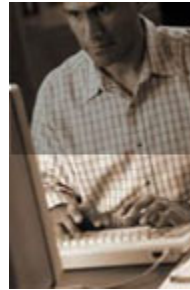


TCAD xp for .NET

© 2006 湖州鸿迪科技发展有限公司



什么是TCAD组件

by 湖州鸿迪科技发展有限公司

您是否为了一个小小的矢量图形的应用程序，而找不到合适的控件，自己从底层编制，工作量又巨大，使用象Mapinfo等大型软件又觉得大材小用、喧宾夺主!？
TCAD矢量组件正是为了您的这种需求而产生的，她是有多个版本，是Delphi, C++ Builder, kyllix, c#的控件，不使用任何DLL, OCX, 简单、快速，安装方便，可绘制、拖动、删除图形，现在您终于可以解决了！

目录

| | |
|----------------------------|-----------|
| Foreword | 11 |
| Part I 使用协议 | 13 |
| Part II TCAD xp 介绍 | 15 |
| 1 TCAD for .NET是什么？ | 15 |
| 2 屏幕截图 | 17 |
| Part III 类型定义 | 22 |
| Part IV MyCAD 类 | 25 |
| 1 属性 | 25 |
| Alpha | 25 |
| ArrowAngle | 25 |
| ArrowLength | 25 |
| ArrowOffset | 25 |
| ArrowStyle | 26 |
| BackgroundBitmap | 26 |
| BackgroundBitmapMode | 26 |
| BackgroundColor | 26 |
| Brush | 27 |
| BrushShow | 27 |
| CrossLine | 27 |
| DragMode | 28 |
| DrawType | 28 |
| GridColor | 28 |
| GridHeight | 28 |
| GridPenSize | 28 |
| GridShow | 29 |
| GridType | 29 |
| GridWidth | 29 |
| HotColor | 30 |
| HotShow | 30 |
| HotSize | 30 |
| LabelValue | 30 |
| LabelXY | 30 |
| LinklineAroundShape | 31 |
| LinklineDrawStyle | 31 |
| OperateAllLayer | 32 |
| PageFoot | 32 |
| PageFootAlignment | 32 |
| PageFootFont | 33 |
| PageFootToBottom | 33 |
| PageHead | 33 |
| PageHeadAlignment | 33 |
| PageHeadFont | 34 |
| PageHeadToTop | 34 |

| | |
|---------------------------------|-----------|
| PageHeight | 34 |
| PageOrientation | 34 |
| PageStyle | 34 |
| PageWidth | 35 |
| Pen | 35 |
| PrintBackground | 35 |
| PrintBorder | 35 |
| PrintBorderToBottom | 35 |
| PrintBorderToLeft | 36 |
| PrintBorderToRight | 36 |
| PrintBorderToTop | 36 |
| Ratio | 36 |
| ResizeEnable | 37 |
| ReturnToSelecting | 37 |
| RotateConstraintDegree | 37 |
| RotateEnable | 37 |
| ShowHotLink | 37 |
| Snap | 37 |
| SnapPixel | 38 |
| SnapShape | 38 |
| UndoRedoSize | 38 |
| UnitType | 38 |
| Version | 38 |
| XYMode | 39 |
| Zoom | 39 |
| 2 方法 | 39 |
| AddImageShapeByCode | 39 |
| AddShapeByCode | 39 |
| AddUserDefineShapeFromLib | 40 |
| AlignBottom | 40 |
| AlignHorizontalCenter | 41 |
| AlignLeft | 41 |
| AlignRight | 41 |
| AlignTop | 41 |
| AlignVerticalCenter | 42 |
| BringToFront | 42 |
| BringToFrontByStep | 42 |
| ClearAllUndoStuff | 43 |
| Copy | 43 |
| CreateLink | 43 |
| Cut | 44 |
| Delete | 44 |
| DeleteAllLayers | 44 |
| DeleteAllShapes | 44 |
| DeleteLayerById | 45 |
| DeleteLayerByName | 45 |
| DeleteSelectedShapes | 45 |
| DeleteShapeById | 46 |
| DeSelectedAllShapesByCode | 46 |
| FlipHorizontal | 46 |
| FlipVertical | 47 |
| GetLayerIdByName | 47 |
| GetLayerIdByNo | 47 |
| GetLayerNameById | 48 |

| | |
|--------------------------------|-----------|
| GetLayerNoById | 48 |
| GetLayerNoByName | 49 |
| GetLayersCount | 49 |
| GetMaxLayerId | 49 |
| GetRootParentShape | 49 |
| GetSelectedShape | 50 |
| GetSelectedShapes | 50 |
| GetSelectedShapesCount | 50 |
| GetShapeById | 51 |
| GetShapeByNo | 51 |
| GetShapeNoById | 51 |
| GetShapesCount | 52 |
| GetShapesCountInALayer | 52 |
| GetWorkingShapesCount | 52 |
| GroupWorkingShapes | 53 |
| InVisibleLayerById | 53 |
| InVisibleLayerByName | 53 |
| LoadFromFile | 54 |
| LoadFromStream | 54 |
| NewLayer | 55 |
| Paste | 55 |
| PopfromUndoRedoShapeList | 55 |
| Preview | 56 |
| Print | 56 |
| SaveToFile | 56 |
| SaveToImage | 57 |
| SaveToStream | 57 |
| SelectAllShapes | 58 |
| SelectShapeByCode | 58 |
| SendToBack | 58 |
| SendToBackByStep | 59 |
| SetLayerNameById | 59 |
| SetLayerNameByName | 59 |
| ShapeMove | 60 |
| ShapeRotate | 60 |
| SizeShape | 60 |
| UngroupShape | 60 |
| VisibleAllLayer | 61 |
| VisibleLayerById | 61 |
| VisibleLayerByName | 62 |
| 3 事件 | 62 |
| ChildShapeSelected | 62 |
| DrawTypeToSelecting | 62 |
| EnterShape | 62 |
| LeaveShape | 63 |
| NodeAdded | 63 |
| NodeDeleted | 63 |
| OnDeleteLayer | 63 |
| OnNewLayer | 64 |
| ShapeAdded | 64 |
| ShapeCodeDragging | 64 |
| ShapeCodeRotating | 64 |
| ShapeDeleted | 65 |
| ShapeMouseDragged | 65 |

| | |
|-------------------------|----|
| ShapeMouseDowning | 65 |
| ShapeMouseDown | 65 |
| ShapeMouseDown | 66 |
| ShapeMouseDown | 66 |
| ShapeMouseDown | 66 |
| ShapeMouseDown | 66 |

Part V 图形类继承关系图 69

Part VI MyShape 类 71

| | |
|----------------------------|----|
| 1 字段 | 71 |
| CenterPoint | 71 |
| ChildShapesNo | 71 |
| LayerId | 71 |
| LinkPoints | 71 |
| LinkShapesNo | 71 |
| ParentShapeNo | 71 |
| Shapeld | 72 |
| ShapeNo | 72 |
| TextOutPoint | 72 |
| ThePoints | 72 |
| 2 属性 | 72 |
| Alpha | 72 |
| Angle | 72 |
| Brush | 73 |
| BrushShow | 73 |
| Caption | 73 |
| CaptionShow | 73 |
| Font | 73 |
| IsFlipHorizontal | 74 |
| IsFlipVertical | 74 |
| Info | 74 |
| Lock | 74 |
| Name | 74 |
| Owner | 74 |
| Pen | 75 |
| ShowUnit | 75 |
| Tag | 75 |
| UserData | 75 |
| Visible | 75 |
| 3 方法 | 75 |
| Assign | 76 |
| ComputeCenterPoint | 76 |
| Dispose | 76 |
| Draw | 76 |
| GetCenterPoint | 76 |
| GetCenterPointInZoom | 76 |
| GetHeight | 77 |
| GetLeftBottom | 77 |
| GetLeftTop | 77 |
| GetLinkPointInZoom | 77 |
| GetLinkPointsCount | 77 |

| | |
|-------------------------|----|
| GetMyHeight | 77 |
| GetMyWidth | 78 |
| GetPoint | 78 |
| GetPointInZoom | 78 |
| GetPointsCount | 78 |
| GetRightBottom | 78 |
| GetRightTop | 79 |
| GetShapeld | 79 |
| GetWidth | 79 |
| HasChildShapes | 79 |
| HasLinkShapes | 79 |
| HasParentShape | 80 |
| LoadFromOldStream | 80 |
| LoadFromStream | 80 |
| SaveToStream | 80 |

Part VII Myline 82

| | |
|-------------------------|----|
| 1 属性 | 82 |
| ArrowAngle | 82 |
| ArrowLength | 82 |
| ArrowOffset | 82 |
| ArrowStyle | 82 |
| 2 方法 | 82 |
| Assign | 82 |
| Draw | 82 |
| GetInfo | 82 |
| LoadFromOldStream | 83 |
| LoadFromStream | 83 |
| MyLine | 83 |
| SaveToStream | 83 |

Part VIII MyLinkLine 85

| | |
|-------------------------|----|
| 1 属性 | 85 |
| LinklineDrawStyle | 85 |
| EndSpNo | 85 |
| EndSpPtId | 85 |
| StartSpNo | 85 |
| StartSpPtId | 85 |
| 2 方法 | 85 |
| Assign | 85 |
| CreateDestLink | 86 |
| CreateSrcLink | 86 |
| Draw | 86 |
| GetEndPoint | 87 |
| GetEndShape | 87 |
| GetStartPoint | 87 |
| GetStartShape | 87 |
| LoadFromOldStream | 87 |
| LoadFromStream | 88 |
| MyLinkLine | 88 |
| RemoveDestLink | 88 |
| RemoveSrcLink | 88 |

| | |
|-----------------------------|------------|
| SaveToStream | 88 |
| Part IX MyPolyLine | 90 |
| 1 方法 | 90 |
| MyPolyLine | 90 |
| Part X MyFreeLine | 92 |
| 1 方法 | 92 |
| MyFreeLine | 92 |
| Part XI MyPolygon | 94 |
| 1 方法 | 94 |
| MyPolygon | 94 |
| Part XII MyRuleLine | 96 |
| 1 属性 | 96 |
| ShowUserInfo | 96 |
| TickStyle | 96 |
| 2 方法 | 96 |
| Assign | 96 |
| Draw | 96 |
| LoadFromOldStream | 96 |
| LoadFromStream | 97 |
| MyRuleLine | 97 |
| SaveToStream | 97 |
| Part XIII MyWaveLine | 99 |
| 1 属性 | 99 |
| WaveHeight | 99 |
| WaveWidth | 99 |
| 2 方法 | 99 |
| Assign | 99 |
| Draw | 99 |
| LoadFromStream | 99 |
| MyWaveLine | 99 |
| SaveToStream | 100 |
| Part XIV MyRectangle | 102 |
| 1 方法 | 102 |
| Draw | 102 |
| GetCenterPoint | 102 |
| GetCenterPointInZoom | 102 |
| GetInfo | 102 |
| Part XV MyEllipse | 104 |
| 1 方法 | 104 |
| Draw | 104 |
| GetCenterPoint | 104 |
| GetCenterPointInZoom | 104 |

| | |
|----------------------------------|------------|
| Part XVI MyLinkPoint | 106 |
| 1 属性 | 106 |
| Size | 106 |
| 2 方法 | 106 |
| Draw | 106 |
| LoadFromOldStream | 106 |
| LoadFromStream | 106 |
| MyLinkPoint | 106 |
| SaveToStream | 107 |
| Part XVII MyLineLinkLine | 109 |
| 1 方法 | 109 |
| Draw | 109 |
| Part XVIII MyImage | 111 |
| 1 属性 | 111 |
| Bitmap | 111 |
| Border | 111 |
| OriginSize | 111 |
| Transparent | 111 |
| 2 方法 | 111 |
| Assign | 111 |
| Dispose | 111 |
| Draw | 111 |
| LoadFromOldStream | 112 |
| LoadFromStream | 112 |
| MyImage | 112 |
| SaveToStream | 112 |
| Part XIX MyRoundRectangle | 114 |
| 1 方法 | 114 |
| MyRoundRectangle | 114 |
| Part XX MyText | 116 |
| 1 属性 | 116 |
| Border | 116 |
| HAlignment | 116 |
| Lines | 116 |
| VAlignment | 116 |
| WordWrap | 116 |
| 2 方法 | 116 |
| Assign | 116 |
| Dispose | 117 |
| Draw | 117 |
| LoadFromOldStream | 117 |
| LoadFromStream | 117 |
| MyText | 117 |
| SaveToStream | 117 |

Part XXI MyElliArc 119

| | |
|----------------------|-----|
| 1 属性 | 119 |
| ArcMode | 119 |
| ArcStyle | 119 |
| 2 方法 | 119 |
| Assign | 119 |
| Dispose | 119 |
| Draw | 119 |
| GetCenterPoint | 119 |
| GetCenterPointInZoom | 120 |
| LoadFromOldStream | 120 |
| LoadFromStream | 120 |
| MyElliArc | 120 |
| SaveToStream | 120 |

Part XXII MyPolyBezier 122

| | |
|------|-----|
| 1 方法 | 122 |
| Draw | 122 |

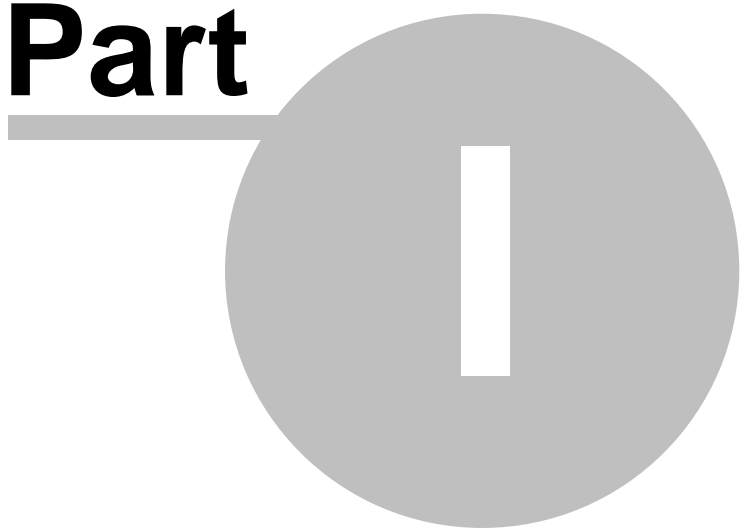
Part XXIII MyUserData 124

| | |
|-------------------|-----|
| 1 属性 | 124 |
| UserDataRecord | 124 |
| 2 方法 | 124 |
| MyUserData | 124 |
| AddKeyAndValue | 124 |
| Assign | 124 |
| ChangeValueByKey | 124 |
| ClearAll | 125 |
| DeleteRecordByKey | 125 |
| GetCount | 125 |
| GetKeyByNo | 125 |
| GetValueByKey | 126 |
| InsertKeyAndValue | 126 |
| RenameKey | 126 |

Part XXIV 关于我们 128

Index 0

Part



1 使用协议

参见合同

Part

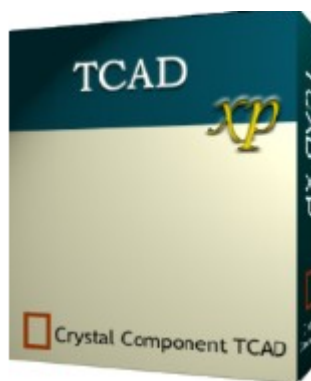


2 TCAD xp 介绍

2.1 TCAD for .NET是什么？

TCADfor.NET

是一个能帮助你开发矢量图形应用的组件，图形能被鼠标或代码控制，使用起来非常简单、高效、功能强大。它将会节省你许多宝贵的时间。



如果你想在应用程序中加入矢量绘图功能，使用OLE模式，许多概念是很难理解的，如果从底层写起，你将发现工作量是非常大，现在，你可以使用TCADfor.NET组件，在应用中仅通过鼠标控制就可以非常简单地创建、使用矢量图形。现在你能轻松的使用她在.NET环境中编程。

主要特点

图形类型

| | |
|--------------------|-------------|
| Line | 线、单箭头线、双箭头线 |
| LinkLine | 连接线 |
| PolyLine | 多义线 |
| Polygon | 多边形 |
| RuleLine | 标尺 |
| Rectangle | 矩形 |
| Ellipse | 圆、椭圆 |
| LinkPoint | 连接点 |
| LineLinkLine | 线的连接点 |
| Image | 图象 |
| Text | 文本 |
| EllliArc | 圆弧、椭圆弧、弦、扇形 |
| PolyBezier | 贝塞而曲线 |
| User define shapes | 用户自定义图形 |

组合/取消组合

支持图层

通过代码创建、移动、旋转、拉伸图形

装入/保存到文件或数据库
方便地创建用户自定义图形
支持4种坐标系

详细信息

通过鼠标操作或代码创建图形
修改已经创建的图形
支持图层、打印、删除、可见、不可见
使用所有颜色
支持连接线形
可以使用不同的画笔和笔刷
用任何已经安装在您的系统里的字体建立文本对象(图形)
具有图形变化的相应事件
使用各种格式的纸张类型(如:A0、A1、A2、A3、A4、letter等等)或自定义类型

能undo并设置undo的容量
能剪切、复制、粘贴、删除图形
控制图形(移到最后、移到最前)

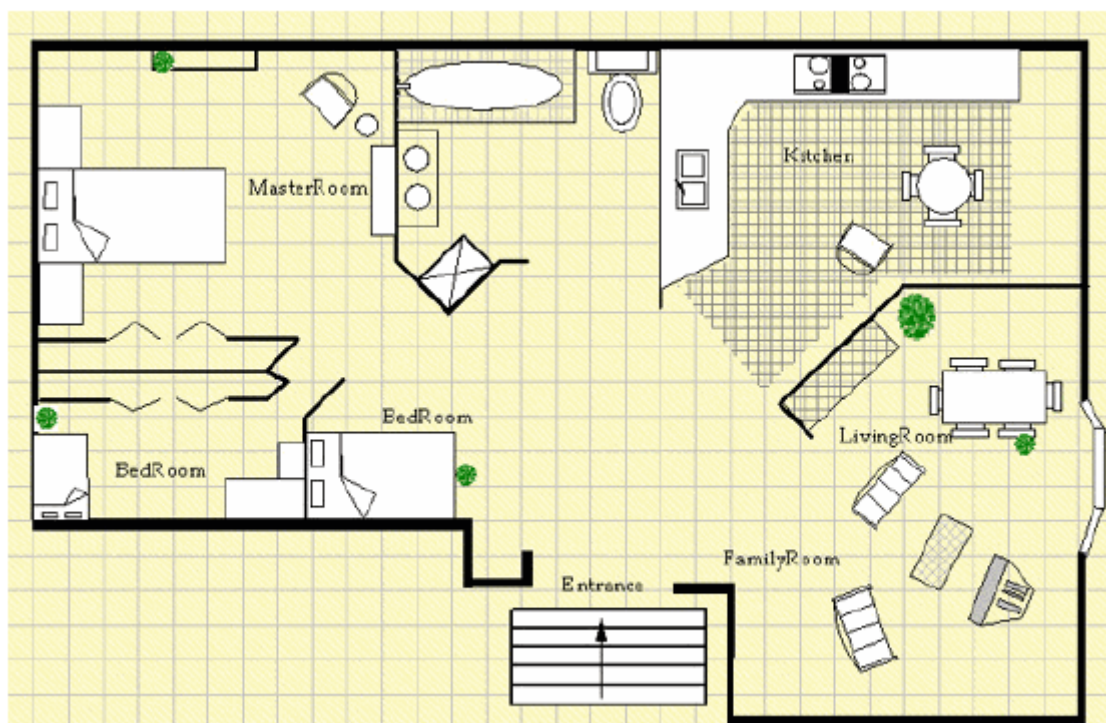
通过鼠标或代码旋转、拖动、缩放图形
任意类型的调整图形
简单去创建用户自定义的联合图形
使鼠标点对齐网格,可以设置网格的宽度和高度
支持24种渐变色

锁定/解锁图形
显示或隐藏图形执点
组合/撤消组合图形
任意比例的图形缩放、浏览
当鼠标移到图形时显示提示

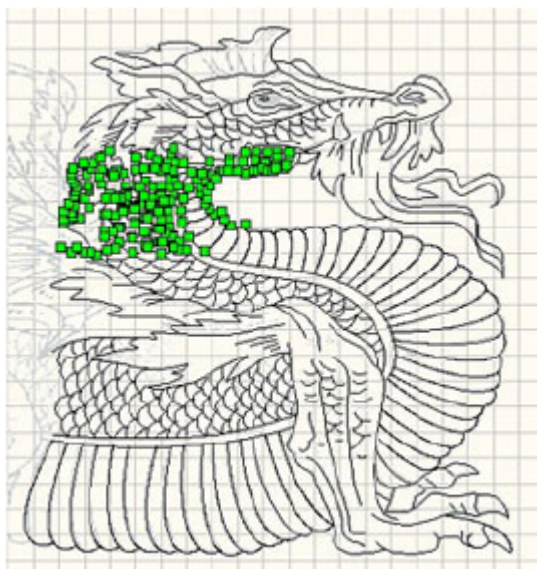
通过文件或流(数据库)保存或打开图形
打印图形

插入图片
像图形一样缩放、旋转、拖动图片
输出图形为WMF、Bitmap、jpg、Dxf格式的文件

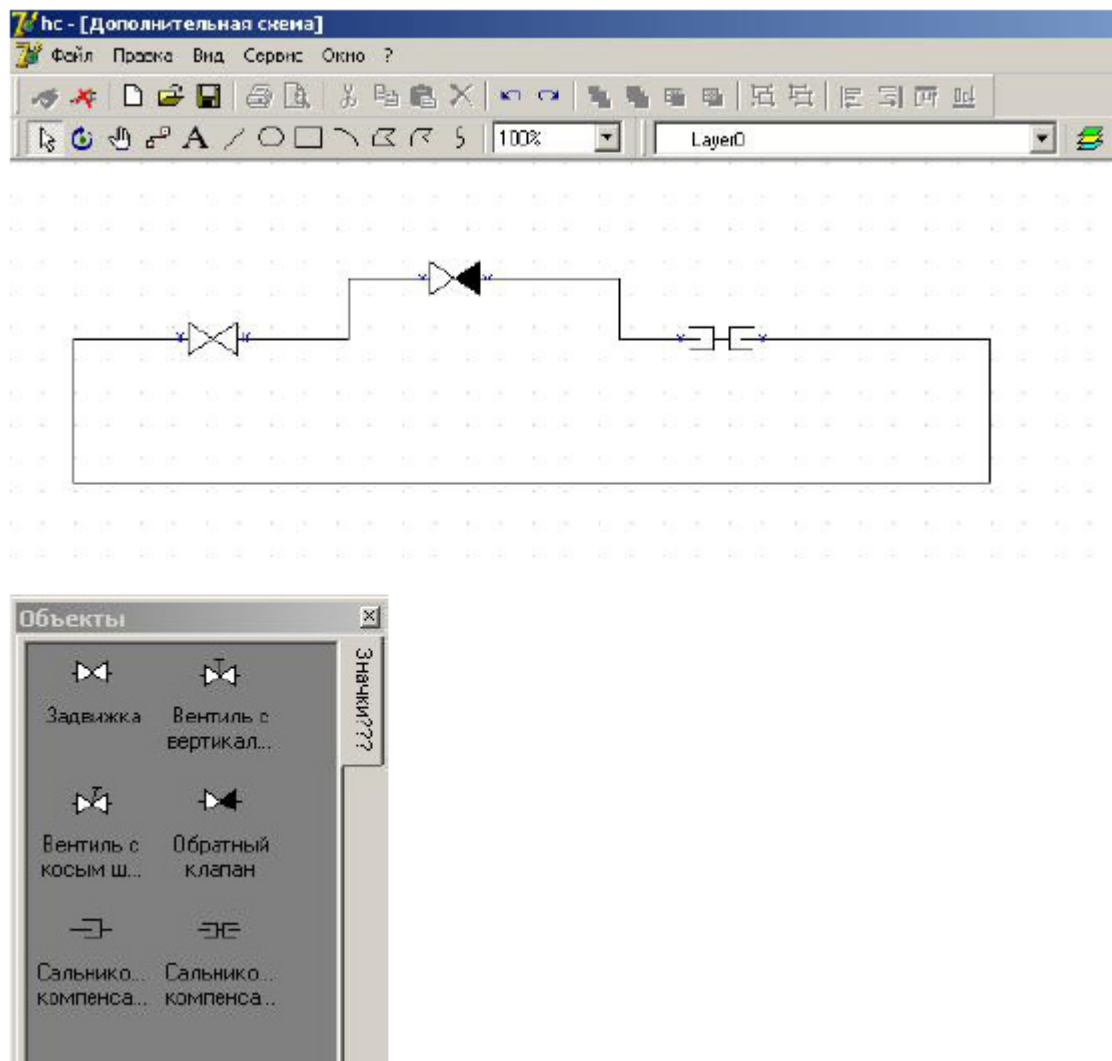
2.2 屏幕截图



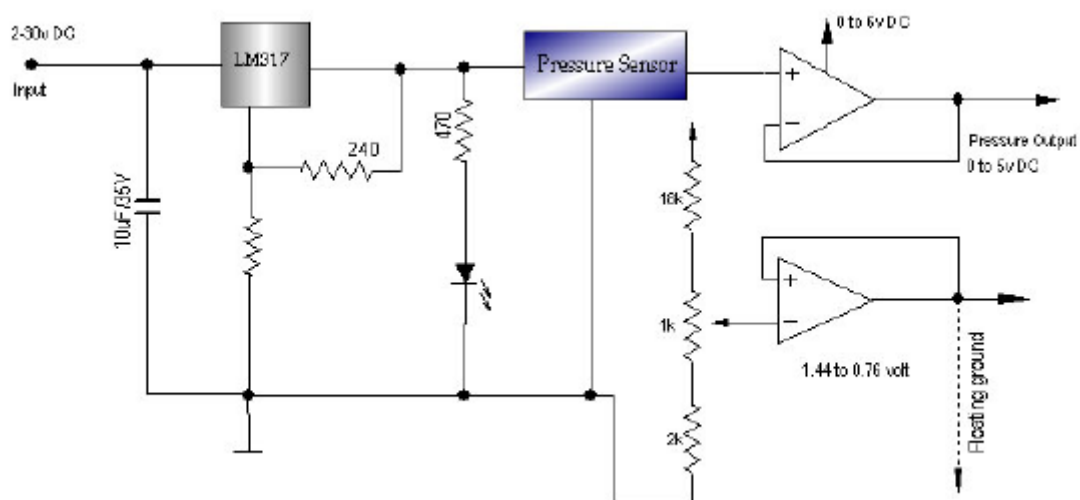
这是一个房屋平面设计图，创建简单快速，没有用到图元库



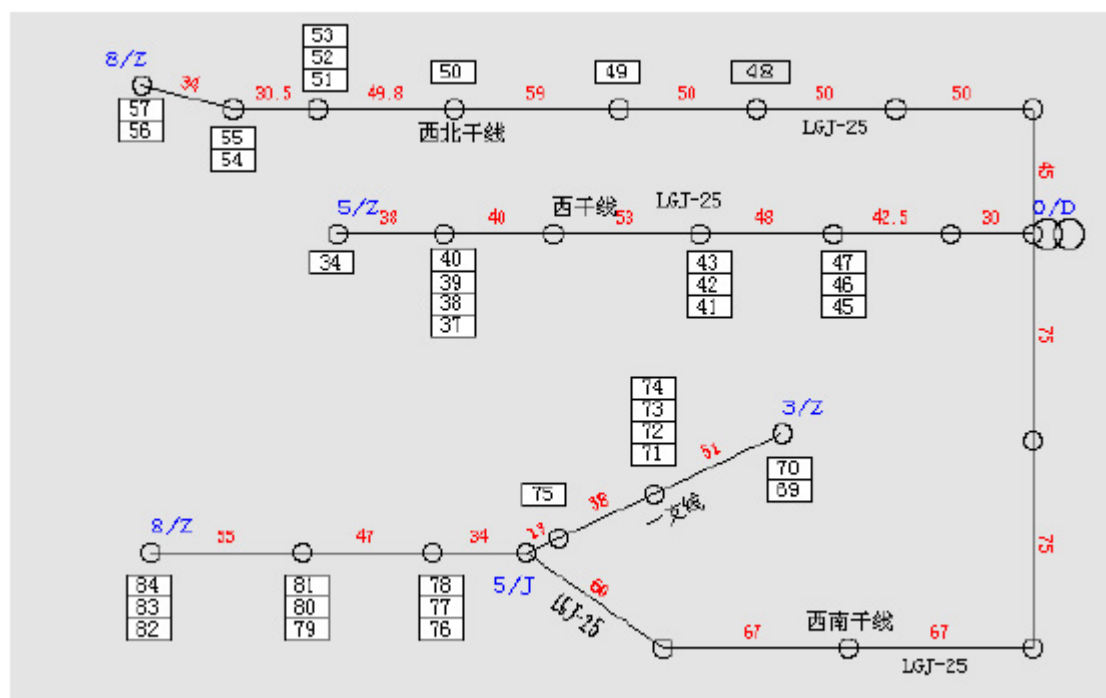
采用TCAD设计的矢量图---龙，最终用于CNC机床加工



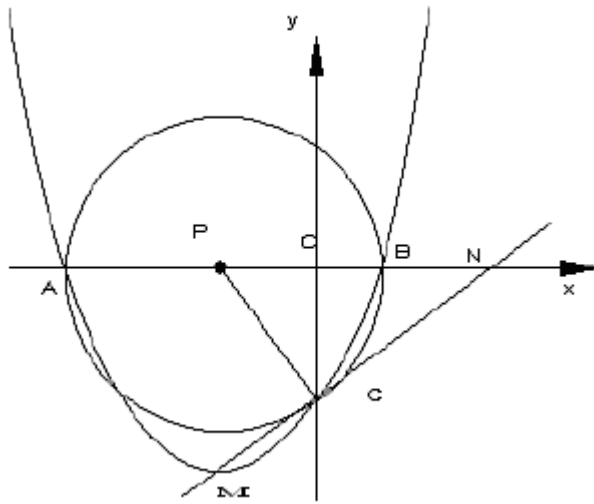
水利设计，由俄罗斯的客户提供



电子线路图，会用到库的功能，支持连接线，TCAD的连接线功能是非常强大的

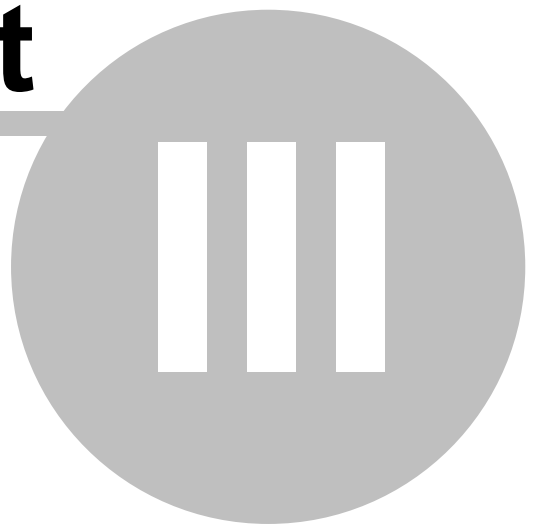


项目：低压电力设备和线损管理
单位：西安工业大学



假如你是老师或是学生，TCAD能帮助你画数学曲线图，你可以复制一个曲线图到MS Word使用数学图形库提高你的工作效率。

Part



3 类型定义

```
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, PolylinePolygonPointAdding, 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

IV

4 MyCAD 类

4.1 属性

4.1.1 Alpha

```
public Byte Alpha {get; set;}
```

描述：
设置图形的透明度，值为 0 到 255.

示例：
`myShape.Alpha = 100;`

4.1.2 ArrowAngle

```
public Int32 ArrowAngle {get; set;}
```

描述：
设置线和双箭头线的箭头角度，其值在 0到 359之间

示例：
`myCAD1.ArrowAngle = 50;`

参见：
[ArrowLength](#)
[ArrowOffset](#)
[ArrowStyle](#)

4.1.3 ArrowLength

```
public Byte ArrowLength {get; set;}
```

描述：
设置线和双箭头线的箭头长度，其值在 10-50之间.

示例：
`myCAD1.ArrowLength = 30;`

参见：
[ArrowAngle](#)
[ArrowOffset](#)
[ArrowStyle](#)

4.1.4 ArrowOffset

```
public Byte ArrowOffset {get; set;}
```

描述：
设置箭头的偏移量，其值在 0到 255之间，默认值为 0

示例：
`myCAD1.ArrowOffset = 0;`



`myCAD1.ArrowOffset = 16;`



参见：

[ArrowAngle](#)
[ArrowLength](#)
[ArrowStyle](#)

4.1.5 ArrowStyle

```
public MyArrowStyle ArrowStyle {get; set;}
```

描述：

设置线和双箭头线的类型；

示例：

```
myCAD1.ArrowStyle = MyArrowStyle.Left;
```

参见：

[ArrowAngle](#)
[ArrowLength](#)
[ArrowOffset](#)

4.1.6 BackgroundBitmap

```
public Bitmap BackgroundBitmap {get; set;}
```

描述：

设置TCAD的背景图片，如果要清除掉，使用 BackgroundBitmap = null;

示例：

```
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;  
}
```

参见：

[BackgroundBitmapMode](#)

4.1.7 BackgroundBitmapMode

```
public MyBackgroundBitmapMode BackgroundBitmapMode {get; set;}
```

描述：

设置背景图片显示样式

示例：

```
myCAD1.BackgroundBitmapMode = MyBackgroundBitmapMode.Center;
```

参见：

[BackgroundBitmap](#)

4.1.8 BackgroundColor

```
public Color BackgroundColor {get; set;}
```

描述：

指定 TM_yCAD背景颜色。

示例：

```
myCAD1.BackgroundColor = Color.White;
```

4.1.9 Brush

```
public Brush Brush {get; set;}
```

描述：

设置 TCAD的笔刷

示例：

如果您想要知道更多关于笔刷，请搜索名为 StartBrush的例子并且读取代码

参见：

[BrushShow](#)

4.1.10 BrushShow

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

描述：

显示/不显示笔刷库

示例：

```
myCAD1.BrushShow = true;
```

参见：

[Brush](#)

4.1.11 CrossLine

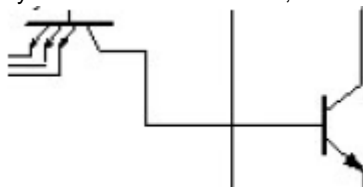
```
public Boolean CrossLine {get; set;}
```

描述：

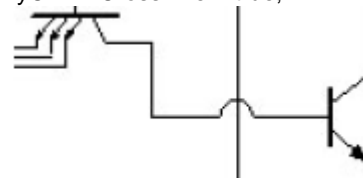
用于连接线功能，当设置为 Ture时显示交叉线特征

示例：

```
myCAD1.CrossLine = false;
```



```
myCAD1.CrossLine = true;
```



参见：

[MyLinkLine](#)

4.1.12 DragMode

```
public MyDragMode DragMode {get; set;}
```

描述：

拖动图形，当设置值为 Horizontal 时图形只能水平方向拖动，当设置值为 Vertical 时图形只能以垂直方向拖动

示例：

```
myCAD1.DragMode = MyDragMode.Horizontal;
```

4.1.13 DrawType

```
public MyDrawType DrawType {get; set;}
```

描述：

设置鼠标绘图类型

示例：

```
myCAD1.DrawType = MyDrawType.Rectangle;
```

4.1.14 GridColor

```
public Color GridColor {get; set;}
```

描述：

设置网格颜色

示例：

```
myCAD1.GridColor = Color.Red;
```

参见：

[GridHeight](#)
[GridPenSize](#)
[GridShow](#)
[GridType](#)
[GridWidth](#)

4.1.15 GridHeight

```
public Int32 GridHeight {get; set;}
```

描述：

设置网格高度。

示例：

```
myCAD1.GridHeight = 50;
```

参见：

[GridColor](#)
[GridPenSize](#)
[GridShow](#)
[GridType](#)
[GridWidth](#)

4.1.16 GridPenSize

```
public Int32 GridPenSize {get; set;}
```

描述：

设置网格笔大小。

示例：
`myCAD1.GridPenSize = 2;`

参见：
[GridColor](#)
[GridHeight](#)
[GridShow](#)
[GridType](#)
[GridWidth](#)

4.1.17 GridShow

`public Boolean GridShow {get; set;}`

描述：
显示或不显示网格

示例：
`myCAD1.GridShow = true;`

参见：
[GridColor](#)
[GridHeight](#)
[GridPenSize](#)
[GridType](#)
[GridWidth](#)

4.1.18 GridType

`public MyGridType GridType {get; set;}`

描述：
设置网格类型。网格有点和线2种形式供您选择。

示例：
`myCAD1.GridType = MyGridType.Pixel;`

参见：
[GridColor](#)
[GridHeight](#)
[GridPenSize](#)
[GridShow](#)
[GridWidth](#)

4.1.19 GridWidth

`public Int32 GridWidth {get; set;}`

描述：
设置网格宽度

示例：
`myCAD1.GridWidth = 50;`

参见：
[GridColor](#)
[GridHeight](#)
[GridPenSize](#)
[GridShow](#)
[GridType](#)

4.1.20 HotColor

`public Color HotColor {get; set;}`

描述：
指定热方块的颜色。

示例：
`myCAD1.HotColor = Color.Yellow;`

参见：
[HotShow](#)
[HotSize](#)

4.1.21 HotShow

`public Boolean HotShow {get; set;}`

描述：
显示或不显示热方块。

示例：
`myCAD1.HotShow = false;`

参见：
[HotColor](#)
[HotSize](#)

4.1.22 HotSize

`public Byte HotSize {get; set;}`

描述：
设置热方块大小。

示例：
`myCAD1.HotSize = 8;`

参见：
[HotColor](#)
[HotShow](#)

4.1.23 LabelValue

`public Label LabelValue {get; set;}`

描述：
显示当前图形的参数，可以显示线类图形的长度，矩形图形的宽和高。

示例：
`myCAD1.LabelValue = Label1;`

参见：
[LabelXY](#)

4.1.24 LabelXY

`public Label LabelXY {get; set;}`

描述：

显示当前鼠标的坐标，随着鼠标的移动而改变；

示例：

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

参见：

[LabelValue](#)

4.1.25 LinklineAroundShape

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

描述：

连接线是否环绕一个组合图形

示例：

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

参见：

[LinklineDrawStyle](#)

4.1.26 LinklineDrawStyle

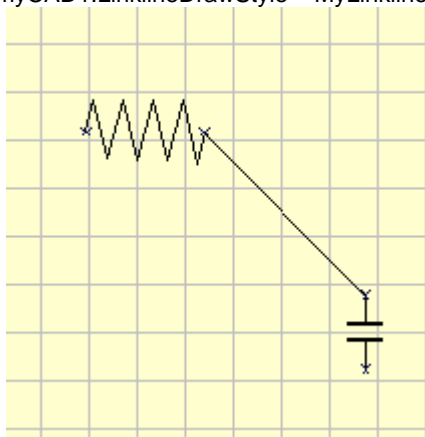
```
public MyLinklineDrawStyle LinklineDrawStyle {get; set;}
```

描述：

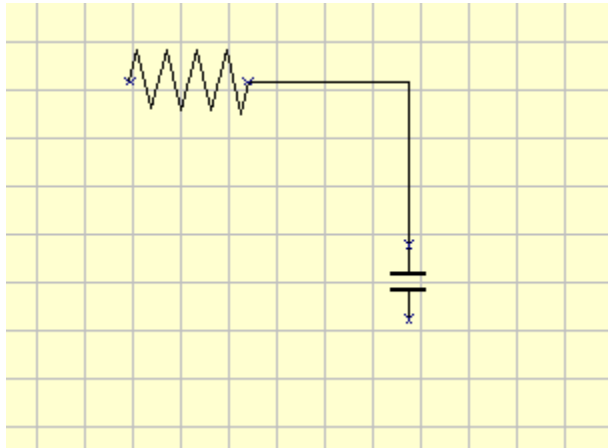
定义连接线类型

示例：

```
myCAD1.LinklineDrawStyle = MyLinklineDrawStyle.Free;
```



```
myCAD1.LinklineDrawStyle = MyLinklineDrawStyle.Horizontal/Vertical;
```



参见：

[LinklineDrawStyle](#)

4.1.27 OperateAllLayer

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

描述：

指定操作对象是所有层。

示例：

```
myCAD1.OperateAllLayer = false;
```

4.1.28 PageFoot

```
public String PageFoot {get; set;}
```

描述：

设置页脚的文字

示例：

```
myCAD1.PageFoot = "TCAD for .NET";
```

参见：

[PageFootAlignment](#)

[PageFootFont](#)

[PageFootToBottom](#)

4.1.29 PageFootAlignment

```
public MyAlignment PageFootAlignment {get; set;}
```

描述：

设置页脚文本对齐方式

示例：

```
myCAD1.PageFootAlignment = MyAlignment.Center;
```

参见：

[PageFoot](#)

[PageFootFont](#)

[PageFootToBottom](#)

4.1.30 PageFootFont

`public Font PageFootFont {get; set;}`

描述：
设置页脚文字字体

示例：
`myCAD1.PageFootFont = myCAD1.Font;`

参见：
[PageFoot](#)
[PageFootAlignment](#)
[PageFootToBottom](#)

4.1.31 PageFootToBottom

`public Byte PageFootToBottom {get; set;}`

描述：
设置页脚文字底部的边距。

示例：
`myCAD1.PageFootToBottom = 20;`

参见：
[PageFoot](#)
[PageFootAlignment](#)
[PageFootFont](#)

4.1.32 PageHead

`public String PageHead {get; set;}`

描述：
设置页眉标题。

示例：
`myCAD1.PageHead = "TCAD for .NET";`

参见：
[PageHeadAlignment](#)
[PageHeadFont](#)
[PageHeadToTop](#)

4.1.33 PageHeadAlignment

`public MyAlignment PageHeadAlignment {get; set;}`

描述：
设置页眉文本对齐方式

示例：
`myCAD1.PageHeadAlignment = MyAlignment.Right;`

参见：
[PageHead](#)
[PageHeadFont](#)
[PageHeadToTop](#)

4.1.34 PageHeadFont

`public Font PageHeadFont {get; set;}`

描述：
设置页眉文本字体

示例：
`myCAD1.PageHeadFont = myCAD1.Font;`

参见：
[PageHead](#)
[PageHeadAlignment](#)
[PageHeadToTop](#)

4.1.35 PageHeadToTop

`public Byte PageHeadToTop {get; set;}`

描述：
设置页眉边距。

示例：
`myCAD1.PageHeadToTop = 20;`

参见：
[PageHead](#)
[PageHeadAlignment](#)
[PageHeadFont](#)

4.1.36 PageHeight

`public Int64 PageHeight {get; set;}`

描述：
设置页面的大小

示例：
`myCAD1.PageHeight = 200;`

参见：
[PageWidth](#)

4.1.37 PageOrientation

`public MyPageOrientation PageOrientation {get; set;}`

描述：
设置画布的方向

示例：
`myCAD1.PageOrientation = MyPageOrientation.Landscape;`

4.1.38 PageStyle

`public MyPageStyle PageStyle {get; set;}`

描述：
设置画布的类型

示例：
`myCAD1.PageStyle = MyPageStyle.A4;`

4.1.39 PageWidth

```
public Int64 PageWidth {get; set;}
```

描述：
设置画布的宽度

示例：
`myCAD1.PageWidth = 200;`

参见：
[PageHeight](#)

4.1.40 Pen

```
public Pen Pen {get; set;}
```

描述：
设置所需要的画笔

示例：
`myCAD1.Pen.Width = 2;`
`myCAD1.Pen.Color = Color.Green;`

4.1.41 PrintBackground

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

描述：
设置是否打印背景。

示例：
`myCAD1.PrintBackground = true;`

参见：
[PrintBorder](#)

4.1.42 PrintBorder

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

描述：
设置是否打印边框，如果选真，TCAD会自动为你加一个边框。

示例：
`myCAD1.PrintBorder = true;`

参见：
[PrintBackground](#)

4.1.43 PrintBorderToBottom

```
public Byte PrintBorderToBottom {get; set;}
```

描述：
设置边框的底部边距

示例：
`myCAD1.PrintBorderToBottom = 50;`

参见：
[PrintBorderToLeft](#)

[PrintBorderToRight](#)
[PrintBorderToTop](#)

4.1.44 PrintBorderToLeft

`public Byte PrintBorderToLeft {get; set;}`

描述：
设置边框的左边距

示例：
`myCAD1.PrintBorderToLeft = 50;`

参见：
[PrintBorderToBottom](#)
[PrintBorderToRight](#)
[PrintBorderToTop](#)

4.1.45 PrintBorderToRight

`public Byte PrintBorderToRight {get; set;}`

描述：
设置边框的右边距

示例：
`myCAD1.PrintBorderToRight = 50;`

参见：
[PrintBorderToBottom](#)
[PrintBorderToLeft](#)
[PrintBorderToTop](#)

4.1.46 PrintBorderToTop

`public Byte PrintBorderToTop {get; set;}`

描述：
设置边框的顶部边距

示例：
`myCAD1.PrintBorderToTop = 50;`

参见：
[PrintBorderToBottom](#)
[PrintBorderToLeft](#)
[PrintBorderToRight](#)

4.1.47 Ratio

`public Double Ratio {get; set;}`

描述：
定义比率。这是一个非常有用的属性，它将被用在显示在 [LabelValue](#)属性中包括线的长度计算和矩形、圆形等其他图形的面积计算中。

示例：
`myCAD1.Ratio = 50;`

参见：
[UnitType](#)

4.1.48 ResizeEnable

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

描述：
设置是否改变图形的大小

示例：
`myCAD1.ResizeEnable = false;`

参见：
[RotateEnable](#)

4.1.49 ReturnToSelecting

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

描述：
当画完图形后绘图状态是否回到 selecting 状态

示例：
`myCAD1.ReturnToSelecting = false;`

4.1.50 RotateConstraintDegree

```
public Int32 RotateConstraintDegree {get; set;}
```

描述：
设置图形旋转时的约束角度，当值为 0 时，可以任意角度旋转

示例：
`myCAD1.RotateConstraintDegree = 30;`

4.1.51 RotateEnable

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

描述：
设置能否旋转图形

示例：
`myCAD1.RotateEnable = false;`

参见：
[ResizeEnable](#)

4.1.52 ShowHotLink

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

描述：
是否在 MyCAD 中显示连接点。

示例：
`myCAD1.ShowHotLink = true;`

4.1.53 Snap

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

描述：
是否对齐鼠标到网格和其他图形，这个功能可以帮助你画图。

示例：

```
myCAD1.Snap = true;
```

参见：

[SnapPixel](#)

4.1.54 SnapPixel

```
public Byte SnapPixel {get; set;}
```

描述：

鼠标和最近网格之间的像素小于这个值时，鼠标会捕获这个网格，这个值不能大于网格宽度和网格高度

示例：

```
myCAD1.Snap = true;
```

参见：

[SnapPixel](#)

4.1.55 SnapShape

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

描述：

当拖动和改变图形大小时，是否对齐到其他图形，有助于画图

示例：

```
myCAD1.SnapShape = true;
```

4.1.56 UndoRedoSize

```
public Byte UndoRedoSize {get; set;}
```

描述：

设置撤消动作保存容量，数值越大，它占用系统资源越大。

示例：

```
myCAD1.UndoRedoSize = 8;
```

4.1.57 UnitType

```
public MyUnitType UnitType {get; set;}
```

描述：

LabelValue 会按照设置的长度单位显示数值，表示是线的长度或 MyRectangle, MyEllisple 图形的面积

示例：

```
myCAD1.UnitType = MyUnitType.Millimeter;
```

4.1.58 Version

```
public String Version {get;}
```

描述：

显示 TCAD 的版本，是只读属性

示例：

```
MessageBox.Show(myCAD1.Version);
```

4.1.59 XYMode

```
public MyXYMode XYMode {get; set;}
```

描述：
有4个模式让你选择适合需求的

示例：
`myCAD1.XYMode = MyXYMode.Mode0;`

4.1.60 Zoom

```
property Zoom:Double
```

描述：
设置缩放比例、宽度、高度和背景图片都将自动地缩放。当缩放比例为1时TMYCAD是以实际大小显示。

示例：
`myCAD1.Zoom = 0.5;`

4.2 方法

4.2.1 AddImageShapeByCode

```
public Int64 AddImageShapeByCode(String shapeName,Point ItPoint,Bitmap b)
```

描述：
通过代码添加一个图片，如果你想增加其他图形，请使用 [AddShapeByCode](#) 或 [AddUserDefineShapeFromLib](#)

参数：
shapeName: 图片名称
ItPoint: 图片在左上角的位置
b: 想增加到TCAD里的图象

返回值：
返回值 ==-1,添加失败。
返回值 >=0,添加成功。

示例：

```
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);
}
```

4.2.2 AddShapeByCode

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

描述：

通过代码添加一个图形，在自动画图方面非常有用，如果你想增加图片请使用 [AddImageShapeByCode](#).

参数：

owner: TMyCAD拥有者
 shapeName: 图形名称
 drawType: 绘制图片类型
 thePoints: 坐标数组
 angle: 图形的角度
 text: 只在 MyText 时使用，设置文本

返回值：

返回值 == -1, 添加失败.
 返回值 >= 0, 添加成功, , 值是新图形的

示例：

```
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,"");
}
```

4.2.3 AddUserDefineShapeFromLib

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

描述：

从库中得到一个用户自定义图形同时将它增加到 TCAD 中，这是 TCAD 一个非常灵活的功能。

参数：

myLibrary: 库的实例
 udShapeName: 用户自定义图形名称
 centerX, centerY: 增加图形的中心位置
 myCAD: 要增加到的 TCAD 实例
 encoding: 字符编码

返回值：

返回值 == -1, 添加失败.
 返回值 >= 0, 添加成功, , 值是新图形的

示例：

```
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");
    }
}
```

4.2.4 AlignBottom

```
public void AlignBottom()
```

描述：

当你选择多个图形时，这个过程能使它们在底端对

示例：

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

4.2.5 AlignHorizontalCenter

```
public void AlignHorizontalCenter()
```

描述：

当你选择多个图形时，这个过程能使它们水平居中。

示例：

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

4.2.6 AlignLeft

```
public void AlignLeft()
```

描述：

当你选择多个图形时，这个过程能使它们在左端对齐。

示例：

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

4.2.7 AlignRight

```
public void AlignRight()
```

描述：

当你选择多个图形时，这个过程能使它们在右端对齐。

示例：

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

4.2.8 AlignTop

```
public void AlignTop()
```

描述：

当你选择多个图形时，这个过程能使它们在顶端对齐

示例：

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

4.2.9 AlignVerticalCenter

```
public void AlignVerticalCenter()
```

描述：

当你选择多个图形时，这个过程能使它们垂直居中

例子：

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

4.2.10 BringToFront

```
public void BringToFront(MyShape tmpShape, Boolean needSaved)
```

描述：

移动选择的图形到最前

参数：

tmpShape: 选择的图形
needSaved: 是否放入 Undo 队列，缺省为 true

示例：

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

参见：

[SendToBack](#)

4.2.11 BringToFrontByStep

```
public void BringToFrontByStep(MyShape tmpShape, Boolean needSaved)
```

描述：

选择的图形向前移一层

参数：

tmpShape: 你选择的图形
needSaved: 是否保存到 undo 队列，用于回退 true

示例：

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

[SendToBackByStep](#)

4.2.12 ClearAllUndoStuff

```
public void ClearAllUndoStuff()
```

描述：

在内存中清空所有的undo队列。在开始一个新绘图时必须使用

示例：

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

4.2.13 Copy

```
public void Copy()
```

描述：

在内存中保存被选中的图形，同时它们也以 bitmap 图片格式保存在剪贴板中，可以在其他的应用软件中使用。

示例：

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

参见：

[Cut](#)
[Paste](#)

4.2.14 CreateLink

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

描述：

在源图形和目标图形之间创建一条连接线。

参数：

linkName: 连接线名称

srcShape: 源图形

srcLinkPtId: 源图形的连接点Id

destShape: 目标图形，这是一个有一个连接点的用户自定义图形

destLinkPtId: 目标图形的连接点Id

返回值：当返回值是 -1

时表示增加失败，返回其它一个大于零的时表示增加连线成功，其返回值是连接线的 ShapeId，您可以用 GetShapeById 获得这个连接线图形的实例。

示例：

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

4.2.15 Cut

`public void Cut()`

描述：

剪切功能。在内存中保存选择的图形，并删除被选择中的图形，同时以 bitmap 图片格式保存在系统的剪贴板中。

示例：

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

参见：

[Copy](#)
[Paste](#)

4.2.16 Delete

`public void Delete()`

描述：

删除被选中的图形。

示例：

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

4.2.17 DeleteAllLayers

`public Boolean DeleteAllLayers()`

描述：

删除所有的图层和所有的图形，当前的图层标识符 (CurrentLayerId) 被设置为 -1.

返回值：

返回 true 表示所在图层已删除，返回 false 表示删除失败

示例：

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

4.2.18 DeleteAllShapes

`public void DeleteAllShapes()`

描述：

删除所有的图形，图层仍旧存在。

示例：

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

```
}
```

4.2.19 DeleteLayerById

```
public Boolean DeleteLayerById(Int32 layerId)
```

描述：

通过图层标识符来删除图层，如果所给出的图层标识符不存在，返回 false，这个函数会删除该图层上的所有的图形。

参数：

layerId: 图层 ID

返回值：

true: 删除成功
false: 删除失败

示例：

```
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();
}
```

参见：

[DeleteLayerByName](#)

4.2.20 DeleteLayerByName

```
public Boolean DeleteLayerByName(String layerName)
```

描述：

通过图层名称来删除图层，如果图层名称不存在，返回 false，这个函数会删除该图层上的所有图形。

参数：

layerName: 图层名称

返回值：

true: 删除成功
false: 删除失败

示例：

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

参见：

[DeleteLayerById](#)

4.2.21 DeleteSelectedShapes

```
public Boolean DeleteSelectedShapes()
```

描述：

删除当前所选择的图形

返回值：

返回true表示删除成功，返回false表示删除失败

示例：

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

4.2.22 DeleteShapeById

```
public void DeleteShapeById(Int64 aShapeId, Boolean NeedSaved)
```

描述：

通过图形标识符删除一个图形或一个组合图形。你不可以用这个过程删除一个子图形。

返回值：

返回true删除成功，返回false删除失败

示例：

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

4.2.23 DeSelectedAllShapesByCode

```
public void DeSelectedAllShapesByCode()
```

描述：

执行这个命令后，所有图形处于没有被选择状态。

示例：

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

4.2.24 FlipHorizontal

```
public void FlipHorizontal(MyShape aShape)
```

描述：

水平翻转图形。

参数：

aShape: 你想翻转这个图形对象。

示例：

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

参见：

[FlipVertical](#)

4.2.25 FlipVertical

```
public void FlipVertical(MyShape aShape)
```

描述：

垂直翻转图形

参数：

aShape: 你想翻转的图形对象。

示例：

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

参见：

[FlipHorizontal](#)

4.2.26 GetLayerIdByName

```
public Int32 GetLayerIdByName(String layerName)
```

描述：

通过图层名称得到图层标识符

参数：

layerName: 图层名称

返回值：

返回值 == -1, 设置图层名称成功。
返回值 >= 0, 设置图层名称失败。

示例：

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

参见：

[GetLayerIdByNo](#)

4.2.27 GetLayerIdByNo

```
public Int32 GetLayerIdByNo(Int32 no)
```

描述：

给出图层序号得到标识符。No是表示某图层在数组中的物理位置，LayerId是逻辑标识符。

参数：

no: 某图层在数组中的物理位置

返回值:

返回值 == -1, 逻辑标识符不存在。
返回值 >= 0, 逻辑标识符存在。

示例:

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

参见:

[GetLayerIdByName](#)

4.2.28 GetLayerNameById

```
public String GetLayerNameById(Int32 layerId)
```

描述:

给出图层的逻辑标识符得到图层名称。

参数:

layerId: 标识符

返回值:

返回值 == "", 标识符设置不成功

示例:

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

4.2.29 GetLayerNoById

```
public Int32 GetLayerNoById(Int32 layerId)
```

描述:

给出标识符得到图层序号得到是表示某图层在数组中的物理位置。 LayerNo 是表示某图层在数组中的物理位置, LayerId是逻辑标识符。

参数:

layerId: 逻辑标识符

返回值:

返回值 == -1, 无法得到某图层在数组中的物理位置。
返回值 >= 0, 能得到某图层在数组中的物理位置。

示例:

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


4.2.30 GetLayerNoByName

```
public Int32 GetLayerNoByName(String layerName)
```

描述：

通过图层名称得到某图层在数组中的物理位置。

描述：

layerName: 图层名称

返回值：

某图层在数组中的物理位置

返回值 == -1, 无法找到这个名称的图层

返回值 >= 0, 找到这个名称的图层。

示例：

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

4.2.31 GetLayersCount

```
public Int32 GetLayersCount()
```

描述：

得到 TMyCAD 当前的图层总数

返回值：

返回图层总数

示例：

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

4.2.32 GetMaxLayerId

```
public Int32 GetMaxLayerId()
```

描述：

得到最大的 LayerId 的值

返回值：

返回最大的图层标示符的值

示例：

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

4.2.33 GetRootParentShape

```
public MyShape GetRootParentShape(MyShape aShape)
```

描述：

得到根目录图形

参数：

aShape: 图形

返回值：

返回到根目录图形

示例：

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

4.2.34 GetSelectedShape

```
public MyShape GetSelectedShape()
```

描述：

得到选择的图形对象。

返回值：

当选择了多个图形时返回 nil；当选择的是一个组合图形的子图形则返回他的父图形

示例：

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

4.2.35 GetSelectedShapes

```
public MyShape[] GetSelectedShapes()
```

描述：

得到选择的图形

返回值：

假如选择多个图形，使用这个函数得到图形数组

示例：

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

4.2.36 GetSelectedShapesCount

```
public Int64 GetSelectedShapesCount()
```

描述：

得到当前选择图形的个数，组合图形算一个图形

返回值：

假如选择多个图形，使用这个函数得到图形数量。

示例：

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

4.2.37 GetShapeById

```
public MyShape GetShapeById(Int64 aShapeId)
```

描述：

通过 ShapeId 得到图形实例。

参数：

aShapeId: 图形标示

返回值：

如果图形标识符不存在，返回空值

示例：

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

参见：

[GetShapeByNo](#)

4.2.38 GetShapeByNo

```
public MyShape GetShapeByNo(Int64 aShapeNo)
```

描述：

通过图形序号得到这个图形。

参数：

aShapeNo: 图形序号

返回值：

如果图形序号不存在，返回空值

示例：

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

参见：

[GetShapeById](#)

4.2.39 GetShapeNoById

```
public Int64 GetShapeNoById(Int64 aShapeId)
```

描述：

通过图形标识符得到图形序号

参数：

aShapeId: 图形标识符

返回值：

如果图形标识符不存在，返回 -1

示例：

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

4.2.40 GetShapesCount

```
public Int64 GetShapesCount()
```

描述：

得到当前 TCAD 的图形总数

返回值：

返回图形总数。

示例：

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

4.2.41 GetShapesCountInALayer

```
public Int64 GetShapesCountInALayer(Int32 layerId)
```

描述：

得到某个图层上的图形总数

参数：

layerId: 图层标识符

返回值：

在图层中返回图形的总数

示例：

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

4.2.42 GetWorkingShapesCount

```
public Int64 GetWorkingShapesCount()
```

描述：

得到被选中的一个或多个图形

返回值：

-1：失败

else : 组合图形的图形个数

示例：

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

4.2.43 GroupWorkingShapes

`public Int64 GroupWorkingShapes()`

描述：

组合被选中的一个图形

返回值：

-1 : 组合图形失败
else : 组合图形的 ShapeID.

示例：

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

参见：

[UngroupShape](#)

4.2.44 InVisibleLayerById

`public void InVisibleLayerById(Int32 layerId)`

描述：

按id设置图层隐藏

参数：

layerId: 图层id

示例：

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

参见：

[InVisibleLayerByName](#)

4.2.45 InVisibleLayerByName

`public void InVisibleLayerByName(String layerName)`

描述：

按名字设置图层隐藏

参数：

layerName: 图层名字

示例：

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

参数：

[InVisibleLayerByld](#)

4.2.46 LoadFromFile

```
public Boolean LoadFromFile(String fileName,String encoding)
```

描述：

从一个文件中装入 TCAD.

描述：

fileName: 想要装入的文件

返回值：

true: 装入成功
false: 装入失败
encoding: 字符编码

示例：

```
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();
    }
}
```

参见：

[SaveToFile](#)

4.2.47 LoadFromStream

```
public Boolean LoadFromStream(FileStream fileStream,String encoding)
```

描述：

从一个流中装入 TCAD

参数：

fileStream: 文件流

返回值：

true: 装入成功
false: 装入失败
encoding: 字符编码

示例：

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

参见：

[SaveToStream](#)

4.2.48 NewLayer

```
public Int32 NewLayer(String IName, Boolean IVisible)
```

描述：

在 TCAD 中增加一个新的图层

参数：

IName: 图层名称

IVisible: 图层是否可见

返回值：

返回图层标示符

示例：

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

4.2.49 Paste

```
public void Paste()
```

描述：

粘贴图形，默认有 4 个像素的偏移。

示例：

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

参数：

[Copy](#)
[Cut](#)

4.2.50 PopfromUndoRedoShapeList

```
public Int32 PopfromUndoRedoShapeList()
```

描述：

撤消一步

返回值：

返回几步骤

示例：

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

4.2.51 Preview

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

描述：

打印预览，可以选全部图层或部分图层。

参数：

IstLayer: 图层名单
previewBitmap: 位图
previewZoom: 预览时的放大倍数

示例：

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

    pictureBox1.Image = b;
}
```

4.2.52 Print

```
public void Print(Int32[] IstLayer, Double previewZoom)
```

描述：

打印。

参数：

IstLayer: 图层名单
previewZoom: 打印比例

实例：

```
private void btnPrint_Click(object sender, System.EventArgs e)
{
    //
    Int32[] IstLayer = new Int32[1];
    IstLayer[0] = 0;
    mainForm.myCAD1.Print(IstLayer, 1);
}
```

4.2.53 SaveToFile

```
public Boolean SaveToFile(String fileName)
```

描述：

保存文件。

参数：

fileName: 文件名称

返回值：

true: 保存成功
false: 保存失败

示例：

```
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);
    }
}
```

参数：

[LoadFromFile](#)

4.2.54 SaveToImage

```
public void SaveToImage(String fileName, ImageFormat imageFormat)
```

描述：

以图片的格式保存。

参数：

fileName: 文件名称
imageFormat: 图片格式

示例：

```
private void menuItem29_Click(object sender, System.EventArgs e)
{
    saveFileDialog1.InitialDirectory = Application.StartupPath;
    saveFileDialog1.Filter = "Bitmap file(*.bmp)|*.bmp";
    saveFileDialog1.FilterIndex = 1;
    saveFileDialog1.DefaultExt = "bmp";
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        myCAD1.SaveToImage(saveFileDialog1.FileName, ImageFormat.Bmp);
    }
}
```

4.2.55 SaveToStream

```
public Boolean SaveToStream(FileStream fileStream)
```

描述：

创建保存在系统上。

参数：

fileStream: 文件流

返回值：

true: 保存成功
false: 保存失败

示例：

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

```
        myCAD1.SaveToStream(myFileStream);  
    }  
参数：  
    LoadFromStream
```

4.2.56 SelectAllShapes

```
public void SelectAllShapes()
```

描述：
选择所有图形

示例：

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

4.2.57 SelectShapeByCode

```
public Boolean SelectShapeByCode(Int64 aShapeId, Boolean removePrevSelectedShape)
```

描述：
用代码选择图形。

参数：
aShapeId: 想要选择的图形标示符
removePreSelectedShape: 是否取消以前选中的图形

返回值：
true: 选择成功
false: 选择失败

示例：

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

4.2.58 SendToBack

```
public void SendToBack(MyShape tmpShape, Boolean needSaved)
```

描述：
将图层置最底层

参数：
tmpShape: 将要移动的图层
needSaved: 是否加入undo的队列

示例：

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

参见：
[BringToFront](#)

4.2.59 SendToBackByStep

```
public void SendToBackByStep(MyShape tmpShape, Boolean needSaved)
```

描述:

将图层置后一层

参数:

tmpShape: 将要置后的图层
needSaved: 是否加入 undo 的队列

示例:

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

参见:

[BringToFrontByStep](#)

4.2.60 SetLayerNameById

```
public void SetLayerNameById(String layerName, Int32 layerId)
```

描述:

按图层的标示符设置名称

参数:

layerName: 想要设置的图层名称
layerId: 图层标示符

示例:

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

参数:

[SetLayerNameByName](#)

4.2.61 SetLayerNameByName

```
public void SetLayerNameByName(String oldName, String newName)
```

描述:

通过图层名称来设置图层名称

参数:

oldName: 旧的图层名字
newName: 想要设置的图层名字

示例:

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

参见:

[SetLayerNameById](#)

4.2.62 ShapeMove

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

描述：
移动一个图形

参数：
tmpShape: 想要移动的图形
dx,dy: 移动的坐标

示例：

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

4.2.63 ShapeRotate

```
public void ShapeRotate(MyShape tmpShape,Single angle)
```

描述：
图形旋转。

参数：
tmpShape: 想要旋转的图形
angle: 旋转的角度(弧度)

示例：

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

4.2.64 SizeShape

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

描述：
改变图形的大小

参数：
tmpShape: 想要改变的图形
selectedHotId: 选择热点的标示符
dragPoint: 目标位置

示例：

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

4.2.65 UngroupShape

```
public void UngroupShape(MyShape tmpShape,Boolean needSaved)
```

描述：
取消一个组合图形

参数：

TMPShape：想要取消组合的图形
NeedSaved：是否放入 Undo 队列，缺省为 true

示例：

```
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.");
    }
}
```

参数：

[GroupWorkingShapes](#)

4.2.66 VisibleAllLayer

```
public void VisibleAllLayer()
```

描述：

使所有图层可见。

示例：

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

参见：

[VisibleLayerById](#)
[VisibleLayerByName](#)

4.2.67 VisibleLayerById

```
public void VisibleLayerById(Int32 layerId)
```

描述：

使指定标识符的图层可见

参数：

layerId: 标示符

示例：

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

参见：

[VisibleAllLayer](#)
[VisibleLayerByName](#)

4.2.68 VisibleLayerByName

`public void VisibleLayerByName(String layerName)`

描述：
使某一图层可见

参数：
layerName: 某一图层名称

示例：

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

参见：
[VisibleAllLayer](#)
[VisibleLayerByld](#)

4.3 事件

4.3.1 ChildShapeSelected

`public event ChildShapeSelectedEventHandler ChildShapeSelected;`

描述：
当选择了一个子图形的事件。

示例：

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

4.3.2 DrawTypeToSelecting

`public event DrawTypeToSelectingEventHandler DrawTypeToSelecting;`

描述：
当回到 DrawTypeToSelecting状态的事件

示例：

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

4.3.3 EnterShape

`public event EnterLeaveShapeEventHandler EnterShape;`

描述：
当鼠标进入 TMxCAD时触发的事件

示例：

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

4.3.4 LeaveShape

`public event EnterLeaveShapeEventHandler LeaveShape;`

描述：

当鼠标离开 TMyCAD 时触发事件。

示例：

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

4.3.5 NodeAdded

`public event NodeAddDeleteEventHandler NodeAdded;`

描述：

当创建 PolyLine or Polygon 时添加一个节点 它将触发该事件

示例：

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

4.3.6 NodeDeleted

`public event NodeAddDeleteEventHandler NodeDeleted;`

描述：

当创建 PolyLine or Polygon 时删除一个节点 它将触发该事件

示例：

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

4.3.7 OnDeleteLayer

`public event LayerOperationEventHandler OnDeleteLayer;`

描述：

当删除一个图层时触发 OnDeleteLayer 事件

示例：

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

```
}

```

4.3.8 OnNewLayer

```
public event LayerOperationEventHandler OnNewLayer;
```

描述：

当增加一个图层时触发 OnNewLayer事件

示例：

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

4.3.9 ShapeAdded

```
public event ShapeAddedEventHandler ShapeAdded;
```

描述：

当增加一个图形时触发 OnShapeAdded事件

示例：

```
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;
        }
    }
}
```

4.3.10 ShapeCodeDragging

```
public event ShapeCodeDragResizeEventHandler ShapeCodeDragging;
```

描述：

当通过代码拖动一个图形时触发 OnShapeCodeDragging事件。

示例：

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

4.3.11 ShapeCodeRotating

```
public event ShapeCodeRotateEventHandler ShapeCodeRotating;
```


描述：

当通过代码旋转图形时触发 OnShapeCodeRotating事件

示例：

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

4.3.12 ShapeDeleted

```
public event ShapeDeletedEventHandler ShapeDeleted;
```

描述：

当删除一个图形时触发 OnShapeDeleted事件

示例：

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

4.3.13 ShapeMouseDragged

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseDragged;
```

描述：

当用鼠标拖动图形时触发 OnShapeMouseDragged事件

示例：

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

4.3.14 ShapeMouseDragging

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseDragging;
```

描述：

当用鼠标拖动图形时触发 OnShapeMouseDragging事件。

示例：

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

4.3.15 ShapeMouseResized

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseResized;
```

描述：

当用鼠标改变图形大小后触发 OnShapeMouseResized事件

示例：

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

4.3.16 ShapeMouseResizing

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseResizing;
```

描述：

当通过鼠标更改图形大小时触发 OnShapeMouseResizing事件

示例：

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

4.3.17 ShapeMouseRotated

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseRotated;
```

描述：

当通过鼠标旋转图形后触发 OnShapeMouseRotated事件

示例：

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

4.3.18 ShapeMouseRotating

```
public event ShapeMouseDragResizeRotateEventHandler ShapeMouseRotated;
```

描述：

当通过鼠标旋转一个图形时触发 OnShapeMouseRotating事件

示例：

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

4.3.19 ShapeSelected

```
public event ShapeSelectedEventHandler ShapeSelected;
```

描述：

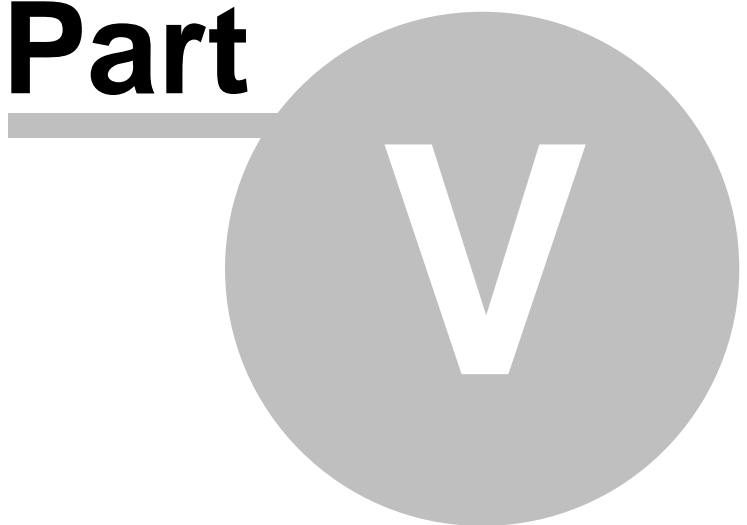
放选择一个图形，触发的事件

示例：

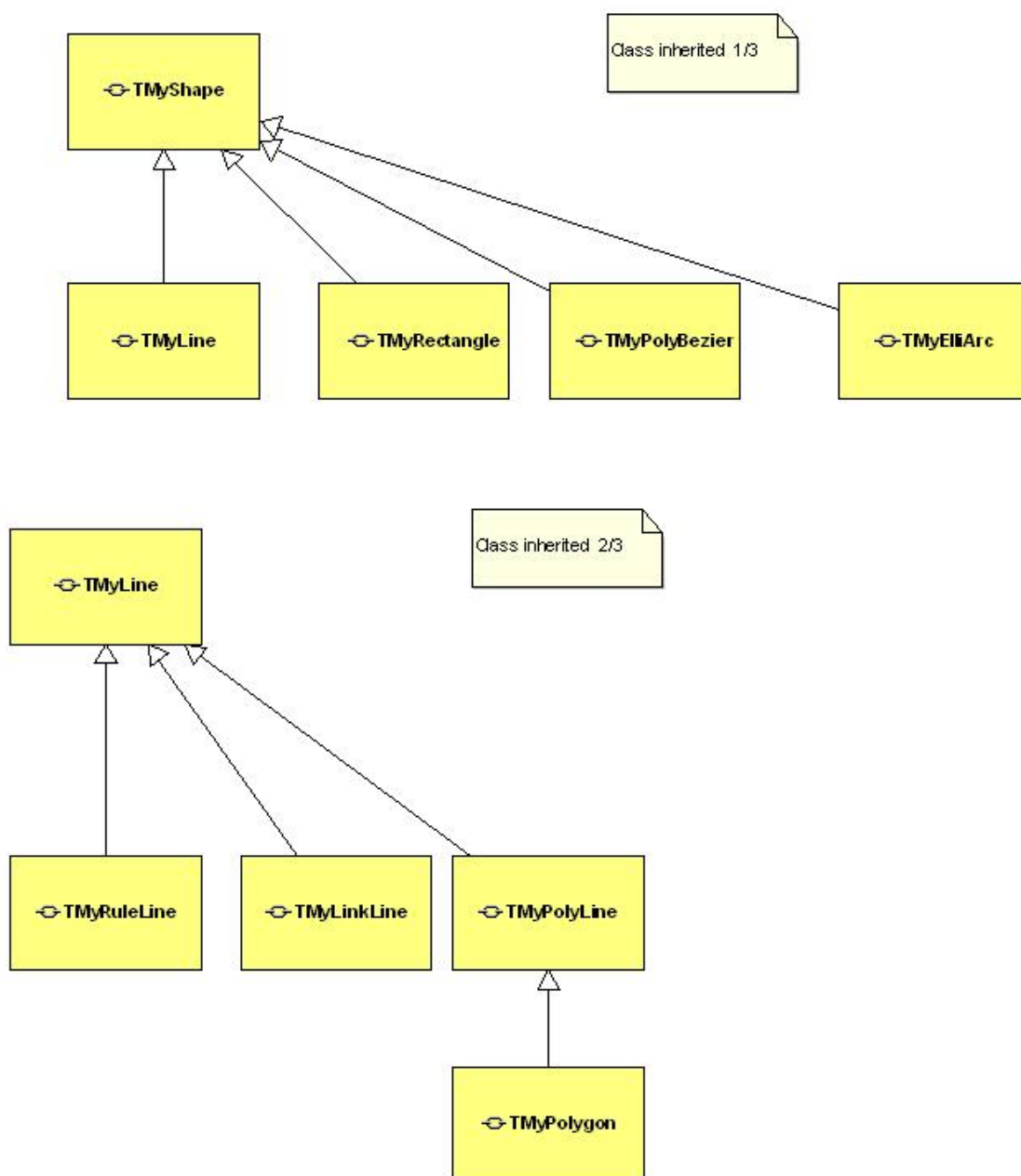
```
private void myCAD1_ShapeSelected(object sender, Codeidea.UltraGraphics.ShapeSelectedEventArgs e)
{
}
```

```
//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



5 图形类继承关系图



Part

VI

6 MyShape 类

6.1 字段

6.1.1 CenterPoint

```
public PointF CenterPoint;
```

描述：

一个图形的中心点，在 TMyCAD 中会自动维护。

6.1.2 ChildShapesNo

```
public Int64 [] ChildShapesNo;
```

描述：

假如一个图形是一个单独的图形，那么它的值是空值，假如这个图形是一个组合图形或自定义图形，在这个父图形的这个字段里会存储子图形的序号(是 ShapeNo, 不是 ShapeId), 你可以用 GetShapeByNo 得到这个子图形。

6.1.3 LayerId

```
public Int32 LayerId;
```

描述：

指定或调整这个图形属于哪个图层，一个图形必须在一个图层上。

6.1.4 LinkPoints

```
public PointF[] LinkPoints;
```

描述：

如果它有链接点，这个域将存放链接点。

6.1.5 LinkShapesNo

```
public Int64[] LinkShapesNo;
```

描述：

假如一个图形和另一个图形存在连接，. 连接线图形的 ShapeNo 会被储存在被连接图形的 LinksShapeNo 属性中，你可以使用 ShapeNo 方便地找到连接线图形

6.1.6 ParentShapeNo

```
public Int64 ParentShapeNo = -1;
```

描述：

父图形的序号(ShapeId)，如果这个图形没有父图形，这个值为 -1

6.1.7 ShapeId

```
public Int64 ShapeId;
```

描述：

图形的标识符，删除或增加一个图形，它不能更改，是一个自增字段域，它在 TMyCAD 中维护，只读

6.1.8 ShapeNo

```
public Int64 ShapeNo;
```

描述：

这个图形的序号，当删除、新增、移前、移后等操作时这个值会改变，在 TMyCAD 中维护，对于你来说它只能读取

6.1.9 TextOutPoint

```
public PointF TextOutPoint;
```

描述：

这是写文本所在的位置，当一个图形创建时，在图形的中下方位置，你可以通过鼠标或代码来改变这个位置

6.1.10 ThePoints

```
public PointF[] ThePoints;
```

描述：

一个图形的位置点数组，不能的图形有不同的位置点个数，它和 XyMode 没有关系。右上角位置是 (0,0)，它是 TMyPoint 类型

6.2 属性

6.2.1 Alpha

```
public Byte Alpha {get; set;}
```

描述：

设置图形的透明度，值为 0 到 255.

示例：

```
myShape.Alpha = 100;
```

6.2.2 Angle

```
public Single Angle {get; set;}
```

描述：

设置图形的角度

示例：

```
图形的角度 = Convert.ToSingle(30 * Math.PI / 180);
```


6.2.3 Brush

`public Brush Brush {get; set;}`

描述：

设置你所需 TMyShape 的画刷

示例：

如果您想要知道更多关于笔刷，请检索名为 StartBrush 的例子并且读取代码。

.

参见：

[BrushShow](#)

6.2.4 BrushShow

`public Boolean BrushShow {get; set;}`

描述：

是否显示 TMyShape 的画刷

示例：

`myShape.BrushShow = true;`

参见：

[Brush](#)

6.2.5 Caption

`public String Caption {get; set;}`

描述：

当创建图形的标记图形时，Name 属性同这个图形的名称，你可以改变它。

示例：

`myShape.Caption = "TCAD for .NET";`

6.2.6 CaptionShow

`public Boolean CaptionShow {get; set;}`

描述：

设置信息提示是否显示，提示信息就是 Caption 属性值

示例：

`myShape.CaptionShow = true;`

6.2.7 Font

`public Font Font {get; set;}`

描述：

TFont 用于设置文本显示风格，TFont 定义了一个指定高度、类型、属性等的集合，TFont 已成为 Windows 惯用的字体。

设置：

`myShape.Font = myCAD1.Font;`

6.2.8 IsFlipHorizontal

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

描述：

改变图形的翻转状态，当设置成 true，图形水平翻转

示例：

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

6.2.9 IsFlipVertical

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

描述：

改变图形的翻转状态，当设置成 true，图形垂直翻转。

示例：

```
myShape.IsFlipVertical = true;
```

6.2.10 Info

```
public String Info {get; set;}
```

描述：

它存储字符串，你可以根据自己需要设置

示例：

```
myShape.Info = "I like TCAD";
```

6.2.11 Lock

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

描述：

锁定图形，热点区域将以灰色显示。

示例：

```
myShape.Lock = true;
```

6.2.12 Name

```
public String Name {get; set;}
```

描述：

在一个应用程序中使用使用 Name 属性来确定一个图形。

示例：

```
myShape.Name = "Shape0";
```

6.2.13 Owner

```
public MyCAD Owner {get; set;}
```

描述：

指定图形属于哪个 TMyCAD

6.2.14 Pen

```
public Pen Pen {get; set;}
```

描述：

设定图形所需的画笔格式

示例：

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

6.2.15 ShowUnit

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

描述：

是否显示单位长度

示例：

```
myShape.ShowUnit = true;
```

6.2.16 Tag

```
public Int32 Tag {get; set;}
```

描述：

设置图形标记

示例：

```
myShape.Tag = 32;
```

6.2.17 UserData

```
public MyUserData UserData {get; set;}
```

描述：

你可以增加你自己的数据，它非常有用，你可以使用这个和数据库相连，说直接将数据存放在这里。

示例：

```
Please read the demo code.
```

6.2.18 Visible

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

描述：

当设置为false时，图形不能被选中、改变大小、旋转、组合。

示例：

```
myShape.Visible = true;
```

6.3 方法

6.3.1 Assign

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

描述：

拷贝一个图形对象

实例：

aShape: The source shape.

6.3.2 ComputeCenterPoint

```
public PointF ComputeCenterPoint()
```

描述：

重新计算出一个图形的中心点。

返回值：

返回中心点

6.3.3 Dispose

```
public virtual void Dispose()
```

描述：

首先释放自己所有的字段域，最后调用父亲的dispose方法

6.3.4 Draw

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

描述：

创建一个图形

参数：

grp: The canvas instance.

6.3.5 GetCenterPoint

```
public virtual PointF GetCenterPoint()
```

描述：

返回一个图形的中心点

返回：

一个图形的中心点

6.3.6 GetCenterPointInZoom

```
public virtual PointF GetCenterPointInZoom()
```

描述：

返回当前缩放比例的中心点

返回值：

返回当前缩放比例的中心点

6.3.7 GetHeight

```
public Single GetHeight()
```

描述：

返回一个图形的外接矩形高度，单位是像素。

返回值：

返回图形的外接矩形高度

6.3.8 GetLeftBottom

```
public PointF GetLeftBottom()
```

描述：

返回一个外矩形的左下角的坐标，单位是 TMyPoint

返回值：

返回左下角的坐标。

6.3.9 GetLeftTop

```
public PointF GetLeftTop()
```

描述：

返回外矩形的左上角的坐标，单位是 TMyPoint

返回：

返回左上角的坐标。

6.3.10 GetLinkPointInZoom

```
public PointF GetLinkPointInZoom(Int32 iPoint)
```

描述：

在当前比例下得么连接点，详细信息请查看 GetLinkPoint 方法

返回值：

返回连接点的缩放

6.3.11 GetLinkPointsCount

```
public Int32 GetLinkPointsCount()
```

描述：

通过某个图形连接点的个数

6.3.12 GetMyHeight

```
public Single GetMyHeight()
```

描述：

得到图形的实际高度，不是外接矩形的高度，它不考虑角度，单位是像素

返回值：

返回实际高度

6.3.13 GetMyWidth

```
public Single GetMyWidth()
```

描述：

得到图形的实际宽度，不是外接矩形的宽度，不考虑角度，单位是像素

返回值：

返回实际宽度。

6.3.14 GetPoint

```
public PointF GetPoint(Int32 iPoint)
```

描述：

通过点标识符得到点

参数：

PointID：给出点标识符，其值必须在 PointID>=0 and PointID<=GetPointsCount-1范围内

返回：

点的值

6.3.15 GetPointInZoom

```
public PointF GetPointInZoom(Int32 iPoint)
```

描述：

得到当前缩放比例的点。详细信息请查看 GetPoint

参数：

iPoint: 你想要的点的顺序

返回值：

返回点的值

6.3.16 GetPointsCount

```
public Int32 GetPointsCount()
```

描述：

得到一个图形的点数

6.3.17 GetRightBottom

```
public PointF GetRightBottom()
```

描述：

返回一个图形的外矩形的右下角边距，单位是 TMyPoint.

返回值：

返回右下角边距

6.3.18 GetRightTop

`public PointF GetRightBottom()`

描述：

返回一个图形的外矩形的右上角的坐标，单位是 TMyPoint

返回值：

返回右上角的坐标.

6.3.19 GetShapeld

`public Int64 GetShapeld()`

描述：

通过图形的标示符

返回值：

图形的标示符

6.3.20 GetWidth

`public Single GetHeight()`

描述：

返回一个图形外接矩形的宽度，单位是像素

返回：

返回图形的宽度

6.3.21 HasChildShapes

`public Boolean HasChildShapes()`

描述：

判断一个图形是否有子图形

返回值：

true：图形有子图形
false：图形没有子图形.

6.3.22 HasLinkShapes

`public Boolean HasLinkShapes()`

描述：

判断一个图形是否有连接的图形

返回值：

true：图形有连接的图形
false：图形没有连接的图形。

6.3.23 HasParentShape

`public Boolean HasParentShape()`

描述：

判断一个图形是否有父图形。

参数：

true：图形有父图形
false：图形没有父图形。

6.3.24 LoadFromOldStream

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

描述：

从流中加载一个图形，像 TmyLine对象等可重载此方法

参数：

fileStream: 文件流

6.3.25 LoadFromStream

`public virtual void LoadFromStream(Stream fileStream)`

描述：

保存一个图形到流，像 TmyLine对象等可覆盖此方法

参数：

fileStream: 文件系统

6.3.26 SaveToStream

`public virtual void SaveToStream(Stream fileStream)`

描述：

保存一个图形到流，像 TmyLine对象等可重载此方法

参数：

fileStream: 文件流

Part

VII

7 Myline

线图形的类，它定义了属性、事件、方法

7.1 属性

7.1.1 ArrowAngle

参考 TMyCAD 的属性。

7.1.2 ArrowLength

参考 TMyCAD 的属性。

7.1.3 ArrowOffset

.参考 TMyCAD 的属性。

7.1.4 ArrowStyle

参考 TMyCAD 的属性。

7.2 方法

7.2.1 Assign

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

描述：

拷贝一个线图形。

参数：

aShape: 图形的来源

7.2.2 Draw

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

描述：

在画布上画一线形

参数：

grp: 画布的句柄

7.2.3 GetInfo

```
public override String GetInfo(Graphics grp)
```

描述：

覆盖基类的 GetInfo，它返回线的长度，改变画布的大小，其值也会改变

参数：

grp: 画布的句柄

返回值:

The info

7.2.4 LoadFromOldStream

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

描述:

从流中加载一个以前图形.

参数:

fileStream: 文件流

7.2.5 LoadFromStream

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

描述:

从流中加载一个图形.

参数:

fileStream: 系统

7.2.6 MyLine

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

描述:

MyLine. 的创建过程

参数:

AOwner: MyCAD 拥有者

7.2.7 SaveToStream

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

描述:

保存一个图形到流

参数:

fileStream: 文件流

Part



8 MyLinkLine

连接线的类，它定义了属性、事件、方法，TCAD xp标准版不支持连线功能

8.1 属性

8.1.1 LinklineDrawStyle

```
public MyLinklineDrawStyle LinklineDrawStyle
```

描述：

定义连接线类型

8.1.2 EndSpNo

```
public Int64 EndSpNo
```

描述：

起始图形的连接点标识符，可读写，如果没有起始图形，其值为-1

8.1.3 EndSpPtId

```
public Int32 EndSpPtId
```

描述：

起始图形的指针。 渲染?1时，则表示没有起始图
|

8.1.4 StartSpNo

```
public Int64 StartSpNo
```

描述：

结束图形的指针，其值为-1表示没有结束图形

8.1.5 StartSpPtId

```
public Int32 StartSpPtId
```

描述：

结束图形连接点的标识符，运行时属性、可读写，如果没有结束图形，其值为-1

8.2 方法

8.2.1 Assign

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

描述：

拷贝一个连接线图形，方法重载父类Assign方法。

参数：

aShape: 源图形

8.2.2 CreateDestLink

```
public Boolean CreateDestLink(Int64 aShapeId,Int32 aShapeLinkId)
```

描述:

它能以水平或垂直的方式连接两个图形，通常用于流程图、电力接线图等

参数:

aShapeLinkId: 图形的连接点标示符

返回值:

true: 创建成功
false: 创建失败

参见:

[CreateSrcLink](#)

8.2.3 CreateSrcLink

```
public Boolean CreateSrcLink(Int64 aShapeId,Int32 aShapeLinkId)
```

描述:

它可以同一个外部图形创建一个连接关系

参数:

aShapeId: 结束图形的标示符
aShapeLinkId: 图形连接点的标示符

返回值:

true: 创建成功
false: 创建失败

参见:

[CreateDestLink](#)

8.2.4 Draw

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

描述:

在 MyCanvas 上画该图形

参数:

grp: 画布的句柄

8.2.5 GetEndPoint

`public PointF GetEndPoint()`

描述：
得到结束端图形的连接点

返回值：

假如没有结束端图形，返回连接线自身的最后一点，否则返回结束端图形连接点。

8.2.6 GetEndShape

`public MyShape GetEndShape()`

描述：
得到结束端图形

返回值：

null: 没有结束图形
else: 结束图形

8.2.7 GetStartPoint

`public PointF GetStartPoint()`

描述：
得到开始端图形的连接点

返回值：

返回起始连接图形的连接点，假如没有起始图形，返回连接线自身的最后一点。

8.2.8 GetStartShape

`public MyShape GetStartShape()`

描述：
得到起始端图形

返回值：

null: 没有起始图形
else: 结束图形。

8.2.9 LoadFromOldStream

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

描述：
从老的流中加载一个图形

参数：
fileStream: 系统

8.2.10 LoadFromStream

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

描述：

从流中加载一个线形图形

参数：

fileStream: 文件流

8.2.11 MyLinkLine

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

描述：

MyLinkLine的创建过程

参数：

AOwner: MyLinkLine的 拥有者， TMyCAD类型

8.2.12 RemoveDestLink

```
public void RemoveDestLink()
```

描述：

取消目标连接的图形

.

8.2.13 RemoveSrcLink

```
public void RemoveSrcLink()
```

描述：

取消源连接的图形

8.2.14 SaveToStream

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

描述：

保存到一个流中

参数：

fileStream: 文件流

Part

IX

9 MyPolyLine

多段线的类，它定义了属性、事件、方法

9.1 方法

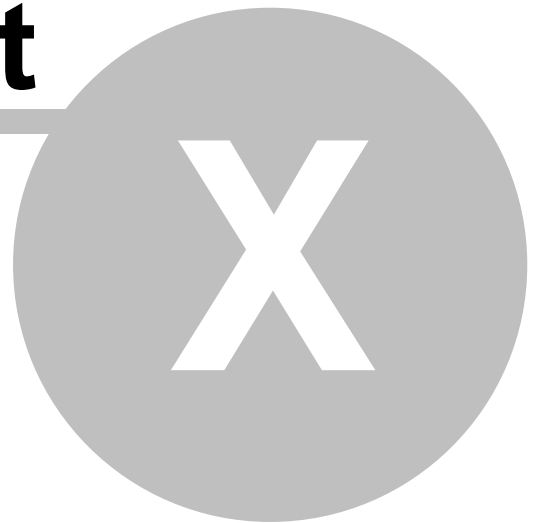
9.1.1 MyPolyLine

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

描述：
MyPolyLine的创建过程

参数：
AOwner: TMyPolyline的拥有者，是TMyCAD的实例

Part



10 MyFreeLine

它是自由线形类,它定义了属性、事件、方法。

10.1 方法

10.1.1 MyFreeLine

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

描述:

MyFreeLine的创建过程

参数:

AOwner: MyFreeLine的拥有者

Part

XI

11 MyPolygon

这是一个贝塞尔曲线图形的类，它定义了属性、事件、方法。

11.1 方法

11.1.1 MyPolygon

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

描述：
MyPolygon创建过程。

参数：
AOwner: MyPolygon的拥有者

Part

XII

12 MyRuleLine

线段图形的类，它定义了属性、事件、方法。

12.1 属性

12.1.1 ShowUserInfo

```
public Boolean ShowUserInfo
```

描述：

当其值为true时，线段显示用户自定义信息，否则显示线段长度信息。

12.1.2 TickStyle

```
public MyTickStyle TickStyle
```

描述：

设置tick类型

12.2 方法

12.2.1 Assign

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

描述：

拷贝一条线段。

参数：

aShape: 一个线段图形

12.2.2 Draw

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

描述：

画标尺图形

参数：

grp: 画布的句柄

12.2.3 LoadFromOldStream

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

描述：

从老的流中加载一条线段图形。

参数：

fileStream: 文件系统

12.2.4 LoadFromStream

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

描述：

从流中加载一条线段。

参数：

fileStream: 文件流

12.2.5 MyRuleLine

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

描述：

创建一个 MyRuleLine. 过程

参数：

AOwner: 该图形的拥有者

12.2.6 SaveToStream

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

描述：

保存到文件流中

参数：

fileStream: 文件流

Part

XIII

13 MyWaveLine

波浪图形的类,它定义了属性、事件、方法。.

13.1 属性

13.1.1 WaveHeight

```
public Byte WaveHeight
```

描述:

设置一条波浪曲线的高度,单位是像素点。

13.1.2 WaveWidth

```
public Byte WaveWidth
```

描述:

设置一条波浪曲线的宽度,单位是像素点。

13.2 方法

13.2.1 Assign

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

描述

复制一条曲线。

参数:

aShape: 源曲线。

13.2.2 Draw

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

描述:

画一个图形

参数:

grp: 画布的句柄

13.2.3 LoadFromStream

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

描述:

从流中装载一个图形。

参数:

fileStream: 文件流

13.2.4 MyWaveLine

```
public MyWaveLine(MCADAOwner)
```

描述:

创建一个 MyWaveLine 过程

参数:

AOwner: 该图形的拥有者

13.2.5 SaveToStream

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

描述:

保存到文件流中

参数:

fileStream: 文件流

Part

XIV

14 MyRectangle

矩形图形的类，它为矩形定义了属性、事件、方法。

14.1 方法

14.1.1 Draw

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

描述：
创建一个矩形图形。

参数：
grp: 画布的句柄

14.1.2 GetCenterPoint

```
public override PointF GetCenterPoint()
```

描述：
得到长方形的中心点

返回值：
返回中心点。

参见：
[GetCenterPointInZoom](#)

14.1.3 GetCenterPointInZoom

```
public override PointF GetCenterPointInZoom()
```

描述：
在缩放时得到长方形的中心点

返回值：
返回缩放时长方形的中心点

参见：
[GetCenterPoint](#)

14.1.4 GetInfo

```
public override String GetInfo(Graphics grp)
```

描述：
得到图形信息。

参数：
grp: 画布的句柄。

返回值：
信息

Part



15 MyEllipse

这是一个椭圆（圆）形的类，它定义了属性、事件、方法。

15.1 方法

15.1.1 Draw

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

描述：

在 MyCanvas 上画椭圆图形。

参数：

grp: 画布的句柄

15.1.2 GetCenterPoint

```
public override PointF GetCenterPoint()
```

描述：

得到椭圆的中心点。

参数：

返回中心点

参见：

[GetCenterPointInZoom](#)

15.1.3 GetCenterPointInZoom

```
public override PointF GetCenterPointInZoom()
```

描述：

按当前缩放比例得到椭圆的中心点。

返回值：

返回缩放时椭圆的中点

参见：

[GetCenterPoint](#)

Part



16 MyLinkPoint

这是一个连接点的类，它定义了属性、事件、方法，它只对库有用

16.1 属性

16.1.1 Size

`public` Byte Size

描述：

得到或设置连接点的大小。

16.2 方法

16.2.1 Draw

`public override void` Draw(Graphics grp)

描述：

画一个连接点图形

参数：

grp: 画布的句柄

16.2.2 LoadFromOldStream

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

描述：

在老的流中装一个连接点图形

参数：

fileStream: 流
encoding: 字符编码

16.2.3 LoadFromStream

`public override void` LoadFromStream(Stream fileStream)

描述：

从流中加载一个连接点。

参数：

fileStream: 文件流

16.2.4 MyLinkPoint

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

描述：

创建一个 MyLinkPoint过程

参数：

AOwner: 拥有者

16.2.5 SaveToStream

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

描述:

保存到文件流中

参数:

fileStream: 文件流

Part

xvi

17 MyLineLinkLine

这是一个连接点的类，它定义了属性、事件、方法

17.1 方法

17.1.1 Draw

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

描述：
创建一个图形

参数：
grp: 画布的句柄

Part



18 MyImage

这是一个图片图形的类，它为一个图片图形定义了属性、事件、方法。

18.1 属性

18.1.1 Bitmap

`public Bitmap Bitmap`

描述：
设置这个图形的位图

18.1.2 Border

`public Boolean Border`

描述：
是否显示图形的边界

18.1.3 OriginSize

`public Boolean OriginSize`

描述：
是否显示图片的真实大小

18.1.4 Transparent

`public Boolean Transparent`

描述：
设置图片是否透明

18.2 方法

18.2.1 Assign

`public override void Assign(MyShape aShape)`

描述：
拷贝一个指定的图片图形。

参数：
aShape: 图片图形。

18.2.2 Dispose

`public virtual void Dispose()`

描述：
先释放自己的字段，再调用继承来（祖先）的dispose过程。

18.2.3 Draw

`public override void Draw(Graphics grp)`

描述：

在 grp 上画一个图片图形

参数:.

grp: 画布的句柄

18.2.4 LoadFromOldStream

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

描述:

从流中加载一个旧的图片图形

参数:

fileStream: 文件流

18.2.5 LoadFromStream

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

描述:

从流中加载一个图片图形

参数:

fileStream: 文件流

18.2.6 MyImage

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

描述:

创建一个 MyImage 过程.

参数:

AOwner: 拥有者

18.2.7 SaveToStream

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

描述:

保存一个图片图形到流中.

参数:

fileStream: 文件流

Part



19 MyRoundRectangle

它定义了圆角长方形的属性，事件，方法

19.1 方法

19.1.1 MyRoundRectangle

```
public MyRoundRectangle(MyCADAOwner)
```

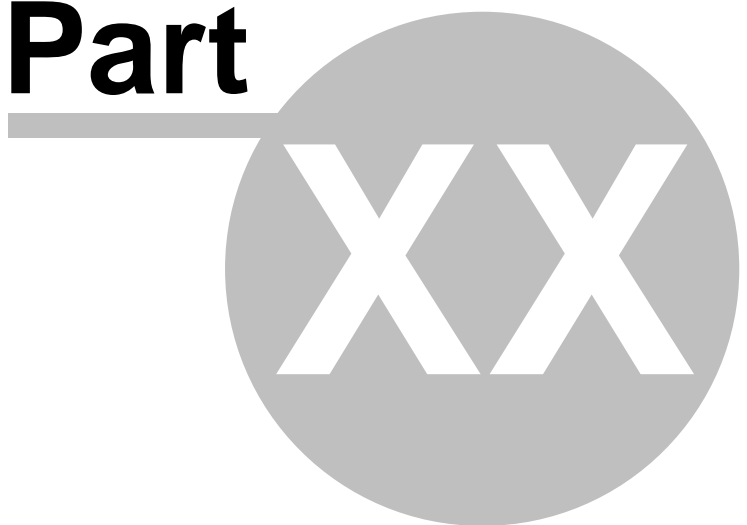
描述：

创建一个 MyRoundRectangle过程。

参数：

AOwner: 拥有者

Part



20 MyText

文本的类，它定义了属性、事件、方法

20.1 属性

20.1.1 Border

`public Boolean Border`

描述：

是否显示图形周围的边框

20.1.2 HAlignment

`public MyAlignment HAlginment`

描述：

决定以外矩形为准，文本的水平对齐方式。

20.1.3 Lines

`public String[] Lines`

描述：

使用Lines在外接矩形内显示多行文本

20.1.4 VAlignment

`public MyVAlignment VAlginment`

描述：

决定以外矩形为准，文本的垂直对齐方式

20.1.5 WordWrap

`public Boolean WordWrap`

描述：

当WordWrap设置为True时，允许多行显示文本，WordWrap为True时，文本长度超出外接矩形时显示超出的文本在新的一行当WordWrap设置为False时，限制文本显示在一行，WordWrap为False时，文本长度超出外接矩形时将显示在矩形外。

20.2 方法

20.2.1 Assign

`public override void Assign(MyShape aShape)`

描述：

拷贝一个文本图形属性

参数：

aShape: 源图形

20.2.2 Dispose

```
public virtual void Dispose()
```

描述：
先释放自己的字段，再调用继承来（祖先）的dispose过程

20.2.3 Draw

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

描述：
在 grp 上画图形

参数：
grp: 画布的句柄

20.2.4 LoadFromOldStream

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

描述：
从流中加载一个旧的图片图形

参数：
fileStream: 文件流

20.2.5 LoadFromStream

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

描述：
从流中加载一个图片图形

参数：
fileStream: 文件流

20.2.6 MyText

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

描述：
创建一个 Mytext 过程

参数：
AOwner: TheMyCAD 拥有者

20.2.7 SaveToStream

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

描述：
保存一个图形到流

参数：
fileStream: 文件流

Part



21 MyElliArc

这是一个圆弧图形的类，它定义了属性、事件、方法

21.1 属性

21.1.1 ArcMode

```
public MyArcMode ArcMode
```

描述：
设置ARC的种类

21.1.2 ArcStyle

```
public MyArcStyle ArcStyle
```

描述：
设置Arc的风格。

21.2 方法

21.2.1 Assign

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

描述：
拷贝一个圆弧图形。
参数：
aShape: 图形的来源

21.2.2 Dispose

```
public virtual void Dispose()
```

描述：
先释放自己的字段，再调用继承来（祖先）的dispose过程。
.

21.2.3 Draw

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

描述：
在grp上画该图形
参数：
grp: 画布的句柄

21.2.4 GetCenterPoint

```
public override PointF GetCenterPoint()
```

描述：
设置 elli arc的中心点
返回值：

返回中心点。

参见：

[GetCenterPointInZoom](#)

21.2.5 GetCenterPointInZoom

```
public override PointF GetCenterPointInZoom()
```

描述：

通过中心点缩放 `elli arc`

返回值：

返回中心点缩

参见：

[GetCenterPoint](#)

21.2.6 LoadFromOldStream

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

描述：

从老的流中加载一个图形

参数：

`fileStream`: 文件流

21.2.7 LoadFromStream

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

描述：

从流中加载一个图片图形

参数：

`fileStream`: 文件流

21.2.8 MyElliArc

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

描述：

创建一个 `MyElliArc`过程。

参数：

`AOwner`: 拥有者

21.2.9 SaveToStream

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

描述：

保存一个图形到流

参数：

`fileStream`: 文件流

Part



22 MyPolyBezier

这是一个贝塞尔曲线图形的类，它定义了属性、事件、方法。

22.1 方法

22.1.1 Draw

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

描述：
在 grp 上画该图形

参数：
grp: 画布的句柄

Part



23 MyUserData

这个类能让你存储一个图形或组合图形自己的数据。

23.1 属性

23.1.1 UserDataRecord

```
public MyUserDataRecord[] UserDataRecord
```

描述：

存储自定义属性

23.2 方法

23.2.1 MyUserData

```
public MyUserData()
```

描述：

初始化内部数据

23.2.2 AddKeyAndValue

```
public Boolean AddKeyAndValue(String newKey,String newValue)
```

描述：

在自定义数据实例中增加一个关键字和值

参数：

newKey: 设置新的密码

newValue: 设置重要的新的密码

返回值：

true：增加成功

false：增加失败

23.2.3 Assign

```
public void Assign(MyUserData source)
```

描述：

拷贝另一个自定义数据集实例。

23.2.4 ChangeValueByKey

```
public Boolean ChangeValueByKey(String originKey,String newValue)
```

描述：

通过关键字来修改其值，如果关键字不存在，返回false，存在返回true。

参数：

originKey: 设置的密码.
newValue: 想要选择的新的密码

返回值:

true: 修改成功
false: 修改失败

23.2.5 ClearAll

```
public Boolean ChangeValueByKey(String originKey,String newValue)
```

描述:

清除所有数据

返回:

true: 清除成功
false: 清除失败

23.2.6 DeleteRecordByKey

```
public Boolean DeleteRecordByKey(String originKey)
```

描述:

通过关键字删除一条记录

参数:

originKey: 您想要删除的关键记录

返回值:

true: 删除成功
false: 删除失败

23.2.7 GetCount

```
public Int32 GetCount()
```

描述:

得到记录总数.

返回:

返回记录

23.2.8 GetKeyByNo

```
public String GetKeyByNo(Int32 n)
```

描述:

通过序号得到关键字, 如果序号不存在, 返回false, 否则返回true.

参数:

n: 序号值

返回:

返回密码

23.2.9 GetValueByKey

```
public String GetValueByKey(String originKey)
```

描述：

通过关键字得到值，如果关键字不存在，返回false，否则返回true。

参数：

originKey: 关键字

返回：

返回关键

23.2.10 InsertKeyAndValue

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

描述：

由被找出的索引插入纪录。

参数：

newKey: 设置新的密码

newValue: 设置新的关键字

index: 其它

before: 纪录插入以后，如果这个参量是错误的，纪录在序号以后将插入

返回：

true:插入成功

false: 插入失败

23.2.11 RenameKey

```
public Boolean RenameKey(String oldKey,String newKey)
```

描述：

重命名一个关键字，如果关键字不存在，返回false，否则返回true

参数：

oldKey: 旧的关键字

newKey: 新的关键字

返回值：

true: 命名成功

false:命名失败

Part



24 关于我们

湖州鸿迪科技发展有限公司自从1998年已经为全球提供优秀的编程控件，深受全球软件开发商的喜爱。我们的目的就是，帮您快速地完成高质量的应用软件的开发，提高您的市场竞争力。感谢您访问我们的网站，因为您是我们的客户，对我们来说非常重要，希望您能打电话或发邮件给我们，咨询我们的产品，我们都将热情地回答您的问题。TCAD能够使您在您的应用软件里加入绘图功能，TCAD已经帮助了世界上许多开发者。

如果您需要与我们联系，您能使用以下方法。

我们这里有客服，请及时与我们沟通，当有要求时打我们的电话，请切记我们上班时间9:00到下午5:00 PM(GMT+8)。

地址：中国浙江省湖州市青铜路699号303室
邮编：313000
电话：+86572-2607144
传真：+86572-2576169
手机：+86 (0) 13511221372
网址：<http://www.codeidea.com>
Email: webmaster@codeidea.com
MSN: webmaster@codeidea.com

矢量图形组件TCAD

<http://www.codeidea.com/cn/>

技术支持

<http://www.codeidea.com/support>

酷得组件仓库

<http://www.cookcode.net>

AVG防木马软件

<http://www.avgchina.com>