TCAD for .NET

for Visual Studio & Delphi.NET

User Manual

TCAD for .NET Edition

Printed: 2007-10-11 in China

Publisher
TCAD Team

Managing Editor
June Shi

Technical Editors
June Shi
Fei Hong Bin

Cover Designer
Fei Hong Bin

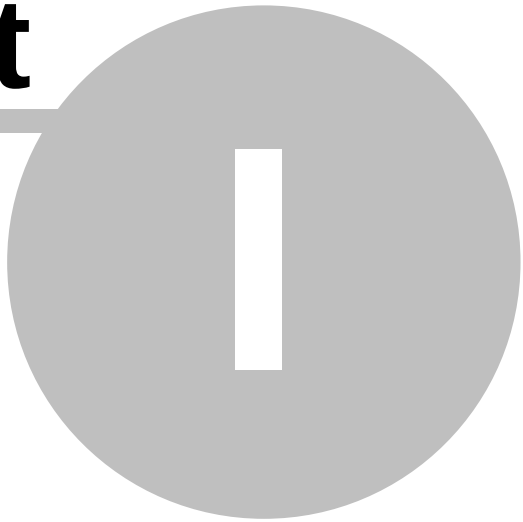
Special thanks to:

All the people who contributed to this document, to mum and dad and grandpa, to my sisters and brothers and mothers in law, to our secretary Kathrin, to the graphic artist who created this great product logo on the cover page (sorry, don't remember your name at the moment but you did a great work), to the pizza service down the street (your daily Capricciosas saved our lives), to the copy shop where this document will be duplicated, and and and...

Last not least, we want to thank EC Software who wrote this great help tool called HELP & MANUAL which printed this document.

Table of Contents

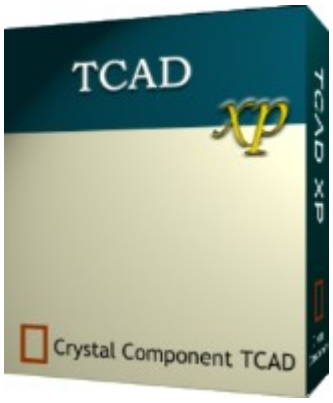
Part



1 Introduction

1.1 What is TCAD for .NET

TCAD for .NET is a component that will help you write vector graphics applications. Shapes can be interacted with by mouse or code. It is easy to use, effective and powerful. It will save you valuable time.



If you want add some CAD-Drawing function into application , using OLE mode, it is too tired to understand its concept..., programming it from zero, There is more work waiting for you! Now, You can using TCAD f or . NET to help you write application.Easy to create and use VECTOR shape in your application developing ,only controled by mouse. Now, You can using TCAD for .NET to help you writing application on .NET environment.

Key Features

Shape Types

Line
LinkLine
PolyLine
Polygon
RuleLine
Rectangle
Ellipse
LinkPoint
LineLinkLine
Image
Text
ElliArc
PolyBezier
User define shapes

Grouping/Ungrouping

Multi-Layers

Create/Move/Resize/Rotate/ shape by code

Save/Load to/from DiskFile/Database

Easy to create user-define shape

Support 4 mode coordinates

Detail infomation

Drawing shapes on the designer canvas by mouse actions or code.
Modifying the drawed shapes.
Support multi-layers, printing/deleting/visible invisible layer(s).
Using all colors possible.
Supporting link line shape
Using different style of pens ,different style of brushes if you need.
Creating text objects with any font installed in the system.
Necessarily shape action related events published.
Using page formats like (A0,A1,A2,A3,A4,letter, etc.) or custom sizes.

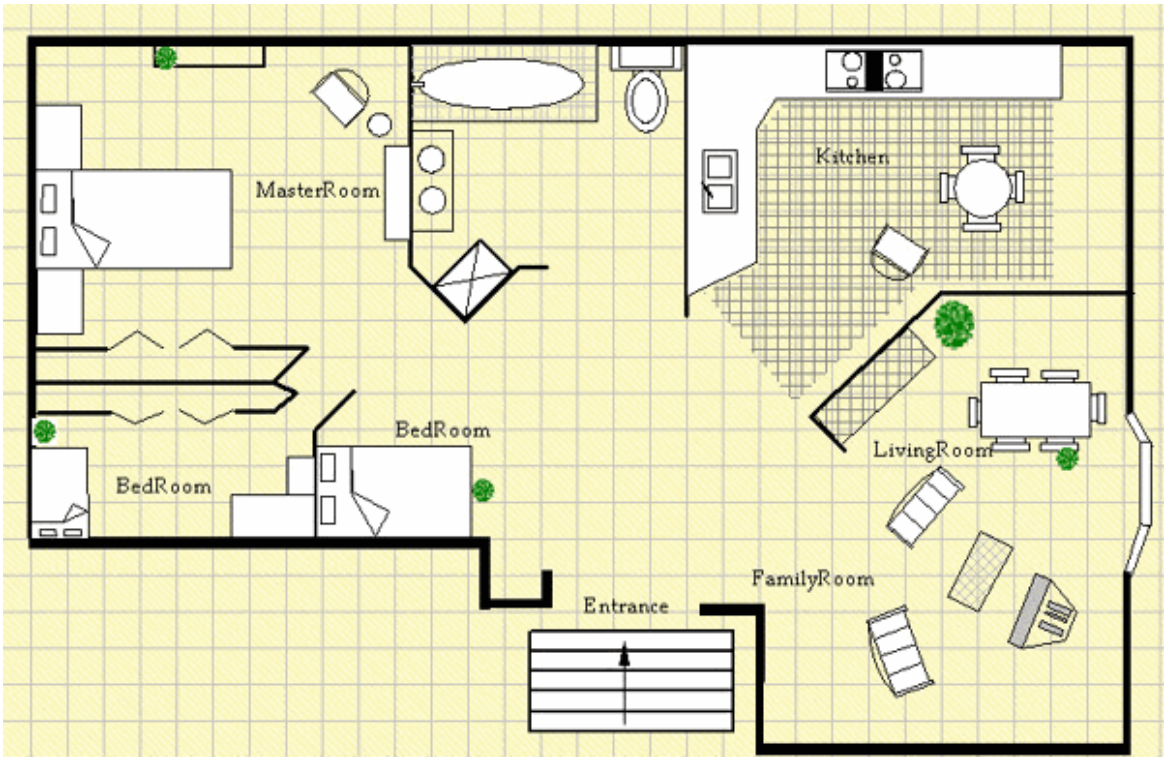
Can Undo and set undo step size
Cutting, copying, pasting and deleting the shapes.
Ordering the shapes(SendToBack, BringToFront, etc.)
Rotating, Dragging and Scaling the shapes by mouse or code.
Aligning the shape in any style.
Easy to create user-define combine shape
Snapping the mouse point to grids,set grid width or height.
Support 24 gradient style fill mode

Locking/Unlocking Shape
Showing HotSpot of a shape or hiding.
Grouping and ungrouping the shapes.
Zooming and panning, viewing the drawing in any scale.
Showing hints when mouse enter a shape.

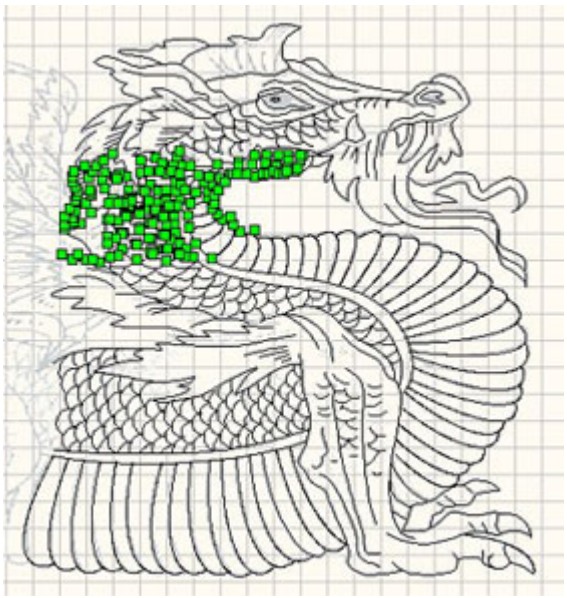
Saving the drawing as disk file or stream(database) and opening it.
Printing the drawing to the printer and/or plotter.

Inserting bitmaps to the drawing.
Scaling, rotating,dragging bitmaps like a shape.
Exporting the drawing as WMF,bitmap,Jpg,dxf(R12) file.

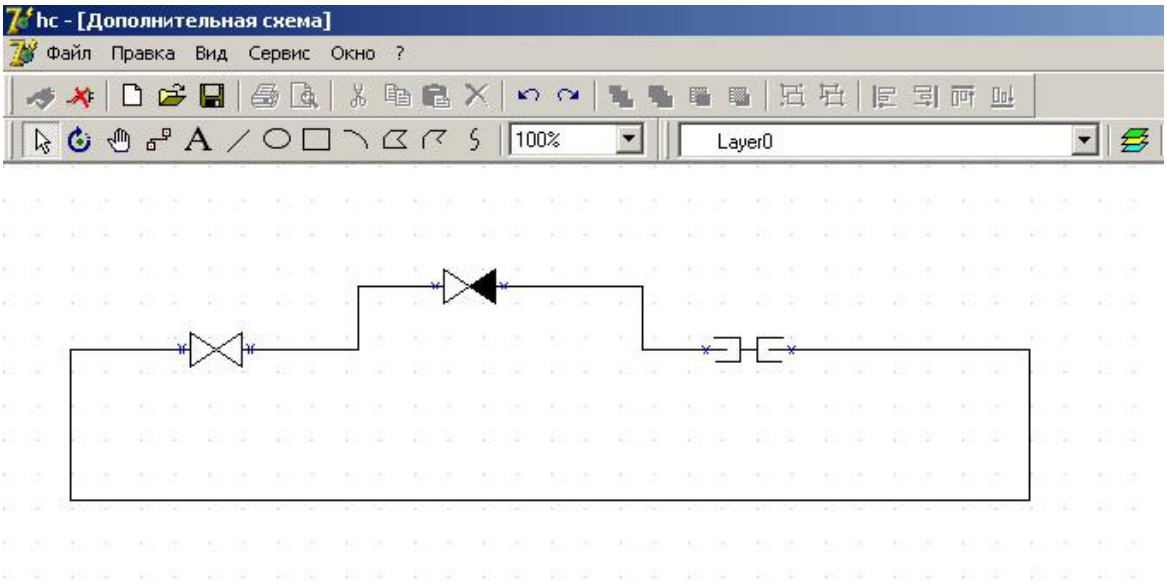
1.2 Application screenshot



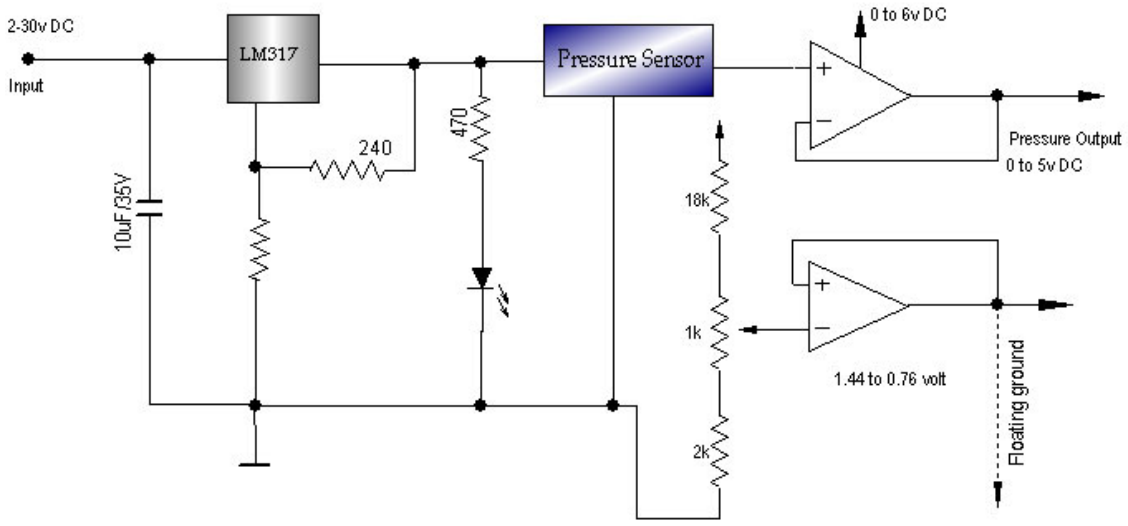
It is a house plan drawing sample,no lib need to draw this,fast and easy.



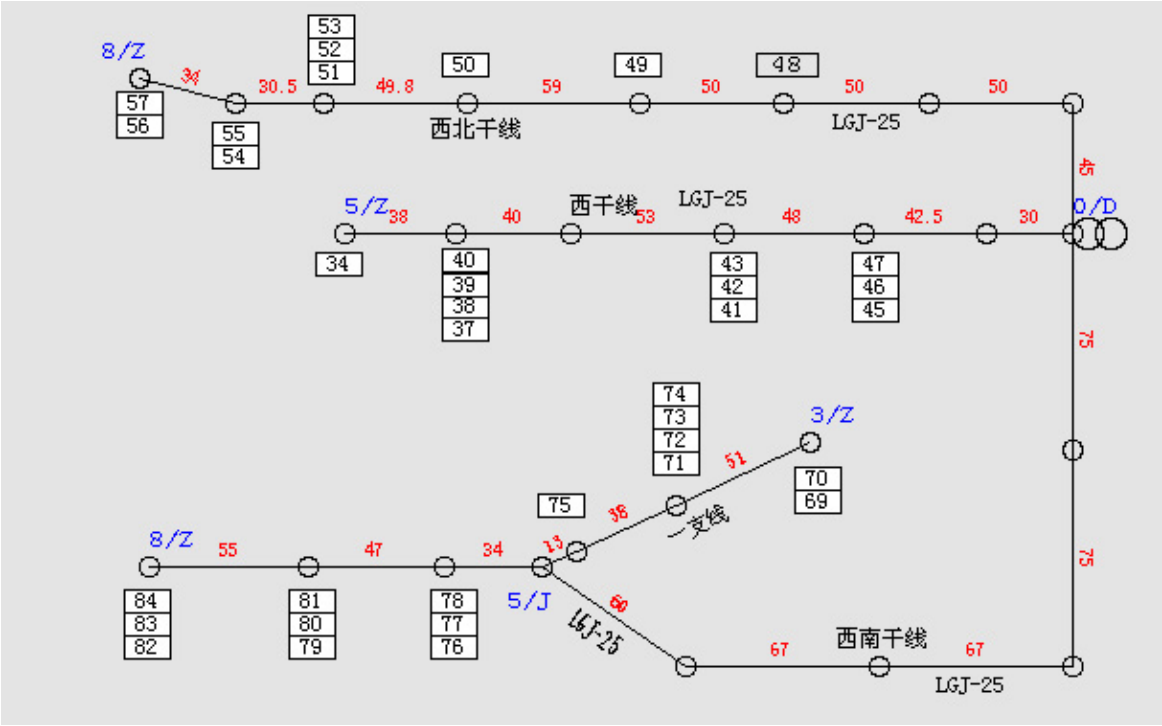
Design of vector drawings from cnc machines .Using TMyElliArc to create this complex dragon.



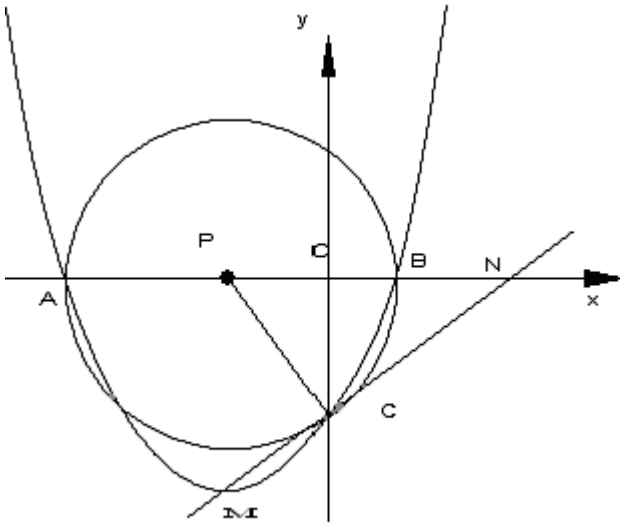
Hydraulic Design



Electric drwaing,library need , and support link line, TCAD is a powerful for " link line style" dr.

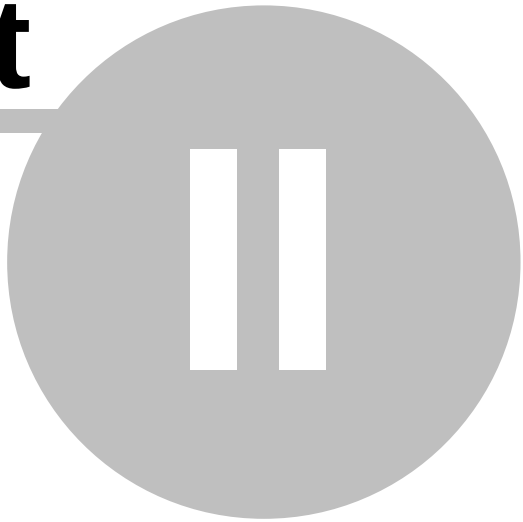


Project:Eletric device manager
Compay:xian college,China



If you are a teacher or student,TCAD can draw math graph,you can copy the graph to MS Word .Creating math graph library,can speed your work.

Part

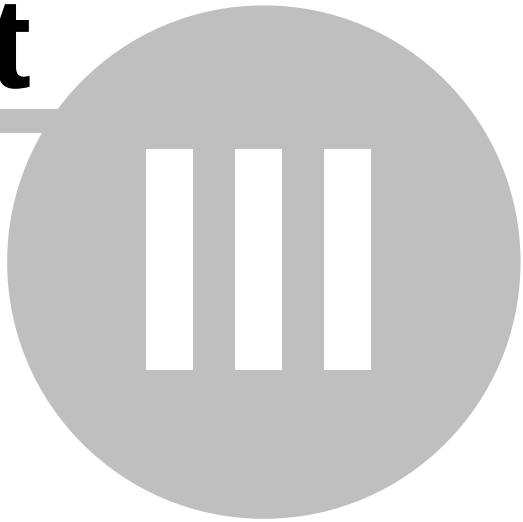


2 Defines

```
public enum MyAlignment
{
	Center,Left,Right
}

public enum MyArcMode
{
	Circle,Ellipse
}

public enum MyArcStyle
{
	Arc,Chord,Sector
}

public enum MyArrowStyle
{
	Double,Left,None,Right
}

public enum MyBackgroundBitmapMode
{
	Center,LeftTop,Stretch,Tiled
}

public enum MyBlockMode
{
	Merge,Import
}

public enum MyDragMode
{
	Both,Horizontal,Vertical
}

public enum MyDrawType
{
Close,Selecting,RotateSelecting,Line,RuleLine,LinkLine,PolyLine,Polygon,PolylinePolygonPointRemoving,P
olylinePolygonPointAdding,Rectangle,Image,Ellipse,Text,LinkPoint,LineLinkLine,PolyBezier,ElliArc,FreeLine
,RoundRectangle,WaveLine
}

public enum MyGridType
{
	Pixel,Line,None
}

public enum MyLinklineDrawStyle
{
	HorizontalVertical,Free
}

public enum MyLinklineEndPoint
{
	Start,End
}

public enum MyLinklineStyle
```

```
{
        Free,Horizontal,Vertical
}

public enum MyPageOrientation
{
        Portrait,Landscape
}

public enum MyPageStyle
{
        A0,A1,A2,A3,A4,A5,B3,B4,B5,Custom
}

public enum MyTickStyle
{
        Line,None
}

public enum MyUnitType
{
        Pixel,Millimeter,Centimeter,Decimeter,Meter,Inch
}

public enum MyVAlignment
{
        Bottom,Middle,Top
}

public enum MyXYMode
{
        Mode0,Mode1,Mode2,Mode3
}
```

Part



3 MyCAD

MyCAD is a powerful 2D drawing component.

3.1 Properties

3.1.1 Alpha

public Byte Alpha {get; set;}

Description:

Set the shape's transparency, value is 0 to 255.

Example:

myShape.Alpha = 100;

3.1.2 ArrowAngle

public Int32 ArrowAngle {get; set;}

value:

Set the Line and PolyLine 's arrow angle. value is between: 0 - 359.

Example:

myCAD1.ArrowAngle = 50;

See also:

ArrowLength
ArrowOffset
ArrowStyle

3.1.3 ArrowLength

public Byte ArrowLength {get; set;}

value:

Set the Line and PolyLine 's arrow Length. value is between: 10-50.

Example:

myCAD1.ArrowLength = 30;

See also:

ArrowAngle
ArrowOffset
ArrowStyle

3.1.4 ArrowOffset

public Byte ArrowOffset {get; set;}

value:

When a Line shape is drawn with an arrow, the ArrowOffset specifies how many pixels from the end of the line the arrow is drawn. Value is between: 0 - 255, default is 0.

Example:

myCAD1.ArrowOffset = 0;

myCAD1.ArrowOffset = 16;

See also:

ArrowAngle
ArrowLength
ArrowStyle

3.1.5 ArrowStyle

public MyArrowStyle ArrowStyle {get; set;}

value:

Set the Line and PolyLine 's arrow style;

Example:

myCAD1.ArrowStyle = MyArrowStyle.Left;

See also:

ArrowAngle
ArrowLength
ArrowOffset

3.1.6 BackgroundBitmap

public Bitmap BackgroundBitmap {get; set;}

value:

Set the background bitmap for MyCAD, And clear it,Set the BackgroundBitmap = null;

Example:

```
openFileDialog1.InitialDirectory = Application.ExecutablePath;
openFileDialog1.Filter = "BMP(*.bmp)|*.bmp|JPEG(*.jpg;*.jpe;*.jpeg)|*.jpg;*.jpe;*.jpeg|All files(*.*)|*.*";
if (openFileDialog1.ShowDialog() == DialogResult.OK)
{
        Bitmap myBitmap = new Bitmap(openFileDialog1.FileName);
        myCAD1.BackgroundBitmap = myBitmap;
}
```

See also:

BackgroundBitmapMode

3.1.7 BackgroundBitmapMode

public MyBackgroundBitmapMode BackgroundBitmapMode {get; set;}

value:

Set the show style of background bitmap.

Example:

myCAD1.BackgroundBitmapMode = MyBackgroundBitmapMode.Center;

See also:

BackgroundBitmap

3.1.8 BackgroundColor

public Color BackgroundColor {get; set;}

value:

Specifies the background color.

Example:

myCAD1.BackgroundColor = Color.White;

3.1.9 Brush

public Brush Brush {get; set;}

value:

Set the brush of MyCAD that you need.

Example:

If you want to know more property about Brush,please check the example named **StartBrush** and read the code.

See also:

BrushShow

3.1.10 BrushShow

public Boolean BrushShow {get; set;}

value:

Show/not show the fill style of Brush.

Example:

myCAD1.BrushShow = true;

See also:

Brush

3.1.11 CrossLine

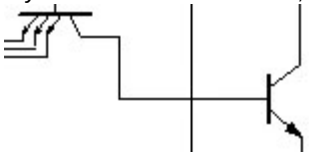
public Boolean CrossLine {get; set;}

value:

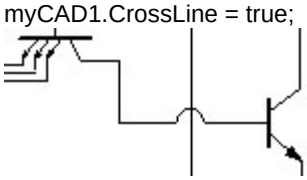
It is used for MyLinkLine drawing feature, when it is true, the drawing speed lower than false;

Example:

myCAD1.CrossLine = false;



myCAD1.CrossLine = true;



See also:

MyLinkLine

3.1.12 DragMode

public MyDragMode DragMode {get; set;}

value:

It can help you drag shape, when value is Horizontal then the shape drag in horizontal when value is Vertical then the shape drag in vertical.

Example:

myCAD1.DragMode = MyDragMode.Horizontal;

3.1.13 DrawType

public MyDrawType DrawType {get; set;}

value:

Set the draw type of mouse drawing.

Example:

myCAD1.DrawType = MyDrawType.Rectangle;

3.1.14 GridColor

public Color GridColor {get; set;}

value:

Set the grid color.

Example:

myCAD1.GridColor = Color.Red;

See also:

GridHeight
GridPenSize
GridShow
GridType
GridWidth

3.1.15 GridHeight

public Int32 GridHeight {get; set;}

value:

Set the grid height.

Example:

myCAD1.GridHeight = 50;

See also:

GridColor
GridPenSize
GridShow
GridType
GridWidth

3.1.16 GridPenSize

public Int32 GridPenSize {get; set;}

value:

Set the grid pen size.

Example:

myCAD1.GridPenSize = 2;

See also:

GridColor
GridHeight
GridShow
GridType
GridWidth

3.1.17 GridShow

public Boolean GridShow {get; set;}

value:

Show or not show the grid.

Example:

myCAD1.GridShow = true;

See also:

GridColor
GridHeight
GridPenSize
GridType
GridWidth

3.1.18 GridType

public MyGridType GridType {get; set;}

value:

Set the grid type.

Example:

myCAD1.GridType = MyGridType.Pixel;

See also:

GridColor
GridHeight
GridPenSize

GridShow
GridWidth

3.1.19 GridWidth

public Int32 GridWidth {get; set;}

value:
Set the grid width.

Example:
myCAD1.GridWidth = 50;

See also:
GridColor
GridHeight
GridPenSize
GridShow
GridType

3.1.20 HotColor

public Color HotColor {get; set;}

value:
Specifies the color of the hot square.

Example:
myCAD1.HotColor = Color.Yellow;

See also:
HotShow
HotSize

3.1.21 HotShow

public Boolean HotShow {get; set;}

value:
Show or not the hot square.

Example:
myCAD1.HotShow = false;

See also:
HotColor
HotSize

3.1.22 HotSize

public Byte HotSize {get; set;}

value:
Set the hot size.

Example:

```
myCAD1.HotSize = 8;
```

See also:

HotColor
HotShow

3.1.23 LabelValue

```
public Label LabelValue {get; set;}
```

value:

Show the parameter about the current shape. It can show the length of a MyLine or height and width of a MyRectangle.

Example:

```
myCAD1.LabelValue = Label1;
```

See also:

LabelXY

3.1.24 LabelXY

```
public Label LabelXY {get; set;}
```

value:

Show current mouse cursor 's coordinate, it is changed by mouse moving;

Example:

```
myCAD1.LabelXY = Label1;
```

See also:

LabelValue

3.1.25 LinklineAroundShape

```
public Boolean LinklineAroundShape {get; set;}
```

value:

Move around a shape or not when drawing a link line to cross a shape.

Example:

```
myCAD1.LinklineAroundShape = true;
```

See also:

LinklineDrawStyle

3.1.26 LinklineDrawStyle

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

3.1.27 OperateAllLayer

public Boolean OperateAllLayer {get; set;}

value:

Specifies the operate object is the all layers or not.

Example:

myCAD1.OperateAllLayer = false;

3.1.28 PageFoot

public String PageFoot {get; set;}

value:

Specifies the foot text of the page.

Example:

myCAD1.PageFoot = "TCAD for .NET";

See also:

PageFootAlignment
PageFootFont
PageFootToBottom

3.1.29 PageFootAlignment

public MyAlignment PageFootAlignment {get; set;}

value:

Specifies the position of page foot.

Example:

myCAD1.PageFootAlignment = MyAlignment.Center;

See also:

PageFoot
PageFootFont
PageFootToBottom

3.1.30 PageFootFont

public Font PageFootFont {get; set;}

value:
Specifies the font of page foot.

Example:
myCAD1.PageFootFont = myCAD1.Font;

See also:
PageFoot
PageFootAlignment
PageFootToBottom

3.1.31 PageFootToBottom

public Byte PageFootToBottom {get; set;}

value:
Specifies the margin of page foot to bottom.

Example:
myCAD1.PageFootToBottom = 20;

See also:
PageFoot
PageFootAlignment
PageFootFont

3.1.32 PageHead

public String PageHead {get; set;}

value:
Specifies the head text of the page.

Example:
myCAD1.PageHead = "TCAD for .NET";

See also:
PageHeadAlignment
PageHeadFont
PageHeadToTop

3.1.33 PageHeadAlignment

public MyAlignment PageHeadAlignment {get; set;}

value:
Specifies the position of page head.

Example:
myCAD1.PageHeadAlignment = MyAlignment.Right;

See also:
PageHead

PageHeadFont
PageHeadToTop

3.1.34 PageHeadFont

public Font PageHeadFont {get; set;}

value:
Specifies the font of page head.

Example:
myCAD1.PageHeadFont = myCAD1.Font;

See also:
PageHead
PageHeadAlignment
PageHeadToTop

3.1.35 PageHeadToTop

public Byte PageHeadToTop {get; set;}

value:
Specifies the margin of page head to top.

Example:
myCAD1.PageHeadToTop = 20;

See also:
PageHead
PageHeadAlignment
PageHeadFont

3.1.36 PageHeight

public Int64 PageHeight {get; set;}

value:
Specifies the height of page.

Example:
myCAD1.PageHeight = 200;

See also:
PageWidth

3.1.37 PageOrientation

public MyPageOrientation PageOrientation {get; set;}

value:
Specifies the orientation of page.

Example:
myCAD1.PageOrientation = MyPageOrientation.Landscape;

3.1.38 PageStyle

public MyPageStyle PageStyle {get; set;}

value:
Specifies the style of page.

Example:
myCAD1.PageStyle = MyPageStyle.A4;

3.1.39 PageWidth

public Int64 PageWidth {get; set;}

value:
Specifies the width of page.

Example:
myCAD1.PageWidth = 200;

See also:
PageHeight

3.1.40 Pen

public Pen Pen {get; set;}

value:
Set the pen of MyCAD that you need.

Example:
myCAD1.Pen.Width = 2;
myCAD1.Pen.Color = Color.Green;

3.1.41 PrintBackground

public Boolean PrintBackground {get; set;}

value:
Specifies the background print or not.

Example:
myCAD1.PrintBackground = true;

See also:
PrintBorder

3.1.42 PrintBorder

public Boolean PrintBorder {get; set;}

value:
Specifies the border print or not.

Example:
myCAD1.PrintBorder = true;

See also:

PrintBackground

3.1.43 PrintBorderToBottom

public Byte PrintBorderToBottom {get; set;}

value:
Specifies the margin of border to bottom.

Example:
myCAD1.PrintBorderToBottom = 50;

See also:
PrintBorderToLeft
PrintBorderToRight
PrintBorderToTop

3.1.44 PrintBorderToLeft

public Byte PrintBorderToLeft {get; set;}

value:
Specifies the margin of border to left.

Example:
myCAD1.PrintBorderToLeft = 50;

See also:
PrintBorderToBottom
PrintBorderToRight
PrintBorderToTop

3.1.45 PrintBorderToRight

public Byte PrintBorderToRight {get; set;}

value:
Specifies the margin of border to right.

Example:
myCAD1.PrintBorderToRight = 50;

See also:
PrintBorderToBottom
PrintBorderToLeft
PrintBorderToTop

3.1.46 PrintBorderToTop

public Byte PrintBorderToTop {get; set;}

value:
Specifies the margin of border to top.

Example:
myCAD1.PrintBorderToTop = 50;

See also:

PrintBorderToBottom
PrintBorderToLeft
PrintBorderToRight

3.1.47 Ratio

public Double Ratio {get; set;}

value:

Specifies the ratio of page. It is very useful, it will be appear at property LabelValue, that is Line's Length and area for MyLine, MyRectangle, MyEllisple and other shapes.

Example:

myCAD1.Ratio = 50;

See also:

UnitType

3.1.48 ResizeEnable

public Boolean ResizeEnable {get; set;}

value:

Specifies the shape can resize or not.

Example:

myCAD1.ResizeEnable = false;

See also:

RotateEnable

3.1.49 ReturnToSelecting

public Boolean ReturnToSelecting {get; set;}

value:

Specifies the DrawType property as Selecting when drawed a shape.

Example:

myCAD1.ReturnToSelecting = false;

3.1.50 RotateConstraintDegree

public Int32 RotateConstraintDegree {get; set;}

value:

Set the constraint degree when rotating,if value is zero,it can rotate freely.

Example:

myCAD1.RotateConstraintDegree = 30;

3.1.51 RotateEnable

public Boolean RotateEnable {get; set;}

value:
Specifies the shape can rotate or not.

Example:
myCAD1.RotateEnable = false;

See also:
ResizeEnable

3.1.52 ShowHotLink

public Boolean ShowHotLink {get; set;}

value:
Whether show the MyCAD displays hot link point for a shape.

Example:
myCAD1.ShowHotLink = true;

3.1.53 Snap

public Boolean Snap {get; set;}

value:
Snap mouse to grid or not, help you drawing.

Example:
myCAD1.Snap = true;

See also:
SnapPixel

3.1.54 SnapPixel

public Byte SnapPixel {get; set;}

value:
The mouse point will be capture to the grid when the pixels low than this value between mouse point and nearest grid. It can not big than GridWidth or GridHeight.

Example:
myCAD1.Snap = true;

See also:
SnapPixel

3.1.55 SnapShape

public Boolean SnapShape {get; set;}

value:
When drag or resize shape, whether snap mouse to other shape, it can help you drawing.

Example:

myCAD1.SnapShape = true;

3.1.56 UndoRedoSize

public Byte UndoRedoSize {get; set;}

value:

Set the size (steps) of undo action saving. It costs system resource.

Example:

myCAD1.UndoRedoSize = 8;

3.1.57 UnitType

public MyUnitType UnitType {get; set;}

value:

Which unit do you use, It will be appear at property LabelValue, that is Line's Length or area for MyLine, MyRectangle, MyEllisple and other shapes.

Example:

myCAD1.UnitType = MyUnitType.Millimeter;

3.1.58 Version

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

3.1.59 XYMode

public MyXYMode XYMode {get; set;}

value:

There four mode to choose, to fit your need.

Example:

myCAD1.XYMode = MyXYMode.Mode0;

3.1.60 Zoom

property Zoom:Double

Description:

Set the zoom value,the width,height, shape and background will resize automatically.When
Zoom is 1 , TMyCAD is showed in 100%.

Example:

myCAD1.Zoom = 0.5;

3.2 Methods

3.2.1 AddBlockfromTCADFile

public Boolean AddBlockfromTCADFile(String fileName,MyBlockMode blockMode)

Description:

Add block from an exist tcad file, this file can be a template drawing.

Parameter:

fileName: The tcad file that you want to add
blockMode: The style of add a image

Return value:

true: Add successfully
false: Add unsuccessfully

Example:

```
private void menuItem27_Click(object sender, System.EventArgs e)
{
        openFileDialog1.InitialDirectory = Application.StartupPath;
        openFileDialog1.Filter = "TCAD drawing file(*.tcad)|*.tcad";
        openFileDialog1.FilterIndex = 1;
        if (openFileDialog1.ShowDialog() == DialogResult.OK)
        {

myCAD1.AddBlockfromTCADFile(openFileDialog1.FileName,MyBlockMode.Merge);
        }
}
```

3.2.2 AddImageShapeByCode

public Int64 AddImageShapeByCode(String shapeName,Point ltPoint,Bitmap b)

Description:

Add image shape by code, if you want add other shape, please use procedure AddShapeByCode or AddUserDefineShapeFromLib

Parameter:

shapeName: Image shape's name
ltPoint: The point of the image's Lfte -Top corner.
b: A image that you want add into TCAD

Return value:

Return value =-1,add shape unsuccessfully.
Return value >=0,add shape successfully,the value is the ShapeId of the new shape

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Bitmap b = new Bitmap(@"d:\images\cover.bmp");
        myCAD1.AddImageShapeByCode("Image1",new Point(100,100),b);
}
```

3.2.3 AddShapeByCode

public Int64 AddShapeByCode(MyCAD owner,String shapeName,MyDrawType drawType,PointF[] thePoints,Single angle,String text)

Description:

Add shape by code, it is useful for automatic drawing. if you want add a image, please use AddImageShapeByCode.

Parameter:

owner: instance of MyCAD
shapeName: Shape's name
drawType: The draw style of a shape
thePoints: Points array
angle: The shape's angle
text: It is of MyText only,set the text

Return value:

Return value =-1,add shape unsuccessfully.
Return value >=0,add shape successfully,the value is the ShapeId of the new shape

Example:

```
private void menuItem14_Click(object sender, System.EventArgs e)
{
        PointF[] thePoints = new PointF[2];
        thePoints[0] = new PointF(100,100);
        thePoints[1] = new PointF(200,200);
        myCAD1.AddShapeByCode(myCAD1,"lineShape",MyDrawType.Line,thePoints,0,"");
}
```

3.2.4 AddUserDefineShapeFromLib

public Int32 AddUserDefineShapeFromLib(MyLibrary myLibrary,String udShapeName,Int32 centerX,Int32 centerY,MyCAD myCAD,String encoding)

Description:

Get from library by user-defined shape's name. and add user-defined shape into MyCAD instance.

Parameter:

myLibrary: Instance of MyLibrary
ud*ShapeName:* User-defined shape's name
centerX,centerY: The shape's center position that you want placed
myCAD: Instance of MyCAD
encoding: The coding of character

Return value:

Return value =-1,add shape unsuccessfully.
Return value >=0,add shape successfully,the value is the ShapeId of the new shape

Example:

```
private void myCAD1_DragDrop(object sender, System.Windows.Forms.DragEventArgs e)
{
        if (e.Data.GetDataPresent(typeof(string)))
        {
                string strData = (string)e.Data.GetData(typeof(string));

myCAD1.AddUserDefineShapeFromLib(myLibrary,strData,clientPoint.X,clientPoint.Y,myCAD1,"gb2312");
        }
}
```

3.2.5 AlignBottom

public void AlignBottom()

Description:

When you select more than one shape, this proceudre can align them bottom.

Example:

```
private void menuAlignBottom_Click(object sender, System.EventArgs e)
{
        myCAD1.AlignBottom();
}
```

3.2.6 AlignHorizontalCenter

public void AlignHorizontalCenter()

Description:

When you select more than one shape, this proceudre can align them horizontal center.

Example:

```
private void menuAlignHorizontalCenter_Click(object sender, System.EventArgs e)
{
        myCAD1.AlignHorizontalCenter();
}
```

3.2.7 AlignLeft

public void AlignLeft()

Description:

When you select more than one shape, this proceudre can align them left.

Example:

```
private void menuAlignLeft_Click(object sender, System.EventArgs e)
{
        myCAD1.AlignLeft();
}
```

3.2.8 AlignRight

public void AlignRight()

Description:

When you select more than one shape, this proceudre can align them right.

Example:

```
private void menuAlignRight_Click(object sender, System.EventArgs e)
{
        myCAD1.AlignRight();
}
```

3.2.9 AlignTop

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper

eget, rutrum a, tempor in, nibh.

3.2.10 AlignVerticalCenter

public void AlignVerticalCenter()

Description:

When you select more than one shape, this proceudre can align them vertical center.

Example:

```
private void menuAlignVerticalCenter_Click(object sender, System.EventArgs e)
{
        myCAD1.AlignVerticalCenter();
}
```

3.2.11 BringToFront

public void BringToFront(MyShape tmpShape,Boolean needSaved)

Description:

Sent the selected shape to front.

Parameter:

tmpShape: The shape that you will change
needSaved: Weather save this operation to undo list

Example:

```
private void menuEditBringToFront_Click(object sender, System.EventArgs e)
{
        myCAD1.BringToFront(myCAD1.GetSelectedShape(),true);
}
```

See also:

SendToBack

3.2.12 BringToFrontByStep

public void BringToFrontByStep(MyShape tmpShape,Boolean needSaved)

Description:

Sent the selected shape to front by a step

Parameter:

tmpShape: The shape that you will change
needSaved: Weather save this operation to undo list

Example:

```
private void menuEditBringToFrontByStep_Click(object sender, System.EventArgs e)
{
        myCAD1.BringToFrontByStep(myCAD1.GetSelectedShape(),true);
}
```

See also:

SendToBackByStep

3.2.13 ClearAllUndoStuff

public void ClearAllUndoStuff()

Description:

Clear all undo stuff in memory,it is be used at start new drawing.

Example:

```
myCAD1.ClearAllUndoStuff();
```

3.2.14 Copy

public void Copy()

Description:

Save selected shape to memory, and they also be saved in bitmap format in clipboard.

Example:

```
private void menuEditCopy_Click(object sender, System.EventArgs e)
{
        myCAD1.Copy();
}
```

See also:

Cut
Paste

3.2.15 CreateLink

public Int64 CreateLink(String linkName,MyShape srcShape,Int32 srcLinkPtId,MyShape destShape,Int32 destLinkPtId)

Description:

To create link line shape between source shape and dest shape.

Parameter:

linkName: The link line's name
srcShape: The source shape
srcLinkPtId: The link point id
destShape: The dest shape
destLinkPtId: The link point id

Return value:

Return value =-1,create link line unsuccessfully
Return value >=0,create link line successfully,the value is the ShapeId of the link line

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.CreateLink("LinkLine",myCAD1.MyShapes[0],0,myCAD1.MyShapes[1],0);
}
```

3.2.16 Cut

public void Cut()

Description:

Save selected shape to memory, delete selected shape, and they also saved in bitmap

format in clipboard.

Example:

```
private void menuEditCut_Click(object sender, System.EventArgs e)
{
        myCAD1.Cut();
}
```

See also:

Copy
Paste

3.2.17 Delete

public void Delete()

Description:

Delete the selected shape.

Example:

```
private void menuEditDelete_Click(object sender, System.EventArgs e)
{
        myCAD1.Delete();
}
```

3.2.18 DeleteAllLayers

public Boolean DeleteAllLayers()

Description:

Delete all layers and all shapes,the CurrentLayerId as -1.

Return value:

true: Delete successfully
false: Delete unsuccessfully

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.DeleteAllLayers();
}
```

3.2.19 DeleteAllShapes

public void DeleteAllShapes()

Description:

Delete all shapes , Layer(s) is still exist.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.DeleteAllShapes();
}
```

3.2.20 DeleteLayerById

public Boolean DeleteLayerById(Int32 layerId)

Description:

Delete layer by the layer's id, if layerId not matched, it returns false. This function will delete shape(s) which belonged this layer.

Parameter:

layerId: Layer id

Return value:

true: Delete successfully
false: Delete unsuccessfully

Example:

```
private void btnDelete_Click(object sender, System.EventArgs e)
{
        Int32 id;
        if (treeView1.SelectedNode.Index > -1)
        {
                id = mainForm.myCAD1.GetLayerIdByNo(treeView1.SelectedNode.Index);
                mainForm.myCAD1.DeleteLayerById(id);
                RefreshIt();
        }
        treeView1.Focus();
}
```

See also:

DeleteLayerByName

3.2.21 DeleteLayerByName

public Boolean DeleteLayerByName(String layerName)

Description:

Delete layer by the layer's name, if layerName not matched, it returns false. this function will delete shape(s) which belonged this layer.

Parameter:

layerName: Layer name

Return value:

true: Delete successfully
false: Delete unsuccessfully

Example:

```
private void btnDelete_Click(object sender, System.EventArgs e)
{
        myCAD1.DeleteLayerByName("Layer0");
}
```

See also:

DeleteLayerById

3.2.22 DeleteSelectedShapes

public Boolean DeleteSelectedShapes()

Description:

Delete current selected shape(s).

Return value:

true: Delete successfully
false: Delete unsuccessfully

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.DeleteSelectedShapes();
}
```

3.2.23 DeleteShapeById

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

3.2.24 DeSelectedAllShapesByCode

public void DeSelectedAllShapesByCode()

Description:

After execute this command, no shape will be in selected state.

Example:

```
private void menuItem7_Click(object sender, System.EventArgs e)
{
        myCAD1.DeSelectedAllShapesByCode();
}
```

3.2.25 FlipHorizontal

public void FlipHorizontal(MyShape aShape)

Description:
Flip a shape horizontal.

Parameter:
aShape: The shape you want to flip

Example:
```
private void menuEditFlipHorizontal_Click(object sender, System.EventArgs e)
{
        myCAD1.FlipHorizontal(myCAD1.GetSelectedShape());
}
```

See also:
FlipVertical

3.2.26 FlipVertical

public void FlipVertical(MyShape aShape)

Description:
Flip a shape vertical.

Parameter:
aShape: The shape you want to flip

Example:
```
private void menuEditFlipVertical_Click(object sender, System.EventArgs e)
{
        myCAD1.FlipVertical(myCAD1.GetSelectedShape());
}
```

See also:
FlipHorizontal

3.2.27 GetLayerIdByName

public Int32 GetLayerIdByName(String layerName)

Description:
Get the layer id by layer name.

Parameter:
layerName: The layer name

Return value:
Return value =-1,get layer id unsuccessfully.
Return value >=0,get layer id successfully,the value is the layer id.

Example:
```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 layerId = myCAD1.GetLayerIdByName("Layer0");
}
```

See also:

GetLayerIdByNo

3.2.28 GetLayerIdByNo

public Int32 GetLayerIdByNo(Int32 no)

Description:

Get the layer id by layer no.

Parameter:

no: The number that you want to get

Return value:

Return value =-1,get layer id unsuccessfully.
Return value >=0,get layer id successfully,the value is the layer id.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 layerId = myCAD1.GetLayerIdByNo(0);
}
```

See also:

GetLayerIdByName

3.2.29 GetLayerNameById

public String GetLayerNameById(Int32 layerId)

Description:

Get the layer name by layer id.

Parameter:

layerId: The layer id

Return value:

Return value ="",get layer name unsuccessfully.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        String layerName = myCAD1.GetLayerNameById(0);
}
```

3.2.30 GetLayerNoById

public Int32 GetLayerNoById(Int32 layerId)

Description:

Get the layer no by layer id.

Parameter:

layerId: The layer id

Return value:

Return value =-1,get layer no unsuccessfully.
Return value >=0,get layer no successfully,the value is the layer no.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 layerNo = myCAD1.GetLayerNoById(0);
}
```

3.2.31 GetLayerNoByName

public Int32 GetLayerNoByName(String layerName)

Description:

Get the layer no by layer name.

Parameter:

layerName: The layer name

Return value:

Return value =-1,get layer no unsuccessfully.
Return value >=0,get layer no successfully,the value is the layer no.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 layerNo = myCAD1.GetLayerNoByName("Layer0");
}
```

3.2.32 GetLayersCount

public Int32 GetLayersCount()

Description:

Get the layers count.

Return value:

return the layers count

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 layerCount = myCAD1.GetLayersCount();
}
```

3.2.33 GetMaxLayerId

public Int32 GetMaxLayerId()

Description:

Get the max layer id.

Return value:

return the max layer id

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 maxLayerId = myCAD1.GetMaxLayerId();
}
```

3.2.34 GetRootParentShape

public MyShape GetRootParentShape(MyShape aShape)

Description:
Get the root parent shape.

Parameter:
aShape: The shape

Return value:
retrun the root parent shape

Example:
```
private void button1_Click(object sender, System.EventArgs e)
{
        MyShape rootShape = myCAD1.GetRootParentShape(myCAD1.GetSelectedShape());
}
```

3.2.35 GetSelectedShape

public MyShape GetSelectedShape()

Description:
Get the selected shape.

Return value:
retrun the selected shape

Example:
```
private void button1_Click(object sender, System.EventArgs e)
{
        MyShape aShape = myCAD1.GetSelectedShape();
}
```

3.2.36 GetSelectedShapes

public MyShape[] GetSelectedShapes()

Description:
Get the selected shapes.

Return value:
retrun the selected shapes

Example:
```
private void button1_Click(object sender, System.EventArgs e)
{
        MyShape[] lstShape = myCAD1.GetSelectedShapes();
}
```

3.2.37 GetSelectedShapesCount

public Int64 GetSelectedShapesCount()

Description:
Get the selected shapes count.

Return value:

retrun the selected shapes count

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 count = myCAD1.GetSelectedShapesCount();
}
```

3.2.38 GetShapeById

public MyShape GetShapeById(Int64 aShapeId)

Description:

Get the shape by id.

Parameter:

aShapeId: Shape id

Return value:

return the shape.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        MyShape tmpShape = myCAD1.GetShapeById(0);
}
```

See also:

GetShapeByNo

3.2.39 GetShapeByNo

public MyShape GetShapeByNo(Int64 aShapeNo)

Description:

Get the shape by no.

Parameter:

aShapeNo: Shape no

Return value:

return the shape.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        MyShape tmpShape = myCAD1.GetShapeByNo(0);
}
```

See also:

GetShapeById

3.2.40 GetShapeNoById

public Int64 GetShapeNoById(Int64 aShapeId)

Description:

Get the shape no by id.

Parameter:

aShapeId: Shape id

Return value:

return the shape no.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 shapeNo = myCAD1.GetShapeNoById(0);
}
```

3.2.41 GetShapesCount

public Int64 GetShapesCount()

Description:

Get the shapes count.

Return value:

return the shapes count.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 count = myCAD1.GetShapesCount();
}
```

3.2.42 GetShapesCountInALayer

public Int64 GetShapesCountInALayer(Int32 layerId)

Description:

Get the shapes count in a layer.

Parameter:

layerId: Layer id

Return value:

return the shapes count in a layer.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 count = myCAD1.GetShapesCountInALayer(0);
}
```

3.2.43 GetWorkingShapesCount

public Int64 GetWorkingShapesCount()

Description:

Get the working shapes count.

Return value:

return the working shapes count.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        Int32 count = myCAD1.GetWorkingShapesCount();
}
```

3.2.44 GroupWorkingShapes

public Int64 GroupWorkingShapes()

Description:

Group the working shapes.

Return value:

Return value =-1,group shapes unsuccessfully.
Return value >=0,group shapes successfully,the value is the group shape id.

Example:

```
private void menuEditGroup_Click(object sender, System.EventArgs e)
{
        myCAD1.GroupWorkingShapes();
}
```

See also:

UngroupShape

3.2.45 InVisibleLayerById

public void InVisibleLayerById(Int32 layerId)

Description:

Set the layer hide by layer id.

Parameter:

layerId: Layer id

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.InVisibleLayerById(0);
}
```

See also:

InVisibleLayerByName

3.2.46 InVisibleLayerByName

public void InVisibleLayerByName(String layerName)

Description:

Set the layer hide by layer name.

Parameter:

layerName: Layer name

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
```

```
        myCAD1.InVisibleLayerByName("Layer0");
}
```

See also:

[InVisibleLayerById]

3.2.47 IsVisibleLayerByID

public Boolean IsVisibleLayerByID(Int32 layerId)

Description:

Is the layer is visible or not.

Parameter:

layerId: Layer id

Return value:

true: Visible
false: Not visible

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.IsVisibleLayerByID(0);
}
```

3.2.48 LoadFromFile

public Boolean LoadFromFile(String fileName,String encoding)

Description:

Load from a file.

Parameter:

fileName: The file you want to load.

Return value:

true: Load successfully
false: Load unsuccessfully
encoding: The coding of character

Example:

```
private void menuFileOpen_Click(object sender, System.EventArgs e)
{
        openFileDialog1.InitialDirectory = Application.StartupPath;
        openFileDialog1.Filter = "TCAD drawing file(*.tcad)|*.tcad";
        openFileDialog1.FilterIndex = 1;
        if (openFileDialog1.ShowDialog() == DialogResult.OK)
        {
                myCAD1.LoadFromFile(openFileDialog1.FileName,"gb2312");
                EditingFileName = openFileDialog1.FileName;
                AdjustCanvas();
        }
}
```

See also:

[SaveToFile]

3.2.49 LoadFromStream

public Boolean LoadFromStream(FileStream fileStream,String encoding)

Description:

Load from a file stream.

Parameter:

fileStream: File stream

Return value:

true: Load successfully
false: Load unsuccessfully
encoding: The coding of character

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.LoadFromStream(myFileStream,"gb2312");
}
```

See also:

SaveToStream

3.2.50 NewLayer

public Int32 NewLayer(String lName,Boolean lVisible)

Description:

Create a new layer.

Parameter:

lName: The layer name
lVisible: The layer visible or not

Return value:

Return the layer id

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.NewLayer("Layer" + Convert.ToString(myCAD1.GetMaxLayerId()+1),true);
}
```

3.2.51 Paste

public void Paste()

Description:

Paste saved shapes to MyCAD, they have 4 pixels offset.

Example:

```
private void menuEditPaste_Click(object sender, System.EventArgs e)
{
        myCAD1.Paste();
}
```

See also:

Copy
Cut

3.2.52 PopfromUndoRedoShapeList

public Int32 PopfromUndoRedoShapeList()

Description:
Undo one step.

Return value:
Return the step number

Example:
```
private void menuEditUndo_Click(object sender, System.EventArgs e)
{
        myCAD1.PopfromUndoRedoShapeList();
}
```

3.2.53 Preview

public void Preview(Int32[] lstLayer,ref Bitmap previewBitmap,Double previewZoom)

Description:
Print preview.

Parameter:
lstLayer: Layer list
previewBitmap: Preview bitmap
previewZoom: Zoom size

Example:
```
private void PrintPreviewForm_Load(object sender, System.EventArgs e)
{
        Int32[] lstLayer = new Int32[1];
        lstLayer[0] = 0;
        Bitmap b = new Bitmap(1,1);
        mainForm.myCAD1.Preview(lstLayer,ref b,1);

        pictureBox1.Image = b;

}
```

3.2.54 Print

public void Print(Int32[] lstLayer,Double previewZoom)

Description:
Print.

Parameter:
lstLayer: Layer list
previewZoom: Zoom size

Example:
```
private void btnPrint_Click(object sender, System.EventArgs e)
{
        //
```

```
            Int32[] lstLayer = new Int32[1];
            lstLayer[0] = 0;
            mainForm.myCAD1.Print(lstLayer,1);
        }
```

3.2.55 SaveToFile

public Boolean SaveToFile(String fileName)

Description:

Save drawing to a file.

Parameter:

fileName: The file name

Return value:

true: Save successfully
false: Save unsuccessfully

Example:

```
private void menuSaveAs_Click(object sender, System.EventArgs e)
{
        saveFileDialog1.InitialDirectory = Application.StartupPath;
        saveFileDialog1.Filter = "TCAD drawing file(*.tcad)|*.tcad";
        saveFileDialog1.FilterIndex = 1;
        saveFileDialog1.DefaultExt = "tcad";
        if (saveFileDialog1.ShowDialog() == DialogResult.OK)
        {
                if (myCAD1.SaveToFile(saveFileDialog1.FileName) == true)
                        MessageBox.Show("MyCAD1 has been saved to " +
saveFileDialog1.FileName);
        }
}
```

See also:

LoadFromFile

3.2.56 SaveToImage

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat

volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

3.2.57 SaveToStream

public Boolean SaveToStream(FileStream fileStream)

Description:

Save drawing to a file stream.

Parameter:

fileStream: The file stream

Return value:

true: Save successfully
false: Save unsuccessfully

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.SaveToStream(myFileStream);
}
```

See also:

LoadFromStream

3.2.58 SelectAllShapes

public void SelectAllShapes()

Description:

Select all shapes.

Example:

```
private void menuItem6_Click(object sender, System.EventArgs e)
{
        myCAD1.SelectAllShapes();
}
```

3.2.59 SelectShapeByCode

public Boolean SelectShapeByCode(Int64 aShapeId,Boolean removePrevSelectedShape)

Description:

Select all shapes.

Parameter:

aShapeId: the shape id that you want to select
removePreSelectedShape: Weather remove the working previous shape or not

Return value:

true: Select successfully
false: Select unsuccessfully

Example:

```
private void menuItem5_Click(object sender, System.EventArgs e)
{
        myCAD1.SelectShapeByCode(1,false);
}
```

3.2.60 SendToBack

public void SendToBack(MyShape tmpShape,Boolean needSaved)

Description:

Sent the selected shape to bakc.

Parameter:

tmpShape: The shape that you will change
needSaved: Weather save this operation to undo list

Example:

```
private void menuEditSendToBack_Click(object sender, System.EventArgs e)
{
        myCAD1.SendToBack(myCAD1.GetSelectedShape(),true);
}
```

See also:

BringToFront

3.2.61 SendToBackByStep

public void SendToBackByStep(MyShape tmpShape,Boolean needSaved)

Description:

Sent the selected shape to back by a step

Parameter:

tmpShape: The shape that you will change
needSaved: Weather save this operation to undo list

Example:

```
private void menuEditSendToBackByStep_Click(object sender, System.EventArgs e)
{
        myCAD1.SendToBackByStep(myCAD1.GetSelectedShape(),true);
}
```

See also:

BringToFrontByStep

3.2.62 SetLayerNameById

public void SetLayerNameById(String layerName,Int32 layerId)

Description:

Set the layer name by layer id

Parameter:

layerName: The layer name that you want to set
layerId: The layer id

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.SetLayerNameById("LayerOK",0);
}
```

See also:

SetLayerNameByName

3.2.63 SetLayerNameByName

public void SetLayerNameByName(String oldName,String newName)

Description:

Set the layer name by layer name

Parameter:

oldName: The old layer name
newName: The layer name that you want to set

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.SetLayerNameByName("oldLayer","newLayer");
}
```

See also:

SetLayerNameById

3.2.64 ShapeMove

public void ShapeMove(MyShape tmpShape,Single dx,Single dy)

Description:

move a shape.

Parameter:

tmpShape: The shape you want to move
dx,dy: The move position

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.ShapeMove(shape0,100,100);
}
```

3.2.65 ShapeRotate

public void ShapeRotate(MyShape tmpShape,Single angle)

Description:

Rotate a shape.

Parameter:

tmpShape: The shape you want to rotate

angle: The rotate angle

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.ShapeRotate(shape0,30);
}
```

3.2.66 SizeShape

public void SizeShape(MyShape tmpShape,Int32 selectedHotId,PointF dragPoint)

Description:

Size a shape.

Parameter:

tmpShape: The shape you want to rotate
selectedHotId: The selected hot id
dragPoint: The new position

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.SizeShape(shape0,3,new Point(200,200));
}
```

3.2.67 UngroupShape

public void UngroupShape(MyShape tmpShape,Boolean needSaved)

Description:

Ungroup the shape.

Parameter:

tmpShape: The grouped shape
needSave: Weather save step to undo list

Example:

```
private void menuEditUngroup_Click(object sender, System.EventArgs e)
{

        if (myCAD1.GetSelectedShapesCount() == 1)
        {
                myCAD1.UngroupShape(myCAD1.GetSelectedShape(),true);
        }
        else
        {
                MessageBox.Show("No shape selected or more shapes selected.");
        }
}
```

See also:

GroupWorkingShapes

3.2.68 VisibleAllLayer

public void VisibleAllLayer()

Description:

Make all layer(s) visible.

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.VisibleAllLayer();
}
```

See also:

VisibleLayerById
VisibleLayerByName

3.2.69 VisibleLayerById

public void VisibleLayerById(Int32 layerId)

Description:

Make layer visible by layer id.

Parameter:

layerId: Layer id

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.VisibleLayerById(0);
}
```

See also:

VisibleAllLayer
VisibleLayerByName

3.2.70 VisibleLayerByName

public void VisibleLayerByName(String layerName)

Description:

Make layer visible by layer name.

Parameter:

layerName: Layer name

Example:

```
private void button1_Click(object sender, System.EventArgs e)
{
        myCAD1.VisibleLayerByName("Layer0");
}
```

See also:

VisibleAllLayer
VisibleLayerById

3.3 Events

3.3.1 ChildShapeSelected

```
public event ChildShapeSelectedEventHandler ChildShapeSelected;
```

Description:

Use ChildShapeSelected to handle when select a child shape.

Example:

```
private void myCAD1_ChildShapeSelected(object sender,
Codeidea.UltraGraphics.ChildShapeSelectedEventArgs e)
{
        MessageBox.Show(e.ChildShape.ShapeId.ToString());
}
```

3.3.2 DrawTypeToSelecting

```
public event DrawTypeToSelectingEventHandler DrawTypeToSelecting;
```

Description:

Use DrawTypeToSelecting to handle when the DrawType return to Selecting state.

Example:

```
private void myCAD1_DrawTypeToSelecting(object sender)
{
        lastButton.BackColor = Color.Transparent;
        btnSelecting.BackColor = Color.Gray;
        lastButton = btnSelecting;
}
```

3.3.3 EnterShape

```
public event EnterLeaveShapeEventHandler EnterShape;
```

Description:

When the mouse enter a shape or a grouped shape, this event trigger.

Example:

```
private void myCAD1_EnterShape(object sender,
Codeidea.UltraGraphics.EnterLeaveShapeEventArgs e)
{
        MessageBox.Show("Enter a shape");
}
```

3.3.4 LeaveShape

```
public event EnterLeaveShapeEventHandler LeaveShape;
```

Description:

When the mouse leave a shape or a grouped shape, this event trigger.

Example:

```
private void myCAD1_LeaveShape(object sender,
Codeidea.UltraGraphics.EnterLeaveShapeEventArgs e)
{
        MessageBox.Show("Leave a shape");
}
```

3.3.5 NodeAdded

public event NodeAddDeleteEventHandler NodeAdded;

Description:

Add a node when drawing PolyLine or Polygon, this event trigger.

Example:

```
private void myCAD1_NodeAdded(object sender,
Codeidea.UltraGraphics.NodeAddDeleteEventArgs e)
{
        MessageBox.Show("Add a node");
}
```

3.3.6 NodeDeleted

public event NodeAddDeleteEventHandler NodeDeleted;

Description:

Del et e a node when drawing PolyLine or Polygon, this event trigger.

Example:

```
private void myCAD1_NodeDeleted(object sender,
Codeidea.UltraGraphics.NodeAddDeleteEventArgs e)
{
        MessageBox.Show("Delete a node");
}
```

3.3.7 OnDeleteLayer

public event LayerOperationEventHandler OnDeleteLayer;

Description:

When del et e a layer, this event trigger.

Example:

```
private void myCAD1_OnDeleteLayer(object sender,
Codeidea.UltraGraphics.LayerOperationEventArgs e)
{
        MessageBox.Show("Delete a layer");
}
```

3.3.8 OnNewLayer

public event LayerOperationEventHandler OnNewLayer;

Description:

When add a layer, this event trigger.

Example:

```
private void myCAD1_OnNewLayer(object sender,
Codeidea.UltraGraphics.LayerOperationEventArgs e)
{
        MessageBox.Show("Add a layer");
}
```

3.3.9 ShapeAdded

```
public event ShapeAddedEventHandler ShapeAdded;
```

Description:

When add a shape, this event trigger.

Example:

```
private void myCAD1_ShapeAdded(object sender, Codeidea.UltraGraphics.ShapeAddedEventArgs
e)
{
        //Add a image shape
        if (e.WorkingShape is MyImage == true)
        {
                openFileDialog1.InitialDirectory = Application.ExecutablePath;
                openFileDialog1.Filter =
"BMP(*.bmp)|*.bmp|JPEG(*.jpg;*.jpe;*.jpeg)|*.jpg;*.jpe;*.jpeg|All files(*.*)|*.*";
                if (openFileDialog1.ShowDialog() == DialogResult.OK)
                {
                        Bitmap myBitmap = new Bitmap(openFileDialog1.FileName);
                        (e.WorkingShape as MyImage).Bitmap = myBitmap;
                }
        }
}
```

3.3.10 ShapeCodeDragging

```
public event ShapeCodeDragResizeEventHandler ShapeCodeDragging;
```

Description:

Dragging a shape when use code, this event trigger.

Example:

```
private void myCAD1_ShapeCodeDragging(object sender,
Codeidea.UltraGraphics.ShapeCodeDragResizeEventArgs e)
{
        MessageBox.Show("ShapeCodeDragging");
}
```

3.3.11 ShapeCodeRotating

```
public event ShapeCodeRotateEventHandler ShapeCodeRotating;
```

Description:

Rotating a shape when use code, this event trigger.

Example:

```
private void myCAD1_ShapeCodeRotating(object sender,
Codeidea.UltraGraphics.ShapeCodeRotateEventArgs e)
{
        MessageBox.Show("ShapeCodeRotating");
}
```

3.3.12 ShapeDeleted

```
public event ShapeDeletedEventHandler ShapeDeleted;
```

Description:

When delete a shape, this event trigger.

Example:

```
private void myCAD1_ShapeDeleted(object sender,
Codeidea.UltraGraphics.ShapeDeletedEventArgs e)
{
        MessageBox.Show("ShapeDeleted");
}
```

3.3.13 ShapeMouseDragged

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseDragged;
```

Description:

When dragged a shape, this event trigger.

Example:

```
private void myCAD1_ShapeMouseDragged(object sender,
Codeidea.UltraGraphics.ShapeMouseDragResizeRotateEventArgs e)
{
        MessageBox.Show("ShapeMouseDragged");
}
```

3.3.14 ShapeMouseDragging

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseDragging;
```

Description:

When dragging a shape, this event trigger.

Example:

```
private void myCAD1_ShapeMouseDragging(object sender,
Codeidea.UltraGraphics.ShapeMouseDragResizeRotateEventArgs e)
{
        MessageBox.Show("ShapeMouseDragging");
}
```

3.3.15 ShapeMouseResized

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseResized;
```

Description:

When resized a shape, this event trigger.

Example:

```
private void myCAD1_ShapeMouseResized(object sender,
Codeidea.UltraGraphics.ShapeMouseDragResizeRotateEventArgs e)
{
        MessageBox.Show("ShapeMouseResized");
}
```

3.3.16 ShapeMouseResizing

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseResizing;
```

Description:

When resizing a shape, this event trigger.

Example:

```
private void myCAD1_ShapeMouseResizing(object sender,
Codeidea.UltraGraphics.ShapeMouseDragResizeRotateEventArgs e)
{
    MessageBox.Show("ShapeMouseResizing");
}
```

3.3.17 ShapeMouseRotated

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseRotated;
```

Description:

When rotated a shape, this event trigger.

Example:

```
private void myCAD1_ShapeMouseRotated(object sender,
Codeidea.UltraGraphics.ShapeMouseDragResizeRotateEventArgs e)
{
    MessageBox.Show("ShapeMouseRotated");
}
```

3.3.18 ShapeMouseRotating

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseRotated;
```

Description:

When rotating a shape, this event trigger.

Example:

```
private void myCAD1_ShapeMouseRotating(object sender,
Codeidea.UltraGraphics.ShapeMouseDragResizeRotateEventArgs e)
{
    MessageBox.Show("ShapeMouseRotating");
}
```

3.3.19 ShapeSelected

```
public event ShapeSelectedEventHandler ShapeSelected;
```

Description:

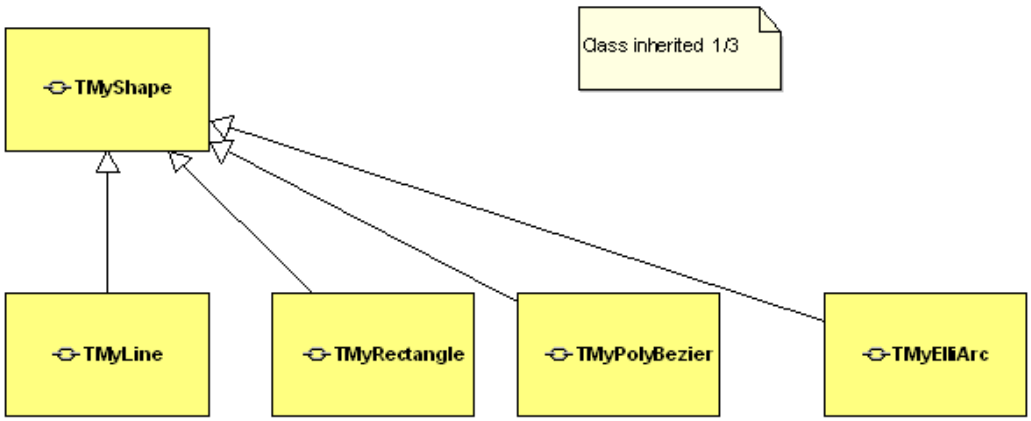
When sel ect a shape, this event trigger.

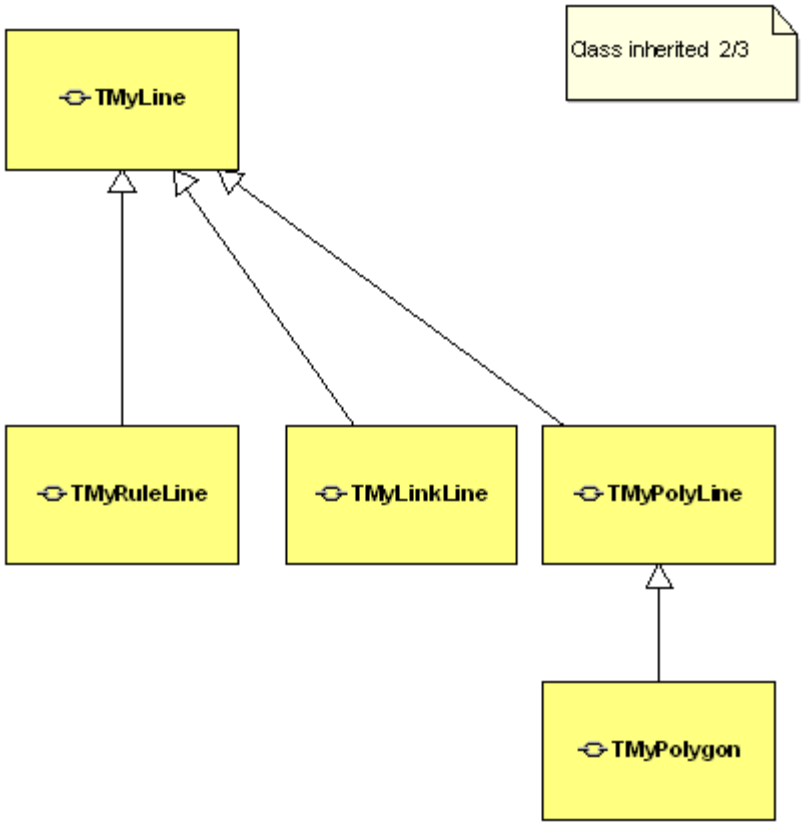
Example:

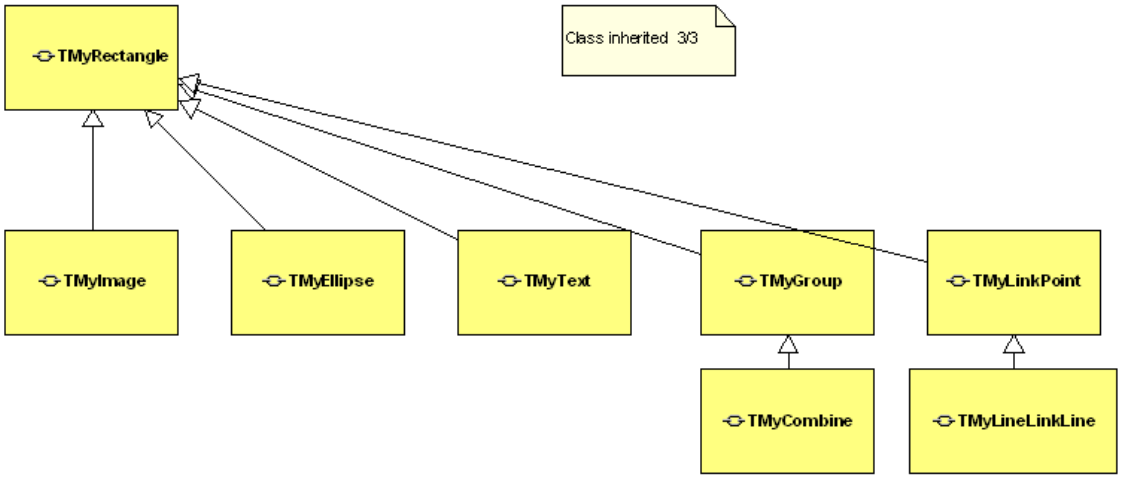
```
private void myCAD1_ShapeSelected(object sender,
Codeidea.UltraGraphics.ShapeSelectedEventArgs e)
{
    //Pen and brush property

    btnPenColor.BackColor = e.SelectedShape.Pen.Color;
    numericPenWidth.Value = Convert.ToDecimal(e.SelectedShape.Pen.Width);
    comboPenDash.SelectedIndex = Convert.ToInt32(e.SelectedShape.Pen.DashStyle);

    checkBrushShow.Checked = e.SelectedShape.BrushShow;
    if (e.SelectedShape.Brush is SolidBrush == true)
        btnBrushColor.BackColor = (e.SelectedShape.Brush as SolidBrush).Color;
}
```

Part

4 Shape class inherited diagram






TMyRectangle
Class inherited 3/3
TMyImage
TMyEllipse
TMyText
TMyGroup
TMyLinkPoint
TMyCombine
TMyLineLinkLine

Part V

5 MyShape

It is a base class of shapes, it defines common properties, events, methods for all shape(s) class.

5.1 Fields

5.1.1 CenterPoint

public PointF CenterPoint;

Description:

It is the center point of a shape.

5.1.2 ChildShapesNo

public Int64 [] ChildShapesNo;

Description:

If the shape is a single shape, this field is empty; if it is a grouped or combined shape, this field will store the child shapes no.

5.1.3 LayerId

public Int32 LayerId;

Description:

It indicate the shape belonged which layer, one shape must belonged one layer.

5.1.4 LinkPoints

public PointF[] LinkPoints;

Description:

This field will store link points if it has link points.

5.1.5 LinkShapesNo

public Int64[] LinkShapesNo;

Description:

If a shape has link shpaes,it will store these link shapes no.

5.1.6 ParentShapeNo

public Int64 ParentShapeNo = -1;

Description:

This field store a parent shape no,the default value is -1,as no parent shape.

5.1.7 ShapeId

public Int64 ShapeId;

Description:

It is the shape's id, delete or add a shape, it is not changed, it is a auto-increment field. it is maintaced by MyCAD and readonly for you.

5.1.8 ShapeNo

public Int64 ShapeNo;

Description:

It is the order of MyShapes.

5.1.9 TextOutPoint

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

5.1.10 ThePoints

public PointF[] ThePoints;

Description:

It is the points of a shape, different shape has different points count.it is not related with XYMode.

5.2 Property

5.2.1 Alpha

public Byte Alpha {get; set;}

Description:

Set the shape's transparency, value is 0 to 255.

Example:

myShape.Alpha = 100;

5.2.2 Angle

public Single Angle {get; set;}

Description:

Set the shape's angle.

Example:

myShape.Angle = Convert.ToSingle(30 * Math.PI / 180);

5.2.3 Brush

public Brush Brush {get; set;}

Description:

Set the brush of shape that you need.

Example:

If you want to know more property about Brush,please check the example named **StartBrush** and read the code.

See also:

BrushShow

5.2.4 BrushShow

public Boolean BrushShow {get; set;}

Description:

Show/not show the fill style of Brush.

Example:

myShape.BrushShow = true;

See also:

Brush

5.2.5 Caption

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer

adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

5.2.6 CaptionShow

public Boolean CaptionShow {get; set;}

Description:

Whether show shape's info.

Example:

```
myShape.CaptionShow = true;
```

5.2.7 Font

public Font Font {get; set;}

Description:

Set the shape's font.

Example:

```
myShape.Font = myCAD1.Font;
```

5.2.8 IsFlipHorizontal

public Boolean IsFlipHorizontal {get; set;}

Description:

Set the shape's horizontal turn.

Example:

```
myShape.IsFlipHorizontal = true;
```

5.2.9 IsFlipVertical

public Boolean IsFlipVertical {get; set;}

Description:

Set the shape's vertical turn.

Example:

myShape.IsFlipVertical = true;

5.2.10 Info

public String Info {get; set;}

Description:

Set the shape's info.

Example:

myShape.Info = "I like TCAD";

5.2.11 Lock

public Boolean Lock {get; set;}

Description:

Set the shape's lock/unlock state.

Example:

myShape.Lock = true;

5.2.12 Name

public String Name {get; set;}

Description:

Set the shape's name.

Example:

myShape.Name = "Shape0";

5.2.13 Owner

public MyCAD Owner {get; set;}

Description:

The shape's owner.

5.2.14 Pen

public Pen Pen {get; set;}

value:

Set the pen of s shape that you need.

Example:

myShape.Pen.Width = 2;
myShape.Pen.Color = Color.Green;

5.2.15 ShowUnit

public Boolean ShowUnit {get; set;}

Description:
Show/not show the length's unit.

Example:
myShape.ShowUnit = true;

5.2.16 Tag

public Int32 Tag {get; set;}

Description:
Set the shape's tag.

Example:
myShape.Tag = 32;

5.2.17 UserData

public MyUserData UserData {get; set;}

Description:
Set the shape's user-defined property.

Example:
Please read the demo code.

5.2.18 Visible

public Boolean Visible {get; set;}

Description:
Set the shape's visible/invisible state.

Example:
myShape.Visible = true;

5.3 Methods

5.3.1 Assign

public virtual void Assign(MyShape aShape)

Description:
Copy a source shape's properties.

Parameter:
aShape: The source shape.

5.3.2 ComputeCenterPoint

public PointF ComputeCenterPoint()

Description:
Computer the center point.

Return value:
Return the center point.

5.3.3 Dispose

public virtual void Dispose()

Description:
First all owned fields be released, finally inherited dispose is called.

5.3.4 Draw

public virtual void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

5.3.5 GetCenterPoint

public virtual PointF GetCenterPoint()

Description:
Get the center point.

Return value:
Return the center point.

5.3.6 GetCenterPointInZoom

public virtual PointF GetCenterPointInZoom()

Description:
Get the center point in zoom.

Return value:
Return the center point in zoom.

5.3.7 GetHeight

public Single GetHeight()

Description:
Get the shape height.

Return value:
return shape height.

5.3.8 GetLeftBottom

public PointF GetLeftBottom()

Description:
Get the shape's left-bottom point.

Return value:
return the left-bottom point.

5.3.9 GetLeftTop

public PointF GetLeftTop()

Description:
Get the shape's left-top point.

Return value:
return the left-top point.

5.3.10 GetLinkPointInZoom

public PointF GetLinkPointInZoom(Int32 iPoint)

Description:
Get the link point in zoom.

Return value:
Return link point in zoom.

5.3.11 GetLinkPointsCount

public Int32 GetLinkPointsCount()

Description:
Get the link points' count.

5.3.12 GetMyHeight

public Single GetMyHeight()

Description:
Get the shape's actual height.

Return value:
return shape's actual height.

5.3.13 GetMyWidth

public Single GetMyWidth()

Description:
Get the shape's actual width.

Return value:
return shape's actual width.

5.3.14 GetPoint

public PointF GetPoint(Int32 iPoint)

Description:
Get the point.

Parameter:
iPoint: The point's order that you want

Return value:
return the point

5.3.15 GetPointInZoom

public PointF GetPointInZoom(Int32 iPoint)

Description:
Get the point in zoom.

Parameter:
iPoint: The point's order that you want

Return value:
return the point in zoom.

5.3.16 GetPointsCount

public Int32 GetPointsCount()

Description:
Get the points' count.

5.3.17 GetRightBottom

public PointF GetRightBottom()

Description:
Get the shape's right-bottom point.

Return value:
return the right-bottom point.

5.3.18 GetRightTop

public PointF GetRightBottom()

Description:
Get the shape's right-bottom point.

Return value:
return the right-bottom point.

5.3.19 GetShapeId

public Int64 GetShapeId()

Description:
Get the shape id.

Return value:
Shape id.

5.3.20 GetWidth

public Single GetHeight()

Description:
Get the shape height.

Return value:
return shape height.

5.3.21 HasChildShapes

public Boolean HasChildShapes()

Description:
A shape has child shapes or not.

Return value:
true: Has child shapes.
false: Do not have child shapes.

5.3.22 HasLinkShapes

public Boolean HasLinkShapes()

Description:
A shape has link shapes or not.

Return value:
true: Has link shapes.
false: Do not have link shapes.

5.3.23 HasParentShape

public Boolean HasParentShape()

Description:
A shape has a parent shape or not.

Return value:
true: Has a parent shape.
false: Do not have a parent shape.

5.3.24 LoadFromOldStream

public virtual void LoadFromOldStream(Stream fileStream, String encoding)

Description:
Load stream from old tcad edition.

Parameter:
fileStream: The stream

5.3.25 LoadFromStream

public virtual void LoadFromStream(Stream fileStream)

Description:
Load stream from

Parameter:
fileStream: The stream

5.3.26 SaveToStream

public virtual void SaveToStream(Stream fileStream)

Description:
Save to stream

Parameter:
fileStream: The stream

Part VI

6 MyLine

It is a class of line shape, it defines properties, events, methods.

6.1 Properties

6.1.1 ArrowAngle

Enter topic text here.

6.1.2 ArrowLength

Enter topic text here.

6.1.3 ArrowOffset

Enter topic text here.

6.1.4 ArrowStyle

Enter topic text here.

6.2 Methods

6.2.1 Assign

public override void Assign(MyShape aShape)

Description:
Copy a source shape's properties.

Parameter:
aShape: The source shape.

6.2.2 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

6.2.3 GetInfo

public override String GetInfo(Graphics grp)

Description:
Get the line's info.

Parameter:
grp: The canvas instance.

Return value:

The info

6.2.4 LoadFromOldStream

```
public virtual void LoadFromOldStream(Stream fileStream,String encoding)
```

Description:
Load stream from old tcad edition.

Parameter:
fileStream: The stream

6.2.5 LoadFromStream

```
public override void LoadFromStream(Stream fileStream)
```

Description:
Load stream from.

Parameter:
fileStream: The stream

6.2.6 MyLine

```
public MyLine(MyCAD AOwner) : base(AOwner)
```

Description:
The create function of MyLine.

Parameter:
AOwner: The MyCAD Instance

6.2.7 SaveToStream

```
public override void SaveToStream(Stream fileStream)
```

Description:
Save to stream.

Parameter:
fileStream: The stream

Part VII

7 MyLinkLine

It is a class of link line shape, it defines properties, events, methods.

7.1 Properties

7.1.1 LinklineDrawStyle

public MyLinklineDrawStyle LinklineDrawStyle

Description:

Set the link line's style.

7.1.2 EndSpNo

public Int64 EndSpNo

Description:

It is a pointer of end shape.
-1: mean no end shape linked.

7.1.3 EndSpPtId

public Int32 EndSpPtId

Description:

it is read / write, at run time only.
The link point id of the end shape.
if EndSpNo is -1, it is must -1.

7.1.4 StartSpNo

public Int64 StartSpNo

Description:

It is a pointer of st ar t shape.
-1: mean no start shape linked.

7.1.5 StartSpPtId

public Int32 StartSpPtId

Description:

it is read / write, at run time only.
The link point id of the st ar t shape.
if StartSpNo is -1, it is must -1.

7.2 Methods

7.2.1 Assign

public override void Assign(MyShape aShape)

Description:

Copy a source shape's properties.

Parameter:
aShape: The source shape.

7.2.2 CreateDestLink

public Boolean CreateDestLink(Int64 aShapeId,Int32 aShapeLinkPtId)

Description:
If can link two shapes, and the path is vertical or horizontal. it is used for flow drawing, electric drawing and more.

Parameter:
aShapeId: the end shape id;
aShapeLinkPtId: the link id of the shape,the shape'id is AShapeid.

Return value:
true: Created success.
false: Created failed.

See also:
CreateSrcLink

7.2.3 CreateSrcLink

public Boolean CreateSrcLink(Int64 aShapeId,Int32 aShapeLinkPtId)

Description:
it can build a (start) relation ship with exist shape.

Parameter:
aShapeId: the end shape id;
aShapeLinkPtId: the link id of the shape,the shape'id is AShapeid.

Return value:
true: Created success.
false: Created failed.

See also:
CreateDestLink

7.2.4 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

7.2.5 GetEndPoint

public PointF GetEndPoint()

Description:
Get the link point in end link shape.

Return value:
If there is no end shape linked, it returns the last point of self; else it returns the Link point id of the linked shape.

7.2.6 GetEndShape

public MyShape GetEndShape()

Description:
Get the end link shape.

Return value:
null: There is no end link shape;
else: The instance of the link shape.

7.2.7 GetStartPoint

public PointF GetStartPoint()

Description:
Get the link point in start link shape.

Return value:
If there is no start shape linked, it returns the last point of self; else it returns the Link point id of the linked shape.

7.2.8 GetStartShape

public MyShape GetStartShape()

Description:
Get the start link shape.

Return value:
null: There is no start link shape;
else: The instance of the link shape.

7.2.9 LoadFromOldStream

public virtual void LoadFromOldStream(Stream fileStream, String encoding)

Description:
Load stream from old tcad edition.

Parameter:
fileStream: The stream

7.2.10 LoadFromStream

public override void LoadFromStream(Stream fileStream)

Description:
Load stream from.

Parameter:
fileStream: The stream

7.2.11 MyLinkLine

public MyLinkLine(MyCAD AOwner) : base(AOwner)

Description:
The create function of MyLinkLine.

Parameter:
AOwner: The MyCAD Instance

7.2.12 RemoveDestLink

public void RemoveDestLink()

Description:
Remove the target linked shape.

7.2.13 RemoveSrcLink

public void RemoveSrcLink()

Description:
Remove the source linked shape.

7.2.14 SaveToStream

public override void SaveToStream(Stream fileStream)

Description:
Save to stream.

Parameter:
fileStream: The stream

Part VIII

8 MyPolyLine

It is a class of polyline shape, it defines properties, events, methods.

8.1 Methods

8.1.1 MyPolyLine

public MyPolyLine(MyCAD AOwner) : base(AOwner)

Description:
The create function of MyPolyLine.

Parameter:
AOwner: The MyCAD Instance

Part

9 MyFreeLine

It is a class of free line shape, it defines properties, events, methods.

9.1 Methods

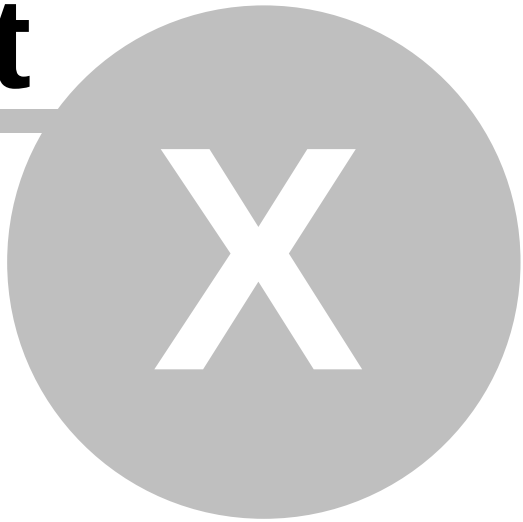
9.1.1 MyFreeLine

public MyFreeLine(MyCAD AOwner) : base(AOwner)

Description:
The create function of MyFreeLine.

Parameter:
AOwner: The MyCAD Instance

Part



10 MyPolygon

It is a class of polygon shape, it defines properties, events, methods.

10.1 Methods

10.1.1 MyPolygon

public MyPolygon(MyCAD AOwner) : base(AOwner)

Description:
The create function of MyPolygon.

Parameter:
AOwner: The MyCAD Instance

Part XI

11 MyRuleLine

It is a class of rule line shape, it defines properties, events, methods.

11.1 Properties

11.1.1 ShowUserInfo

public Boolean ShowUserInfo

Description:
When it is true, UserInfo showed on the rule line, else show the length string computed by MyRuleline.

11.1.2 TickStyle

public MyTickStyle TickStyle

Description:
Set the rule line's style.

11.2 Methods

11.2.1 Assign

public override void Assign(MyShape aShape)

Description:
Copy a source shape's properties.

Parameter:
aShape: The source shape.

11.2.2 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

11.2.3 LoadFromOldStream

public virtual void LoadFromOldStream(Stream fileStream, String encoding)

Description:
Load stream from old tcad edition.

Parameter:
fileStream: The stream

11.2.4 LoadFromStream

```
public override void LoadFromStream(Stream fileStream)
```

Description:
Load stream from.

Parameter:
fileStream: The stream

11.2.5 MyRuleLine

```
public MyRuleLine(MyCAD AOwner) : base(AOwner)
```

Description:
The create function of MyRuleLine.

Parameter:
AOwner: The MyCAD Instance

11.2.6 SaveToStream

```
public override void SaveToStream(Stream fileStream)
```

Description:
Save to stream.

Parameter:
fileStream: The stream

Part XII

12 MyWaveLine

It is a class of wave line shape, it defines properties, events, methods.

12.1 Properties

12.1.1 WaveHeight

```
public Byte WaveHeight
```

Description:
Set the wave line's height.

12.1.2 WaveWidth

```
public Byte WaveWidth
```

Description:
Set the wave line's width.

12.2 Methods

12.2.1 Assign

```
public override void Assign(MyShape aShape)
```

Description:
Copy a source shape's properties.

Parameter:
aShape: The source shape.

12.2.2 Draw

```
public override void Draw(Graphics grp)
```

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

12.2.3 LoadFromStream

```
public override void LoadFromStream(Stream fileStream)
```

Description:
Load stream from

Parameter:
fileStream: The stream

12.2.4 MyWaveLine

```
public MyWaveLine(MyCAD AOwner)
```

Description:
The create function of MyWaveLine.

Parameter:
AOwner: The MyCAD Instance

12.2.5 SaveToStream

```
public override void SaveToStream(Stream fileStream)
```

Description:
Save to stream

Parameter:
fileStream: The stream

Part XIII

13 MyRectangle

It is a class of rectangle shape, it defines properties, events, methods.

13.1 Methods

13.1.1 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

13.1.2 GetCenterPoint

public override PointF GetCenterPoint()

Description:
Get the center point of the rectangle.

Return value:
Return the center point.

See also:
GetCenterPointInZoom

13.1.3 GetCenterPointInZoom

public override PointF GetCenterPointInZoom()

Description:
Get the center point of the rectangle in zoom.

Return value:
Return the center point in zoom.

See also:
GetCenterPoint

13.1.4 GetInfo

public override String GetInfo(Graphics grp)

Description:
Get the line's info.

Parameter:
grp: The canvas instance.

Return value:
The info

Part XIV

14 MyEllipse

It is a class of ellipse shape, it defines properties, events, methods.

14.1 Methods

14.1.1 Draw

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

14.1.2 GetCenterPoint

public override PointF GetCenterPoint()

Description:
Get the center point of the ellipse.

Return value:
Return the center point.

See also:
GetCenterPointInZoom

14.1.3 GetCenterPointInZoom

public override PointF GetCenterPointInZoom()

Description:
Get the center point of the ellipse in zoom.

Return value:
Return the center point in zoom.

See also:

GetCenterPoint

Part XV

15 MyLinkPoint

It is a class of link point shape, it defines properties, events, methods.

15.1 Properties

15.1.1 Size

public Byte Size

Description:
Set the size of the link point.

15.2 Methods

15.2.1 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

15.2.2 LoadFromOldStream

public virtual void LoadFromOldStream(Stream fileStream, String encoding)

Description:
Load stream from old tcad edition.

Parameter:
fileStream: The stream

15.2.3 LoadFromStream

public override void LoadFromStream(Stream fileStream)

Description:
Load stream from

Parameter:
fileStream: The stream

15.2.4 MyLinkPoint

public MyLinkPoint (MyCAD AOwner) : base(AOwner)

Description:
The create function of MyLinkPoint.

Parameter:
AOwner: The MyCAD Instance

15.2.5 SaveToStream

```
public override void SaveToStream(Stream fileStream)
```

Description:
Save to stream

Parameter:
fileStream: The stream

Part XVI

16 MyLineLinkLine

It is a class of line link line shape, it defines properties, events, methods.

16.1 Methods

16.1.1 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

Part XVI

17 MyImage

It is a class of image line shape, it defines properties, events, methods.

17.1 Properties

17.1.1 Bitmap

public Bitmap Bitmap

Description:
Set this shape's bitmap.

17.1.2 Border

public Boolean Border

Description:
Whether show a border around the shape or not.

17.1.3 OriginSize

public Boolean OriginSize

Description:
Whether show real size of image or not.

17.1.4 Transparent

public Boolean Transparent

Description:
set the image transparent or not.

17.2 Methods

17.2.1 Assign

public override void Assign(MyShape aShape)

Description:
Copy a source shape's properties.

Parameter:
aShape: The source shape.

17.2.2 Dipose

public virtual void Dispose()

Description:
First all owned fields be released, finally inherited dispose is called.

17.2.3 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

17.2.4 LoadFromOldStream

public virtual void LoadFromOldStream(Stream fileStream, String encoding)

Description:
Load stream from old tcad edition.

Parameter:
fileStream: The stream

17.2.5 LoadFromStream

public override void LoadFromStream(Stream fileStream)

Description:
Load stream from.

Parameter:
fileStream: The stream

17.2.6 MyImage

public MyImage(MyCAD AOwner) : base(AOwner)

Description:
The create function of MyImage.

Parameter:
AOwner: The MyCAD Instance

17.2.7 SaveToStream

public override void SaveToStream(Stream fileStream)

Description:
Save to stream.

Parameter:
fileStream: The stream

Part

18 MyRoundRectangle

It is a class of round rectangle shape, it defines properties, events, methods.

18.1 Methods

18.1.1 MyRoundRectangle

public MyRoundRectangle(MyCAD AOwner)

Description:

The create function of MyRoundRectangle.

Parameter:

AOwner: The MyCAD Instance

Part XIX

19 MyText

It is a class of text shape, it defines properties, events, methods.

19.1 Properties

19.1.1 Border

public Boolean Border

Description:
Whether show a border arount the shape.

19.1.2 HAlignment

public MyAlignment HAlginment

Description:
Set the horizontal alignment of the text.

19.1.3 Lines

public String[] Lines

Description:
Set the text of the shape.

19.1.4 VAlignment

public MyVAlignment VAlginment

Description:
Set the vertical alignment of the text.

19.1.5 WordWrap

public Boolean WordWrap

Description:
Set WordWrap to true to allow the text to display multiple line of text.
When WordWrap is true, text that is too wide for the outer rectanger control wraps at the right margin and continues in additional lines.
Set WordWrap to false to limit the label to a single line. When WordWrap is false, text that is too wide for the outer rectanger appears outside.

19.2 Methods

19.2.1 Assign

public override void Assign(MyShape aShape)

Description:
Copy a source shape's properties.

Parameter:
aShape: The source shape.

19.2.2 Dipose

public virtual void Dispose()

Description:
First all owned fields be released, finally inherited dispose is called.

19.2.3 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

19.2.4 LoadFromOldStream

public virtual void LoadFromOldStream(Stream fileStream, String encoding)

Description:
Load stream from old tcad edition.

Parameter:
fileStream: The stream

19.2.5 LoadFromStream

public override void LoadFromStream(Stream fileStream)

Description:
Load stream from.

Parameter:
fileStream: The stream

19.2.6 MyText

public MyText (MyCAD AOwner) : base(AOwner)

Description:
The create function of MyText.

Parameter:
AOwner: The MyCAD Instance

19.2.7 SaveToStream

public override void SaveToStream(Stream fileStream)

Description:
Save to stream.

Parameter:
fileStream: The stream

Part

20 MyElliArc

It is a class of arc shape, it defines properties ,events,methods.

20.1 Properties

20.1.1 ArcMode

public MyArcMode ArcMode

Description:
Set the mode of Arc.

20.1.2 ArcStyle

public MyArcStyle ArcStyle

Description:
Set the style of Arc.

20.2 Methods

20.2.1 Assign

public override void Assign(MyShape aShape)

Description:
Copy a source shape's properties.

Parameter:
aShape: The source shape.

20.2.2 Dipose

public virtual void Dispose()

Description:
First all owned fields be released, finally inherited dispose is called.

20.2.3 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

20.2.4 GetCenterPoint

public override PointF GetCenterPoint()

Description:
Get the center point of the elli arc.

Return value:
Return the center point.

See also:
GetCenterPointInZoom

20.2.5 GetCenterPointInZoom

public override PointF GetCenterPointInZoom()

Description:
Get the center point of the elli arc in zoom.

Return value:
Return the center point in zoom.

See also:
GetCenterPoint

20.2.6 LoadFromOldStream

public virtual void LoadFromOldStream(Stream fileStream, String encoding)

Description:
Load stream from old tcad edition.

Parameter:
fileStream: The stream

20.2.7 LoadFromStream

public override void LoadFromStream(Stream fileStream)

Description:
Load stream from

Parameter:
fileStream: The stream

20.2.8 MyElliArc

public MyElliArc(MyCAD AOwner) : base(AOwner)

Description:
The create function of MyElliArc.

Parameter:
AOwner: The MyCAD Instance

20.2.9 SaveToStream

public override void SaveToStream(Stream fileStream)

Description:
Save to stream

Parameter:
fileStream: The stream

Part XXI

21 MyPolyBezier

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

21.1 Methods

21.1.1 Draw

public override void Draw(Graphics grp)

Description:
Draw the shape.

Parameter:
grp: The canvas instance.

Part XXI

22 MyUserData

This class lets you store yourself data for a single or group shape.

22.1 Property

22.1.1 UserDataRecord

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

22.2 Methods

22.2.1 MyUserData

public MyUserData()

Description:

Initialize the inner data.

22.2.2 AddKeyAndValue

public Boolean AddKeyAndValue(String newKey,String newValue)

Description:

Append a new user define property.

Parameter:

newKey: The new key you set.
newValue: The new value you set.

Return:

true: Append successfully.
false: Append unsuccessfully.

22.2.3 Assign

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

22.2.4 ChangeValueByKey

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

22.2.5 ClearAll

public Boolean ChangeValueByKey(String originKey,String newValue)

Description:
Clear all data.

Return:
true: Clear successfully.
false: Clear unsuccessfully.

22.2.6 DeleteRecordByKey

public Boolean DeleteRecordByKey(String originKey)

Description:
Delete a record by key,if not match the key,delete unsuccessfully.

Parameter:
originKey: The record's key that you want to delete.

Return:
true: Delete successfully.
false: Delete unsuccessfully.

22.2.7 GetCount

public Int32 GetCount()

Description:
Get the count of the user record.

Return:
Return the count.

22.2.8 GetKeyByNo

public String GetKeyByNo(Int32 n)

Description:
Get key by order of number.

Parameter:
n: The order of number.

Return:
Return the key.

22.2.9 GetValueByKey

public String GetValueByKey(String originKey)

Description:
Get value by key.

Parameter:
originKey: The key you set.

Return:
Return the value.

22.2.10 InsertKeyAndValue

public Boolean InsertKeyAndValue(String newKey,String newValue,Int32 index,Boolean before)

Description:
Insert a record by located index.

Parameter:
newKey: The new key you set.
newValue: The new value you set.
index: the order
before: insert record before or after,if this parameter is false,the record will insert after the order number.

Return:
true: Insert successfully.
false: Insert unsuccessfully.

22.2.11 RenameKey

public Boolean RenameKey(String oldKey,String newKey)

Description:
Rename a key.

Parameter:
oldKey: The record's key that you want to rename.
newKey: The new key you want to set.

Return:
true: Delete successfully.
false: Delete unsuccessfully.

Part XII

23 About Crystal Component

About us

HuZhou HongDi science and technology development co.,ltd. Zhejiang, China, specializes in developing and supporting expert graphical visualization components - TCAD for adding vector drawing function into your applications. TCAD has been helping many software developers around the world.

The Delphi , C++ Builders , kylix , .NET components are special products. Now we have an opportunity to help each registered user by e-mail. We answer any questions and give some recommendations on more sophisticated use of our products' features. Typically we answer messages in 24 hours, but depending on singularity and difficulty of your question it may take a bit longer. You can affect our products' modification and development by e-mailing us about your needs.

If you need to contact us, you can do so using one of the following methods. We are here to help, so do not hesitate to communicate with us when required.When calling us by phone, please remember that our offices are open from 9:00AM to 5:00PM (GMT +8).

Location

Address: Room A303-304# QingTong Road 699# Science & technology development service center HuZhou Zhejiang China
Phone: +86 572 2607144
Fax: +86 572 2576169
Mobile: +86 (0) 13511221372
Website: http://www.codeidea.com
Email: webmaster@codeidea.com
MSN: webmaster@codeidea.com

Crystal

www.codeidea.com