGIS.Shapefile, IndexOne As Integer, IndexTwo As Integer, Optional cBack As MapWinGIS.ICallback) As MapWinGIS.Shape

Parameters

Shapes	The shapefile containing the two shapes to be merged to create the new shape.
IndexOne	The index of the first shape to be merged.
IndexTwo	The index of the second shape to be merged.
cBack	Optional. The ICallback object which will receive progress and error messages while the shapes are being merged.
Arg2	description.

Sample Code

```
Private Sub UtilShapeMerge()
    Dim utils As New MapWinGIS.Utils()
    Dim sf As New MapWinGIS.Shapefile()
    Dim newshape As New MapWinGIS.Shape()
    'Create a new shape from shape 0 and shape 1 in the shapefile
    newshape = utils.ShapeMerge(sf, 0, 1, Me)
End Sub
```

3.18.2.12 ShapeToShapeZ

Creates a new shapefile with z values added from an elevation grid.
See also Grid and Shapefile

VB.NET Usage

Function ShapeToShapeZ(Shapefile As MapWinGIS.Shapefile, Grid As MapWinGIS.Grid, Optional cBack As MapWinGIS.ICallback) As MapWinGIS.Shapefile

Parameters

Shapefile	The shapefile to be converted to a new shapefile with z values.
Grid	The elevation grid to get the z values from.
cBack	Optional. The ICallback object which will receive progress and error messages while z values are being added to the shapefile.
ReturnValue	The new shapefile containing z values obtained from the elevation grid.

Sample Code

```
Private Sub ShapeToShapeZ()
```

```
    Dim utils As New MapWinGIS.Utils()
    Dim sf As New MapWinGIS.Shapefile(), sfZ As New MapWinGIS.Shapefile()
    Dim grid As New MapWinGIS.Grid()
    'Create polygon shapefile from the grid
    sf = utils.GridToShapefile(grid)
    'Create polygon Z shapefile by adding elevation data from grid to the shapefile
    sfZ = utils.ShapeToShapeZ(sf, grid, Me)
  End Sub
```

3.18.2.13 ShapefileToGrid

This function is not yet implemented. Converts a shapefile to a grid.

VB.NET Usage

Function ShapefileToGrid(Shpfile As MapWinGIS.Shapefile, Optional UseShapefileBounds As Boolean, Optional GrdHeader As MapWinGIS.GridHeader, Optional Cellsize As Double, Optional UseShapeNumber As Boolean, Optional SingleValue As Short) As MapWinGIS.Grid

Parameters

Shpfile	The shapefile to be converted into a grid.
UseShapefileBounds	Optional. A boolean value representing whether or not the grid will have the same extents as the shapefile. The default is True.
GrdHeader	Optional. The grid header to use to create the new grid. If UseShapefileBounds is set to True, the extents supplied in the grid header will be ignored when the grid is created.
Cellsize	Optional. The cell size of the new grid. The default cell size is 30.
UseShapeNumber	Optional. Specifies that the value of each cell in the grid should be the shape index. The default is True.
SingleValue	Optional. The value to use when creating the grid. This only applies when UseShapeNumber is set to False. The default value is 1.
ReturnValue	The grid created from the shapefile.

Sample Code

```
  Private Sub Name()
    'code
  End Sub
```

3.18.2.14 TinToShapefile

Creates a shapefile from a TIN object.

VB.NET Usage

Function TinToShapefile(Tin As MapWinGIS.Tin, Type As MapWinGIS.ShpfileType, Optional cBack As MapWinGIS.ICallback) As MapWinGIS.Shapefile

Parameters

Tin	The TIN object to be used to create a new shapefile.
Type	The type of the shapefile to be created.
cBack	Optional. The ICallback object which will receive progress and error messages while the shapefile is created from the TIN.
ReturnValue	The new shapefile created from the TIN.

Sample Code

```
  Private Sub TinToShapefile()
    Dim utils As New MapWinGIS.Utils()
    Dim tin As New MapWinGIS.Tin()
    Dim sf As New MapWinGIS.Shapefile()
    'Open a TIN from file
    tin.Open("C:\test.tin")
    'Create a shapefile from the TIN
    sf = utils.TinToShapefile(tin, MapWinGIS.ShpfileType.SHP_POLYGON, Me)
  End Sub
```

3.18.2.15 hBitmapToPicture

Converts an hBitmap to an IPictureDisp object..

VB.NET Usage

Function hBitmapToPicture(hBitmap As Integer) As stdole.IPictureDisp

Parameters

hBitmap	A device context handle to a bitmap.
ReturnValue	An IPictureDisp object created from the bitmap.

Sample Code

```
'Use the user32 function "LoadImage" from user32.dll
    Declare Function LoadImage Lib "user32" Alias "LoadImageA" (ByVal hInst As Integer, ByVal lpsz As String, ByVal un1
As Integer, ByVal n1 As Integer, ByVal n2 As Integer, ByVal un2 As Integer) As Integer
    '...
    Private Sub Utils_hBitmap()
    Dim hBitmap As Integer
    Dim iPictDisp As stdole.IPictureDisp
    Dim utils As New MapWinGIS.Utils()
    Dim image As New MapWinGIS.Image()
    'Constants used to get an hBitmap
    Const IMAGE_BITMAP As Integer = 0
    Const LR_LOADFROMFILE As Integer = &H10
    Const LR_CREATEDIBSECTION As Integer = &H2000
    'Load a bitmap from file storing a handle to the bitmap in hBitmap
    hBitmap = LoadImage(0, "C:\test.bmp", IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE Or LR_CREATEDIBSECTION)
    'Get an IPictureDisp object from the hBitmap
    iPictDisp = utils.hBitmapToPicture(hBitmap)
    'Set the picture for the image to the IPictureDisp object created from the hBitmap
    image.Picture = iPictDisp
    'Add the image to the map
    Map1.AddLayer(image, True)
End Sub
```

3.18.3 Properties

3.18.3.1 Area

Returns the area of the polygon shape. For mulit-part polygons which may contain counter-clockwise holes, the area of the holes will be subtracted from that of the surrounding clockwise portions.

VB.NET Usage

ReadOnly Property Area(Shape As MapWinGIS.Shape) As Double

Parameters

Shape	The polygon shape for which the area is required.
ReturnValue	The area of the polygon shape.

Sample Code

```
Private Sub UtilsArea()
    Dim utils As New MapWinGIS.Utils()
    Dim shape As New MapWinGIS.Shape()
    Dim area As Double
    'Get the area of the polygon shape
    area = utils.Area(shape)
End Sub
```

3.18.3.2 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.18.3.3 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
    Inherits System.Windows.Forms.Form
        'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
    Implements MapWinGIS.ICallback
      '...
 #Region "ICallback Members"
     Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
              'Display the error message in a label
             LabelError.Text = ErrorMsg
    End Sub
    Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
            'Display the progress in a label
        Label1.Text = "Progress: " + Str(Percent) + "%"
            'Display the message in a label
         Label2.Text = Message
     End Sub
 #End Region
              '...
```

3.18.3.4 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
    If k = "Map1" Then
        'Set the map's key to "My Map1"
        Map1.Key = "My Map1"
    Else
```

```
        'Set the map's key to "Map1"
        Map1.Key = "Map1"
    End If
End Sub
```

3.18.3.5 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.18.3.6 Length

Gets the length of the line shape.

VB.NET Usage

ReadOnly Property Length(Shape As MapWinGIS.Shape) As Double

Parameters

Shape	The line shape for which the length is required.
ReturnValue	The length of the line shape.

Sample Code

```
Private Sub UtilsLength()
    Dim utils As New MapWinGIS.Utils()
    Dim shape As New MapWinGIS.Shape()
    Dim length As Double
    'Get the length of the line shape
    length = utils.Length(shape)
End Sub
```

3.18.3.7 Perimeter

Gets the perimeter of the polygon shape.

VB.NET Usage

ReadOnly Property Perimeter(Shape As MapWinGIS.Shape) As Double

Parameters

Shape	The polygon shape for which the perimeter is required.
ReturnValue	The perimeter of the polygon shape.

Sample Code

```
Private Sub UtilsPerimeter()
    Dim utils As New MapWinGIS.Utils()
    Dim shape As New MapWinGIS.Shape()
    Dim perimeter As Double
    'Get the perimeter of the polygon shape
    perimeter = utils.Perimeter(shape)
End Sub
```

3.19 Vector

A vector object is used to represent the light source for a grid color scheme.

The functions and properties are listed below. Clicking on each will yield a description of the function and its arguments as well as sample code where applicable.

3.19.1 Functions

3.19.1.1 CrossProduct

Calculates the vector cross product between the current vector object and the specified vector.

VB.NET Usage

Function CrossProduct(V As MapWinGIS.Vector) As MapWinGIS.Vector

Parameters

V	The vector to calculate the cross product with the current vector object.
ReturnValue	The cross product of the current vector object and the specified vector.

Sample Code

```
Private Sub VectorCrossProduct()
    Dim vector As New MapWinGIS.Vector(), vector2 As New MapWinGIS.Vector(), crossproduct As New
MapWinGIS.Vector()
    'Get the cross product of vector and vector2
    crossproduct = vector.CrossProduct(vector2)
End Sub
```

3.19.1.2 Dot

Calculates the dot product of the current vector object with the specified vector.

VB.NET Usage

Function Dot(V As MapWinGIS.Vector) As Double

Parameters

V	The vector to use to calculate the dot product with the current vector object.
ReturnValue	The dot product of the current vector object and the specified vector.

Sample Code

```
Private Sub VectorDotProduct()
    Dim vector As New MapWinGIS.Vector(), vector2 As New MapWinGIS.Vector()
    Dim dotproduct As Double
    'Get the dot product of vector and vector2
    dotproduct = vector.Dot(vector2)
End Sub
```

3.19.2 Subs

3.19.2.1 Normalize

Normalizes the vector.

VB.NET Usage

Sub Normalize()

Parameters

None

Sample Code

```
Private Sub VectorNormalize()
    Dim vector As New MapWinGIS.Vector()
```

```
    'Normalize the vector
    vector.Normalize()
  End Sub
```

3.19.3 Properties

3.19.3.1 ErrorMsg

Retrieves the error message associated with the specified error code.

VB.NET Usage

ReadOnly Property get_ErrorMsg(ErrorCode As Integer) As String

Parameters

ErrorCode	The error code for which the error message is required.
ReturnValue	The error message description for the specified error code.

Sample Code

```
  Private Sub ErrorMessage()
    Dim errorCode As Integer
    'Set the error code
    errorCode = 10
    'Display message box giving error message for error code
    MsgBox(Map1.get_ErrorMsg(errorCode))
  End Sub
```

3.19.3.2 GlobalCallback

The global callback is the interface used by MapWinGIS to pass progress and error events to interested applications.

VB.NET Usage

Property GlobalCallback() As Object

Parameters

ReturnValue	The global callback used by MapWinGIS to pass progress and errors.

Sample Code

```
Public Class Form1
  Inherits System.Windows.Forms.Form
    'To use the MapWinGIS callback to receive errors and messages, you must implement the MapWinGIS.ICallback
interface
  Implements MapWinGIS.ICallback
    '...
#Region "ICallback Members"
   Public Sub myError(ByVal KeyOfSender As String, ByVal ErrorMsg As String) Implements MapWinGIS.ICallback.Error
        'Display the error message in a label
        LabelError.Text = ErrorMsg
  End Sub
  Public Sub Progress(ByVal KeyOfSender As String, ByVal Percent As Integer, ByVal Message As String) Implements
MapWinGIS.ICallback.Progress
        'Display the progress in a label
    Label1.Text = "Progress: " + Str(Percent) + "%"
        'Display the message in a label
     Label2.Text = Message
   End Sub
#End Region
        '...
```

3.19.3.3 Key

The key may be used by the programmer to store any string data associated with the object.

VB.NET Usage

Property Key() As String

Parameters

ReturnValue	The key in string format.

Sample Code

```
Private Sub MapKey()
    Dim k As String
    'Get the map's key
    k = Map1.Key
    'Check if the map's key is "Map1"
    If k = "Map1" Then
        'Set the map's key to "My Map1"
        Map1.Key = "My Map1"
    Else
        'Set the map's key to "Map1"
        Map1.Key = "Map1"
    End If
End Sub
```

3.19.3.4 LastErrorCode

Retrieves the last error generated in the object.

VB.NET Usage

ReadOnlyProperty LastErrorCode() As Integer

Parameters

ReturnValue	The integer error code for the last error generated in the object.

Sample Code

```
Private Sub LastErrorCode()
    Dim errorCode As Integer
    'Get the last error in the map
    errorCode = Map1.LastErrorCode
    'Display message box giving error message for the last error in the map
    MsgBox(Map1.get_ErrorMsg(errorCode))
End Sub
```

3.19.3.5 i

Gets or sets the i component of the vector.

VB.NET Usage

Property i() As Double

Parameters

ReturnValue	The i component of the vector.

Sample Code

```
Private Sub Vectori()
    Dim vector As New MapWinGIS.Vector()
    Dim i As Double
    'Set the value of i
    i = 100
    'Set the i component for the vector
    vector.i = i
    'Get the i component for the vector
    i = vector.i
End Sub
```

3.19.3.6 j

Gets or sets the j component of the vector.

VB.NET Usage

Property j() As Double

Parameters

ReturnValue	The j component of the vector.

Sample Code

```
Private Sub Vectorj()
    Dim vector As New MapWinGIS.Vector()
    Dim j As Double
    'Set the value of j
    j = 100
    'Set the j component for the vector
    vector.j = j
    'Get the j component for the vector
    j = vector.j
End Sub
```

3.19.3.7 K

Gets or sets the k component of the vector.

VB.NET Usage

Property k() As Double

Parameters

ReturnValue	The k component of the vector.

Sample Code

```
Private Sub Vectork()
    Dim vector As New MapWinGIS.Vector()
    Dim k As Double
    'Set the value of k
    k = 100
    'Set the k component for the vector
    vector.k = k
    'Get the k component for the vector
    k = vector.k
End Sub
```

---

4 Enumerations

Following is a list of enumerations contained in the MapWinGIS ActiveX control. These enumerations are used by specific functions listed in the objects in the preceeding section.

4.1 AmbiguityResolution

The ambiguity resolution enumerations are used to determine how ambiguities are resolved when creating a shapenetwork. Note: Distance to outlet is the only ambiguity resolution technique implemented at this time.

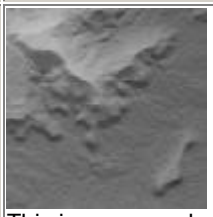
Values

Name	Summary
Z_VALUE	Uses the z value of the end points of an ambiguous shape to determine what direction water would flow while creating a shapenetwork. Note: Using this enumerated value has no effect because distance to outlet is the only ambiguity resolution technique implented at this time.
DISTANCE_TO_OUTLET	Uses the distance to the outlet from the end points of an ambiguous shape to determine what direction water would flow while creating a shapenetwork.
NO_RESOLUTION	Uses no ambiguity resolution technique when there is an ambiguous shape when creating a shapenetwork. The algorithm chooses a random direction of flow for the ambiguous shape. Note: Using this enumerated value has no effect because distance to outlet is the only ambiguity resolution technique implented at this time.

4.1.1 ColoringType

The coloring type enumerations are used to determine how a grid color break will be displayed.

Values

Name	Summary
Hillshade	 This is an example of a grid color break with a hillshade coloring type. The higher elevations in the grid are colored by light gray, and the lower elevations in the grid are colored by the dark gray. Notice how the shaded slopes help distinguish elevated terrain. This gives the most realistic coloring for the grid color scheme the break is added to.
Gradient	 This is an example of a grid color break with a gradient coloring type. The higher elevations in the grid are colored by light gray, and the lower elevations in the grid are colored by the dark gray.
Random	 This is an example of a grid color break with a random coloring type. This is not implemented at this time. When this coloring type is used for a grid color break, the low value color will be used to color the entire grid.

4.2 FieldType

The field type enumerations are used to indicate what type of data will be stored in the field.

Values

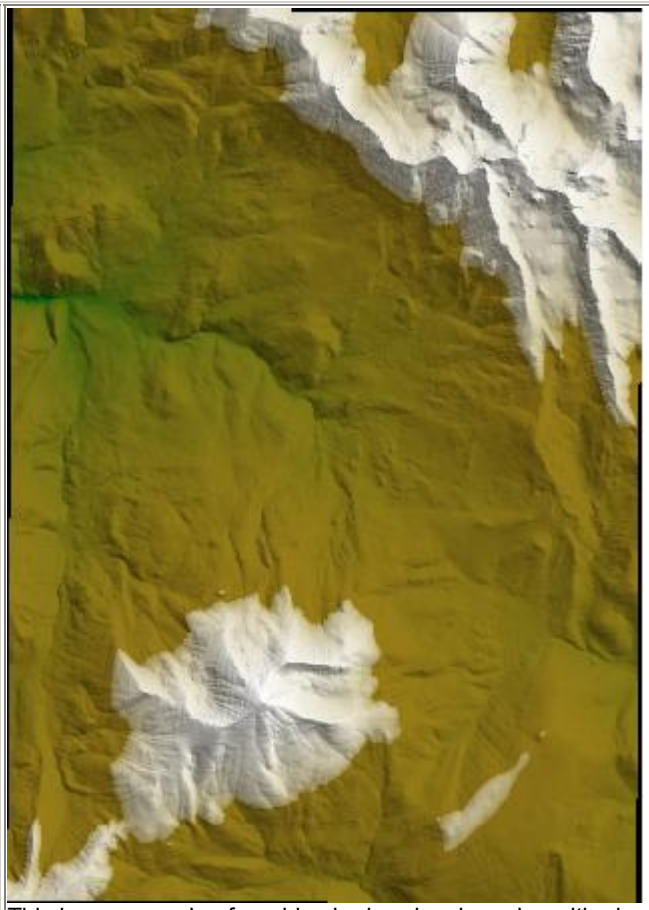
Name	Summary
STRING_FIELD	String data will be the type of data to be stored in the Field when its type is set to STRING_FIELD.
INTEGER_FIELD	Integer data will be the type of data to be stored in the Field when its type is set to INTEGER_FIELD.
DOUBLE_FIELD	Double data will be the type of data to be stored in the Field when its type is set to DOUBLE_FIELD.

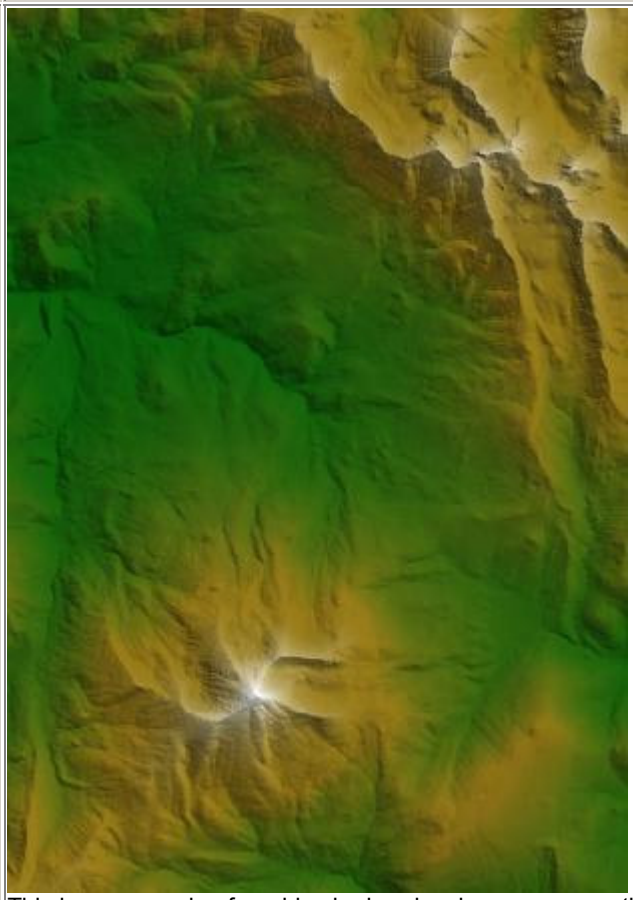
4.2.1 GradientModel

The gradient model enumerations are used to determine the gradient model to be used for a grid color break.

Values

Name	Summary

Logorithmic	 This is an example of a grid color break using a logorithmic gradient model. This produces the slowest variation in color for the grid color scheme the color break is added to.
Linear	 This is an example of a grid color break using a linear gradient model. This produces a faster variation in color for the grid color scheme the color break is added to. Notice how it produces faster variation in the gradient colors than the logorithmic gradient model,

	and slower variation in the gradient colors than the exponential gradient model.
Exponential	 This is an example of a grid color break using an exponential gradient model. This produces the fastest variation in color for the grid color scheme the color break is added to.

4.3 GridDataType

The grid data type enumerations are used to indicate the type of data stored in the grid.
Values

Name	Summary
InvalidDataType	This specifies that the values stored in the grid are an invalid data type values. Use this as the grid's data type when you know the data is not of any other type listed below.
ShortDataType	This specifies that the values stored in the grid are Short data type values.
LongDataType	This specifies that the values stored in the grid are Long data type values.
FloatDataType	This specifies that the values stored in the grid are Float data type values.
DoubleDataType	This specifies that the values stored in the grid are Double data type values.
UnknownDataType	This specifies that the values stored in the grid are Unknown data type values. Use this as the grid's data type when you are unsure of the type of data stored in the grid.

4.4 GridFileType

The grid file type enumerations indicate the file type for the grid.
Values

Name	Summary
InvalidGridFileType	This specifies that the grid is of a file type not supported by MapWinGIS.
Ascii	This specifies that the grid is an ASCII grid.
Binary	This specifies that the grid is of a binary grid.
Esri	This specifies that the grid is of an ESRI grid.
Sdts	This specifies that the grid is of an SDTS grid.
UseExtension	This specifies that the grid type can be determined by the file extension used in the grid's filename.

4.5 ImageType

The image type enumerations indicate the image type for the image.
Values

Name	Summary
BITMAP_FILE	This specifies that the image is in bitmap image format.
GIF_FILE	This specifies that the image is in GIF image format.
USE_FILE_EXTENSION	This specifies that the image format can be found by using the file extension used in the image's filename.
PPM_FILE	This specifies that the image is in portable pixmap format.

4.6 PolygonOperation

Polygon operation enumerations are used to determine which operation is to be used when clipping polygons with the Utils function ClipPolygon. Beside each enumerated operation is an example of clipping a triangle and square where the triangle covers part of the square as shown below in Figure 1.

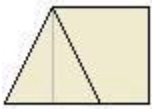
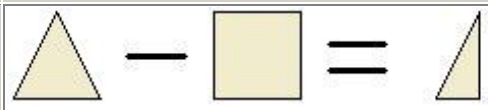
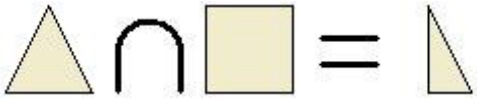
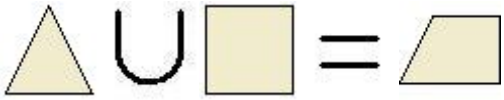


Figure 1: The triangle hides part of the square.

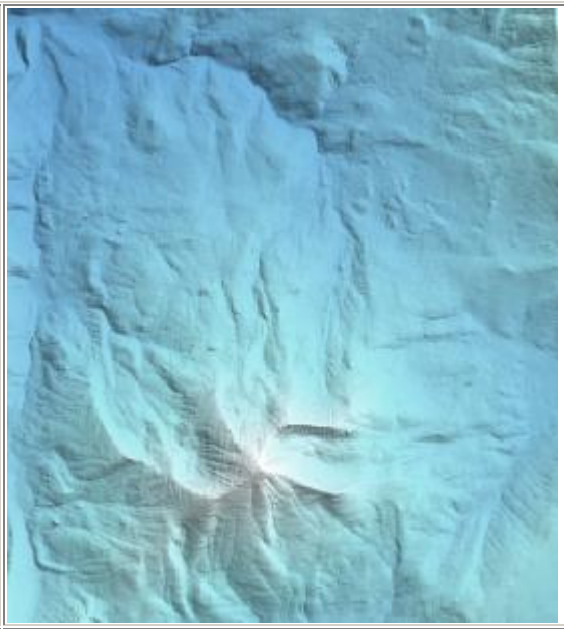
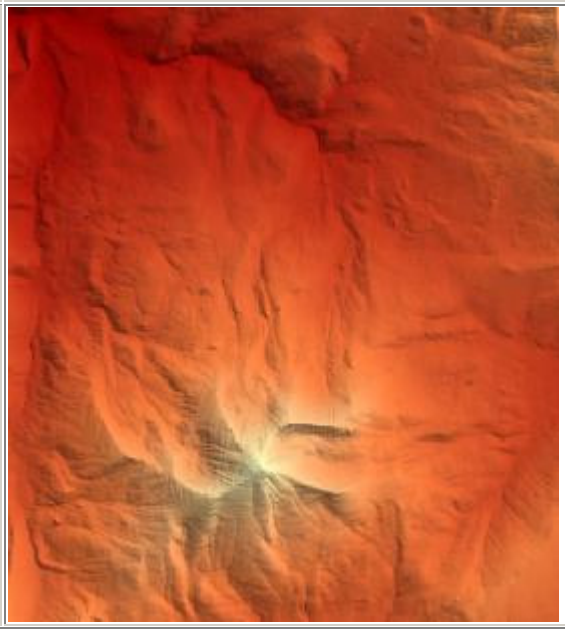
Name	Summary
DIFFERENCE_OPERATION	
INTERSECTION_OPERATION	
EXCLUSIVEOR_OPERATION	
UNION_OPERATION	

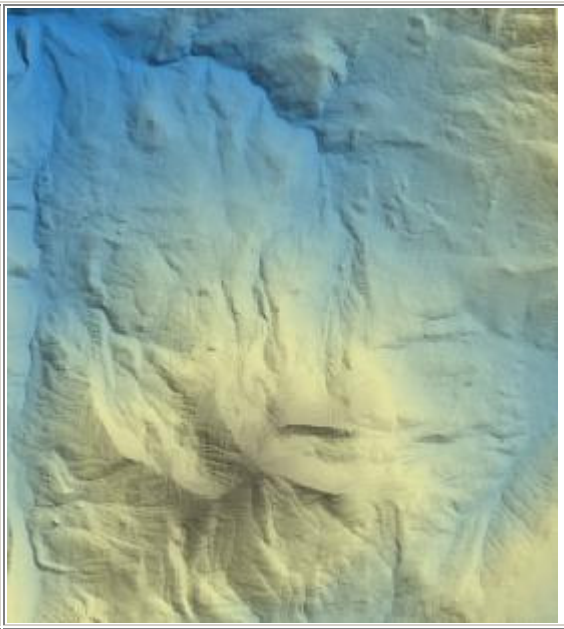
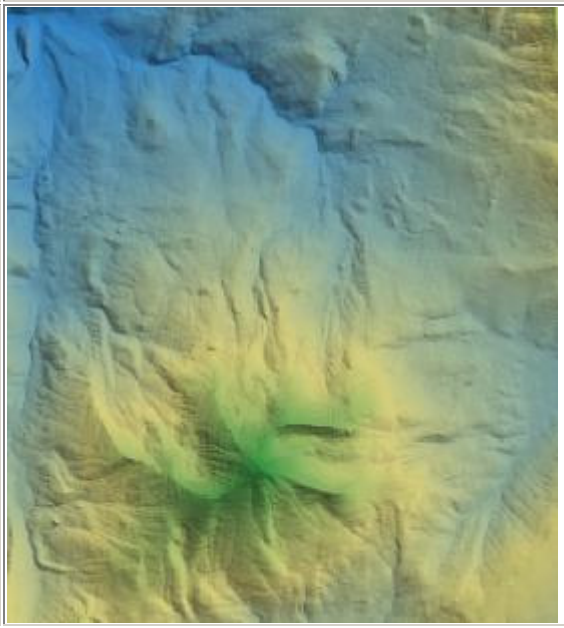
4.7 PredefinedColorScheme

The predefined color scheme's allow you to quickly create a grid color scheme without creating your own color breaks.
Values

Name	Example

FallLeaves	
SummerMountains	
Desert	

Glaciers	
Meadow	
ValleyFires	

DeadSea	
Highway1	

4.8 SelectMode

The select mode enumerations are used to determine the behavior of selecting shapes within a shapefile.
Values

Name	Summary
INTERSECTION	This specifies that shapes in the shapefile must only intersect with the bounding box to be selected.
INCLUSION	This specifies that shapes in the shapefile must be completely enclosed by the bounding box to be selected.

4.9 ShpfileType

The shapefile type enumerations are used to determine the the type of a shape and also the type of shapes that can be inserted into the shapefile. The shapefile type of a shape determines how the points added to the shape are interpreted to draw the desired geometric object. Note: Multipatch shapefiles are not currently supported by MapWindow.
Values

Name	Summary
SHP_NULLSHAPE	This specifies that there is no geometric data in the shape. Null shapes

	are often used as place holders when a shapefile is created, and are later populated with geometric data.
SHP_POINT	This specifies that the shapefile is a Point shapefile, and that only shapes of this type are contained in the shapefile. The points are stored as double precision values (x, y).
SHP_POLYLINE	This specifies that the shapefile is a Polyline shapefile, and that only shapes of this type are contained in the shapefile. The Polyline is defined by an ordered array of points of double precision values (x,y). These points are contained in one or more parts.
SHP_POLYGON	This specifies that the shapefile is a Polygon shapefile, and that only shapes of this type are contained in the shapefile. The Polygon is defined by an ordered array of points of double precision values (x,y). These points are contained in one or more parts. A polygon may have interior parts which are not filled. These parts contain points ordered in a counter-clockwise direction. Parts that are to be filled are ordered in a clockwise direction. The outer part for the polygon must be ordered in a clockwise direction.
SHP_MULTIPOINT	This specifies that the shapefile is a MultiPoint shapefile, and that only shapes of this type are contained in the shapefile. The points are stored as double precision values (x, y) in an array.
SHP_POINTZ	This specifies that the shapefile is a PointZ shapefile, and that only shapes of this type are contained in the shapefile. The points are stored as double precision values (x, y, z) with an associated double precision Measure value (M).
SHP_POLYLINEZ	This specifies that the shapefile is a PolylineZ shapefile, and that only shapes of this type are contained in the shapefile. The Polyline is defined by an ordered array of points of double precision values (x,y,z) with an associated Measure value (M) for each point. These points are contained in one or more parts.
SHP_POLYGONZ	This specifies that the shapefile is a PolygonZ shapefile, and that only shapes of this type are contained in the shapefile. The Polygon is defined by an ordered array of points of double precision values (x,y,z) with an associated Measure value (M) for each point. These points are contained in one or more parts. A polygon may have interior parts which are not filled. These parts contain points ordered in a counter-clockwise direction. Parts that are to be filled are ordered in a clockwise direction. The outer part for the polygon must be ordered in a clockwise direction.
SHP_MULTIPOINTZ	This specifies that the shapefile is a MultipointZ shapefile, and that only shapes of this type are contained in the shapefile. The points are stored as double precision values (x, y, z) in an array with an associated Measure value (M) for each point.
SHP_POINTM	This specifies that the shapefile is a PointM shapefile, and that only shapes of this type are contained in the shapefile. The points are stored as double precision values (x, y) with an associated Measure value (M).
SHP_POLYLINEM	This specifies that the shapefile is a PolylineM shapefile, and that only shapes of this type are contained in the shapefile. The Polyline is defined by an ordered array of points of double precision values (x,y) with an associated Measure value (M) for each point. These points are contained in one or more parts.
SHP_POLYGONM	This specifies that the shapefile is a PolygonM shapefile, and that only shapes of this type are contained in the shapefile. The Polygon is defined by an ordered array of points of double precision values (x,y) with an associated Measure value (M) for each point. These points are contained in one or more parts. A polygon may have interior parts which are not filled. These parts contain points ordered in a counter-clockwise direction. Parts that are to be filled are ordered in a clockwise direction. The outer part for the polygon must be ordered in a clockwise direction.
SHP_MULTIPOINTM	This specifies that the shapefile is a MultipointM shapefile, and that only shapes of this type are contained in the shapefile. The points are stored as double precision values (x, y) in an array with an associated Measure value (M) for each point.
SHP_MULTIPATCH	This specifies that the shapefile is a Multipatch shapefile, and that only shapes of this type are contained in the shapefile.

	The Multipatch is defined by a group of parts. There is an ordered array of points which is divided into parts. Each part has a part type (Triangle Strip, Triangle Fan, Outer Ring, Inner Ring, First Ring, or Ring). The part type defines how the points in the specified part are interpreted to draw the desired geometric shape. Note: Multipatch shapefiles are not currently supported by MapWindow.

4.10 SplitMethod

The split method enumerations are used to determine what criteria is used when the TIN creation algorithm decides whether or not to split a triangle.

Values

Name	Summary
InscribedRadius	This specifies that the inscribed radius will determine whether or not a specific triangle will be split while creating a TIN. Inscribed radius refers to the method of drawing the largest circle you can within a triangle. The radius of this circle is compared to a given minimum radius allowed. If the radius of the inscribed circle is less than the given minimum radius, then the triangle will not be split.
AngleDeviation	This specifies that the angle deviation will determine whether or not a specific triangle will be split while creating a TIN. The angle deviation method refers to comparing the angles in a triangle to a given value in degrees of the smallest allowable angle in the TIN. When a triangle is tested to see if it should be split, each of the three angles in the triangle is compared with the given smallest allowed angle. If any of these angles is smaller than the smallest allowed angle, the triangle will not be split.

4.11 tkCursor

The tkCursor enumerations specify the cursors that can be used with the map.

Values

Name	Summary
crsrAppStarting	
crsrArrow	
crsrCross	
crsrHelp	
crsrIBeam	
crsrMapDefault	
crsrNo	
crsrSizeAll	
crsrSizeNESW	
crsrSizeNS	
crsrSizeNWSE	
crsrSizeWE	
crsrUpArrow	
crsrWait	
crsrUserDefined	User defined cursor.

4.12 tkCursorMode

The tkCursorMode enumerations are used to select the cursor mode for the map.

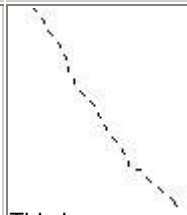
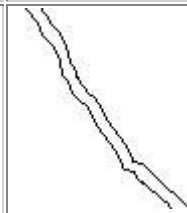
Values

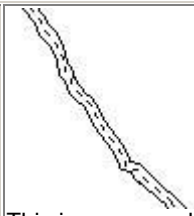
Name	Summary
cmZoomIn	
cmZoomOut	
cmPan	
cmSelection	
cmNone	

4.12.1 tkLineStipple

The tkLineStipple enumerations are used to determine the stipple used when drawing a line shapefile in the map.

Values

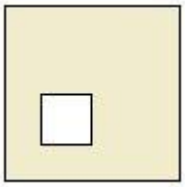
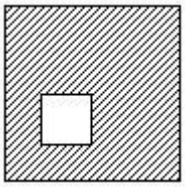
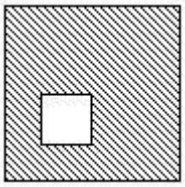
Name	Summary
lsCustom	 This is an example of a line shapefile drawn with a custom line stipple. To use a custom line stipple for a shapefile displayed in the map, you must set the UDLineStipple property for the layer in the map.
lsDotted	 This is an example of a line shapefile drawn with a dotted line stipple.
lsDashed	 This is an example of a line shapefile drawn with a dashed line stipple.
lsDashDotDash	 This is an example of a line shapefile drawn with a dash-dot-dash line stipple.
lsDoubleSolid	 This is an example of a line shapefile drawn with a double solid line stipple.

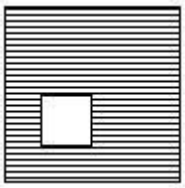
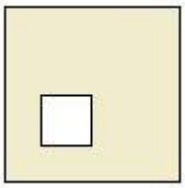
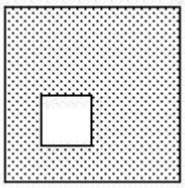
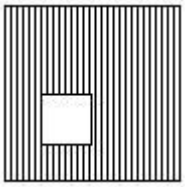
lsDoubleSolidPlusDash	 This is an example of a line shapefile drawn with a double solid-plus dash line stipple. This line stipple is intended to mimic a road with a dashed centerline.
lsNone	 This is an example of a line shapefile drawn with a no line stipple.
lsTrainTracks	 This is an example of a line shapefile drawn with a train track line stipple.

4.13 tkFillStipple

The tkFillStipple enumerations are used to determine the fill stipple of polygon shapefiles for a layer in map. Note: The custom fill stipple is not implemented as yet.

Values

Name	Summary
fsCustom	 This is an example of a polygon shapefile drawn with a custom fill stipple. Note: Custom fill stipple is not implemented as of MapWindow 3.0. No fill stipple will be drawn if fsCustom is used.
fsDiagonalDownLeft	 This is an example of a polygon shapefile drawn with a diagonal-down-left fill stipple.
fsDiagonalDownRight	 This is an example of a polygon shapefile drawn with a diagonal-down-right fill stipple.

fsHorizontalBars	 This is an example of a polygon shapefile drawn with a horizontal-bars fill stipple.
fsNone	 This is an example of a polygon shapefile drawn with no fill stipple.
fsPolkaDot	 This is an example of a polygon shapefile drawn with a polka-dot fill stipple.
fsVerticalBars	 This is an example of a polygon shapefile drawn with a vertical-bars fill stipple.

4.14 tkDrawReferenceList

The draw reference list enumerations are used to determine how the drawing coordinates will be referenced when drawing on the map.

Values

Name	Summary
dlScreenReferencedList	This specifies that screen coordinates will be used to draw on the map. The coordinates will be measured in pixel units from the upper left corner of the map display.
dlSpatiallyReferencedList	This specifies that spatial coordinates will be used to draw on the map. The coordinates will be measured in projected map coordinates.

4.15 tkDrawMode

The Draw Mode enumerations are used to determine how points will be drawn in a multipatch shapefile. Note: Multipatch shapefiles are not currently supported in MapWindow.

Values

Name	Summary
dmPoints	The points in the part will be drawn as individual points.
dmLines	The points in the part will be drawn with lines between each point.
dmLineLoop	The points in the part will be drawn with lines between each point with an extra line connecting the first and last point in the part.
dmLineStrip	The points in the part will be drawn as a line strip.
dmTriangles	The points in the part will be drawn as triangles between groups of three points.

dmTriangleStrip	The points in the part will be drawn as a triangle strip with triangles drawn between each point.
dmTriangleFan	The points in the part will be drawn as a triangle fan with the first point as the center of the fan and the other points creating the arch in the fan.
dmQuads	The points in the part will be drawn as quads between groups of four points.
dmQuadStrip	The points in the part will be drawn as a quad strip.
dmPolygon	The points in the part will be drawn as a polygon.

4.16 tkPointType

The point type enumerations are used to determine how points are drawn in a point shapefile in a layer in the map.
Values

Name	Summary
ptCircle	
ptDiamond	
ptImageList	[[1]]
ptSquare	
ptTriangleUp	
ptTriangleDown	
ptTriangleLeft	

ptTriangleRight	
ptUserDefined	

4.17 tkLockMode

The lock mode enumerations determine whether the map is locked or unlocked.
Values

Name	Summary
lmUnlock	This specifies that the map is unlocked.
lmLock	This specifies that the map is locked.

4.18 tkHJustification

The horizontal justification enumerations determine the horizontal justification of text displayed in a label on the map.
Values

Name	Summary
hjLeft	This specifies that the text is to be displayed with left justification.
hjCenter	This specifies that the text is to be diplayed with center justification.
hjRight	This specifies that the text is to be displayed with right justification.