TCAD for VCL

v2013

User Manual

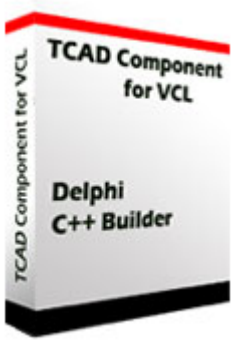
http://www.codeidea.com

Table of Contents

1. Introduction

1.1 What is TCAD

TCAD 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 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 to help you writing application.

Key Features

Shape Types
Line
BLine
RuleLine
Polyline
Polygon
PolyBezier
Rectangle
Arc
Ellipse
Text
Bitmap
User-Define shapes

Grouping/Ungrouping

Multi-Layers

Create/ Move /Rotate shape by code

Save/Load to/From DiskFile/Database

Easy to create user-define shape

Support 4 mode coordinates

Detail information

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.

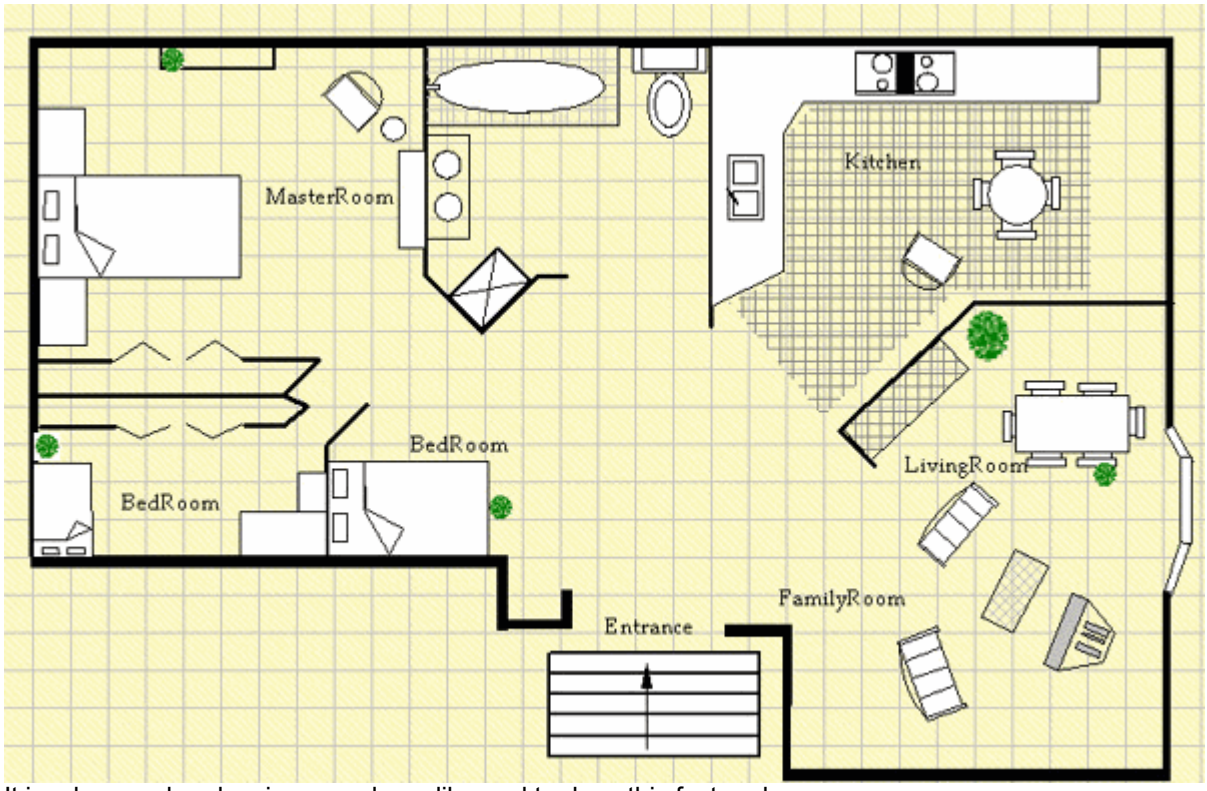
Can Undo and set undo step size
Cutting, copying, pasting and deleting the shapes.
Ordering the shapes(SendToBack, BringToFront, etc.)
Rotating, Dragging and Scaling the shapes by mouse or code.
Aligning the shape in any style.
Easy to create user-define combine shape
Snapping the mouse point to grids,set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

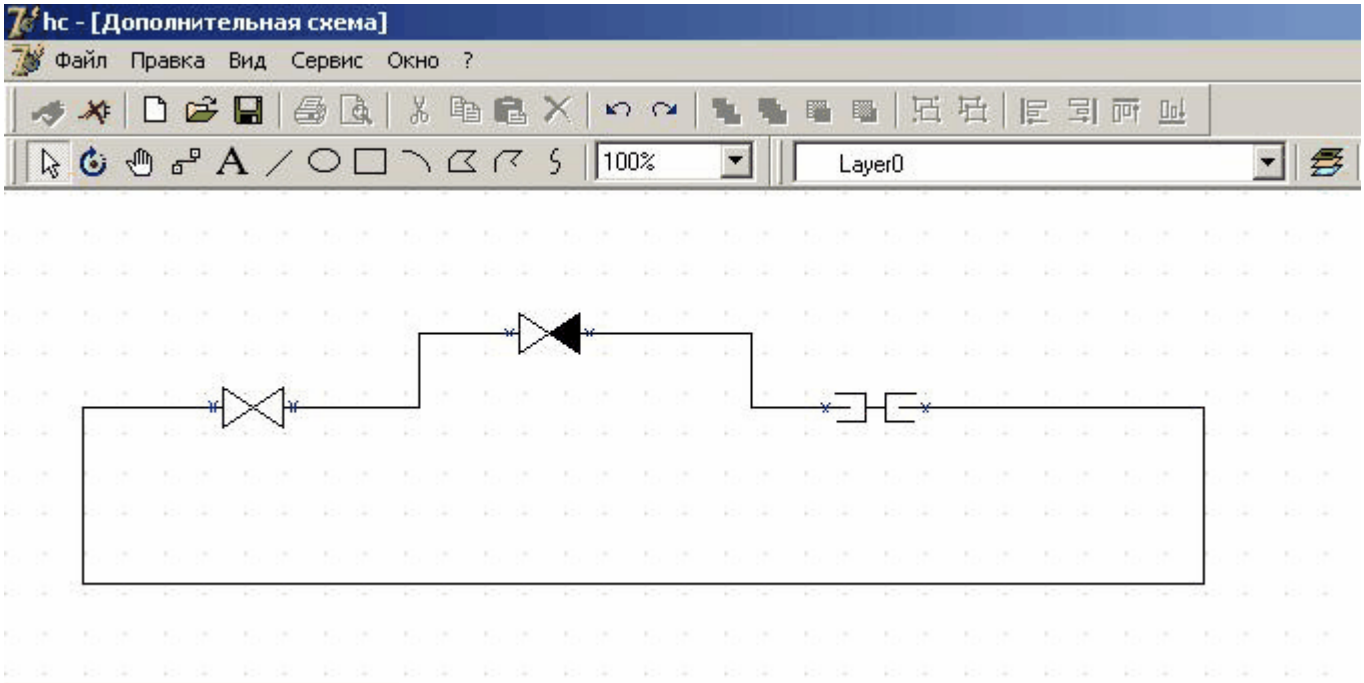
Inserting bitmaps to the drawing.
Scaling, rotating,dragging bitmaps like a shape.
Exporting the drawing as WMF,bitmap,Jpg,dxf(R12) file.

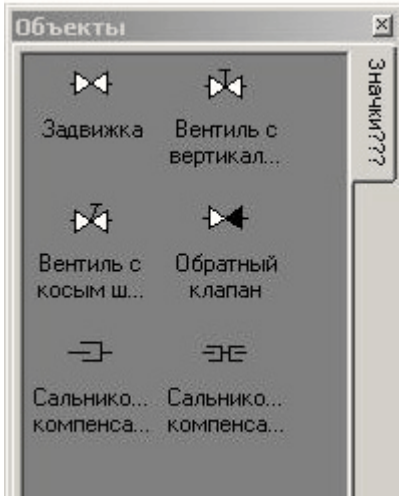
1.2 Application ScreenShot



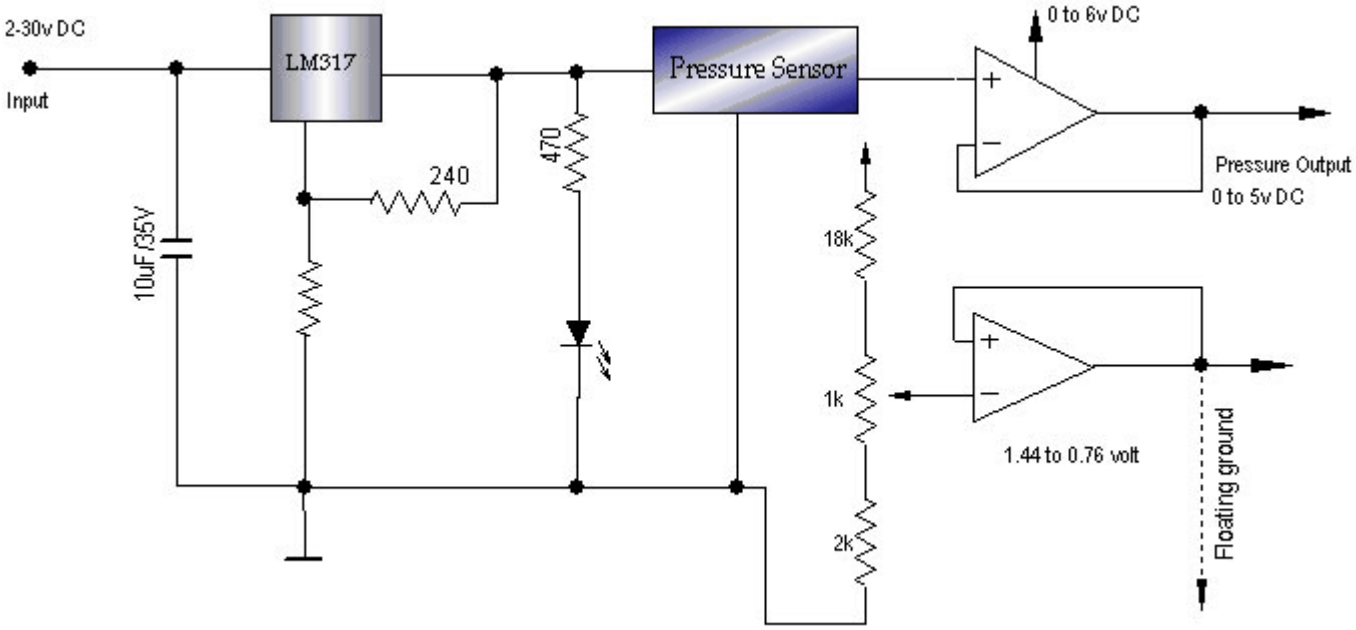
It is a house plan drawing sample,no lib need to draw this,fast and easy.

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



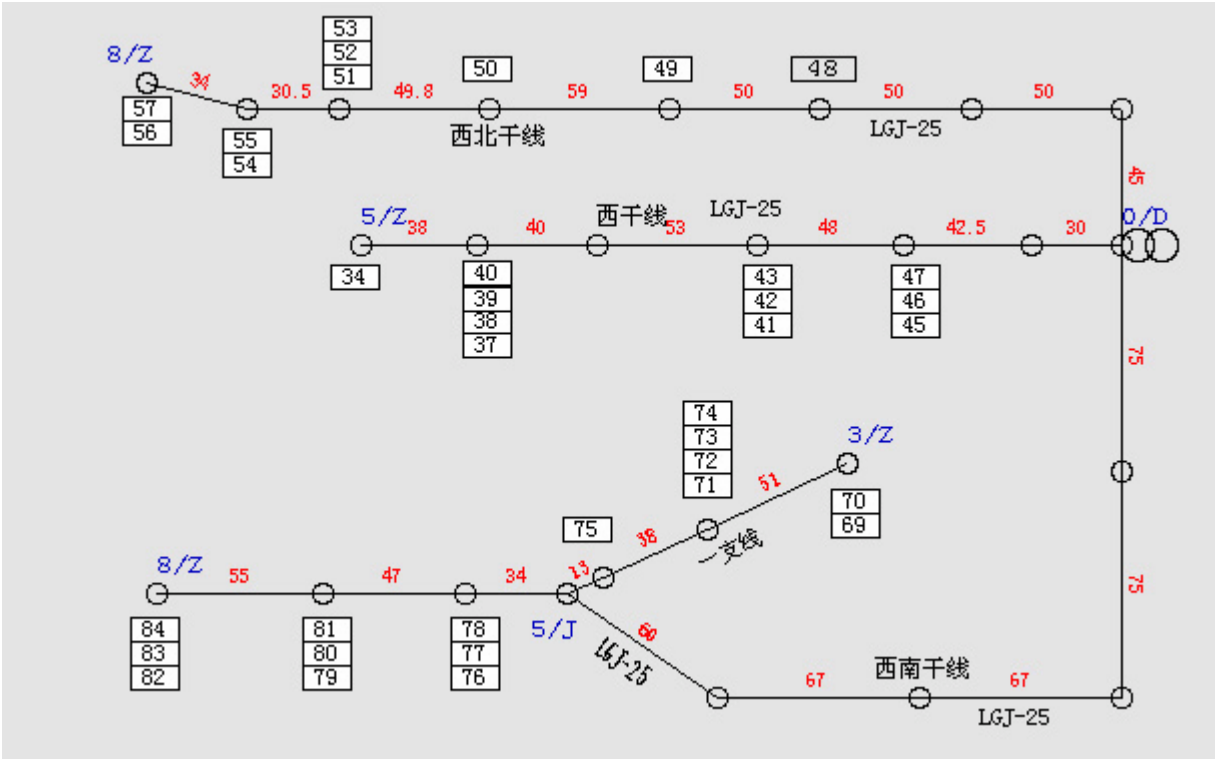


Hydraulic Design

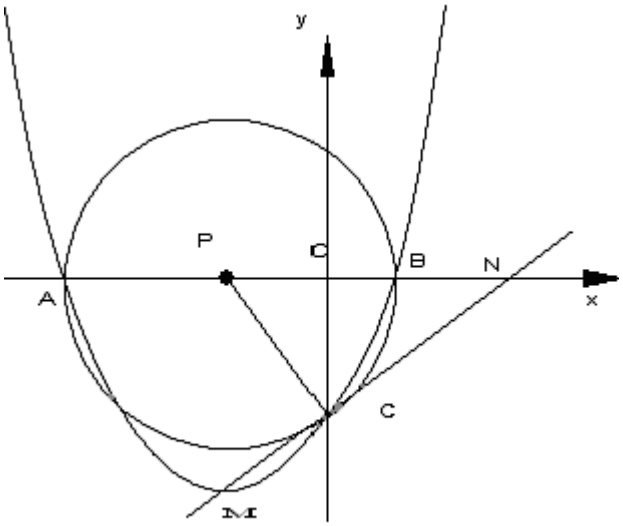


Electric drwaing,library need , and support link line, TCAD is a powerful for drawing.

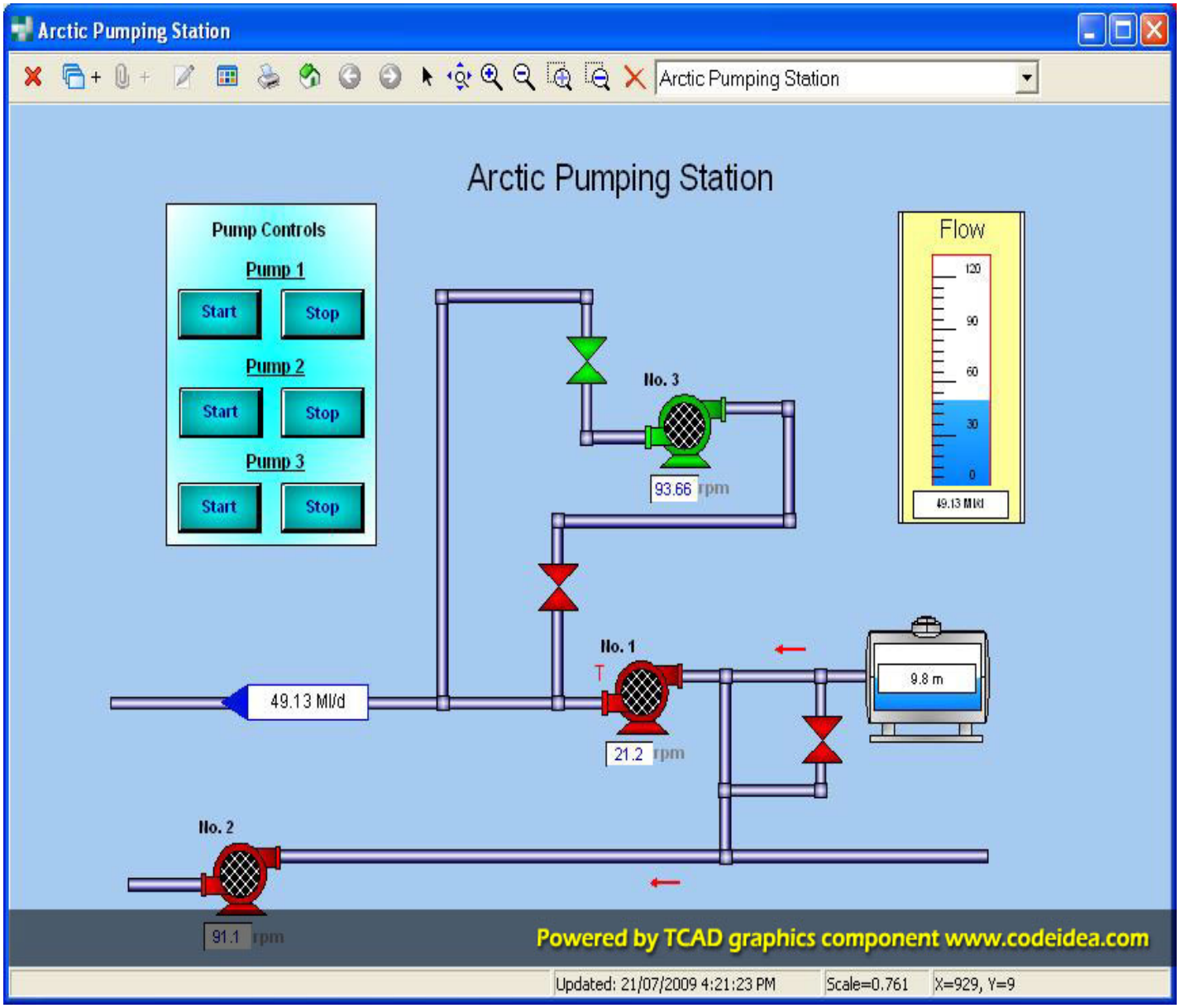
nk line style?

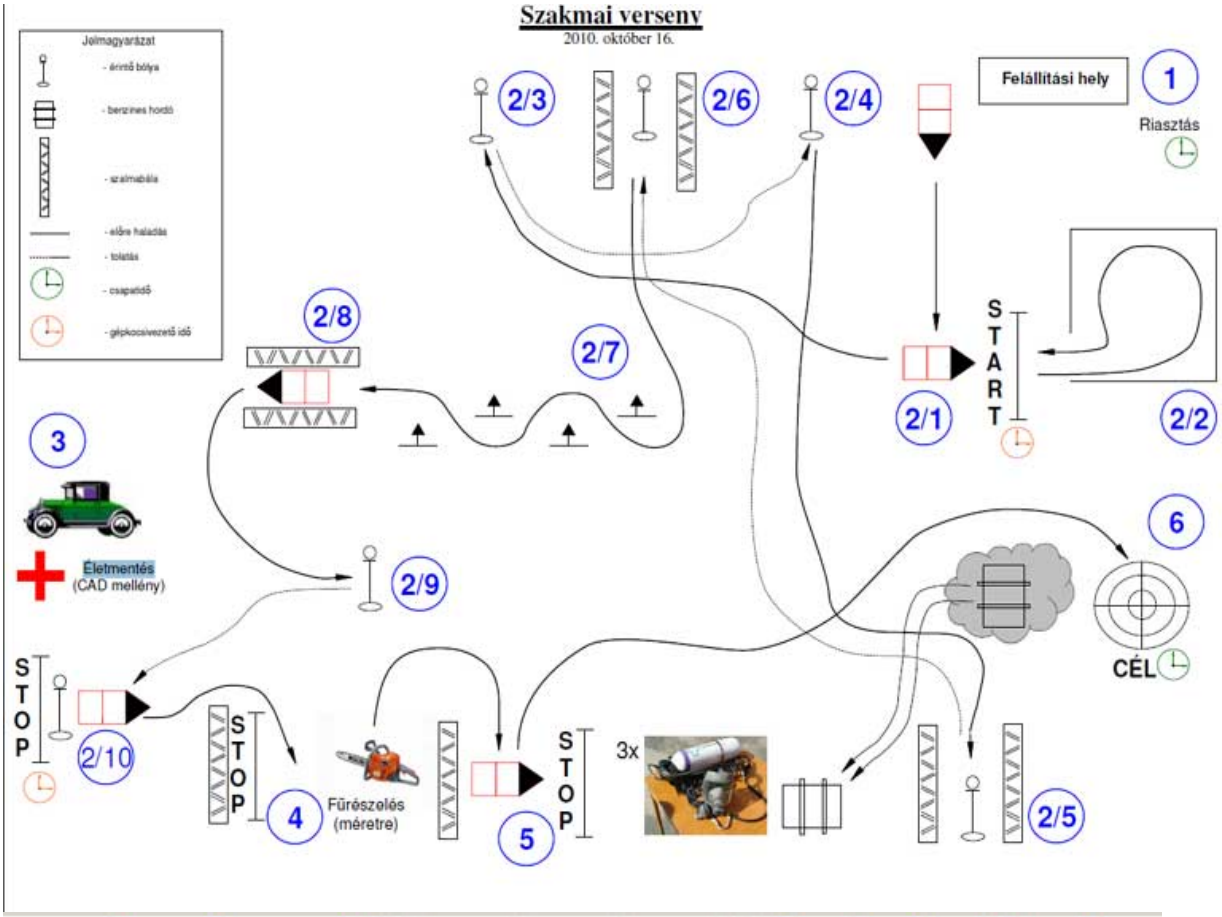


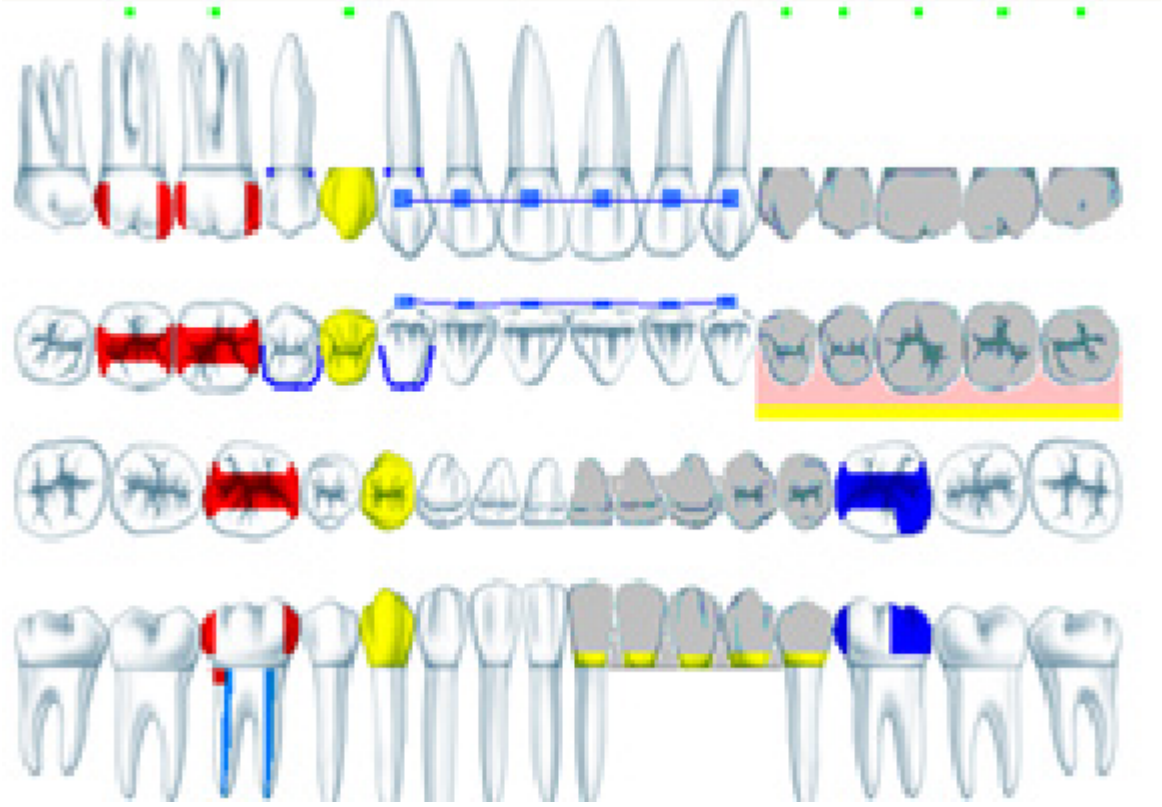
Project:Eletric device manager
Compay:Xi An electric industry college,China

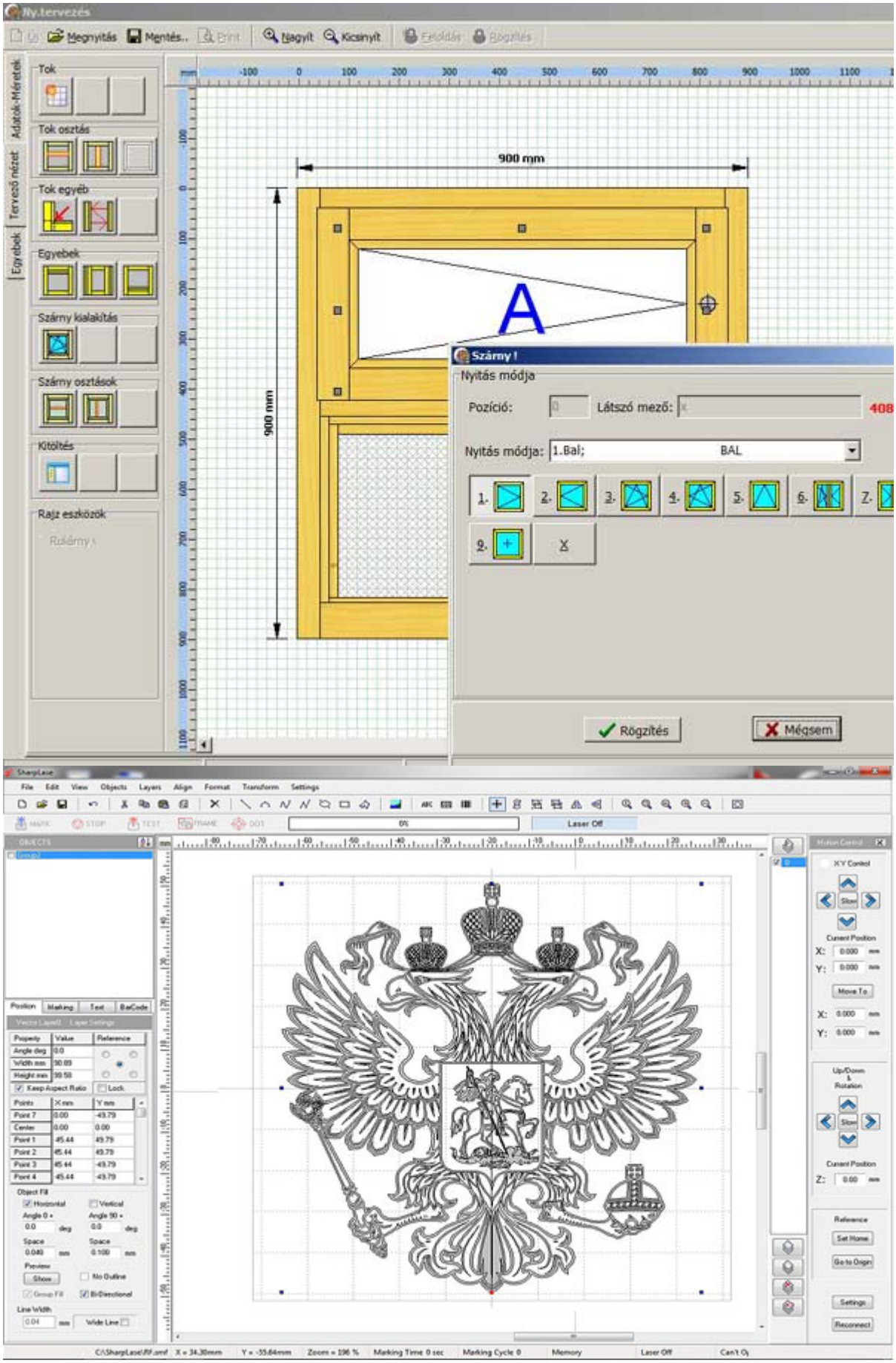


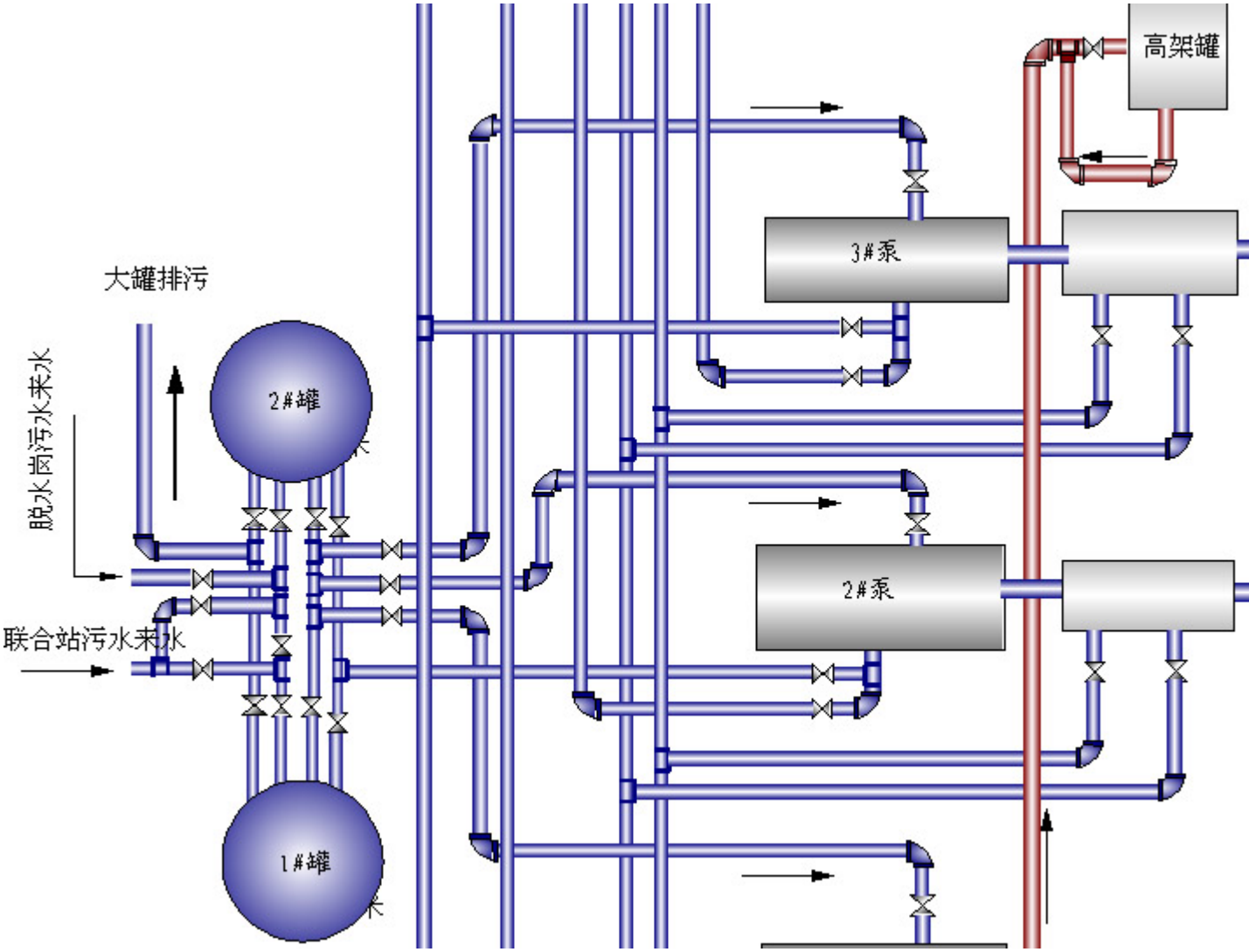
If you are a teacher or student,TCAD can draw math graph,you can copy the graph to MS Word .
Creating math graph library,can speed your job.


Arctic Pumping Station
Arctic Pumping Station
Arctic Pumping Station
Pump Controls
Pump 1
Start
Stop
Pump 2
Start
Stop
Pump 3
Start
Stop
Flow
120
90
60
30
0
No. 3
93.66 rpm
No. 1
21.2 rpm
49.13 Ml/d
9.8 m
No. 2
91.1 rpm
Powered by TCAD graphics component www.codeidea.com
Updated: 21/07/2009 4:21:23 PM
Scale=0.761
X=929, Y=9


Szakmai verseny
2010. október 16.
Jelmagyarázat
Felállítási hely
1
Riasztás
2/1
START
2/2
2/3
2/4
2/5
2/6
2/7
2/8
2/9
2/10
3
Életmentés
(CAD mellény)
4
Fűrészelés
(méretre)
5
6
CÉL
STOP
3x




Ny.tervezés
Megnyitás
Mentés..
Nagyít
Kicsinyít
Tok
Tok osztás
Tok egyéb
Egyebek
Szárny kialakítás
Szárny osztások
Kitöltés
Rajz eszközök
Adatok-Méretek
Tervező nézet
Egyebek
900 mm
900 mm
A
Szárny !
Nyitás módja
Pozíció:
Látszó mező:
Nyitás módja:
1.Bal;
BAL
Rögzítés
Mégsem


高架罐
3#泵
大罐排污
2#罐
脱水岗污水来水
2#泵
联合站污水来水
1#罐

2. Defines

```
 TMyPoint = record
        x:single;
        y:single;
  end;

 TDrawTool =
(SpLinkLine,SpSelecting,SpRotateSelecting,SpLine,SpRuleLine,SpPolyLine,SpRectangle,SpLinkPoint,SpLin
eLinkLine,SpEllipse,
                SpText,SpImage,SpPolygon,SpPolyBezier,SpElliArc,SpClose,SpPolylinePolygonPointRemovi
ng,SpPolylinePolygonointAdding,
                 SpDragWhole);
 TPageStyle = (A0,A1,A2,A3, A4,  A5, B3, B4, B5, CustomerPage);

  TDragMode= (dgHorz,dgVert,dgBoth);

  TXYMode =(Mode0,Mode1,Mode2,Mode3);

  TGridType =(gPixel,gLine,gNone);

  TUnits = (pixel, mm,dm,m, inch);

  TBkBitmapMode = (Tiled,Stretch, Center, LeftTop);

  TArrowStyle = (ANone,ALeft,ARight,ADouble);

  TBlockLayerMode= (Merge,Import);
```

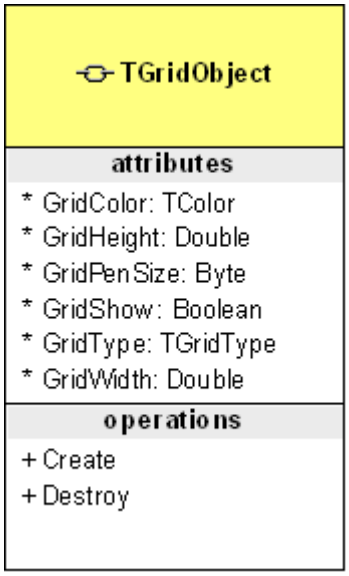
```
TGradientStyle = ( gsRadialC, gsRadialT, gsRadialB, gsRadialL,
  gsRadialR, gsRadialTL, gsRadialTR, gsRadialBL, gsRadialBR, gsLinearH,
  gsLinearV, gsReflectedH, gsReflectedV, gsDiagonalLF, gsDiagonalLB,
  gsDiagonalRF, gsDiagonalRB, gsArrowL, gsArrowR, gsArrowU, gsArrowD,
  gsDiamond, gsButterfly,gsNone);
```

```
 TArrUserDataRecord=array of Record
   Key:string;
   Value:String;
 END;

  TPenStyleJoinType = (psjtROUND,psjtBEVEL,psjtMITER,psjtMASK);

  TPenStyleEndCapType = (psectROUND,psectSQUARE,psectFLAT,psectMASK);
```

3. TGridObject



3.1 GridColor

property GridColor:TColor;

Description:

Set the grid color .

Example:

```
MyCAD1.GridOperation.GridColor:=clBlack;
```

See also:
GridWidth
GridHeight
GirdType
GridPenSize

3.2 GridHeight

property GridHeight:byte;

Description:

The height of grid , it is in pixels unit.

Example:

```
MyCAD1.GridOperation.GridHeight:=12;
```

See also:
GridWidth
GirdType
GridShow

3.3 GridInZoom

property GridInZoom:Boolean;

Description:

The grid size will be related with zooming when you set this value as True.

Example:

```
MyCAD1.GridOperation.GridInZoom := false;
```

3.4 GridPenSize

property GridPenSize:Byte;

Description:

The grid line or pixel size

Example:

```
MyCAD1.GridOperation.GridPenSize := 3;
```

See also:
GridWidth
GridHeight
GirdType
GridColor

3.5 GridShow

property GridShow:Boolean;

Description:

The grid show or not.

Example:

```
MyCAD1.GridOperation.GridShow := false;
```

See also:
GridWidth
GridHeight
GirdType

3.6 GridType

property GridType:**TGridType**

Description:

The grid is made of pixel or line .

Example:

```
MyCAD1.GridOperation.GridType:=gLine;
```

See also:

GridWidth
GridHeight
GridShow

3.7 GridWidth

property GridWidth:byte;

Description:

The width of grid , the unit is in pixels

Example:

```
MyCAD1.GridOperation.GridWidth:=12;
```

See also:

GridHeight
GirdType
GridShow

4. TMyCAD

TMyCAD is a powerful 2D drawing component, it is can be used at delphi and c++ Builder.

4.1 ClassDiagram

4.1.1 Part1 Properties

4.1.2 Part2 Events

4.1.3 Part3 Methods

4.2 Events

4.2.1 OnActionToolToSelecting

TActionToolToSelecting = procedure() of object;

property OnActionToolToSelecting: TActionToolToSelecting

Description:

Use OnActionToolToSelecting to handle when the ShapeTool return to SpSelecting state,

Example:

```
procedure TMainFrm.MyCAD1ActionToolToSelecting;
begin
  SelectBtn.Down := true;
end;
```

4.2.2 OnChildShapeSelected

TChildhapeSelected = procedure (SelectedChildShape: TMyShape; var AcceptSelect:boolean) of object;
property OnChildShapeSelected: TChildhapeSelected

Description:
Use OnChildShapeSelected to handle when you click the combine shape.

Example:

```
procedure TMainFrm.MyCAD1ChildShapeSelected(SelectedChildShape: TMyShape; var
AcceptSelect: Boolean);
begin
                AcceptSelect:=false;
end;
```

4.2.3 OnClick

Please read Delphi or C++ Builder Help file.

4.2.4 OnDblClick

Please read Delphi or C++ Builder Help file.

4.2.5 OnDeleteLayer

TLayerOp = procedure(LayerId: integer; LayerName: string; Visible: Boolean) of object;

property OnDeleteLayer: TLayerop

Parameters:

LayerID
The id of this layer.

LayerName
The name of this layer.

Visible
This layer is visible or not.

Description:

Use OnDeleteLayer to handler a layer be deleted.

4.2.6 OnDragDrop

property OnDragDrop: TDragDropEvent;

Delphi syntax:

type TDragDropEvent = procedure(Sender, Source: TObject; X, Y: Integer) of object;
property OnDragDrop: TDragDropEvent;

C++ syntax:

typedef void __fastcall (__closure *TDragDropEvent)(System::TObject* Sender, System::TObject* Source, int X, int Y);
__property TDragDropEvent OnDragDrop = {read=FOnDragDrop, write=FOnDragDrop};

Description:

Occurs when the user drops an object being dragged.

Use the OnDragDrop event handler to specify what happens when the user drops an object. The Source parameter of the OnDragDrop event is the object being dropped, and the Sender is the control the object is being dropped on. The X and Y parameters are the coordinates of the mouse positioned over the control.

4.2.7 OnDragOver

Occurs when the user drags an object over a control.

Delphi syntax:

type
  TDragOverEvent = procedure(Sender, Source: TObject; X, Y: Integer; State: TDragState; var Accept: Boolean) of object;
property OnDragOver: TDragOverEvent;

C++ syntax:

typedef void __fastcall (__closure *TDragOverEvent)(System::TObject* Sender, System::TObject* Source, int X, int Y, TDragState State, bool &Accept);
__property TDragOverEvent OnDragOver = {read=FOnDragOver, write=FOnDragOver};

Description

Use an OnDragOver event to signal that the control can accept a dragged object so the user can drop or dock it.

Within the OnDragOver event handler, change the Accept parameter to false to reject the dragged object. Leave Accept as true to allow the user to drop or dock the dragged object on the control.

To change the shape of the cursor, indicating that the control can accept the dragged object, change the

value of the DragCursor property for the control before the OnDragOver event occurs.

The Source is the object being dragged, the Sender is the potential drop or dock site, and X and Y are screen coordinates in pixels. The State parameter specifies how the dragged object is moving over the control.

Note: Within the OnDragOver event handler, the Accept parameter defaults to true. However, if an OnDragOver event handler is not supplied, the control rejects the dragged object, as if the Accept parameter were changed to false.

4.2.8 OnWholeDragged

TWholeDragOperation = procedure of object;

property OnWholeDragged: TWholeDragOperation

Description:

Use OnWholeDragged to handler a operation after drag all shapes.

4.2.9 OnMouseDown

Please read Delphi or C++ Builder Help file.

4.2.10 OnMouseEnter

property OnMouseEnter:TNotifyEvent

Description:

When the mouse enter TMyCAD, this event trigger.

Example:

```
procedure TForm1.MyCAD1MouseEnter(Sender: TObject);
begin
 Memo1.Lines.Add('---------Enter TCAD Event---------------');
end;
```

See Also:

OnMouseLeave

4.2.11 OnMouseEnterShape

TEnterLeaveShape = procedure (AShape:TMyShape) of object;

property OnMouseEnterShape:TNotifyEvent

Description:

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

Example:

```
procedure TForm1.MyCAD1MouseEnterShape(AShape: TMyShape);
begin
  Memo1.Lines.Add('---------Enter Shape Event---------------');
  Memo1.Lines.Add('Enter shape, id is : ' + inttostr(AShape.ShapeID));
end;
```

See Also:

OnMouseLeaveShape

4.2.12 OnMouseLeave

property OnMouseLeave:TNotifyEvent

Description:

When the mouse leave TMyCAD, this event trigger.

Example:

```
procedure TForm1.MyCAD1MouseLeave(Sender: TObject);
begin
 Memo1.Lines.Add('---------Leave TCAD Event---------------');
end;
```

See Also:

OnMouseEnter

4.2.13 OnMouseLeaveShape

TEnterLeaveShape = procedure (AShape:TMyShape) of object;

property OnMouseLeaveShape:TEnterLeaveShape

Description:

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

Example:

```
procedure TForm1.MyCAD1MouseLeaveShape(ShapeID, LayId: Integer;
  AShape: TMyShape);
begin
   Memo1.Lines.Add('--------- Leave Shape Event---------------');
   Memo1.Lines.Add('Leave shape, id is : ' + inttostr(AShape.ShapeID));
end;
```

See also:

OnMouseEnterShape

4.2.14 OnMouseMove

Please read Delphi or C++ Builder Help file.

4.2.15 OnMouseUp

Please read Delphi or C++ Builder Help file.

4.2.16 OnNewLayer

TLayerOp = procedure(LayerId: integer; LayerName: string; Visible: Boolean) of object;

property OnNewLayer: TLayerOp ;

Description:

Use OnNewLayer to handle a new layer be added.

4.2.17 OnPaint

Please read Delphi or C++ Builder Help file.

4.2.18 OnShapeAdded

TShapeAdded = procedure(LayerId: integer; ShapeID: integer;ShapeName:string; AShape: TMyShape; ShapeCount: integer) of object;

property OnShapeAdded: TShapeAdded;

Description:

Use OnShapeAdded to handler the event of shape be added .

Examples:

```
procedure TMainFrm.MyCAD1ShapeAdded( var AShape: TMyShape; ShapeCount: Integer);
begi n
 EventMemo.Lines.Add('---------Add Event');
  EventMemo.Lines.Add('You Add a Shape,it is :' + ShapeName + ',in Layer ' +
    inttostr(LayerId) + ', it is  a ' + AShape.ClassName + ',Now CAD has ' +
    inttostr(ShapeCount) + 'Shapes');
end;
```

4.2.19 OnShapeCodeDragging

TShapeCodeDraggingSizing = procedure (AShape:TMyShape; dx,dy:Single) of object;

property OnShapeCodeDragging

Description:

When a Shape is Dragging by code , this event be trigger.

Example:

```
procedure TForm1.MyCAD1ShapeCodeRotating(AShape: TMyShape;dx,dy: single);
begin
  Memo1.Lines.Add('---------code Drag Event---------------');
  Memo1.Lines.Add('ShapeId is: '+Inttostr(AShape.ShapeId)+' '+AShape.Name +' rotating ');
end;
```

See Also:

Move

4.2.20 OnShapeCodeRotating

TShapeCodeRotating = procedure (AShape:TMyShape; AAngle:Single) of object;

property OnShapeCodeRotating

Description:

When a Shape is Rotating by code , this event will be trigger.

Example:

```
procedure TForm1.MyCAD1ShapeCodeDragging( AShape:TMyShape;AAngle:single);
begin
  Memo1.Lines.Add('---------code Rotate Event---------------');
  Memo1.Lines.Add('ShapeId is: '+Inttostr(AShape.ShapeId)+' '+AShape.Name +' rotating ');
end;
```

See Also:

Rotate

4.2.21 OnShapeDeleted

TShapeDeleted = procedure (ShapeID:integer;LayerId:integer;AShape:TMyShape) of object ;

property OnShapeDeleted: TShapeDelected ;

Description:

Use OnShapeDelected to handle the event of shape be delected.

4.2.22 OnShapeMouseDragged

TShapeDraggingSizingRotated = procedure (AShape:TMyShape;FromPoint:TPoint;var ToPoint:TPoint) of object;

property OnShapeMouseDragged:TShapeDraggingSizingRotating

Description:

When a Shape is dragging by mouse , this event be trigger.

Example:

```
procedure TMainFrm.MyCAD1ShapeMouseDragged(AShape: TMyShape; FromPoint:TPoint; var
ToPoint: TPoint);
begin
Memo1.Lines.Add('---------Dragged Event---------------');
  Memo1.Lines.Add('ShapeId is: '+Inttostr(AShape.ShapeId)+AShape.Name +' dragged ,'+
'From x:'+ inttostr(FromPoint.x)+' y:' +inttostr(FromPoint.y)+
'To x:'+ inttostr(ToPoint.x)+' y:' +inttostr(ToPoint.y));
end;
```

See Also:

Move

4.2.23 OnShapeMouseDragging

TShapeDraggingSizingRotated = procedure (AShape:TMyShape;FromPoint:TPoint;var ToPoint:TPoint) of object;

property OnShapeMouseDragging:TShapeDraggingSizingRotating

Description:

When a Shape is dragging by mouse , this event be trigger.

Example:

```
procedure TForm1.MyCAD1ShapeMouseDragging( AShape: TMyShape; FromPoint:TPoint; var
ToPoint: TPoint);
begin
  Memo1.Lines.Add('---------Dragged Event---------------');
  Memo1.Lines.Add('ShapeId is: '+Inttostr(AShape.ShapeId)+AShape.Name +' are dragging!'
end;

end;
```

See Also:

Move

4.2.24 OnShapeMouseResized

TShapeDraggingSizingRotating= procedure (AShape:TMyShape;FromPoint:TPoint;var ToPoint:TPoint) of object;
property OnShapeMouseResizing:TShapeDraggingSizingRotating

Description:

When a shape is resizing by mouse, this event be trigger.

Example:

```
procedure TForm1.MyCAD1ShapeMouseResized(AShape: TMyShape; FromPoint:TPoint;var ToPoint:
TPoint);
begin
  Memo1.Lines.Add('---------Resized Event---------------');
  Memo1.Lines.Add('ShapeId is: '+Inttostr(AShape.ShapeId)+AShape.Name +' resized,'+
'From x:'+ inttostr(FromPoint.x)+' y:' +inttostr(FromPoint.y)+
' To x:' + inttostr(ToPoint.x)+' y:' +inttostr(ToPoint.y));
end;
```

4.2.25 OnShapeMouseResizing

TShapeDraggingSizingRotating = procedure (AShape:TMyShape;FromPoint:TPoint;ToPoint:TPoint) of object;

property OnShapeMouseResizing:TShapeDraggingSizingRotating

Description:

When a shape is resizing by mouse, this event be trigger.

Example:

```
procedure TForm1.MyCAD1ShapeMouseResizing(AShape: TMyShape; FromPoint:TPoint;var ToPoint:
TPoint);
begin
  Memo1.Lines.Add('---------Resizing Event---------------');
  Memo1.Lines.Add('ShapeId is: '+Inttostr(AShape.ShapeId)+AShape.Name +' resizing ,'+
'From x:'+ inttostr(FromPoint.x)+' y:' +inttostr(FromPoint.y)+
'To x:'+ inttostr(ToPoint.x)+' y:' +inttostr(ToPoint.y));
end;
```

4.2.26 OnShapeMouseRotated

TShapeDraggingSizingRotating= procedure (AShape:TMyShape;FromPoint:TPoint;ToPoint:TPoint) of object;

property OnShapeMouseRotated:TShapeDraggingSizingRotating

Description:

When a Shape is rotated by mouse, this event be trigger.

Example:

```
procedure TForm1.MyCAD1ShapeMouseRotated(AShape: TMyShape; FromPoint:TPoint,var ToPoint:
TPoint);
begin
  Memo1.Lines.Add('---------Rotated Event---------------');
  Memo1.Lines.Add('ShapeId is: '+Inttostr(AShape.ShapeId)+AShape.Name +' rotated ');
end;
```

See Also:

Rotate

4.2.27 OnShapeMouseRotating

TShapeDraggingSizingRotating= procedure (AShape:TMyShape;FromPoint:TPoint;ToPoint:TPoint) of object;

property OnShapeMouseRotating:TShapeDraggingSizingRotating

Description:

When a shape is rotating by mouse, this event be trigger.

Example:

```
procedure TForm1.MyCAD1ShapeMouseRotating(AShape: TMyShape; FromPoint:TPoint;var ToPoint:
TPoint);
begin
  Memo1.Lines.Add('---------Rotating Event---------------');
  Memo1.Lines.Add('ShapeId is: '+Inttostr(AShape.ShapeId)+AShape.Name +' rotating ');
end;
```

See Also:

Rotate

4.2.28 OnShapeSelected

TShapeSelected = procedure (LastSelectedShape:TMyShape; SelectedShape: TMyShape) of object;

property OnShapeSelected: TShapeSelected ;

Description:

Use OnShapeSelected to handle a shape is selected and also can get the old shape 's infomation.

Examples:

```
procedure TMainFrm.MyCAD1ShapeSelected(LastSelectedShape, SelectedShape: TMyShape);
begin
  Memo1.Lines.Add('---------Select Event---------------');
if LastSelectedShape = nil then
     Memo1.Lines.Add('No last shape selected')
else
     Memo1.Lines.Add(' Last shape id is ' + inttostr(LastSelectedShape.ShapeID)+
',in Layer ' + inttostr(LastSelectedShape.LayerId) +
' , it is  a ' + LastSelectedShape.ClassName);

     Memo1.Lines.Add('Now you selected ' + inttostr(SelectedShape.ShapeID)+
',in Layer ' + inttostr(LayerId) +
', it is  a ' + SelectedShape.ClassName);
end;
```

4.3 Methods

function AddImageShapeByCode(TheName:String;LeftTop:TPoint;ABitmap:TBitmap):integer;

Description:

Add image shape by code, if you want add other shape , Please use procedure **AddShapeByCode** or **AddCmbShapeByCode**

Return Value:
-1: Add failed
else(>=0): The new shape's shapeId.

Parameter
TheName:Image shape's name
LeftTop:the point of the image's Lfte-Top corner.
ABitmap:a image that you want add into TCAD

Example:

```
var
    myBitmap:TBitmap;
begin
{Create a temp bitmap to load form a disk file}
   MyBitmap := TBitmap.Create;
{Load from files}
   Mybitmap.LoadFromFile( ExtractFilePath(Application.ExeName) +'images\1'+'.bmp');

{a layer0 already exist by default}
{Add Image into TMyCAD by code}
    MyCAD1.AddImageShapeByCode('MyImage',MyPoint(50,100),MyBitmap);
{free it because no use}
    MyBitmap.Free;
end;
```

See also:

AddShapeByCode
AddUserDefineShapefromLib
OnShapeAdded

4.3.1 AddImageShapeByCode

function AddImageShapeByCode(TheName:String;LeftTop:TPoint;ABitmap:TBitmap):integer;

Description:

Add image shape by code, if you want add other shape , Please use procedure **AddShapeByCode** or **AddCmbShapeByCode**

Return Value:
-1: Add failed
else(>=0): The new shape's shapeId.

Parameter
TheName:Image shape's name
LeftTop:the point of the image's Lfte-Top corner.
ABitmap:a image that you want add into TCAD

Example:

```
var
    myBitmap:TBitmap;
begin
{Create a temp bitmap to load form a disk file}
   MyBitmap := TBitmap.Create;
{Load from files}
   Mybitmap.LoadFromFile( ExtractFilePath(Application.ExeName) +'images\1'+'.bmp');

{a layer0 already exist by default}
{Add Image into TMyCAD by code}
    MyCAD1.AddImageShapeByCode('MyImage',MyPoint(50,100),MyBitmap);
{free it because no use}
    MyBitmap.Free;
end;
```

See also:

AddShapeByCode
AddUserDefineShapefromLib
OnShapeAdded

4.3.2 AddShapeByCode

function AddShapeByCode(Owner:TComponent;ShapeStyle:TDrawTool; ShapeName:string;ThePoints: array of TMyPoint;TheAngle:Extended=0;OnlyForText:String=''): integer;

Description:

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

Return Value:
0: Ok
-1: incorrect points count

Parameter:
Owner:instance of TMyCAD
ShapeStyle : Please see TDrawTool
ShapeName: shape's name
ThePoints: Points array
TheAngle: The Angle of this new shape
OnlyForText: it is for TMyText shape only

Example:

```
  MyCAD1.AddShapeByCode(MyCAD1,spEllipse, 'EllipseShape',[MyPoint(231, 211), MyPoint
(340,211), MyPoi nt ( 340, 350),MyPoint(231,350)]);
end;
```

See also:
AddImageShapeByCode
AddCmbShapeByCode
OnShapeAdded

4.3.3 AddBlockfromTCADFile

function AddBlockfromTCADFile(const FileName:string; BlockLayerMode:TBlockLayerMode):Boolean;

Description:

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

Return Value:
true : add success
false : add failed

Parameter
TheName:Image shape's name
LeftTop:the point of the image's Lfte-Top corner.
ABitmap:a image that you want add into TCAD

See also:

Defines

4.3.4 AddUserDefineShapefromLib

function AddUserDefineShapefromLib(ALibManager:TLibManager;
UDShapeName:string;CenterX,CenterY:integer;OwnerCAD:TMyCAD): Integer;

Description:

Get from library by UDshapename . and add user-defined shape into TMyCAD instance,it is a important function in TMyCAD

Return Value:

-1: Add failed
else(>=0): The new shape's shapeId.

Parameter:

ALibManager:the instance of a library
UDShapeName : the name of a combine shape;
CenterX,CenterY:the shape's center position that you want placed;
OwnerCAD:this shape will be added .

Example:

```
var
     mLibManager:TLibManager;
begin
if (FileExists(FileName)) and (IsTCADLibFile(FileName)) then
begin
        mLibManager:=TLibManager.Create;
```

```
        mLibManager.LoadFromFile(LibFileName);
        MyCAD1.AddUserDefineShapefromLib( mLibManager, 'MyShapename',100,100,MyCAD1);
        mLibManager.Free;
end;
end;
```

See also:

AddShapeByCode
AddImageShapeByCode
OnShapeAdded

4.3.5 AlignBottom

procedure AlignBottom;

Description:

When you select more than one shapes, this procedure can align them to bottom.

Example:

```
MyCAD1.Alignbottom;
```

4.3.6 AlignHorizontalCenter

procedure AlignHorizontalCenter;

Description:

When you select more than one shapes, this procedure can align them horizontal center.

Example:

```
MyCAD1.AlignHorizontalCenter;
```

4.3.7 AlignLeft

procedure AlignLeft;

Description:

When you select more than one shapes, this procedure can align them left.

Example:

```
MyCAD1.AlignLeft;
```

4.3.8 AlignRight

procedure AlignRight;

Description:

When you select more than one shapes, this procedure can align them right.

Example:

```
MyCAD1.AlignRight;
```

4.3.9 AlignTop

procedure AlignTop;

Description:

When you select more than one shapes, this procedure can align them to top.

Example:

```
MyCAD1.AlignTop;
```

4.3.10 AlignVerticalCenter

procedure AlignVerticalCenter;

Description:

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

Example:

```
MyCAD1.AlignVerticalCenter;
```

4.3.11 AverageHeight

procedure AverageHeight(shapeArray: TArrMyShape);

Description:

Average height in shapes, more than two shapes.

Example:

```
MyCAD1.AverageHeight(MyCAD1.WorkingShapes);
```

See Also:

AverageWidth

4.3.12 AverageWidth

procedure AverageWidth(shapeArray: TArrMyShape);

Description:

Average width in shapes, more than two shapes.

Example:

```
MyCAD1.AverageWidth(MyCAD1.WorkingShapes);
```

See Also:

AverageHeight

4.3.13 BringToFront

procedure BringToFront(AShape:TMyShape;NeedSave:boolean=true);

Description:

Sent the selected shape to front

Parameter:

AShape:the shape that you will changed;
NeedSave:used by internal.

Example:

```
MyCAD1.BringToFront(MyCAD1.GetSelectedShape);
```

See Also:

SendToBack
BringToFrontByStep

4.3.14 BringToFrontByStep

procedure BringToFrontByStep(AShape:TMyShape;NeedSave:boolean=true)

Description:

Sent the selected shape to front by a step

Parameter:

AShape:the shape that you will changed;
NeedSave:used by internal.

Example:

```
MyCAD1.BringToFront(MyCAD1.GetSelectedShape);
```

See Also:

SendToBack
BringToFront

4.3.15 ClearAllUndoStuff

procedure ClearAllUndoStuff

Description:

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

Example:

```
MyCAD1.ClearAllUndoStuff;
```

See also:

UndoRedoSize

4.3.16 ClosePolygon

procedure ClosePolygon

Description:

Finish TMyPolygon drawing.

Example:

```
MyCAD1.ClosePolygon;
```

4.3.17 Copy

procedure Copy();

Description:

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

Example:

```
   MyCAD1.SelectShapeBycode(0);
//Select other shape
   MyCAD1.SelectShapeBycode(1,false);
   MyCAD1.Copy;
   MyCAD1.Paste;
```

See also:

Paste
Cut

4.3.18 CopyToClipBoardAsWmf

procedure CopyToClipBoardAsWmf

Description:

Save selected shape be saved in wmf format in clipboard.

Example:

```
   MyCAD1.SelectShapeBycode(0);
//Select other shape
   MyCAD1.SelectShapeBycode(1,false);
   MyCAD1.CopyToClipBoardAsWmf;
```

See also:

Paste
Cut
Copy

4.3.19 Create

constructor Create(AOwner: TComponent);

Description:

Create TMyCAD; There is a layer be created,named "Layer0".

Example:

```
var
  MyCAD:TMyCAD;
begin
 MyCAD:=TMyCAD.Create(Form1);
 MyCAD.Parent:=Form1;
end;
```

See also:

Destroy

4.3.20 CreateLink

function CreateLink(ALinkShapeName:string;SrcShape:TMyShape; SrcLinkPtId:integer; DestShape:TMyShape;DestLinkPtId:integer): Integer;

Description:

To create linkline shape between SrcShape and DestShape.

Parameter:

ALinkShapeName : The linkline shape's shapename;
SrcShape: The source shape,it is a userdefine shape, and has linkpoint in it;
SrcLinkPtId: The link point id.
DestShape: The destination shape,it is a userdefine shape, and has linkpoint in it;
DestLinkPtId: The link point id.

Return value:
-1: add failed
else(>=0) : add success, it is the LinkLinkshape's ShapeId.

See Also

DeleteAllShapes

4.3.21 Cut

procedure Cut();

Description:

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

Example:

```
MyCAD1.SelectShapeBycode(0);
MyCAD1.Cut;
MyCAD1.Paste;
```

See also:

Copy
Paste

4.3.22 DeleteAllLayers

procedure DeleteAllLayers(): **Boolean**;

Description:

Delete all layers and delete all shapes, CurrentLayerId is -1.

Return value:

true: the all layers was delected
false: deleted failed

See Also

DeleteAllShapes

4.3.23 DeleteAllShapes

procedure DeleteAllShapes;

Description:

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

See Also:

DeleteAllLayers

4.3.24 DeleteLayerByID

function DeleteLayerByID(ALayerID: integer): **Boolean**;

Description:

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

See also:

DeleteLayerByName

4.3.25 DeleteLayerByName

function DeleteLayerByName(ALayerName: string): **Boolean**;

Description:

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

See also:

DeleteLayerByID

4.3.26 DeleteSelectedShape

function DeleteSelectedShape:boolean;

Description:

Delete current selected shape(s).

Return Value:
true : delete success
false : delete failed

4.3.27 DeleteShapeByID

function DeleteShapeByID(ShapeID:Integer):**Boolean**;

Description:

Delete a shape or a grouped shape, Shape's ID , start from zero ;when you delete one or more shapes, the shapeId will not be chaged.

Sample:

```
procedure Form1.Button1Click(Send:TObject)
begin
  DeleteShapeByID(1)
end;
```

Return Value:
true : delete success
false : delete failed

See Also:

DeleteSelectedShape
DeleteAllShapes

4.3.28 DeSelectedAllShapesByCode

procedure DeselectedAllShapesByCode

Description:

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

Example:

```
MyCAD1.DeselectedAllShapesByCode;
```

See also:

SelectShapeByCode

4.3.29 Destroy

destructor Destroy;

Description:

Destroy the instance of TMyCAD, all layers,all shapes belonged it will be destoryed too;

See also:

Create

4.3.30 DrawAllShape

procedure DrawAllShape(MyCanvas:TCanvas;ARect:TRect);

Description:

Redraw all shapes.

Parameter:

MyCanvas : The canvas that you want draw on;
ARect: The region, 15 shape out of this region, it will not be draw.This is for speed drawing.

4.3.31 FlipHoriz

procedure FlipHoriz(AShape:TMyShape)

Description:

Flip a shape in horizontally .if AShape is a group shape,the childshapes that belonged it will be fliped.

See also:

TMyShape.IsFlipVert

4.3.32 FlipVert

procedure FlipVert(AShape:TMyShape)

Description:

Flip a shape in vertically . if AShape is a group shape,the childshapes that belonged it will be fliped too.

See also:

TMyShape.IsFlipHorz

4.3.33 GetLayerIdByName

function GetLayerIdByName(ALayerName: string): **integer**;

Description:

Get layer's Id and it's name is ALayerName, if do not match ALayerName, return -1;

Sample:

```
procedure For m1. But t on1Cl i ck( Send: TObj ect )
begin
  ShowMessage(' The Layers name is:'+'MyName' +' its layer Id is: '+Inttostr(MyCAD1.
GetLayerIdByName('MyName')));
end;
```

See Also:
GetLayerNameById
GetLayerNobyId

4.3.34 GetLayerIdByNo

function GetLayerIdByNo(ALayerNo: integer): **integer**;

Description:

Get layer's Id by No, if do not match No, return -1;

Sample:

```
procedure Form1.Button1Click(Send:TObject)
var
 a:integer;
begin
 a:=GetLayerIdByNo(4);

  ShowMessage(' The Layers No is:'+'4' +' its layer Id is: '+Inttostr(a));
end;
```

See Also:

GetLayerNameById
GetLayerNobyId

4.3.35 GetLayerNameByID

function GetLayerNameByID(ALayerId: integer): **string**;

Description:

Get layer's Name using it's Id , if do not match id, return '';

Sample:

```
procedure Form1.Button1Click(Send:TObject)
begin
  ShowMessage(' The Layers Id is:'+'8' +' Name is: '+GetLayerNameByID(8);
end;
```

See Also:

GetLayerIdByName
GetLayerNobyId

4.3.36 GetLayerNoById

function GetLayerNoById(ALayerId: integer): **integer**;

Description:

Get layer's No by Id, if do not match Id, return -1;

Sample:

```
procedure Form1.Button1Click(Send:TObject)
begin
  ShowMessage(' The Layers Id is:'+'4' +' its layer No is: '+Inttostr(MyCAD1.
GetLayerNoById(4)));
end;
```

See Also:

GetLayerIdByNo

4.3.37 GetLayerNoByName

function GetLayerNoByName(ALayerName: string): **integer**;

Description:

Get layer's No and it's name is ALayerName, if do not match ALayerName, return -1;

Sample:

```
procedure Form1.Button1Click(Send:TObject)
begin
with MyCAD1 do
  ShowMessage(' The Layers name is:'+'MyName' +' layer No is: '+Inttostr(GetLayerNoByName
('MyName')));
end;
```

See Also:
GetLayerIdByNo
GetLayerNameById
GetLayerNobyId

4.3.38 GetLayersCount

function GetLayersCount(): **integer**;

Description:

How many layers in TMyCAD.

Sample:

```
procedure Form1.Button1Click(Send:TObject)
begin
  ShowMessage(' There are '+Inttostr(MyCAD1.GetLayersCount)+'lagers in TMyCAD1!');
end;
```

See Also:

GetShapesCount
GetShapesCountInALayer

4.3.39 GetMaxLayerId

function GetMaxLayerId(): **integer;**

Description:

Get max layer's Id, It is not the count of layers,if there is no layer existed,it return -1.

Example:

```
procedure Form1.Button1Click(Send:TObject)
begin
  ShowMessage(' The max layer Id is  '+Inttostr(MyCAD1.GetMaxLayerID));
end;
```

See Also:

GetLayersCount

4.3.40 GetMemShapesCount

function GetMemShapesCount: integer;

Description:

Get the count of memorize shapes in undo list.

Example:

```
EditPaste1.Enabled:=  MyCAD1.GetMemShapesCount > 0;
```

See also:

GetShapesCount

4.3.41 GetSelectedShape

function GetSelectedShape: TMyShape;

Description:

Get the selected shape object.

Return Value:

When more than two shapes selected , return nil, When a grouped shape selected, return the first shape

Example:

```
Shape1:= MyCAD1.GetSelectedShape;
```

See Also:

GetSelectedShapes

4.3.42 GetSelectedShapes

function GetSelectedShapes: TArrMyShape;

Description:

Get the selected shapes array.

Return Value:

If select more than two shapes selected ,Please use this function to get seleceted shape.

Example:

```
Shape1:= MyCAD1.GetSelectedShapes[0];
```

See Also:

GetSelectedShape

4.3.43 GetSelectedShapesCount

function GetSelectedShapesCount:integer;

Description:

Get how many shape were selected, a grouped shape is one shape .

See also:

GetSelectedShape
GetSelectedShapes

4.3.44 GetShapebyID

function GetShapeByID(Id:integer): TMyShape;

Description:

Know a shape's ID , Get the shape object.

Return Value:

if Id is not exist, return nil;

Example:

```
Shape1:= MyCAD1.GetShapeById(0);
```

See Also:

GetShapeByName
GetShapeByNo

4.3.45 GetShapeByName

function GetShapeByName(const AName:String): TMyShape;

Description:

Know a shape's name , Get the shape.

Return Value:
if there is no shape named AName , return nil;

Example:

```
Shape1:= GetShapeByName('Shape100');
```

See also:

GetShapeById

4.3.46 GetShapebyNo

function GetShapeByNo(AShapeNo:integer): TMyShape;

Description:

Know a shape's No , get the object.

Return Value:
if AShapeNo is not exist, return nil;

Example:

```
Shape1:= MyCAD1.GetShapeByNo(10);
```

See Also:

GetShapeByName
GetShapeByID

4.3.47 GetShapeNoById

function GetShapeNoById(AShapeId:Cardinal): Integer;

Description:

Know a shape's ID , get the shape no.

Return Value:

if Id is not exist, return -1;

Example:

```
ANo := MyCAD1.GetShapeNoById(0);
```

See Also:

GetShapeByName
GetShapeByNo

4.3.48 GetShapesCount

function GetShapesCount: integer;

Description:

Get the count of shapes that you have added .

Example:

```
ShowMessage('There are ' + IntToStr(MyCAD1.GetShapesCount)+ ' in your form!')
```

See also:

GetShapesCountInALayer

4.3.49 GetShapesCountInALayer

function GetShapesCountInALayer(ALayerId: integer): **integer**;

Description:

Get the amount of the shapes on a layerId;

See also:

GetShapesCount

4.3.50 GetShapesByLayerId

function GetShapesByLayerId(layerId: Byte): TArrMyShape;

Description:

Get all the shapes in a layer.

Parameter:

layerId: the layer id

Return Value:

the shapes array

4.3.51 GroupWorkingShape

function GroupWorkingShape: Integer;

Description:

Group one or more selected shape.

Return Value:

-1: Group shape failed
else:The ShapeID

Example:

```
MyCAD1.GroupWorkingShape;
```

See Also:

UnGroupShape

4.3.52 InVisibleLayerById

procedure InVisibleLayerByID(LayerId: integer);

Description:

set the layer hide.

See also:

InVisibleLayerByName

4.3.53 InVisibleLayerByName

procedure InVisibleLayerByName(LayerName: string);

Description:

set the layer hide.

See also:

InVisibleLayerById

4.3.54 IsLinked

function IsLinked(AShape,BShape:TMyShape): Integer;

Description:

To sure there is linking relationshape between AShape and BShape. it there is more than one linking relationship between , it is onyl return the first.

Return Value:

-1:there is no link
>=0 : it is linkline shape id .

4.3.55 IsTCADFile

function IsTCADFile(FileName:String):Boolean;

Description:

Determine a file is in TCAD file format or not .

Example:

```
if MyCAD1.IsTCADFile( OpenDialog1.FileName) then
begin
ifnot MyCAD1.LoadFromFile(OpenDialog1.FileName) then
        ShowMessage('Error when read file');
end
else
   ShowMessage( OpenDi al og1. Fi l eName +'is not a TCAD format file');
end;
```

4.3.56 IsVisibleLayerByID

function IsVisibleLayerByID(LayerId: integer): **Boolean**;

Description:

Is the layer is visible or not .

Return Value:

true: it is visible
false:it is invisble or LayerId is not existed.

4.3.57 LoadFromFile

function LoadFromFile(FileName:String):Boolean;

Description:

Load from a file.

Example:

```
MyCAD1.LoadFromFile('c:\SavedFile.tcad');
```

See also:

SavetoFile

4.3.58 LoadFromStream

function LoadFromStream(Stream:TStream):Boolean;

Description:

Load from the stream, you can read from the database's field or TCP/IP stream.

Example:

```
MyCAD1.LoadFromStream(myStream);
```

See also:

SaveToStream

4.3.59 LockUnLockforShapes

procedure LockUnLockforShapes(AShape:TMyShape,Value:boolean);

Description:

Prevent the shape or the grouped shape be moved or resized by mouse, but can moved by code. After execute this command, AShape.Locked is true and display the gray hotspot when you selected this shape.

Parameter:
AShape:a shape;
Vaule: true,lock it;
false,unlock it.

4.3.60 MergeLayers

function MergeLayers(LayerId1,LayerId2:integer): Boolean;

Description:

Merge Layer2 to Layer1 and remove the layer2, CurrentLayerId is Layer1.

Return :
false: failed
true: merge success

Example:

```
if MyCAD1.NewLayer (1,2) then
  ShowMessage('Merge ok!');
```

See also:
CurrentLayerId

4.3.61 Move

procedure Move(AShape:TMyShape;Dx,Dy:integer);

Description:

move a shape or a grouped shape by code.OnShapeDragging event tigged.

Example:

```
 AShape:= MyCAD1.GetSelectedShape;
if AShape <> nil then
   MyCAD1.Move(AShape,20,40);
```

See Also:

OnShapeDragginga

4.3.62 NewLayer

function NewLayer(ALayerName: string; aVisible: Boolean = true): integer;

Description:

Add a layer for TMyCAD, return new layerid, CurrentLayerid is new layerid .

Return :
-1: failed
else: return LayerId.

Example:

```
if MyCAD1.NewLayer (' mapLayer')> -1then
  ShowMessage('Add ok');
```

See also:
CurrentLayerId

4.3.63 Paste

procedure Paste();

Description:

Paste saved shapes to TMyCAD, they have 4 pixels offset .
Example:

```
MyCAD1.SelectShapeBycode(0);
MyCAD1.Cut;
MyCAD1.Paste;
```

See also:

Copy
Cut
PasteFromMyCAD

4.3.64 PasteFromMyCAD

procedure PasteFromMyCAD(AMyCAD: TMyCAD);

Description:

Paste saved shapes to other TMyCAD.

Example:

```
MyCAD1.PasteFromMyCAD(MyCAD2);
```

See also:

Paste

4.3.65 PopfromUndoRedoShapeList

procedure PopfromUndoRedoShapeList

Description:

Undo one step .

Example:

```
MyCAD1.PopfromUndoRedoShapeList;
```

See also:

UndoRedoSize

4.3.66 Print

procedure Print(ALayers: array of Integer; UserScale: double = 1.0);

Description:

Print to printer.

Parameter
ALayer: if you want print all layer, please use [], else [1,2,3], or you create array and set layerId that you want print
UserScale:the scale.

Example:

Delphi

```
//print all shapes in 50% scale.
MyCAD1.Print ([],0.5);
```

C++ Builder

```
//print all shapes in 100% scale.
void__fastcall TMainFrm::BitBtn1Click(TObject *Sender)
{
int p[1];
 p[0]=NULL;
 MyCAD1->Print(p,0,1.0);
}

//print the shapes in 0,1 layers in 100% scale.
void__fastcall TMainFrm::BitBtn1Click(TObject *Sender)
{
int p[2];
 p[0]=0; //First Layer Id
 p[1]=1; //Second Layer Id
```

```
 MyCAD1->Print(p,2,1.0);
}
```

See also:

PrintPreview

4.3.67 PrintPreview

procedure PrintPreview(ALayers: array OF Integer; ABitmap:TBitmap; ScaleforPreview:double = 1.0);

Description:

Preview the drawing on the Bitmap.

Parameter
ALayer: if you want print all layer, please use [], else [1,2,3], or you create array and set layerId that you want print
ABitmap: that you draw on.
ScaleforPreview:the scale.

Example:

Delphi:

```
//print preview all shapes in 100% scale.
MyCAD1.PrintPeview ([],Image1.Picture.Bitmap,1.0);

 int p[1];
 p[0]=0;
  MyCAD1->PrintPreview(p,1,Form1->Image1->Picture->Bitmap,1);
  Form1->Show();
}
//---------------------------------------------------------------------------
```

C++ Builder:

```
//print preview all shapes in 100% scale.
void__fastcall TMainFrm::BitBtn1Click(TObject *Sender)
{
int p[1];
 p[0]=NULL;
MyCAD1->PrintPreview(p,0,Image1->Picture->Bitmap,1.0);
}
//print preview the shapes in 0,1 layers in 100% scale.
void__fastcall TMainFrm::BitBtn1Click(TObject *Sender)
{
int p[2];
 p[0]=0; //First Layer Id
 p[1]=1; //Second Layer Id
 MyCAD1->Print(p,2,Image1->Picture->Bitmap,1.0);
}
```

See also:

Print

4.3.68 RenameShapename

procedure RenameShapeName(const OldName,NewName:String);

Description:

Know a shape's name , Set the shape new name.

Example:

```
MyCAD1.RenameShapeName('Shape100','MyNewShape');
```

4.3.69 Rotate

procedure Rotate (AShape:TMyShape;AAngle:Single);

Description:

Rotate a shape or a combine shape by code. AAngle in radians. This command is more useful for automatic processing.

Example:

This example is for rotating shape 30 degree.

```
 AShape:= MyCAD1.GetSelectedShape;
if AShape <> nil then
   MyCAD1.Rotate(AShape, 30*pi/180);
```

4.3.70 SaveToBmp

procedure SaveToBmp(BmpFileName:String);

Description:

Save TMyCAD drawing to a disk file , it is in BMP format .

Example:

```
MyCAD1.SavetoBMP('c:\SavedFile.bmp');
```

See also:

LoadFromFile
SavetoStream
SaveToWMF

SaveToJPG

4.3.71 SaveToBmp-2

procedure SaveToBmp(Bmp:TBitmap; NewWidth, NewHeight: integer); overload;

Description:

Save TMyCAD drawing to a disk file in new ratio, it is in BMP format . if NewWidth/ Width > NewHeight / Height , then the zoom according NewWidth/ Width, else according NewHeight / Height.

Example:

```
MyCAD1.SavetoBMP('c:\SavedFile.bmp',100,200);
```

See also:

LoadFromFile
SavetoStream
SaveToWMF
SaveToJPG

4.3.72 SaveToDxf

procedure SaveToDXF(DXFFileName:String);

Description:

Save CAD drawing to a disk file , it is in AutoCAD R12 DXF format .

Example:

```
CAD1.SavetoDXF('c:\SavedFile.dxf');
```

See also:

LoadFromFile
SavetoStream
SaveToBmp
SaveToJPG

4.3.73 SaveToFile

function SaveToFile(FileName:String):Boolean;

Description:

Save TMyCAD drawing content to a disk file, it is in TCADformat .

Return value:

true: the drawing saved to the file.
false:the drawing not saved.

Example:

```
CAD1.SavetoFile('c:\SavedFile');
```

See also:

LoadFromFile
SavetoStream

4.3.74 SaveToJpg

procedure SaveToJpeg(JpegFileName:String;Quality:integer=80)

Description:

Save CAD drawing to a disk file , it is in JPEG format , set parameter Quality can change the quality of the image,default is 80%.

Example:

```
CAD1.SavetoJPG('c:\SavedFile.jpg');
```

See also:

LoadFromFile
SavetoStream
SaveToBmp
SaveToWMF

4.3.75 SaveToStream

function SaveToStream(Stream:TStream):Boolean;

Description:

Save to the stream, you can save all content into the database's field or memory stream.

Example:

MyCAD1. SaveToSt r eam(mySt r eam) ;

See also:

LoadFromStream
SaveToFile

4.3.76 SaveToWmf

procedure SaveToWmf(WMFFileName:String);

Description:

Save CAD drawing to a disk file , it is in WMF format .

Example:

```
CAD1.SavetoWMF('c:\SavedFile.WMF');
```

See also:

LoadFromFile
SavetoStream
SaveToBmp
SaveToJPG

4.3.77 SelectAllShapes

procedure SelectAllShapes;

Description:

Select all shapes

Notes:

if the shape is grouped with other shapes, the hotspot for that group appear.

Examples:

```
MyCAD1.SelectAllShapes;
```

See also:

SelectAllShapesByLayerId
SelectShapeByCode

4.3.78 SelectAllShapesByLayerId

procedure SelectAllShapesByLayerId(const ALayerId:integer);

Description:

Specifies the layer id to select all shapes in this layer.

Notes:

if the shape is grouped with other shapes, the hotspot for that group appear.

Examples:

```
MyCAD1.SelectAllShapesByLayerId(0);
```

See also:

SelectAllShapes
SelectShapeByCode

4.3.79 SelectShapeByCode

function SelectShapeByCode(ShapeId:integer;RemovePrevSelectedShape:boolean=true): Boolean;

Description:

Select a shape by code like mouse actions. Use this function if, for example, you want to perform operations on a certain shape, which require the use of GetSelectedShapeId, if RemovePrevSelectedShape is true, do not clear last selected shape hot spot.

Return value:
true: the shape is selected
false: there is no shape with this Id,

Notes:

if the shape is grouped with other shapes, the hotspot for that group appear.
This function can effect the GetSelectedShapeId

Examples:

```
MyCAD1.SelectShapeByCode(2);
```

See also:

SelectAllShapes
SelectAllShapesByLayerId

4.3.80 SendtoBack

procedure SendToBack(AShape:TMyShape;NeedSave:Boolean=true);

Description:

Sent the selected shape to background .

Parameter:

AShape:the shape that you will changed;
NeedSave:used by internal.

Example:

```
MyCAD1.SendToBack(MyCAD1.GetSelectedShape);
```

See Also:

BringToFrontByStep
BringToFront

4.3.81 SendToBackByStep

procedure SetLayerNameByID(const NewLayerName: string; ALayerID: integer);

Description:

Change the**procedure** SendToBackByStep(AShape:TMyShape;NeedSave:boolean=true)

Description:

Sent the selected shape to background by a step

Parameter:

AShape:the shape that you will changed;
NeedSave:used by internal.

Example:

```
MyCAD1.SendToBackByStep(MyCAD1.GetSelectedShape);
```

See Also:

SendToBack
BringToFront

4.3.82 SetLayerNameById

procedure SetLayerNameByName(const NewLayerName, OldLayerName: string);

Description:

Change the layer's name by layer's id.

4.3.83 SetLayerNameByName

procedure SetLayerNameByName(const NewLayerName, OldLayerName: string);

Description:

Change the layer's name by old layer's name.

4.3.84 SetMyImage

function SetMyImage(ShapeID: integer; ABitmap: TBitmap): **Boolean**;

Description:

Add a bitmap to TMyImage's shape, it is important.

Example:

```
if AShape.ClassName = 'TMyImage'then
begin
if OpenPictureDialog1.Execute then
begin
      myBitmap := TBitmap.Create;
      mybitmap.LoadFromFile(OpenPictureDialog1.FileName);
if MyCAD1.SetMyImage(ShapeId, MyBitmap) then
        Memo1.Lines.Add('Bitmap be loaded into Shape' + Inttostr(ShapeID))
else
        Memo1.Lines.Add('Bitmap NOT be loaded into Shape' + Inttostr(ShapeID));
      MyBitmap.Free;
end;
end;
```

Return value:

true: add image success
false:add image failed.

4.3.85 SizeShape

procedure SizeShape(AShape:TMyShape;ASelectedHotId:integer;AMovPt:TMyPoint);

Description:

Size a shape or a grouped shape,it is like mouse action of resizing a shape.

Parameter:

AShape:shape that you will size it.
ASelectedHotId:it is hotspot id.
AMovPt:the destination point;

Example:

```
MyCAD1.SizeShape(MyCAD1.GetSelectedShape,0,MyPoint(100,100));
```

See also:

Move

4.3.86 Tilt

function Tilt(operateShape:TMyShape;centerPoint,fromPoint,toPoint:TMyPoint;Constrain: Boolean):single;

Description:

Rotate a shape by code.

Parameter:

operateShape: shape that you will rotate .
centerPoint: specify the center point of shape.
fromPoint: the origin point;
toPoint: the destination point;
Constrain:constraint the degree or not.

Return:
return the rotated angle value.

Example:

```
var
  oShape:TMyShape;
begin
  oShape := MyCAD1.GetSelectedShape;
  MyCAD1.Tilt(oShape,MyPoint(100,100),MyPoint(200,200),MyPoint(300,300),false);
end;
```

4.3.87 UnGroupShape

procedure UngroupShape(AShape: TMyShape;NeedSaved:boolean= true);

Description:

unGroup have grouped shapes.

Parameter:

AShape:shape that you will ungroup it.

NeedSaved:it is for undo procedure,normal is true, do not change it,it is used by internal.

Example:

```
MyCAD1.UnGroupShape(MyCAD1.GetSelectedShape);
```

See also:

GroupWorkingshape

4.3.88 VisibleAllLayer

procedure VisibleAllLayer;

Description:

Make all layer(s) visible

See Also:

VisibleLayerById
VisibleLayerByName
InVisibleLayerById
InVisibleLayerByName

4.3.89 VisibleLayerByID

procedure VisibleLayerByID(LayerID: integer);

Description:

Make a layer visible by layer's id.

See Also:

VisibleAllLayer
VisibleLayerByName
InVisibleLayerById
InVisibleLayerByName

4.3.90 VisibleLayerByName

procedure VisibleLayerByName(LayerName: string);

Description:

Make a layer visible by layer's name.

See Also:

VisibleAllLayer
VisibleLayerById
InVisibleLayerById
InVisibleLayerByName

4.4 Properties

4.4.1 ArrowAngle

property ArrowAngle:integer

Description:

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

Example:

```
MyCAD1.ArrowAngle := 180 ;
```

See also:

ArrowOffset
ArrowStyle

4.4.2 ArrowLength

property ArrowLength:Byte

Description:

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

See also:

ArrowAngle
ArrowOffset
ArrowStyle

4.4.3 ArrowOffset

property ArrowOffset:byte

Description:

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
ArrowStyle

4.4.4 ArrowStyle

property ArrowStyle:**TArrowStyle**

Description:

Set the Line and PolyLine 's arrow style;

ANone: it is a line;
ALeft: one arrow at the left end of the line or polyline;
ARight: one arrow at the right end of the line or polyline;
ADouble: a double-arrow line or polyline

See also:

ArrowAngle
ArrowOffset

4.4.5 BkBitmap

property BkBitmap:TBitmap

Description:

Set the background bitmap for TMyCAD, for clear it , BkBtimap:=nil;

Example:

The example show assign a bitmap that you load to MyCAD1.

```
var
```

```
   mybitmap:TBitmap;
begin
   if OpenPictureDialog1.Execute then
begin
        myBitmap:=TBitmap.Create;
        mybitmap.LoadFromFile(OpenPictureDialog1.FileName);
        MyCAD1.BkBitmap:=mybitmap;
        MyBitmap.Free;
end;
end;
```

See also:

BkBitmapMode

4.4.6 BkBitmapMode

property BkBitmapMode:TBkBitmapMode

TBkBitmapMode = (Tiled,Stretch,Center,LeftTop);

Description:

Set the background bitmap show style.

Tiled : if the bitmap size low than TMyCAD's size , two or more this bitmap drawed on TMyCAD
Stretch:Bitmap be stretch to fit at TMyCAD
Center: Bitmap is in center of TMyCAD
LeftTop:Bitmap be showed from TMyCAD's Left,Top

See Also:
BkBitmapMode

4.4.7 Brush

property Brush:TBrush;

Description:

Set the brush of TMyCAD that you need;

4.4.8 Canvas

property Canvas:TCanvas;

Description:

Canvas allows drawing on the TMyCAD by providing a TCanvas object for this purpose. Any canvas operation is valid on TMyCAD, and draw operation doesn't change TMyShapes object, the content will be clear when paint event occured.A canvas object is created automatically for the TMyCAD and the property is read-only.

See also:

OnPaint

4.4.9 ColorOfBackground

property ColorOfBackground:TColor

Description:

specify the background color .

Example:

```
MyCAD1.ColorOfBackground := clBule;
```

See Also:

ColorOfHot

4.4.10 ColorOfHot

property ColorOfHot:TColor

Description:

Specify the color of hot square;

Example:

```
MyCAD1.ColorofHot:=clRed;
```

See also:

ColorofBackground

4.4.11 CrossLine

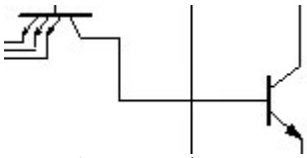
property CrossLine:Boolean;

Description:

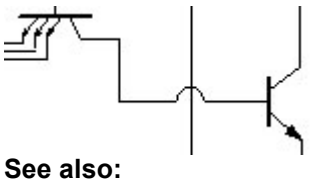
It is used for TMyLinkLine drawing feature, when it is true,the drawing speed lower than false;

Example:

```
MyCAD1.CrossLine:=false;
```



```
MyCAD1.CrossLine:=true;
```



See also:

TMyLinkLine

4.4.12 CurrentLayerId

property CurrentLayerId :integer;

Description:

Get the current layer's ID, it is start from 0; New shape(s) will belong to this layer.

Example:

```
MyCAD1.CurrentLayerID:=2;
ShowMessage (IntToStr('You are drawing on layerID:'+MyCAD1.CurrentLayerID));
MyCAD1.CurrentLayerID:=3;
ShowMessage (IntToStr('You are drawing on layerID:'+MyCAD1.CurrentLayerID));
// if there is no layerid is 3,MyCAD1.CurrentLayerID still equal 2
```

See also:

NewLayer

4.4.13 DisableTextInput

property DisableTextInput:Boolean;

Description:

Set as true,it will hide text input dialog when you draw a text shape by mouse.

Example:

```
MyCAD1.DisableTextInput:=true;
```

4.4.14 DiskFileVersion

It is readonly field,for TMyCAD version compatibility.

4.4.15 DragMode

property DragMode:**TDragMode**

Description:

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

Example:

```
MyCAD1.DragMode:=dgHorz;
```

See Also:
TDragMode= (dgHorz,dgVert,dgBoth);

4.4.16 DragTrace

property DragTrace: Boolean
Description:

Showing the drag trace or not when you dragging a shape.

Example:

```
MyCAD1.DragTrace:=false;
```

4.4.17 Enable

property Enabled: Boolean;

Description

Whether the TCAD control responds to mouse, keyboard, and timer events. but still effected by code action.

Use Enabled to change the availability of the TMyCAD to the user. To disable a control, set Enabled to false. Disabled TMyCAD appear dimmed. If Enabled is false, the TMyCAD ignores mouse, keyboard, and timer events.

To re-enable a control, set Enabled to true. The TMyCAD is no longer dimmed, and the user can use the TMyCAD.

See also

Visible

4.4.18 Font

property Font:TFont;

Description:

Set the font.

4.4.19 GridOperation

property GridOperation:TGridOperation

Description:

set the grid of TMyCAD.

4.4.20 HotShow

property HotShow:boolean;

Description:

true: the hotspot showed normally
false: the hotspot hide, even a shape be selected.

Example:

Delphi syntax:

```
MyCAD1.HotShow := true;
```

C++ syntax:

```
MyCAD1->HotShow = true;
```

4.4.21 HotSize

property HotSize:byte

Description:

can set the hotsize' size.

Delphi syntax:

```
MyCAD1.HotSize := 6;
```

C++ syntax:

```
MyCAD1->HotSize = 6;
```

4.4.22 LabelValue

property LabelValue:TLable

Description:

Show the parameter about the current shape. It can display the length of a TMyLine or height and width of a TMyRectangle

Example:

```
MyCAD1.LabelValue:=Form1.Label2;
```

See also:

LabelXY

4.4.23 LabelXY

property LabelXY:TLabel

Description:

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

Example:

Delphi syntax:

```
MyCAD.LabelXy:=Form1.Label1;
```

C++ syntax:

```
MyCAD->LabelXy =Form1->Label1;
```

See also:

LabelValue

4.4.24 LinkLineDrawStyle

property LinkLineDrawStyle:TLinkLineDrawStyle

Description:

define linkline style.

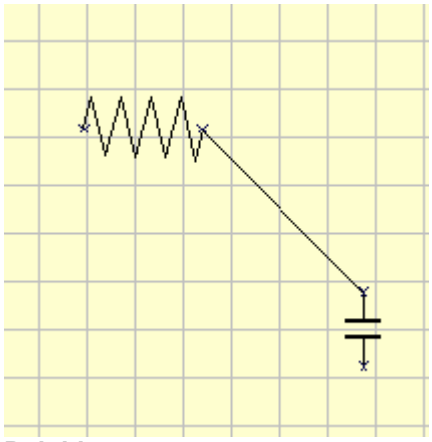
Example:

Delphi syntax:

```
MyCAD1.LinkLineDrawStyle:=lldsFree;
```

C++ syntax:

```
MyCAD1->LinkLineDrawStyle =lldsFree;
```

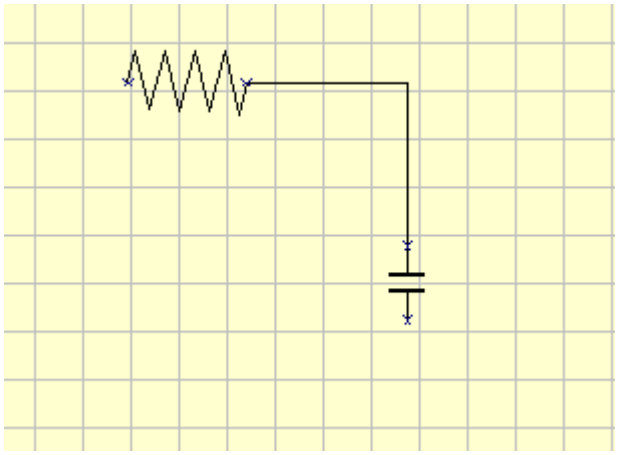


Delphi syntax:

```
MyCAD1.LinkLineDrawStyle:=lldsHV;
```

C++ syntax:

```
MyCAD1->LinkLineDrawStyle =lldsHV;
```



4.4.25 LockBound

property LockBound: Boolean
Description:

This property specify the lock bound out or not when you drag a shape.

Example:

```
MyCAD1.LockBound:=true;
```

4.4.26 MouseEffect

property MouseEffect:Boolean;
Description:

Allow user-define cursor. .

Example:

```
MyCAD1.MouseEffect := true;
```

4.4.27 OperateAllLayer

property OperateAllLayer:boolean;

Description:

Is you want mouse action effects only to current layer or all layers. default is true;

Example:

```
MyCAD1.OperateAllLayer :=true;
```

4.4.28 PageFoot

property PageFoot:string;

Description:

Set the printing page foot.

Example:

Delphi syntax:

```
MyCAD1.PageFoot := 'Designed by hongbin.fei, 2004.12';
```

C++ syntax:

```
MyCAD1->PageFoot = "Designed by hongbin.fei, 2004.12";
```

See also:

PageHead

4.4.29 PageFootAlignment

property PageFootAlignment: TAlignment;

Description:

Determines how the text is aligned within the page foot.

Use Alignment to change the way the text is formatted by the TMyCAD control. Alignment can take one of the following values:

Value	Meaning
taLeftJustify	Text is left-justified: Lines all begin at the left edge of the control.
taCenter	Text is centered in the control.
taRightJustify	Text is right-justified: Lines all end at the right edge of the control.

See also:

PageHeadAlignment

4.4.30 PageFootFont

property PageFootFont:TCADFont;

Description:

Set the font of page foot.

4.4.31 PageFootToBottom

property PageHeadToBottom: integer;

Description:

The space of page foot to bottom, it is in pixel unit.

See also:

PageHeadToTop

4.4.32 PageHead

property PageHead:string;

Description:

Set the page title.

Example:

Delphi syntax:

```
MyCAD1.PageHead := 'TCAD magic drawing';
```

C++ syntax:

```
MyCAD1->PageHead = "TCAD magic drawing";
```

See also:

PageFoot

4.4.33 PageHeadAlignmen

property PageHeadAlignment: TAlignment;

Description:

Determines how the text is aligned in the page head.

Use Alignment to change the way the text is formatted by the TCAD control. Alignment can take one of the following values:

Value	Meaning
taLeftJustify	Text is left-justified: Lines all begin at the left edge of the control.
taCenter	Text is centered in the control.
taRightJustify	Text is right-justified: Lines all end at the right edge of the control.

See also:

PageFootAlignment

4.4.34 PageHeadFont

property PageHeadFont:TCADFont;

Description:

Set the font style of page head.

4.4.35 PageHeadToTop

property PageHeadToTop: integer;

Description:

The space of page title to top, it is in pixel unit.

See also:

PageFootToBottom

4.4.36 PageHeight

property PageHeight:Cardinal

Description:

Set the Canvas(Page)'s Height, it is in unit cm. If you change the PageHeight ,PageStyle is changed automatic.

Example:

```
MyCAD1.PageHeight := 210;
```

See also:

PageStyle
PageWidth

4.4.37 PageOrientation

property PageOrientation:TPrinterOrientation ;

Description:

The Canvas's direction is Portrait or Landscape.then PageWidth and PageHeight will be changed(swaped) unless if PageStyle is CustomerPage.

Example:

```
MyCAD1.PageOrientation := poLandscape;
```

See also:

PageStyle

4.4.38 PageStyle

property PageStyle: TPageStyle;

Description:

Set the Canvas(Page)'s style,(A0,A1,A2,A3,A4,A5,B3,B4,B5,CustomerPage),

then PageWidth and PageHeight will be changed.

Example:

```
MyCAD1.PageStyle:= A4;
```

See also:

Defines

4.4.39 PageWidth

property PageWidth:Cardinal

Description:

Set the Canvas(Page)'s Width, it is in unit cm. If you change the PageStyle,then it is changed automatic.

Example:

```
MyCAD1.PageWidth := 120;
```

See also:

PageStyle
PageHeight

4.4.40 Pen

property Pen:TPen

Description:

Set the pen that you need;

4.4.41 PenStyleEndCapType

property PenStyleEndCapType:TPenStyleEndCapType

Description:

Set the pen style type of the line end cap.

Value:

psectROUND
psectSQUARE
psectFLAT
psectMASK

4.4.42 PenStyleJoinType

property PenStyleJoinType:TPenStyleJoinType

Description:

Set the pen style type of the line joining.

Value:

psjtROUND
psjtBEVEL
psjtMITER
psjtMASK

4.4.43 PenWidthRelateZoom

property PenWidthRelateZoom:Boolean

Description:

Set the pen width associate zoom, when you zoom the canvas, the pen width can change together.

4.4.44 PreZoom

property PreZoom: Double;

Description:
This property stores a preview zoom value.

See also:
Zoom

4.4.45 PrintABorder

property PrintABorder:Boolean ;

Description:

Is a border when printpreview or print.

See Also:

PrintBackground

4.4.46 PrintABordertoBottom

property PrintABordertoBottom: integer;

Description:

The space of the border to bottom, it is in pixel unit.

See also:

PrintABordertoLeft
PrintABordertoTop
PrintABordertoRight

4.4.47 PrintABordertoLeft

property PrintABordertoLeft: integer;

Description:

The space of the border to left, it is in pixel unit.

See also:

PrintABordertoTop
PrintABordertoBottom
PrintABordertoRight

4.4.48 PrintABordertoRight

property PrintABordertoRight: integer;

Description:

The space of the border to right, it is in pixel unit.

See also:

PrintABordertoLeft
PrintABordertoTop
PrintABordertoBottom

4.4.49 PrintABordertoTop

property PrintABordertoTop: integer;

Description:

The space of the border to top, it is in pixel unit.

See also:

PrintABordertoLeft
PrintABordertoBottom
PrintABordertoRight

4.4.50 PrintBackground

property PrintBackground:Boolean ;

Description:

Print canvas's background,if true,print will like the screen.

Example:

```
MyCAD1.PrintBackground:=false;
```

See Also:

PrintABorder

4.4.51 Ratio

property Ratio:double

Description:

Define the ratio ,It will be effected at property LabelValue, that is Line's Length and area for TMyLine,TMyRectangle,TMyEllisple and other shape.

Example:

```
MyCAD1.Ratio:=100;  // 1:100
```

See also:

TheUnit

4.4.52 ResizeEnable

property ResizeEnable:boolean;

Description:

Can resizing shape or not, it is very useful for the case of don't allow resize shape

See also:

RotateEnable

4.4.53 ReturnToSelecting

property ReturnToSelecting: **Boolean**

Description:
if it is true, mean The ShapeTool will return to [selecting] status automatically after you drawed a shape, trigger a event
OnActionToolToSelecting, else, ReturnToSelecting is false , the ShapeTool still on draw status, You can draw same
shapes consecutively.

Example:

```
MyCAD1.ReturnToSelecting:=false;
```

You can draw same shapes consecutively.

See Also:

OnActionToolToSelecting

4.4.54 RotateConstraintDegree

property RotateConstraintDegree:Integer;
Description:

Set the constraint degree for action rotating,if value as zero,it can rotate freely.

4.4.55 RotateEnable

property RotateEnable:boolean;

Description:

Can rotating shape or not, it is very useful for the case of don't allow rotate shape

See also:

ResizeEnable

4.4.56 Shapetool

property ShapeTool:**TDrawTool**

Description:

Set the current's Tool, you can change it when you need.

Example:

```
// Set Line draw tool
MyCAD1. ShapeTool : =SpLi ne;

// Set Link Line draw tool
MyCAD1.ShapeTool:=SpLinkLine;
```

```
// this code only allowed under TCADxp-PRO or TCADxp-ENT
```

Note:

If set shapetool =spClose , to close TCAD ?draw,drag,resize,rotate fucntion;

See also:

Defines

4.4.57 ShowHint

property ShowHint: Boolean;

Description:

Determines whether the TMyCAD displays a Shape Caption when the mouse pointer rests momentarily on it.

Example:

Delphi syntax:

```
MyCAD1.ShowHint  := true ;
```

C++ syntax:

```
MyCAD1->ShowHint  = true ;
```

4.4.58 ShowHotLink

property ShowHotLink: Boolean;

Description:

Whether show the TMyCAD displays hot link point for a shape.

Example:

Delphi syntax:

```
MyCAD1. ShowHot Li nk  : = true ;
```

C++ syntax:

```
MyCAD1->ShowHotlink  = true ;
```

4.4.59 Snap

property Snap:Boolean ;

Description:

Snap mouse to grid or not, help drawing .

Example:

```
MyCAD1.Snap := true;
```

See also:

SnapPixels
GridWidth
GridHeight

4.4.60 SnapPixels

property SnapPixles:Byte

Description:

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.SnapPixels := 3 ;
```

See also:

Snap
GridWidth
GridHeight

4.4.61 SnapShape

property SnapShape:Boolean ;

Description:

When Drag or Resize shape, whether snap current shape to other shape , it can help you drawing .

Example:

```
MyCAD1.SnapShape := true;
```

See also:

SnapPixels

4.4.62 TheUnit

property TheUnit:**TUnits**

Description:

Which unit do you use, It will be appear at property LabelValue, that is Line's Length or area for TMyLine,TMyRectangle,TMyEllisple...

Example:

```
MyCAD1.TheUnit := inch
```

4.4.63 UndoRedoSize

property UndoRedoSize:byte

Description:

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

Example:

```
MyCAD1.Zoom:= 0.50 ;
```

See also

PopfromUndoRedoShapeList

4.4.64 UserData

property UserData:TUserData;

Description

You can add yourself data .

Example:

```
UserData.AddKeyandValue('Weight','20kg');
```

See also:

UserData

4.4.65 Version

property Version: string;

Description:

Store the version of TMyCAD, it is readonly.

Example:

```
ShowMessage( 'Installed TMyCAD version is:'+MyCAD1.Version);
```

4.4.66 Visible

property Visible: Boolean;

Description

Determines whether the TMyCAD appears on screen.

Use the Visible property to control the visibility of the control at runtime. If Visible is true, the control appears. If Visible is false, the control is not visible.

Calling the Show method sets the control's Visible property to true. Calling the Hide method sets it to false.

See also

Enable

4.4.67 XYMode

property XYMode

Description:

There 4 mode to choose, to fit your need.

Please read **TXyMode** in defines .

4.4.68 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%.

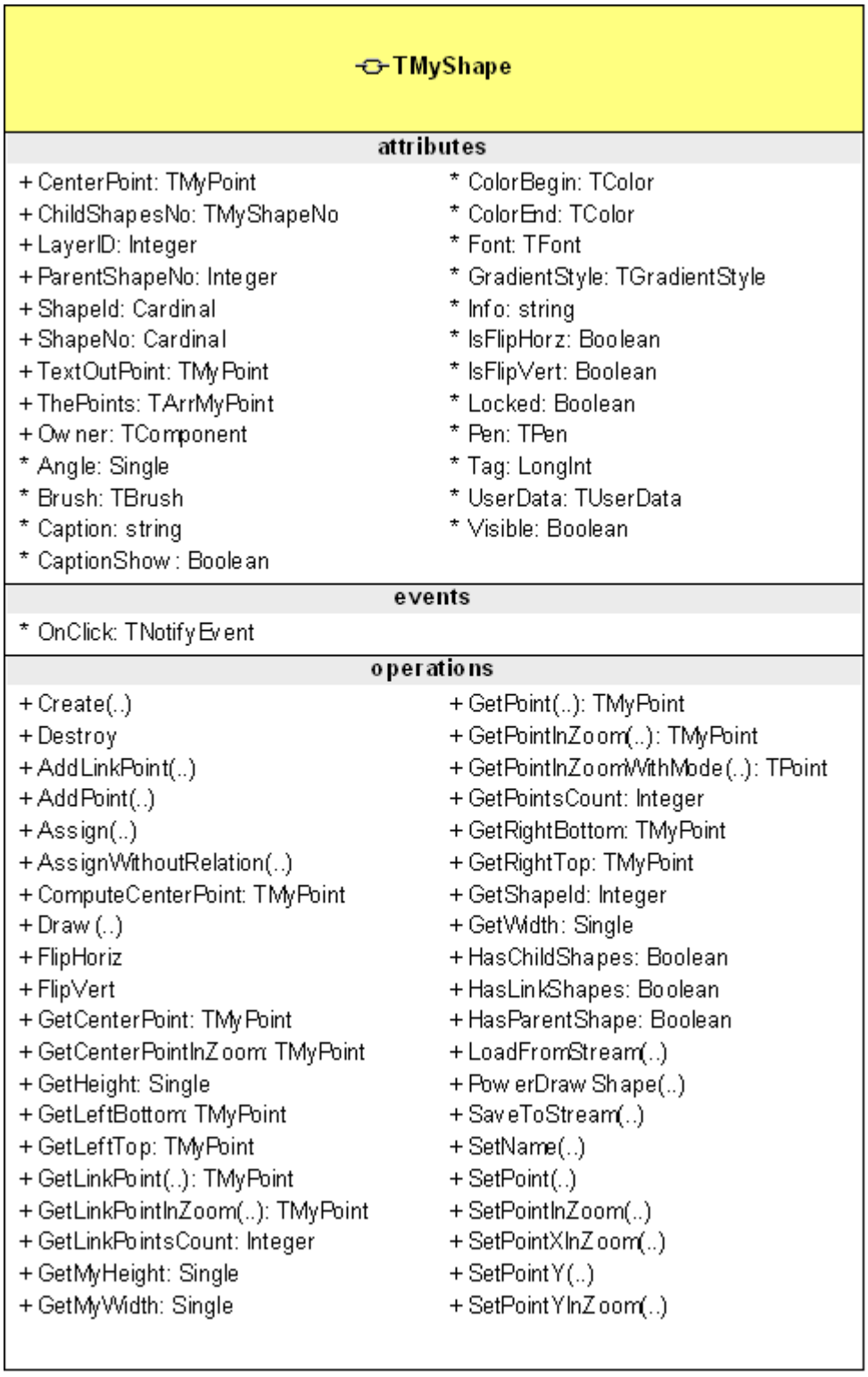
Example:

```
MyCAD1.Zoom:= 0.50 ;
```

5. TMyShape

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

5.1 ClassDiagram



5.2 Fields

5.2.1 CenterPoint

CenterPoint:TMyPoint;

Description:

It is the CenterPoint for a shape, and maintained by TMyCAD.

5.2.2 ChildShapesNo

ChildShapesNo:TMyShapeNo;

Description:

If the shape is a single shape , this field is -1;if it is a grouped or combined shape, this field will store the child shape no.

5.2.3 LayerID

LayerID:integer

Description:

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

5.2.4 ParentShapesNo

ParentShapesNo:integer;

Description:

It is the parent shape's no., if a shape has not a parent , it is -1.

5.2.5 ShapeId

ShapeId:integer;

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 TMyCAD and readonly for you.

5.2.6 ShapeNo

ShapesNo:integer;

Description:

It is the order of MyShapes, delete, add, sendback,bringtofront,sendbackbystep,bringtofrontbystep ,will change it.
it is maintaced by TMyCAD and readonly for you.

5.2.7 TextOutPoint

TextOutPoint:TMyPoint;

Description:

It is text out position, when a shape created, it is at center - bottom of a shape, you can change it by mouse or code.

5.2.8 ThePoints

ThePoints:TArrMyPoint

Description:
It is the points of a shape, different shape has different points count.it is not related with XyMode.
LeftTop is (0,0). it is base on TMyPoint type.

5.3 Methods

5.3.1 Assign

procedure Assign(Source: TMyShape); virtual;

Description:

Copy a source shape properties.

Example:

AShape.Assign(BShape);

See also:

Create

5.3.2 ComputeCenterPoint

function ComputeCenterPoint: TMyPoint;

Description:

It can compute the shape's centerpoint.

Returns:

The center point of a shape.

5.3.3 Create

constructor Create(AOwner: TMyCAD); virtual;

Description:

Internal data structure is initialized.

See also:

Destroy

5.3.4 Destroy

destructor Destroy; override;

Description:

First all owned fields be released, finally inherited Destroy is called.

See also:

Create

5.3.5 Draw

procedure Draw(MyCanvas:TCanvas); virtual;

Description:

Draw a shape on the MyCanvas.

5.3.6 GetCenterPoint

function GetCenterPoint: TMyPoint;

Description:

It returns the shape's centerpoint.

Returns:

The center point of a shape.

See Also:

GetCenterPointInZoom

5.3.7 GetCenterPointInZoom

function GetCenterPointInZoom: TMyPoint;

Description:

It returns the shape's centerpoint in current zoom.

See Also:

GetCenterPoint

5.3.8 GetHeight

function GetHeight: Single;

Description:

It return a shape's Outside-Rectangle 's Height,it is in pixel unit.

5.3.9 GetLeftBottom

function GetLeftBottom: Single;

Description:

It return a shape's Outside-Rectangle's Left-Bottom position,it is in pixel unit.

5.3.10 GetLeftTop

function GetLeftTop: Single;

Description:

It return a shape's Outside-Rectangle's Left-Top position,it is in pixel unit.

5.3.11 GetLinkPoint

function GetLinkPoint(PointID: integer): TMyPoint;

Description

Get the shape's link point by PointId.

Parameter:

PointID : the point id of the link point that you will retrieve.
it is PointId>=0 and PointId<= GetLInkPointsCount-1

Returns:

the link point.

Example:

Get the first link point of a shape.

```
IF AShape.GetLinkPointsCount >0THEN
         APoint:=AShape.GetLinkPoint(0);
```

5.3.12 GetLinkPointInZoom

function GetLinkPointinZoom(PointID: integer): TMyPoint;

Description

Get the link point in current zoom value.Please see **GetLinkPoint**to get more information.

5.3.13 GetMyHeight

function GetMyHeight: Single;

Description:

It return a shape's actual height,it leave out the angle, the value is in pixel unit.

5.3.14 GetMyWidth

function GetMyWidth: Single;

Description:

It return a shape's actual width,it leave out the angle, the value is in pixel unit.

5.3.15 GetPoint

function GetPoint(PointID: integer): TMyPoint; public

Description

Get the shape's point by PointId.

Parameter:

PointID : the point id of the point that you will retrieve. PointId>=0 and PointId<= GetPointsCount-1

Returns:

the point.

Example:

Get the first point of a shape.

```
IF AShape.GetPointsCount >0THEN
         APoint:=AShape.GetPoint(0);
```

5.3.16 GetPointInZoom

function GetPointInZoom(PointID: integer): TMyPoint; public

Description

Get the point in current zoom value.Please see **GetPoint** to get more information.

5.3.17 GetPointsCount

function GetPointsCount: Integer; public

Description:

Get how many points of a shape.

5.3.18 GetRightBottom

function GetRightBottom: Single;

Description:

It return a shape's Outside-Rectangle's Right-Bottom position,it is in pixel unit.

5.3.19 GetRightTop

function GetRightTop: Single;

Description:

It return a shape's Outside-Rectangle's Right-Top position,it is in pixel unit.

5.3.20 GetShapeId

function GetShapeId:integer

Description:

To get the shape id of a shape.

Note:

The ShapeId is start from Zero.

5.3.21 GetWidth

function GeWidth: Single;

Description:

It return a shape's Outside-Rectangle's width,it is in pixel unit.

5.3.22 HasChildShapes

function HasChildShapes: Boolean;

Description

To know a shape has child shape(s) or not.

Return:
true : a shape has child shape(s).
fasle : a shape has no child shape(s).

5.3.23 HasLinkShapes

function HasLinkShapes: Boolean;

Description

To know a shape has linkline shape(s) or not.

Return:
true : a shape has linkline shape(s).
fasle : a shape has no linkline shape(s).

5.3.24 HasParentShape

function HasParentShape: Boolean;

Description

To know a shape has parent shape(s) or not.

Return:
true : a shape has parent shape(s).
fasle : a shape has no parent shape(s).

5.3.25 IsClickedMe

function IsClickedMe(MyCanvas:TCanvas;APoint: TPoint): Boolean; virtual;

Description:

Is the point inside the shape or not.

5.3.26 LoadFromStream

procedure LoadFromStream(AStream:TStream); virtual;

Description:

Load a shape from a stream, other class such as TMyline override it.

See also:

SaveToStream of TMyShape

5.3.27 SaveToStream

procedure SaveToStream(AStream:TStream); virtual;

Description:

It can save a shape to a stream, other class such as TMyline override it.

See Also:

LoadFromStream of TMyShape

5.4 Properties

5.4.1 Angle

property Angle:single;

Description:

Set the Angle of TMyShape that you need; it is in rad.

Example:

AShape.Angle:=0.222;

See Also:

Rotate(TMyCAD)

5.4.2 Brush

property Brush:TBrush;

Description:

Set the brush of TMyShape that you need;

Note:
In tcad of xp.b edition,there's a new function that support brush.bitmap to fill a closed shape.

Code example:

```
var
  tmpBitmap:TBitmap;
begin
  tmpBitmap := TBitmap.Create;
  tmpBitmap.LoadFromFile('d:\test\tmp.bmp');
  MyCAD1.GetSelectedShape.Brush.Bitmap :=  tmpBitmap;
  MyCAD1.Repaint;
end;
```

5.4.3 Caption

property Caption:string;

Description:

When a shape be created, it is same as name,you can change it.

Example:

AShape.Caption:='it is a line';

See Also:

CaptionShow

5.4.4 Captionshow

property CaptionShow:boolean;

Description:

Set the hint show or not. the hint is caption.

Example:

AShape.CaptionShow:= true;

See Also:

Caption

5.4.5 ColorBegin

property ColorBegin: TColor;

Description:

Set the begin color if you use the gradient style.

See Also:

Color End
GradientStyle

5.4.6 ColorEnd

property ColorEnd: TColor;

Description

Set the end color if you use the gradient style.

See Also:

ColorBegin
GradientStyle

5.4.7 Font

property Font:TCADFont;

Description:

TCADFont describes font characteristics used when displaying text. TCADFont defines a set of characters by specifying the logheight and logwidth, font family (typeface), attributes (such as bold or italic) and so on.TCADFont encapsulates a windows logical font. it is used for display info field.

See Also:

Info

5.4.8 GradientStyle

property GradientStyle:TGradientStyle;

Description:

To set gradient style

Example:

AShape.GradientStyle:=gsRadialC;

5.4.9 HotShow

property HotShow:Boolean;

Description:

true: the hotspot showed normally
false: the hotspot hide, even a shape be selected.

Example:

Delphi syntax:

```
shape.HotShow := true;
```

C++ syntax:

```
shape->HotShow = true;
```

5.4.10 Info

property Info:string;

Description:

It store string, you can set it as your need.

Example:

AShape.Info:='The is a fun shape'

5.4.11 IsFlipHorz

property IsFlipHorz: boolean;

Description

change the shape flipping state, when it is true, the shape will be flip in horizontally.

See also

TMyCAD.FlipHorz

5.4.12 IsFlipVert

property IsFlipVert: boolean;

Description

change the shape flipping state,when it is true, the shape will be flip in vertically.

See also

TMyCAD.FlipVert

5.4.13 Locked

property Locked:boolean;

Description:

If a shape be locked,the hotspot show in gray color.

5.4.14 LogHeight

property LogHeight: Integer

Description:

Set font height, LogUse should set true.

See Also:

LogUse
LogWidth

5.4.15 LogUse

property LogUse: Boolean

Description:

Set true, LogHeight and LogWidth are valid.

See Also:

LogHeight
LogWidth

5.4.16 LogWidth

property LogWidth: Integer

Description:

Set font width, LogUse should set true.

See Also:

LogHeight
LogUse

5.4.17 MultiInfo

property MultiINfo : TStrings;

Note: if MultiInfo's value is nil,TMyShape's info property is enabled,you can use it.

5.4.18 MultiInfoAlignment

property MultiInfoAlignment:TAlignment;

Description:

Control the MultiInfo's value alignment,it's base the longest string item's width.

Value	**Meaning**
taLeftJustify	Text is left-justified: Lines all begin at the left edge of the control.
taCenter	Text is centered in the control.
taRightJustify	Text is right-justified: Lines all end at the right edge of the control.

5.4.19 Name

property Name: TComponentName;

Specifies the name of the component as referenced in code.

Description:

Use Name to change the name of a shape to reflect its purpose in the current application. By default, the new shape will be named "shape1","shape2", and so on.

Warning:
Changing Name at runtime causes any references to the old name to become undefined. Any subsequent code that uses the old name will cause an exception.

5.4.20 Owner

property Owner : TMyCAD;

Description:
The shape is belonged with which instance of TMyCAD.

5.4.21 Pen

property Pen:TPen

Description:

Set the pen of TMyShape that you need;

5.4.22 PenStyleEndCapType

property PenStyleEndCapType:TPenStyleEndCapType

Description:

Set the pen style type of the line end cap.

Value:
psectROUND
psectSQUARE
psectFLAT
psectMASK

5.4.23 PenStyleJoinType

property PenStyleJoinType:TPenStyleJoinType

Description:

Set the pen style type of the line joining.

Value:
psjtROUND
psjtBEVEL
psjtMITER
psjtMASK

5.4.24 ResizeEnable

property ResizeEnable:boolean;

Description:

Can resizing shape or not, it is very useful for the case of don't allow resize shape

5.4.25 RotateEnable

property RotateEnable:Boolean;

Description:

Can rotating shape or not, it is very useful for the case of don't allow rotate shape.

5.4.26 Tag

property Tag: Longint;

Description

Tag has no predefined meaning. The Tag property is provided for the convenience of developers. It can be used for storing an additional integer value or it can be typecast to any 32-bit value such as a component reference or a pointer.

5.4.27 UserData

property UserData:TUserData;

Description

You can add yourself data , it is very usesful.

Example:

```
UserData.AddKeyandValue('Weight','20kg');
```

See also:

TUserData

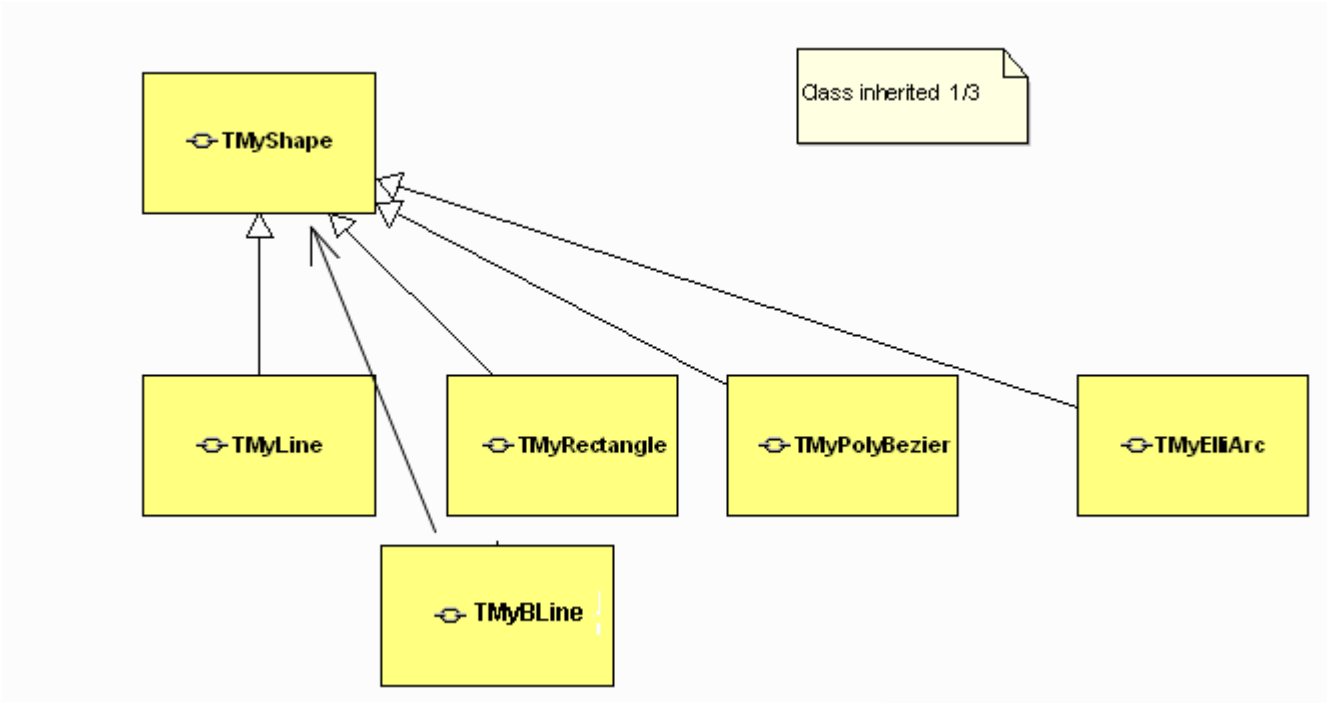
5.4.28 Visible

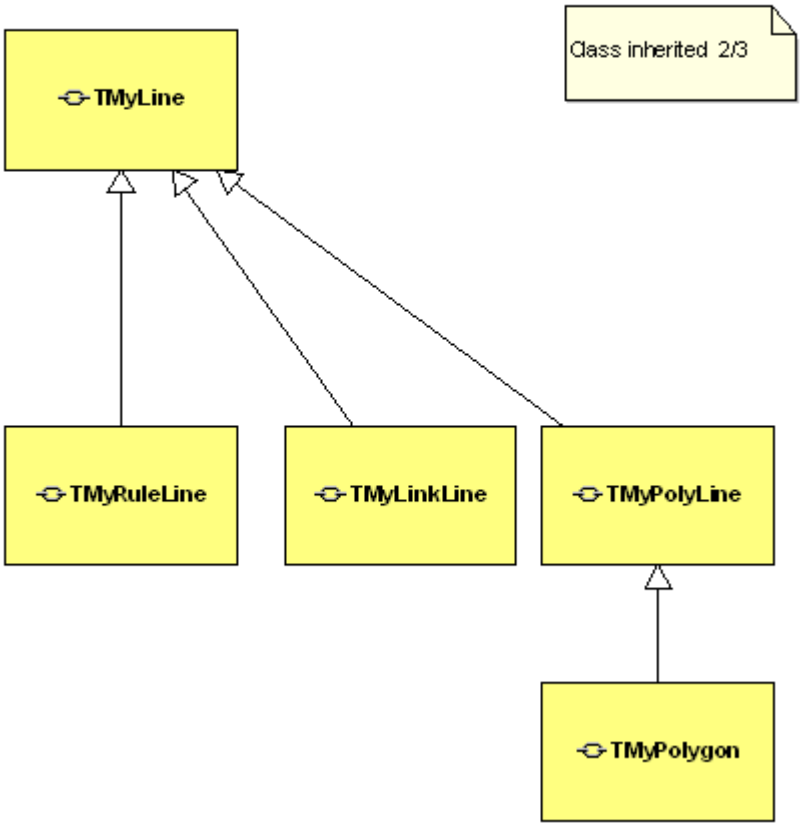
property Visible: boolean;

Description

when it is false,the shape cannot be select,resize and rotate and group.

6. Shape Class Inherited Diagram

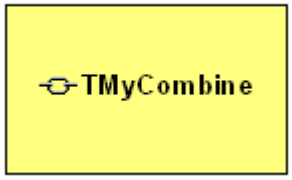




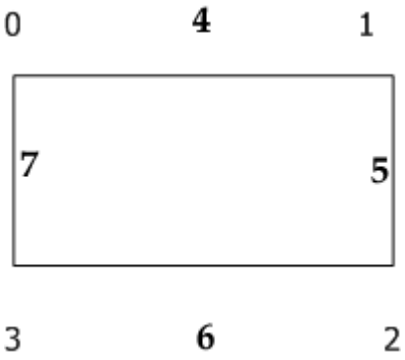
7. TMyCombine

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

7.1 ClassDiagram



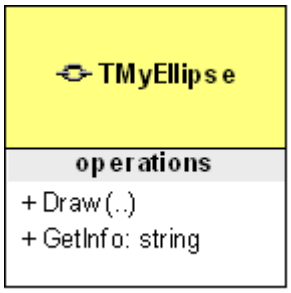
The points Id order:



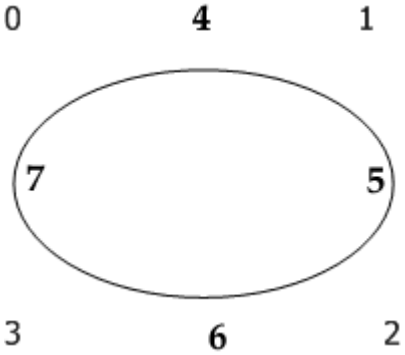
8. TMyEllipse

It is a class of Ellipses(Circle) shape, it defines properties ,events,methods for a Ellipses.

8.1 ClassDiagramofTMyEllipse



The points Id order:



8.2 Methods

8.2.1 Draw

procedure Draw(MyCanvas:TCanvas); override;

Description:

Draw an ellipse shape on the canvas.

8.2.2 GetCenterPoint

function GetCenterPoint: TMyPoint;override

Description:

function GetCenterPoint overrides inherited GetCenterPoint.,It returns the ellipse shape's centerpoint .

See Also:

GetCenterPointInZoom of TMyEllipse

8.2.3 GetCenterPointInZoom

function GetCenterPointInZoom: TMyPoint;override

Description:

function GetCenterPointInZoom overrides inherited GetCenterPointInZoom,It returns the ellipse shape's centerpoint in current zoom.

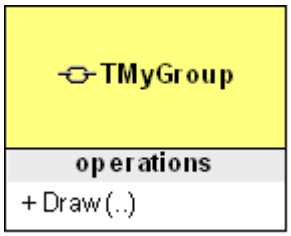
See Also:

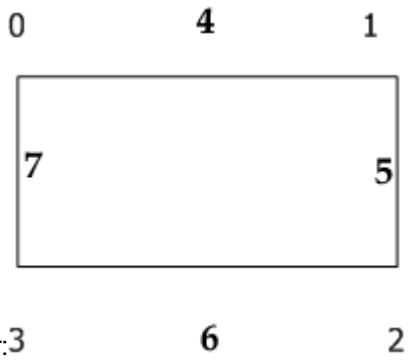
GetCenterPoint of TMyEllipse

9. TMyGroup

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

9.1 ClassDiagram





The points Id order:

9.2 Methods

9.2.1 Draw

procedure Draw(MyCanvas:TCanvas); override;

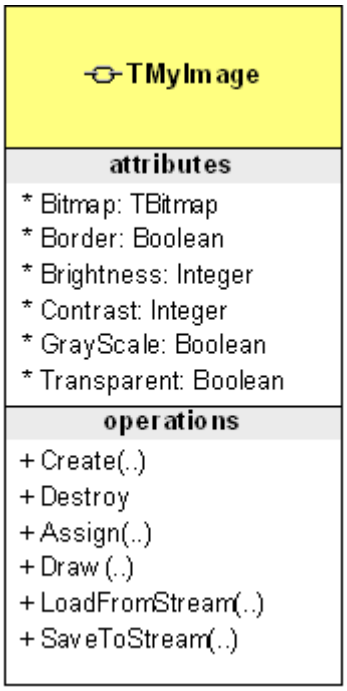
Description:

Draw a group shape on the MyCanvas.This methd calls child shape to draw on the MyCanvas.

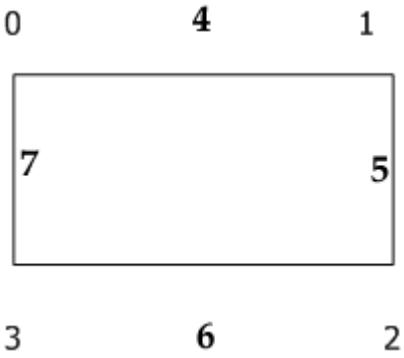
10. TMyImage

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

10.1 Classdiagram



The points Id order:



10.2 Properties

10.2.1 Bitmap

property Bitmap:TBitmap

Description:

Set the bitmap for TMyImage shape, for clear it , Btimap:=nil;

Example:

The example show assign a bitmap that you load to TMyImage.

```
var
   mybitmap:TBitmap;
begin
   if OpenPictureDialog1.Execute then
begin
        myBitmap:=TBitmap.Create;
        mybitmap.LoadFromFile(OpenPictureDialog1.FileName);
        (AShape as TMyImage).Bitmap:=mybitmap;
        MyBitmap.Free;
end;
end;
```

10.2.2 Border

property Border:boolean

Description:

Set the a border around the bitmap ;

Example:

```
AShape.Pen.Width:=2;
AShape.Pen.Color:=clRed;
AShape.Border:=true;
```

10.2.3 Brightness

property Brightness:integer

Description:

Set the the brightness for a bitmap , it is -255<= Brightness<=255;

Example:

```
AShape.Brightness:=45;
```

See also:

Contrast

10.2.4 Contrast

property Constrast:integer;

Description:

Set the the constrast for a bitmap , it is -100<= constrast<=100;

Example:

```
AShape.constrast:=45;
```

See also:

Brightness

10.2.5 Grayscale

property Grayscale:boolean;

Description:

Grayscale a color bitmap.

Example:

```
AShape.Grayscale:=true;
```

See also:

Brightness
Contrast

10.2.6 Transparent

property Transparent:Boolean

Description:

Set the Transparent for bitmap of TMyImage shape.

10.3 Methods

10.3.1 Assign

procedure Assign(Source: TMyShape); override;

Description:

Copy a source image shape properties.

Example:

AShape.Assign(BShape);

10.3.2 Create

constructor Create(AOwner: TMyCAD); override;

Description:

Internal data structure is initialized and Bitmap be created.

10.3.3 Destroy

destructor Destroy; override;

Description:

First all owned fields be released, finally inherited Destroy is called.

See also:

10.3.4 Draw

procedure Draw(MyCanvas:TCanvas); override;

Description:

Draw a image shape on the MyCanvas.

10.3.5 LoadFromStream

procedure LoadFromStream(AStream:TStream); override;

Description:

Load an image shape from a stream.

See also:

SaveToStream

10.3.6 SaveToStream

procedure SaveToStream(AStream:TStream); override;

Description:

It can save a shape to a stream.

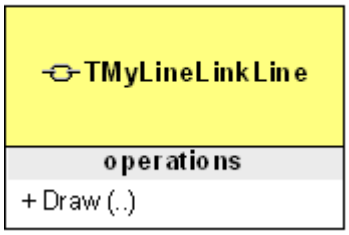
See also:

LoadFromStream

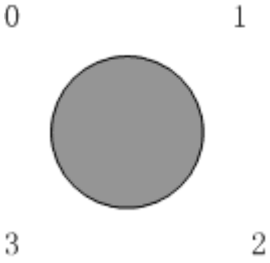
11. TMyLineLinkLine

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

11.1 ClassDiagram



The points Id order:



11.2 Methods

11.2.1 Draw

procedure Draw(MyCanvas:TCanvas); override;

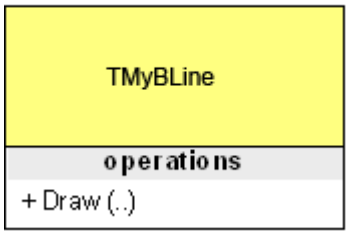
Description:

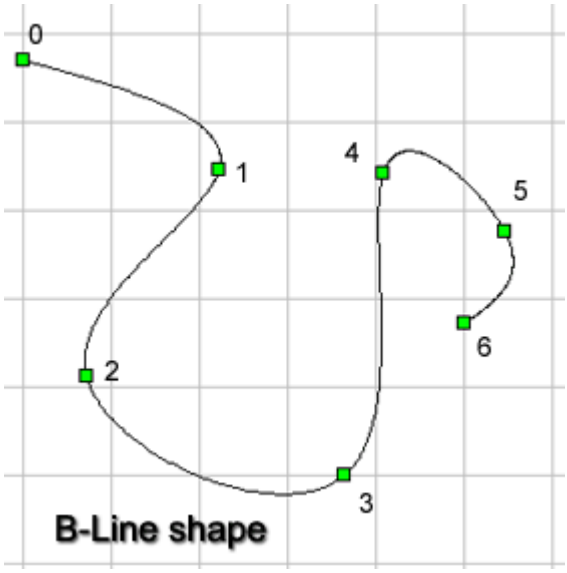
Draw a line link line shape on the MyCanvas.

12. TMyBLine

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

12.1 ClassDiagram





12.2 Methods

12.2.1 Draw

procedure Draw(MyCanvas:TCanvas); override;

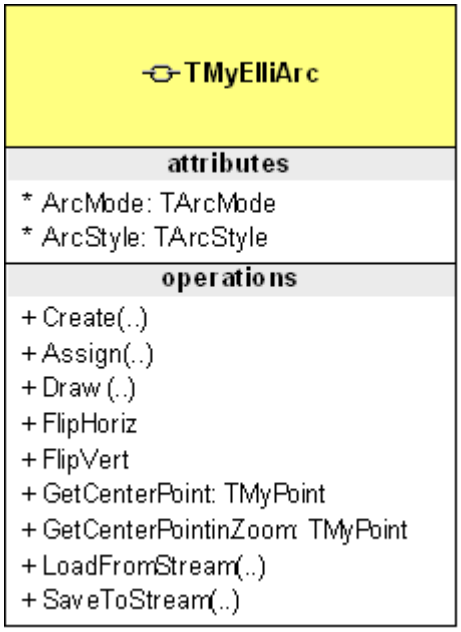
Description:

Draw B-Line shape on the MyCanvas.

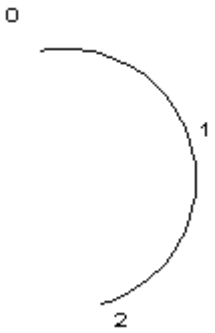
13. TMyElliArc

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

13.1 ClassDiagram



The points Id order:



13.2 Properties

property ArcMode:TArcMode;

Description:

There are two choice for arc mode, amCircle and amEllipse.

amCircle:

amEllipse:

13.2.1 ArcMode

Description:
There are two choice for arc mode, amCircle and amEllipse.
amCircle:

amEllipse:

13.2.2 ArcAtyle

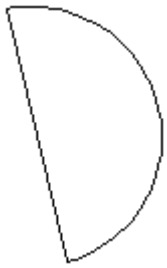
property ArcStyle:TArcStyle;

Description:

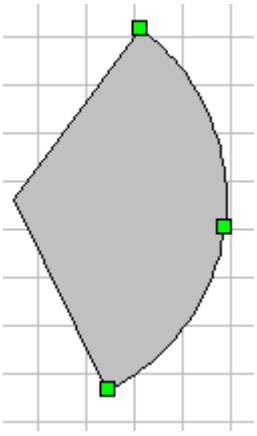
There are 3 choices for this property only when ArcMode is amCircle.

asArc:

asChord:



asSector:



13.3 Methods

13.3.1 Assign

procedure Assign(Source: TMyShape); override;

Description:

Copy a source arc shape properties.

Example:

AShape.Assign(BShape);

13.3.2 Create

constructor Create(AOwner: TMyCAD); override;

Description:

Internal data structure is initialized and ArcStyle is asArc ;ArcMode is amCircle;

13.3.3 Draw

procedure Draw(MyCanvas:TCanvas); override;

Description:

Draw a arc hape on the MyCanvas.

13.3.4 GetCenterPoint

function GetCenterPoint: TMyPoint;override

Description:

function GetCenterPoint overrides inherited GetCenterPoint.,It returns the shape's centerpoint .

See Also:

13.3.5 GetCenterPointInZoom

function GetCenterPointInZoom: TMyPoint;override

Description:

function GetCenterPointInZoom overrides inherited GetCenterPointInZoom,It returns the shape's centerpoint in current zoom.

See Also:

13.3.6 LoadFromStream

procedure LoadFromStream(AStream:TStream); override;

Description:

Load a arc shape from a stream.

See also:

13.3.7 SaveToStream

procedure LoadFromStream(AStream:TStream); override;

Description:

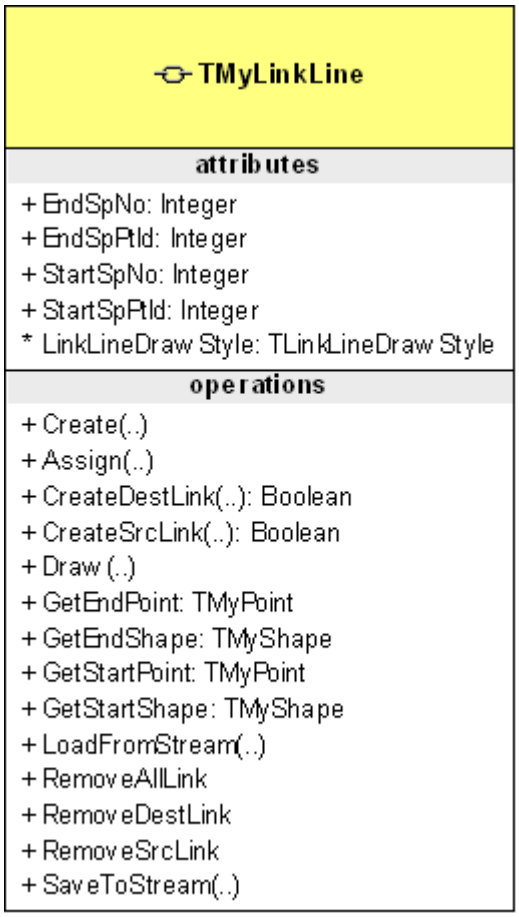
Load a arc shape from a stream.

See also:

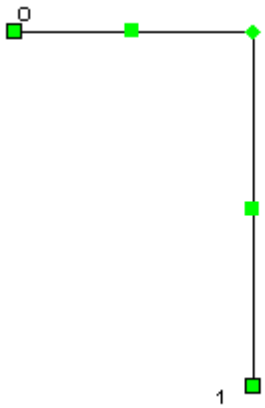
14. TMyLinkLine

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

14.1 ClassDiagram



The points Id order:



14.2 Properties

14.2.1 LinkLineDrawStyle

property LinkLineDrawStyle:TLinkLineDrawStyle

Description:

define linkline draw style.

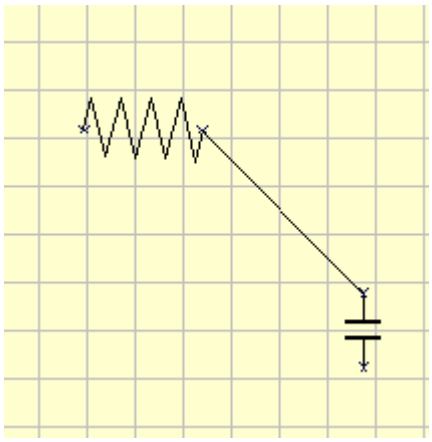
Example:

Delphi syntax:

```
AShape.LinkLineDrawStyle:=lldsFree;
```

C++ syntax:

```
AShape->LinkLineDrawStyle =lldsFree;
```



Delphi syntax:

```
AShape.LinkLineDrawStyle:=lldsHV;
```

C++ syntax:

14.2.2 StartSpPtId

Property StartSpPtId:integer;

Description:
it is read / write, at run time only.The link point id of the start shape.
if StartSpNo is -1,it is must -1.

14.2.3 StartSpNo

Property StartSpNo:integer;

Description:

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

14.2.4 EndSpNo

Property EndSpNo:integer;

Description:

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

14.2.5 EndSpPtId

Property EndSpPtId:integer;

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

14.3 Methods

14.3.1 Create

constructor Create(AOwner: TMyCAD); override;

Description:

Constructor Create overrides the inherited Create. First inherited Create is called, then the internal data structure is initialized.LinkLineDrawStyle is lldsHV.

14.3.2 Assign

procedure Assign(Source: TMyShape); override;

Description:

Copy a source line shape properties.procedure Assign overrides inherited Assign.

Example:

AShape.Assign(BShape);

14.3.3 Draw

procedure Draw(MyCanvas:TCanvas); override;

Description:

Draw on the MyCanvas.

14.3.4 LoadFromStream

procedure LoadFromStream(AStream:TStream); override;

Description:

Load a link line shape from a stream.

14.3.5 CreateDestLink

function CreateDestLink(AShapeId:integer;AShapeLinkPtID:integer): Boolean;

Description:

AShapeId : the end shape id;
AShapeLinkPtID : the link id of the shape,the shape'id is AShapeid.

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

Returns:

true : created success;
false : created failed.

See also:

CreateSrcLink

14.3.6 CreateSrcLink

function CreateSrcLink(AShapeId:integer;AShapeLinkPtID:integer): Boolean;

Description:

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

Parameter:
AShapeId : the start shape id;
AShapeLinkPtID : the link id of the shape,the shape'id is AShapeid.

Returns:
true : created success;

false : created failed.

See also:

CreateDestLink

14.3.7 GetEndPoint

function GetEndPoint: TMyPoint;

Description:

Get the link point in end link shape.

Return(s):
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.

14.3.8 GetEndShape

function GetEndShape: TMyShape;

Description:

Get the end link shape.

Return(s):
nil : there is no end link shape;
else : the instance of the link shape.

14.3.9 GetStartPoint

function GetStartPoint: TMyPoint;

Description:

Get the link point in start link shape.

Return(s):
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.

14.3.10 RemoveDestLink

procedure RemoveDestLink;

Description:

remove the target linked shape.

14.3.11 RemoveSrcLink

procedure RemoveSrcLink;

Description:

Remove the source linked shape.

14.3.12 GetStartShape

function GetStartShape: TMyShape;

Description:

Get the start link shape.

Return(s):
nil : there is no start link shape;
else : the instance of the link shape.

14.3.13 RemoveAllLink

procedure RemoveAllLink;

Description:

remove the all linked shape.

14.3.14 Savetostream

procedure SaveToStream(AStream:TStream); override;

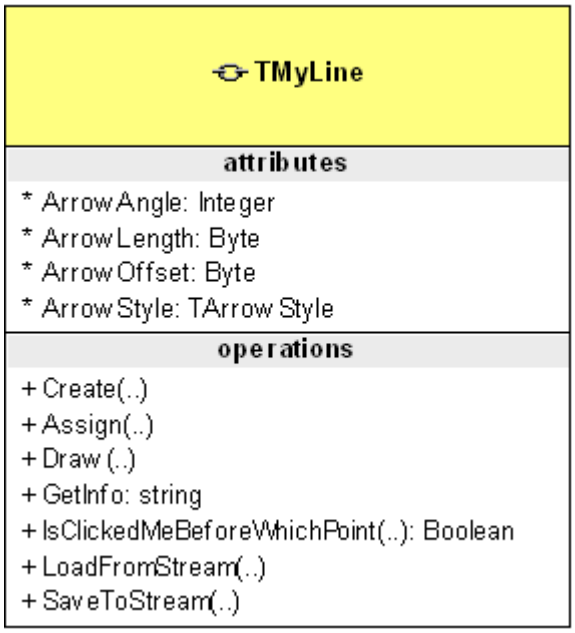
Description:

It can save a link line shape to a stream.

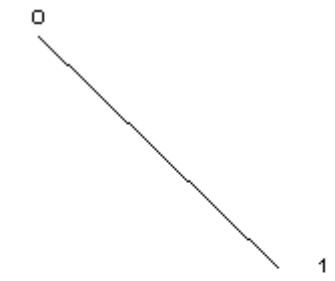
15. TMyLine

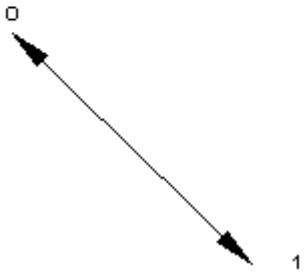
It is a class of line shape, it defines properties ,events,methods.

15.1 ClassDiagram



The points Id order:





15.2 Properties

15.2.1 ArrowAngle

property ArrowAngle:integer

Description:

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

Example:

```
MyCAD1.ArrowAngle := 10 ;
```

15.2.2 ArrowLength

property ArrowLength:Byte

Description:

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

15.2.3 ArrowOffset

property ArrowOffset:byte

Description:

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.

15.2.4 ArrowStyle

property ArrowStyle:**TArrowStyle**

Description:

Set the Line and PolyLine 's arrow style;

ANone: it is a line;
ALeft: one arrow at the left end of the line or polyline;
ARight: one arrow at the right end of the line or polyline;
ADouble: a double-arrow line or polyline

15.3 Methods

15.3.1 Assign

procedure Assign(Source: TMyShape); override;

Description:

Copy a source line shape properties.

Example:

AShape.Assign(BShape);

15.3.2 Create

constructor Create(AOwner: TMyCAD); override;

Description:

Internal data structure is initialized and ArrowOffset, ArrowLength, ArrowAngle, ArrowStyle will be equal TMyCAD's.

15.3.3 Draw

procedure Draw(MyCanvas:TCanvas); override;

Description:

Draw on the MyCanvas.

15.3.4 GetInfo

function GetInfo: string;override

Description:

function GetInfo overrides inherited GetInfo.,It returns the line shape's length.Change the owner's TheUnit, the area will be changed.

See Also:

TheUnit of TMyCAD

15.3.5 LoadFromStream

procedure LoadFromStream(AStream:TStream); override;

Description:

Load a line shape from a stream.

See also:

15.3.6 SaveToStream

procedure SaveToStream(AStream:TStream); override;

Description:

Save a line shape to a stream.

See also:

16. TMyPolyBezier

It is the class of polybezier shape,a continuous bezier shape. it defines properties ,events,methods.

16.1 Methods

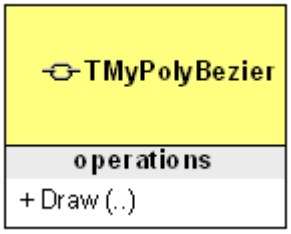
16.1.1 Draw

procedure Draw(MyCanvas:TCanvas); override;

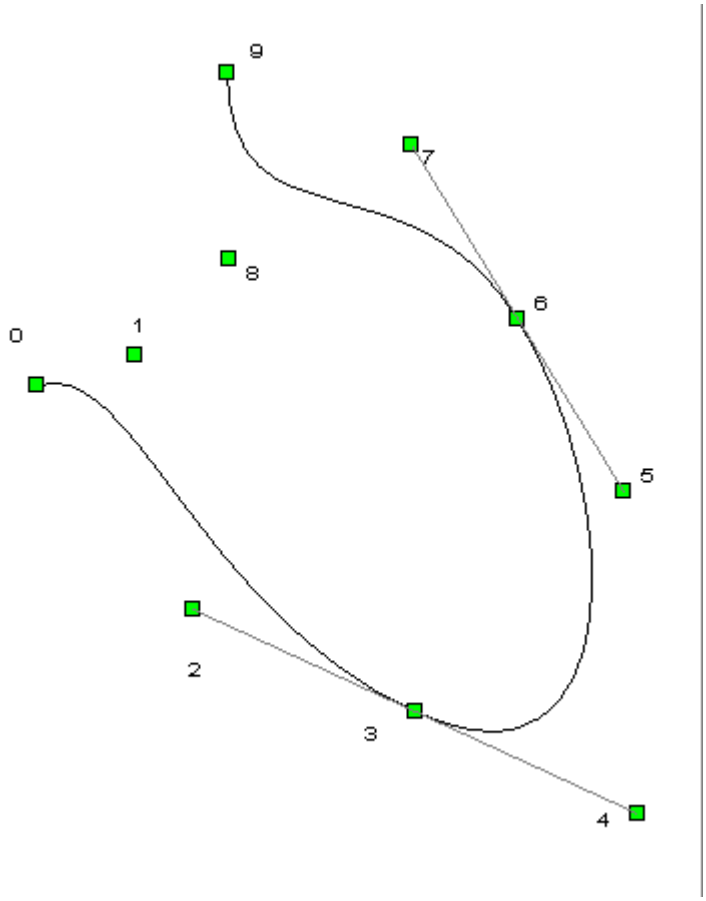
Description:

Draw on the MyCanvas.

16.2 ClassDiagram



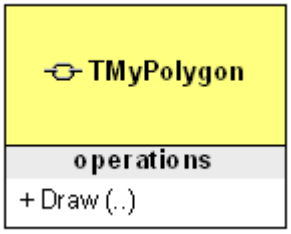
The points Id order like TMyLine.


9
7
8
6
0
1
5
2
3
4

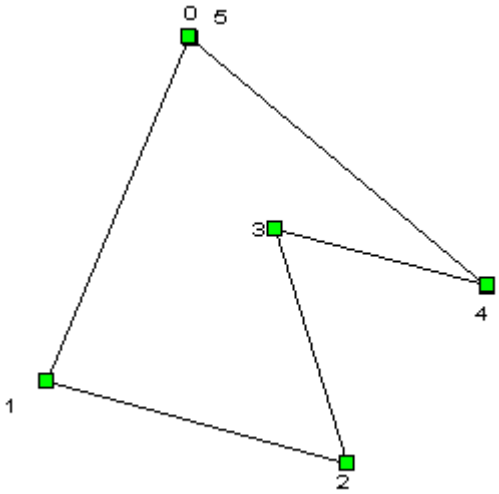
17. TMyPolygon

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

17.1 ClassDiagram



The points Id order like TMyPolyline, and it has more than two point.



17.2 Methods

17.2.1 Draw

procedure Draw(MyCanvas:TCanvas); override;

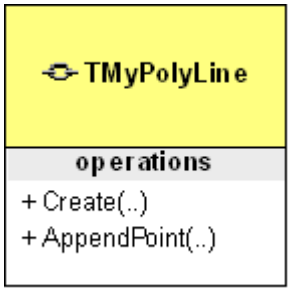
Description:

Draw on the MyCanvas.

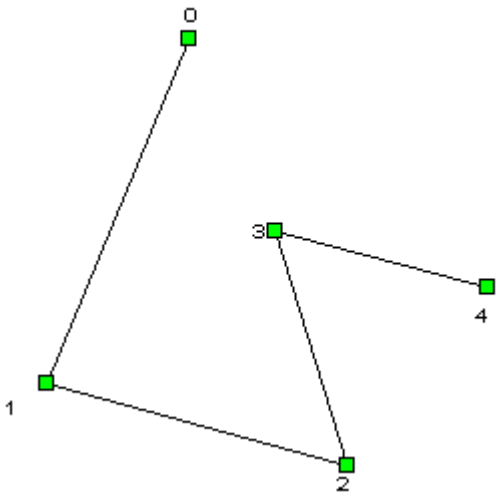
18. TMyPolyLine

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

18.1 ClassDiagram



The points Id order like TMyLine.



18.2 Methods

18.2.1 Create

constructor Create(AOwner: TMyCAD); override;

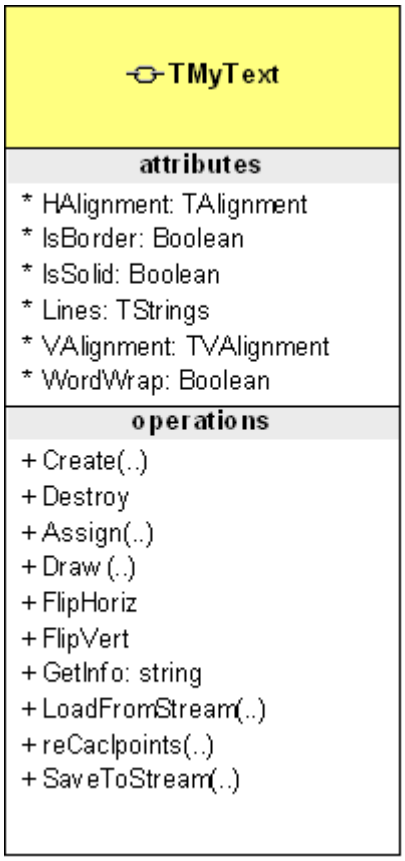
Description:

Internal data structure is initialized and ArrowStyle default is asNone.

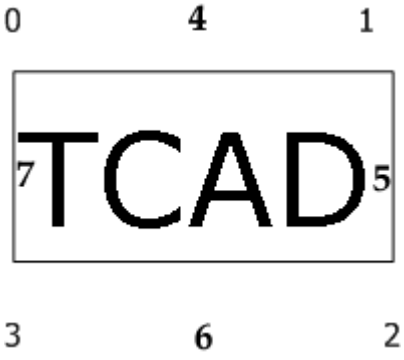
19. TMyText

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

19.1 ClassDiagram



The points Id order:



19.2 Properties

19.2.1 HAlignment

property HAlignment:TAlignment

Description:

Determines how the text is horizontal aligned within the outer rectangle.

Use Alignment to change the way the text is formatted by the TMyText control. Alignment can take one of the following values:

Value	Meaning
taLeftJustify	Text is left-justified: Lines all begin at the left edge of the control.
taCenter	Text is centered in the control.
taRightJustify	Text is right-justified: Lines all end at the right edge of the control.

See also:

VAlignment

19.2.2 IsBorder

property IsBorder:Boolean;

Description:

When it is true,there is a rectangle around the text.

19.2.3 IsSolid

property IsSolid:Boolean;

Description:

when it is true,there is a solid text.

19.2.4 Lines

property Lines: TStrings

Description

Use Lines to manipulate text in an outer rectangle on a line-by-line basis. Lines is a TStrings object, so the TStrings methods may be used for Lines to perform manipulations such as counting the lines of text, adding new lines, deleting lines, or replacing lines with new text.

Note: if Lines' value is nil,TMyText's info property is enabled,you can use it.

19.2.5 NestingColor

property NestingColor: Boolean

Description:

Set the text's nesting color.

See also:

NestingText

19.2.6 NestingText

property NestingText: Boolean;

Description:

Set the text's nesting color, when you set true, draw text use nesting color.

See also:

NestingColor

19.2.7 VAlignment

property VAlignment:TVAlignment;

Description:

Determines how the text is vertical aligned within the outer rectangle.

Use VAlignment to change the way the text is formatted by the TMyText control. VAlignment can take one of the following values:

Value	**Meaning**
vaTop	Text is top: Lines all begin at the top edge of the control.
vaMiddle	Text is centered between top and bottom in the control.
vaBottom	Text is bottom: Lines all end at the bottom edge of the control.

See also:

HAlignment

19.2.8 WordWrap

property WordWrap:Boolean;

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.

19.3 Methods

19.3.1 Assign

procedure Assign(Source: TMyShape); override;

Description:

Copy a source text shape properties.

Example:

AShape.Assign(BShape);

19.3.2 SaveToStream

Description:

Copy a source text shape properties.

Example:

AShape.Assign(BShape);

19.3.3 LoadFromStream

procedure LoadFromStream(AStream:TStream); override;

Description:

Load a text shape from a stream.

See also:

SaveToStream

19.3.4 Create

constructor Create(AOwner: TMyCAD); override;

Description:

Internal data structure is initialized and IsBorder is false,IsSolid is true.

See also:

Destroy

19.3.5 Destroy

destructor Destroy; override;

Description:

First all owned fields be released, finally inherited Destroy is called.

See also:

Create

19.3.6 Draw

procedure Draw(MyCanvas:TCanvas); override;

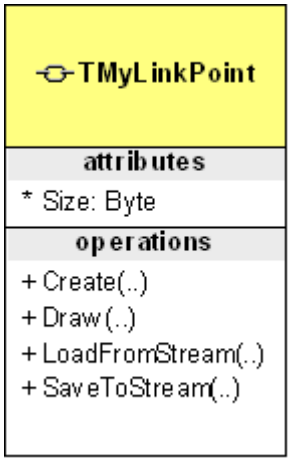
Description:

Draw a text shape on the MyCanvas.

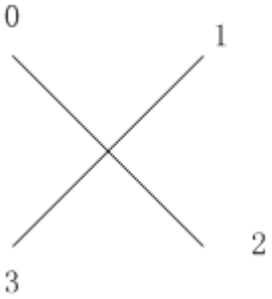
20. TMyLinkPoint

It is a class of linkpoint shape, it defines properties ,events,methods ,it used by Library Manger Tools ONLY.

20.1 ClassDiagram



The points Id order:



20.2 Properties

20.2.1 Size

property Size:byte

Description:

set the size **.**

20.3 Methods

20.3.1 SaveToStream

procedure SaveToStream(AStream:TStream); override;

Description:

Save a linkpoint shape to a stream.

See also:

20.3.2 LoadFromStream

procedure LoadFromStream(AStream:TStream); override;

Description:

Load a linkpoint shape from a stream.

See also:

20.3.3 Create

constructor Create(AOwner: TMyCAD); override;

Description:

Internal data structure is initialized and size is 8 pixel.

20.3.4 Draw

procedure Draw(MyCanvas:TCanvas); override;

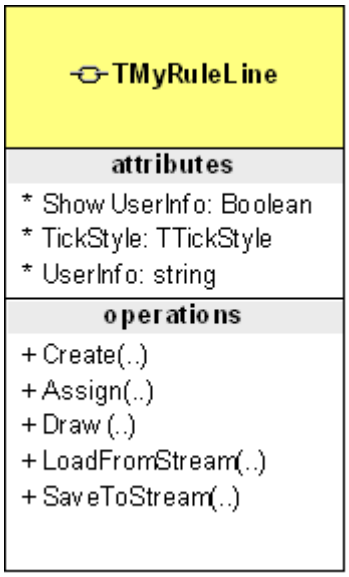
Description:

Draw a linkpoint shape on the MyCanvas.

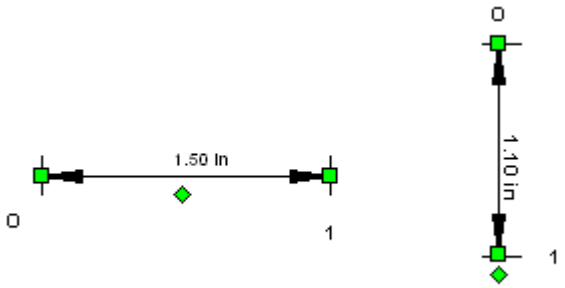
21. TMyRuleLine

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

21.1 ClassDiagram



The points Id order:



21.2 Properties

21.2.1 UserInfo

property UserInfo:string;

Description:

set the userinfo string.

See also:

ShowUserInfo

21.2.2 Showuserinfo

property ShowUserInfo:boolean;

Description:

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

21.2.3 TickStyle

property TickStyle:TTickStyle;

Description:

TTickStyle=(tsLine,tsNone);

When it is tsLine, the TMyRuleline has a two small line beside two end point;
when it is tsNone, the TMyRuleline has no small line beside two end point;

21.3 Methods

21.3.1 Assign

procedure Assign(Source: TMyShape); override;

Description:

Copy a source TMyRuleLine shape properties.

Example:

AShape.Assign(BShape);

21.3.2 Create

constructor Create(AOwner: TMyCAD); override;

Description:

Internal data structure is initialized and ArrowStyle is ADouble, UserInfo is '', ShowUserInfo is False, TickStyle is tsLine;

21.3.3 LoadFromStream

procedure LoadFromStream(AStream:TStream); override;

Description:

Load a rule line shape from a stream.

See also:

21.3.4 Draw

procedure Draw(MyCanvas:TCanvas); override;

Description:

Draw on the MyCanvas.

21.3.5 SaveToStream

procedure SaveToStream(AStream:TStream); override;

Description:

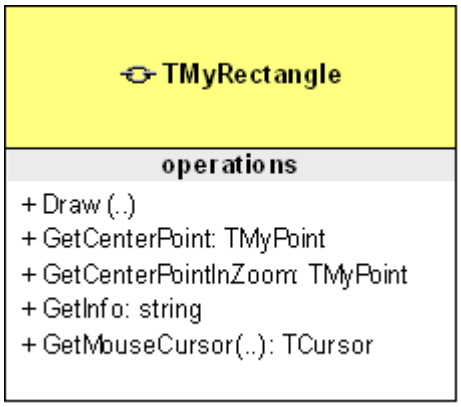
Save a rule line shape to a stream.

See also:

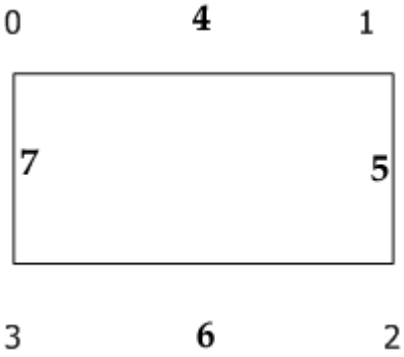
22. TMyRectangle

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

22.1 ClassDiagram



The points Id order:



22.2 Properties

property ShowSideHot:Boolean

Description:

This property specify the showing the side hot of a TMyRectangle object.

Example:

```
MyCAD1.ShowSideHot:=true;
```

22.2.1 ShowSideHot

property ShowSideHot:Boolean

Description:

This property specify wether the side hot suare of a TMyRectangle show or not.

Example:

```
MyCAD1.ShowSideHot:=true;
```

22.2.2 AssociateSideResizing

property AssociateSideResizing: Boolean

Description:

This property specify the relationship when resizing the side of shape.

Example:

```
MyCAD1.AssociateSideResizing:=true;
```

22.3 Methods

22.3.1 Draw

procedure Draw(MyCanvas:TCanvas); override;

Description:

Draw a rectangle shape on the MyCanvas.

22.3.2 GetCenterPoint

function GetCenterPoint: TMyPoint;override

Description:

function GetCenterPoint overrides inherited GetCenterPoint.,It returns the rectangle shape's centerpoint .

See Also:

GetCenterPointInZoom of TMyRectangle

22.3.3 GetCenterPointInZoom

function GetCenterPointInZoom: TMyPoint;override

Description:

function GetCenterPointInZoom overrides inherited GetCenterPointInZoom,It returns the rectangle shape's

centerpoint in current zoom.

See Also:

GetCenterPoint of TMyRectangle

22.3.4 GetInfo

function GetInfo: string;override

Description:

function GetInfo overrides inherited GetInfo.,It returns the rectangle shape's area.Change the owner's TheUnit, and Ratio the area will be changed.

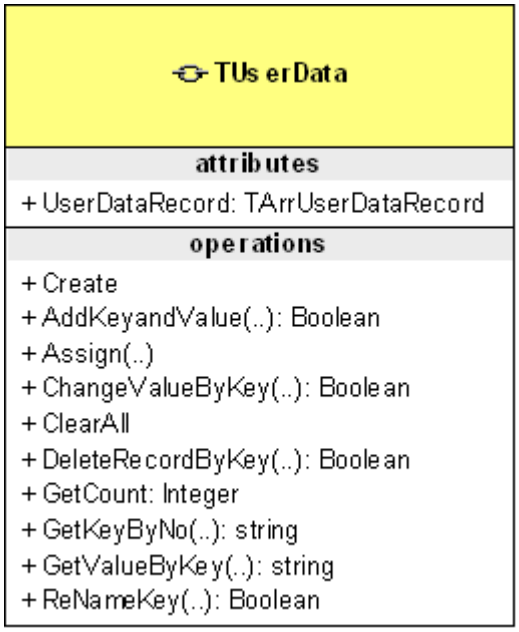
See Also:

TheUnit of TMyCAD

23. TUserData

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

23.1 ClassDiagram



23.2 Properties

23.2.1 UserDataRecords

property UserDataRecords:TArrUserDataRecord;

Description:

Store userdefine properties

See also:

Defines

23.3 Methods

23.3.1 Create

constructor Create;

Description:

Internal data structure is initialized ;

23.3.2 AddKeyAndValue

funcrion AddKeyAndValue(Const AKey,AValue:string):boolean;

Description:

Add key and value to the TUserData instance;

Parameter:

AKey: a key for the user define property, it is a string;
AValue: the value, it is a string;

Returns:

true: add success;
false: add failed, if the key is existed, will return false;

Examples:

```
UserData1.AddKeyandValue('Name','John');
```

23.3.3 Assign

procedure Assign(Source:TUserData)

Description:

Copy a instance of TUserData.

Parameter:

Source: other exist instance of TUserData

Examples:

```
UserData1.Assign(UserData0);
//Userdata1's data will be lost and rewrite
```

23.3.4 ChangeValueByKey

function ChangeValueByKey(const AKey:string;AValue:string):boolean;

Description:

Change the value by key. if AKey do not match , it return false else return true;

Parameter:

AKey: exist key;
AValue: new value ;

Examples:

```
If  UserData1.ChangeValueByKey('Name','Rose') THEN
 ShowMessage('Changed!');
```

23.3.5 ClearAll

procedure ClearAll;

Description:

Clear all data;

23.3.6 DeleteRecordByKey

function DeleteRecordByKey(AKey:string): Boolean

Description:

Delete a record by key;

Parameter:
AKey: the exist key;

Returns:
true: delete success;
false:delete failed, a key cannot be found,

23.3.7 GetCount

function GetCount: Integer;

Description:

Get how many records existed.

23.3.8 GetKeyByNo

function GetKeyByNo(const ANo:integer): string;

Description:

Get the key string by no. if no do not match , it return false else return true;

Parameter:

ANo: the record id in the records, it is from zero.

Examples:

```
ShowMessage('the first key name is: ' + UserData1.GetKeyByNo(0));
```

23.3.9 GetValueByKey

function GetValueByKey(const AKey:string): string;

Description:

Get the value by key. if no do not match , it returns empty string.

Parameter:

AKey: the key string in the records.

Examples:

```
ShowMessage('the first key name is: ' + UserData1.GetKeyByNo(0)+' Value:
'+UserData1.GetValueByKey(UserData1.GetKeyByNo(0) ));
```

23.3.10 ReNameKey

function ReNameKey(OldKey,NewKey:string):boolean;

Description:

Rename a key name. if no do not match , it return false else return true;

Parameter:

AKey: the key string in the records.

24. TCAD File Format

The TCAD file format of TCAD VCL Component :

name	length	note
The Lenght of ShapeName	4	
ShapeName	not fixed	
ShapeId	4	
layerId	4	
parentshape	4	
ChildShapesCount	4	
ChildShapes	4*Count	
LinkShapesCount	4	
LinkShapes	4*Count	
LinkShapesCount	4	
ThePointsCount	4	
ThePoints[?].x	10	
ThePoints[?].y	10	
LinkShapesCount	4	
ThePointsCount	4	
LinkPoints[?].x	10	
LinkPoints[?].y	10	
TextoutPoint.x	10	
TextoutPoint.y	10	
Tag	4	
Angle	10	
Brush.Bitmap	not fixed	
Brush.Color	4	
Brush.Style	1	
CaptionLength	4	
Caption	not fixed	
Begincolor	4	
EndColor	4	
Font name length	4	
Font name	not fixed	
Font charset	1	
Font Color	1	
Font Height	4	
Font Size	4	
Font.Pitch	4	
FFont.PixelsPerInch	4	
Font Style	1	
Font.LogWidth	1	
Font.LogHeight	1	
GradientStyle	1	
HotSize	1	
HotSpot display	1	
Info length	4	
Info	not fixed	
Pen.Color	4	
Pen.Mode	1	
Pen.style	1	
Pen.width	4	

25. About us

HongDi science technology development co.,ltd. provide you with useful components and we hope makes it easier for you to create great application with us. we have been serving the image and graphic components since 1998.

Our website:
You can get all information of TCAD graphics component from our web.

http://www.codeidea.com

E-Mail:

hongbin.fei
webmaster@codeidea.com

yuefen.yao
support@codeidea.com

Telephone:

+86 572 2607144(Phone)
+86 (0)13511221372 (Mobile)

Address:

Room 1113 Building B JinMao Square
No.251 QiLiTing Road
HuZhou, Zhejiang 313000
China

Index

A

B

C

G

H

I

L

M

R

S

U

V

W

X

Z