

TCAD 2002

for Borland Delphi &C++ Builder



Help handbook

1998-2002 True Image Component Company

published on 2002.10.8

Introduction

What's TCAD2002

by hongbin.fei

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 , controled by mouse or code!

Now, You can using TCAD to help you writing application. You can create CAD application fast and easy!

TCAD2002 for Delphi & C++ Builder

1998-2002 True Image Component Company

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: Aug,2002 in China

Publisher

hongbin.fei

Technical Editors

hongbin.fei

wei.li

Cover Designer

yuefen.yao

Team Coordinator

hongbin.fei

Production

hongbin.fei

Special thanks to:

All the people who contributed to this document, to mum and dad , my mothers in law, to our secretary Rain, to the graphic artist who created this great product logo on the cover page ,to the copy shop where this document will be duplicated, and and and...

Table of Contents

Foreword	1
Part I Introduction	3
1 What is TCAD2002.6	3
Part II Defines	5
Part III Properties	7
1 ArrowAngle	8
2 ArrowCount	8
3 ArrowLength	8
4 BkBitmapMode	8
5 Brush	9
6 ColorOfBackground	9
7 ColorOfHot	10
8 CurrentLayerId	10
9 Font	10
10 GridHeight	11
11 GridShow	11
12 GridWidth	11
13 HintColor	12
14 HintShow	12
15 LabelXY	12
16 LableValue	13
17 Locked	13
18 PageHeight	13
19 PageOrientation	14
20 PageStyle	14
21 PageWidth	14
22 Pen	15
23 PrintABorder	15
24 PrintBackground	15
25 ReturnToSelecting	15
26 ShapeTool	16
27 Snap	16
28 TheUnit	17
29 Ratio	17
30 SnapPixels	17
31 Zoom	18
Part IV Methods	20

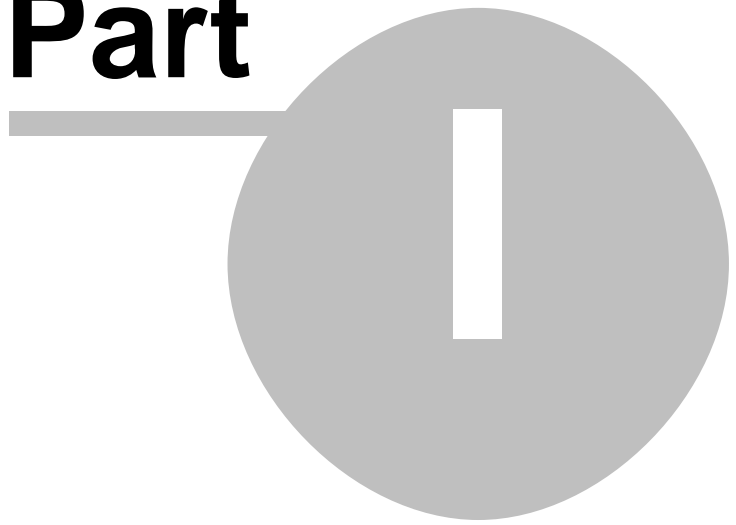
1	AddImageShapeByCode	21
2	AddShapeByCode	22
3	AlignBottom	22
4	AlignLeft	22
5	AlignRight	23
6	AlignTop	23
7	BringToFront	23
8	ChangeAllTextFontSize	24
9	Create	24
10	DeleteAllLayers	24
11	DeleteAllShapes	25
12	DeleteLayerByID	25
13	DeleteLayerByName	25
14	DeleteSelectedShape	25
15	DeleteShapeByID	26
16	Destroy	26
17	DrawAllShape	26
18	GetLayerIdByName	27
19	GetLayerIdByNo	27
20	GetLayerNameByID	28
21	GetLayerNoById	28
22	GetLayerNoByName	28
23	GetLayersCount	29
24	GetMaxLayerId	29
25	GetSelectedShapeID	29
26	GetShapeByName	30
27	GetShapesCount	30
28	GetShapesCountInALayer	30
29	GroupWorkingShape	31
30	InVisibleLayerById	31
31	InVisibleLayerByName	31
32	IsTCADFile	32
33	IsVisibleLayerById	32
34	LoadFromFile	33
35	LoadFromStream	33
36	LockShape	33
37	Move	34
38	NewLayer	34
39	Rotate	34
40	Print	35
41	PrintLayers	35
42	PrintPreview	35

43	PrintPreviewLayers	36
44	RenameShapename	36
45	SaveToFile	37
46	SaveToStream	37
47	SendtoBack	37
48	SetBkBitmap	38
49	SetCurrentLayerId	38
50	SetLayerNameById	39
51	SetLayerNameByName	39
52	SetMyImage	39
53	UnGroupShape	40
54	UnLockShape	40
55	VisibleAllLayer	40
56	VisibleLayerByID	41
57	VisibleLayerByName	41
Part V Events		43
1	OnActionToolToSelecting	43
2	OnDeleteLayer	43
3	OnMouseEnter	44
4	OnMouseEnterShape	44
5	OnMouseLeave	44
6	OnMouseLeaveShape	45
7	OnNewLayer	45
8	OnShapeAdded	45
9	OnShapeDragging	46
10	OnShapeResizing	46
11	OnShapeSelected	46
Part VI About Crystal Component		49
Index		50

I



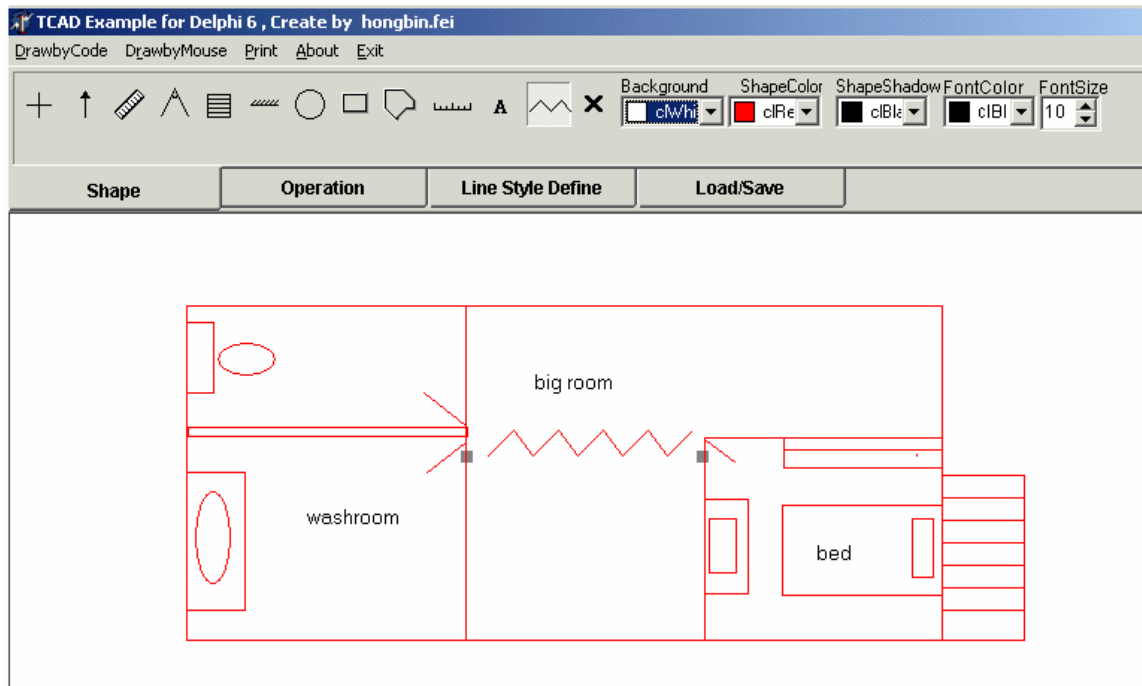
Part



1 Introduction

1.1 What is TCAD2002.6

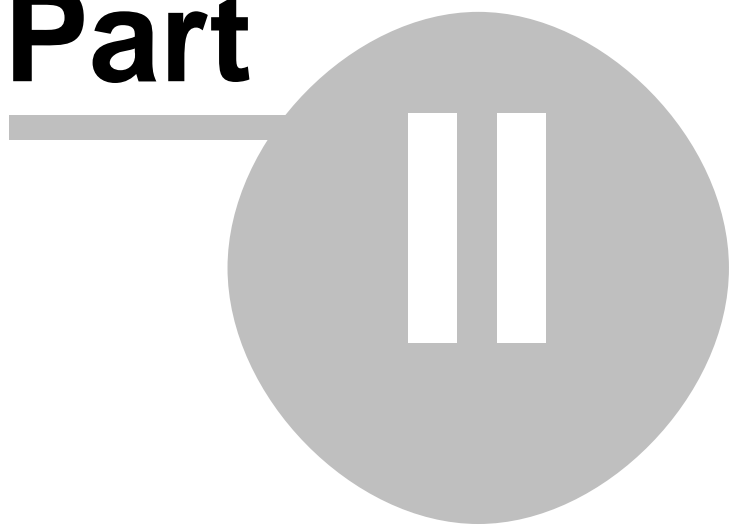
It is a component of Delphi&C++Builder for vector application developing. it can save your times.



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.

- Easy to create and use VECTOR shape in your application developing ,only controled by mouse;
- Draw,Drag,Resize,Rotate the Shapes
- Can delete a shape or all shapes
- Allow save to file or database field, easy to recall it.
- Can add shape by code
- Support printview and print.
- Select or customer define page(or canvas) size .
- Show grid , set grid width or height
- Snap to grid , easy to draw shape
- Group/Ungroup shapes
- WhenPress [Ctrl],there is a square or circle ,horz./vert. line appear.
- Support layer,visible or invisible a layer, print one or more layer,delete layer
- Can set the ratio.
- Can Lock/Unlock Shape
- Can show hint(Shapename) when mouse enter a shape.

Part



2 Defines

TDrawTool = (SpSelecting, SpLine, SpRuleLine, SpSector, SpPolyLine,
SpPolygon,SpRectangle,SpEllipse, SpWallLine,
SpWaveLine, SpWallRect, SpText ,SpArc,SpImage);

TUnits = (pixel, mm, inch);

TPageStyle =(A3,A4,A5,B3,B4,B5,CustomerPage);

TCADFont = class(TFont)

property LogHeight:integer

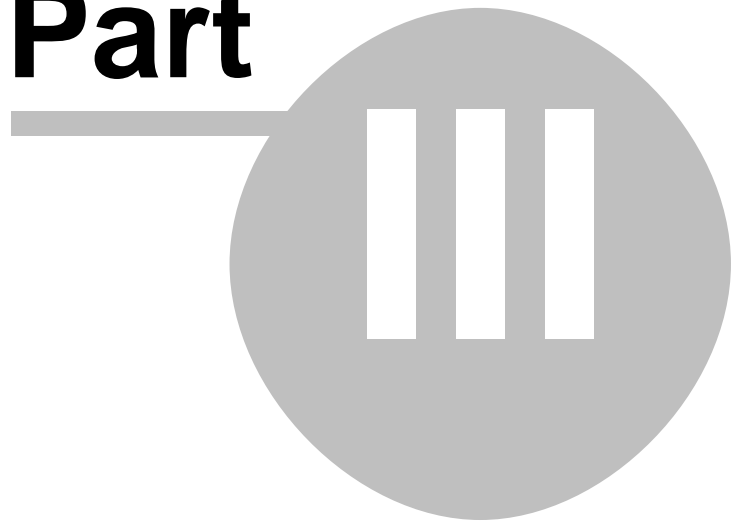
property LogWidth:integer

property BkColor:TColor

end;

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

Part



3 Properties

property Enabled;
property Visible;

property [CurrentLayerId](#): integer;
property [ReturnToSelecting](#): Boolean

property [LabelXY](#): TLabel
property [LabelValue](#): TLabel
property [ArrowAngle](#): double
property [ArrowCount](#): integer
property [ArrowLength](#): byte
property [BkBitmapMode](#): TBkBitmapMode
property [ShapeTool](#): TDrawTool
property [ColorOfBackGround](#): TColor
property [ColorOfHot](#): TColor
property [Font](#): TCADFont
property [Pen](#): TPen
property [Brush](#): TBrush
property [Zoom](#): Double
property [TheUNIT](#): TUnits
property [Ratio](#): double

property [HintShow](#): Boolean;
property [HintColor](#): TColor;

property [GridWidth](#): integer
property [GridHeight](#): integer
property [GridShow](#): Boolean
property [Snap](#): Boolean
property [SnapPixels](#): byte

property [PageWidth](#): Cardinal
property [PageHeight](#): Cardinal
property [PageStyle](#): TPageStyle
property [PageOrientation](#): TPrinterOrientation

property [PrintBackground](#): Boolean
property [PrintABorder](#): Boolean

property [Locked](#): Boolean

3.1 ArrowAngle

property ArrowAngle:double

Description:

Set the Line and PolyLine 's arrow angle.

value is between: 0 - 0.5

3.2 ArrowCount

property ArrowCount:integer

Description:

Set the Line and PolyLine 's arrow count;

value is : 0,1,2

ArrowCount is 0 , the line or polyline have no arrow
ArrowCount is 1 , one arrow at the end of the linear polyline;
ArrowCount is 2 , a double-arrow line or polyline

3.3 ArrowLength

property ArrowLength:Byte

Description:

Set the Line and PolyLine 's arrow Length.

value is between: 10-50

3.4 BkBitmapMode

Description:

Set the background bitmap show style.

Tiled : if the bitmap size low than TCAD's size , two or more this bitmap drawn on TCAD

Stretch:Bitmap be stretch to fit at TCAD

Center: Bitmap is in center of TCAD

LeftTop:Bitmap be showed from TCAD's Left,Top

CADSizeAccordingMe:The TCAD 's size according background bitmap file.

Q: How to load a background image, then size the tcad component to the exact size of the image.

A:Set BkBitmapMode = CADSizeAccordingMe , then Load the bitmap .

See Also:

[TBkBitmapMode](#) in defines

3.5 Brush

property Brush:TBrush;

Description:

Set the Brush that you need;

3.6 ColorOfBackground

property ColorOfBackground:TColor

Description:

The color of Canvas.

Example:

```
CAD1.Shadow:=true;  
CAD1.ColorOfBackground:=clWhite;
```

See Also:

3.7 ColorOfHot

property ColorOfHot:TCColor

Description:

The color of hot square;

Example:

```
CAD1.ColorofHot:=clWhite;
```

See also:

3.8 CurrentLayerId

CurrentLayerId :integer;

Description:

Get the current layer's ID. New shape would belong to this layer.

Example:

```
ShowMessage(IntToStr('You are draw on layerID:'+MyCAD1.CurrentLayerID));
```

See also:

[i_etCurrentLayerID](#)

3.9 Font

property Font:TCADFont;

Description:

Set the [spText](#)'s Font,it is differ with TFont, added some properties.

Example:

```
Mycad1.Font.LogHeight:=SpinEdit2.Value;  
Mycad1.Font.LogWidth:=SpinEdit2.Value;
```

See also:

3.10 GridHeight

property GridHeight:integer;

Description:

The height of grid , it is in pixels

Example:

```
CAD1.GridHeight:=12;
```

See also:

[GridWidth](#)

3.11 GridShow

property GridShow:Boolean;

Description:

The grid is show or not.

Example:

```
CAD1.GridShow := false;
```

See also:

[GridWidth](#)

[GridHeight](#)

3.12 GridWidth

property GridWidth:integer;

Description:

The width of grid , the unit is in pixels

Example:

```
CAD1.GridWidth:=12;
```

See also:

[GridHeight](#)

3.13 HintColor

property HintColor:TColor

Description:

The color of Hint window's background.

Example:

```
CAD1.HitColor:=clWhite;
```

See Also:

3.14 HintShow

property HintShow:Boolean;

Description:

When mouse move over the shape or the grouped shape, show the shape's name in a popped window.

See Also:

RenameShapeName

3.15 LabelXY

property LabelXY:TLabel

Description:

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

Example:

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

See also:

3.16 LableValue

property LabelValue:TLabel

Description:

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

Example:

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

See also:

[LabelXY](#)

3.17 Locked

property Locked : Boolean

Description:

Whether a shape or grouped shape can be moved or renamed. it readonly.

See also:

[UnlockShape](#)

[LockeShape](#)

3.18 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](#)

3.19 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](#)

3.20 PageStyle

property PageStyle: TPageStyle;

Description:

Set the Canvas(Page)'s style,(A3,A4,A5,B3,B4,B5,CustomerPage), then PageWidth and PageHeight will be changed.

Example:

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

See also:

[Defines](#)

3.21 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:

[PageStyle](#)
[PageHeight](#)

3.22 Pen

property Pen:TPen

Description:

Set the pen that you need;

3.23 PrintABorder

property PrintABorder:Boolean ;

Description:

Is border showed or printed when printpreview or print to printer;

3.24 PrintBackground

property PrintBackground:Boolean ;

Description:

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

Example:

MyCAD1.PrintBackground:=false;

3.25 ReturnToSelecting

property ReturnToSelecting: **Boolean**

Description:

if it is true, mean The ShapeTool will return to [selecting] status automatically after you drew a shape, trigger a event OnActionToolToSelecting, else, ReturnToSelecting is false , the ShapeTool still on draw status,

Example:

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

You can draw same shapes consecutively.

See Also:

[OnActionToolToSelecting](#)

3.26 ShapeTool

property ShapeTool:(SpSelecting, SpLine,SpRuleLine,SpSector, SpPolyLine, SpPolygon,SpRectangle, SpEllipse,SpWallLine,SpWallRect,SpText,SpArc,Splmage); it is enum type.

Description:

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

Example:

```
CAD1.ShapeTool:=SpLine;  
{  
  // Drawing line shape in TCAD  
}
```

```
CAD1.ShapeTool=SpSelecting;
```

See also:

[Defines](#)

3.27 Snap

property Snap:Boolean ;

Description:

Snap to grid or not.

Example:

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

3.28 TheUnit

property TheUnit:(mm,inch,Pixel)

Description:

Which unit do you use, It will be appear at property [LabelValue](#), that is Line's Length and area for TMyLine,TMyRectangle,TMyEllisple.

Example:

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

3.29 Ratio

property Ratio:double

Description:

Define the ratio , it is very useful,It will be appear at property [LabelValue](#), that is Line's Length and area for TMyLine,TMyRectangle,TMyEllisple.

Example:

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

3.30 SnapPixels

property SnapPixles:Byte

Description:

The pixels between mouse point and nearest grid .It can not big than gridwidth or gridheight.

Example:

```
MyCAD1.SnapPixels := 3 ;
```

3.31 Zoom

property Zoom:Double

Description:

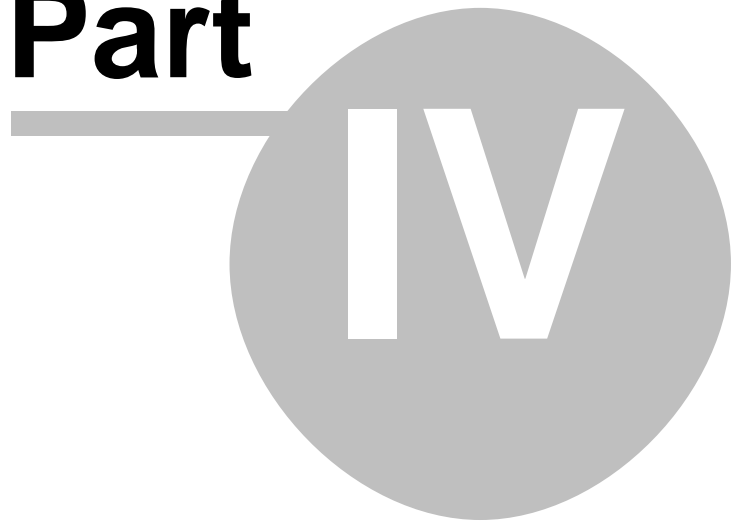
Set the zoom value, the shape and background will resize automatically

Example:

```
CAD1.Zoom:= 0.50 ;
```

See also:

Part



4 Methods

```

constructor Create(AOwner: TComponent); override;
destructor Destroy; override;

procedure PrintPreview(PreviewBitmap:TBitmap;Scale:Double=1.0);
procedure Print(Scale:double=1.0);
procedure PrintPreviewLayers(ALayers: array of Integer; TheResult: TBitmap; ScaleforPrint:
double = 1.0);
procedure PrintLayers(ALayers: array of Integer; Scale: double = 1.0);

function SetCurrentLayerId(ALayerID: integer): Boolean;
function NewLayer(ALayerName: string; aVisible: Boolean = true): integer;
function GetLayersCount(): integer;
function GetMaxLayerId(): integer;
function GetLayerIdByName(ALayerName: string): integer;
function GetLayerNameById(ALayerId: integer): string;
function GetLayerNoByName(ALayerName: string): integer;
function GetLayerNoById(ALayerId: integer): integer;
function GetLayerIdByNo(ALayerNo: integer): integer;

function DeleteLayerByName(ALayerName: string): Boolean;
function DeleteLayerById(ALayerID: integer): Boolean;
function DeleteAllLayers: Boolean;

procedure VisibleLayerByName(LayerName: string);
procedure VisibleLayerById(LayerId: integer);

procedure VisibleAllLayer;
function IsVisibleLayerById(LayerId: integer): Boolean;
procedure InVisibleLayerByName(LayerName: string);
procedure InVisibleLayerById(LayerId: integer);
procedure SetLayerNameByName(const NewLayerName, OldLayerName: string);
procedure SetLayerNameById(const NewLayerName: string; ALayerID: integer);
function GetShapesCountInALayer(ALayerId: integer): integer;

function GetShapeByName(const AName:String): TMyShape;
procedure RenameShapeName(const OldName,NewName:String);

procedure alignLeft;
procedure AlignRight
procedure AlignBottom
procedure AlignTop
function AddShapeByCode( ShapeStyle:TDrawTool; ThePoints: array of
TPoint;OnlyForText:String=""): integer;
procedure SendToBack
procedure BringToFront
function GetShapesCount:integer;
function GetSelectedShape():integer;
function DeleteShape(ShapelD:Integer):Boolean;
function DeleteSelectedShape():Boolean;
procedure DeleteAllShapes;
procedure DrawAllShape;

```

```
procedure GroupWorkingShape();
  procedure UngroupShape
  procedure SetBkBitmap(const Value: TBitmap);
  procedure SetBkBitmap(const Value: TBitmap);
  procedure LockShape(AShape:TMyShape);
  procedure UnLockShape(AShape:TMyShape);
  function SaveToStream(Stream: TStream):Boolean;
  function LoadFromStream(Stream: TStream):Boolean;
  function SaveToFile(FileName:String):Boolean;
  function LoadFromFile(FileName:String):Boolean;
  function IsTCADFile(FileName:String):Boolean;
```

4.1 AddImageShapeByCode

function

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

Description:

Add image shape by code, if you want add other shape, please use AddShapeByCode.

Return Value:

0: Add ok
else: Add failed

Parameter

TheName:Image shape's name
LeftTop:the point of the image's Left-Top corner.
ABitmap:a image that you want add into TCAD
TheAngle:Rotated angle, it is in radians.

Example:

```
var
myBitmap:TBitmap;
begin
  {Create a temp bitmap to load from 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',Point(50,100),MyBitmap,0{rotated angle});

  MyCAD1.AddShapeByCode(spEllipse,'EllipseName',[Point(20,80),Point(80,80),Point(80,160),
Point(20,160)],0);

  {free it because no use}
```

See also

[AddShapeByCode](#)
[OnShapeAdded](#)

4.2 AddShapeByCode

function AddShapeByCode(ShapeStyle:TDrawTool; ShapeName:string;ThePoints: array of TPoint;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: too few points
-2: too more points

Parameter:

ShapeStyle : Please see [DrawTool](#)
ShapeName: shape's name

Example:

```
if  
MyCAD1.AddShapeByCode(spEllipse,'EllipseName',[Point(20,80),Point(80,80),Point(80,160),Point(20,160)],0) = 0 then
```

```
ShowMessage('Add a shape finished!')
```

See also

[AddImageShapeByCode](#)
[OnShapeAdded](#)

4.3 AlignBottom

procedure AlignBottom;

Description:

When you select more than one shape, this procedure can align them Bottom.

Example:

```
MyCAD1.Alignbottom;
```

4.4 AlignLeft

procedure AlignLeft;

Example:

```
MyCAD1.AlignLeft;
```

4.5 AlignRight

```
procedure AlignRight;
```

Description:

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

Example:

```
MyCAD1.AlignRight;
```

4.6 AlignTop

```
procedure AlignTop;
```

Description:

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

Example:

```
MyCAD1.AlignTop;
```

4.7 BringToFront

```
procedure BringToFront
```

Description:

Sent the selected shape to front

Example:

```
MyCAD1. BringToFront;
```

See Also:

[SendToBack](#)

4.8 ChangeAllTextFontSize

procedure ChangeAllTextFontSize(FontHeight,FontWidth: integer)

Description:

Change **all** spText's font size.

Example:

MyCAD1. ChangeAllTextFontSize(100,50);

4.9 Create

constructor Create(AOwner: TComponent); **Create TMyCAD;**

Description:

Create TMyCAD;

4.10 DeleteAllLayers

function DeleteAllLayers(): **Boolean;**

Description:

Delete all layer and delete all shapes.

See Also

DeleteAllShapes

4.11 DeleteAllShapes

procedure DeleteAllShapes;

Description:

Delete all shapes. The layers still exist and not change.

See Also:

[DeleteAllLayers](#)

4.12 DeleteLayerByID

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

Description:

Delete Layer by the layer's Id , if ALayerId not matched , it returns false.

4.13 DeleteLayerByName

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

Description:

Delete Layer by the layer's name , if ALayerName not matched , it returns false.

4.14 DeleteSelectedShape

function DeleteSelectedShape

Description:

Delete current selected shape, if no shape or more shapes be selected, it returns false.

4.15 DeleteShapeByID

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

Description:

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

Is is effect only for alone shape, not grouped shape.

Sample:

```
procedure Form1.Button1Click(Sender:TObject)
begin
  DeleteShapeID(1)
end;
```

See Also:

[DeleteSelectedShape](#)

[DrawAllShape](#)

4.16 Destroy

destructor Destroy;

Description:

Destroy the instance of TMyCAD;

4.17 DrawAllShape

procedure DrawAllShape

Description:

Redraw all shapes.

4.18 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 Form1.Button1Click(Send:TObject)
begin
  ShowMessage(' The Layers name is:'+MyName' +' its layer Id is: '+Inttostr(MyCAD1.
  GetLayerIdByName('MyName')));
end;
```

See Also:

4.19 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:

4.20 GetLayerNameByID

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

Description:

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

Sample:

```
procedure Form1.Button1Click(Send:TObject)
var
  A:string;
begin
  A:=GetLayerNameByID(8);
  if A<>'' then
    ShowMessage(' The Layers Id is:'+8' + its layer Name is: '+A;
end;
```

See Also:

4.21 GetLayerNoByld

function GetLayerNoByld(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.GetLayerNoByld(4)));
end;
```

See Also:

4.22 GetLayerNoByName

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

Description:

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

Sample:

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

4.23 GetLayersCount

function GetLayersCount(): **integer**;

Description:

How many layers in TMyCAD

Sample:

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

See Also:

4.24 GetMaxLayerId

function GetMaxLayerId(): **integer**;

Description:

Get max layer's Id, It is not the count of layers

Sample:

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

See Also:

4.25 GetSelectedShapeID

function GetSelectedShapeID():**integer**;

Description:

Return selectedShape'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
  ShowMessage('now,the No.'+inttostr(GetSelectedShapeID())+'be selected!')
```

See Also:

[DeleteShape](#)

[DeleteSelectedShape](#)

[DrawAllShape](#)

4.26 GetShapeByName

```
function GetShapeByName(const AName:String): TMyShape;
```

Description:

Know a shape's name , Get the shape.

Return Value:

A Shape or the first shape in a grouped shape

Example:

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

4.27 GetShapesCount

```
function GetShapesCount:integer;
```

Description:

Get the count of shapes that you drew.

Example:

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

4.28 GetShapesCountInALayer

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

Description:

get the amount of the shapes on a layerId;

4.29 GroupWorkingShape

```
procedure GroupWorkingShape;
```

Description:

Group more selected shape.when you select one shape, no effect;

Example:

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

4.30 InVisibleLayerById

```
procedure InVisibleLayerById(LayerId: integer);
```

Description:

set the layer hide.

4.31 InVisibleLayerByName

```
procedure InVisibleLayerByName(LayerName: string);
```

Description:

set the layer hide.

4.32 IsTCADFile

```
function IsTCADFile(FileName:String):Boolean;
```

Description:

Determine a file is TCAD format or not .

Example:

```
if MyCAD1.IsTCADFile( OpenFileDialog1.FileName) then  
begin  
  if not MyCAD1.LoadFromFile(OpenDialog1.FileName) then  
    ShowMessage('Error when read file');  
  end  
  else  
    ShowMessage(OpenDialog1.FileName +'is not a TCAD format file');  
end;
```

4.33 IsVisibleLayerByID

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

Description:

Is the layer is visible or not .

Return Value:

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

4.34 LoadFromFile

```
function LoadFromFile(FileName:String):Boolean;
```

Description:

Load from a file.

Example:

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

See also:

[SaveToFile](#)

4.35 LoadFromStream

```
function LoadFromStream(Stream:TStream):Boolean;
```

Description:

Load from the stream, you can read from the database's field.

Example:

```
CAD1.LoadFromStream(myStream);
```

See also:

[SaveToStream](#)

4.36 LockShape

```
procedure LockShape(AShape:TMyShape);
```

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 disappear the hotspot when you selected

See Also:

[UnlockShape](#)
[Locked](#)

4.37 Move

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

Description:

move a shape or a grouped shape by code.

Example:

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

See Also:

[OnShapeDragging](#)

4.38 NewLayer

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

Description:

Add a layer for TMyCAD, if failed return -1 ,else it is return LayerId.

Example:

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

See also:

4.39 Rotate

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

Description:

Rotate a shape or a combine shape by code.
AAngle in radians.

This example can rotate shape 30 degree.

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

4.40 Print

```
procedure Print(Scale:double=1.0);
```

Description:

Print the Canvas to printer.

Example:

```
CAD1.Print (0.5);
```

See also:

[PrintPreview](#)

4.41 PrintLayers

```
procedure PrintLayers(ALayers: array of Integer; Scale: double = 1.0);
```

Description:

Print the one or more layers to printer.

Example:

```
Form1.mycad1.PrintLayers(form1.MyCAD1.CurrentLayerId,1.0);
```

See also:

[Print](#)
[PrintPreview](#)
[PrintPreViewLayers](#)

4.42 PrintPreview

```
procedure PrintPreview(PreviewBitmap:TBitmap;Scale:Double=1.0);
```

Print the Canvas to printer.

Example:

```
CAD1.PrintPreview (Image1.Picture.Bitmap,1.0);
```

See also:

[Print](#)

4.43 PrintPreviewLayers

procedure PrintPreviewLayers(ALayers: array of Integer; TheResult: TBitmap; ScaleforPrint: double = 1.0);

Description:

You can select one or more layer to print to the printer.

Example:

[PrintPreview](#)
[Print](#)

4.44 RenameShapename

procedure RenameShapeName(const OldName,NewName:String);

Description:

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

Example:

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

4.45 SaveToFile

```
function SaveToFile(FileName:String):Boolean;
```

Description:

Save CAD content to file, it is in TCAD 2002 format .

Example:

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

See also:

[LoadFromFile](#)

4.46 SaveToStream

```
function SaveToream(Stream:TStream):Boolean;
```

Description:

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

Example:

```
CAD1.SaveToStream(myStream);
```

See also:

[LoadFromStream](#)

4.47 SendtoBack

```
procedure SentToBack
```

Description:

Sent the selected shape to background .

```
MyCAD1.SendToBack;
```

See Also:

[BringToFront](#)

4.48 SetBkBitmap

```
procedure SetBkBitmap(const Value: TBitmap);
```

Description:

Set the background bitmap for TCAD, for clear it , using SetBkBitmap(nil);

Example:

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

4.49 SetCurrentLayerId

```
function SetCurrentLayerId(ALayerID: integer): Boolean;
```

Description:

Change the current layer Id .

Examples:

```
if ComboBoxEx1.ItemIndex > -1 then  
begin  
  Id := MyCAD1.GetLayerIdByNo(ComboBoxEx1.ItemIndex);  
  MyCAD1.SetCurrentLayerId(id);  
end;
```

See Also

4.50 SetLayerNameByid

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

Description:

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

4.51 SetLayerNameByName

procedure SetLayerNameByName(const NewLayerName, OldLayerName: string);

Description:

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

4.52 SetMyImage

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

Description:

Add a bitmap to TMyImage's shape.

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;
```

4.53 UnGroupShape

`procedure UngroupShape`

Description:

unGroup have grouped shapes.

Example:

```
MyCAD1.UnGroupShape( MyCAD1.GetSelectedGroupShape);
```

4.54 UnLockShape

`procedure UnLockShape(AShape:TMyShape);`

Description:

After execute this command, the shape or the grouded shape can be moved or resized .

AShape.Locked is false.

See Also:

[LockShape](#)
[Locked](#)

4.55 VisibleAllLayer

`procedure VisibleAllLayer;`

Description:

Make all layer visible

See Also:

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

4.56 VisibleLayerByID

```
procedure VisibleLayerByID(LayerID: integer);
```

Description:

Make a layer visible by layer's id.

See Also:

[VisibleAllLayer](#)

[VisibleLayerByName](#)

4.57 VisibleLayerByName

```
procedure VisibleLayerByName(LayerName: string);
```

Description:

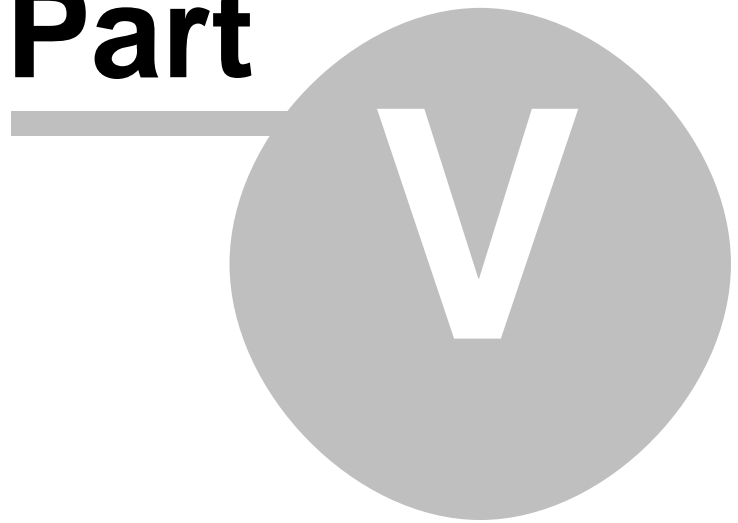
Make a layer visible by layer's name.

See Also:

[VisibleAllLayer](#)

[VisibleLayerByld](#)

Part



5 Events

property OnMouseDown
property OnMouseMove
property OnMouseUp
property OnPaint

property [OnActionToolToSelecting](#)
property [OnShapeSelected](#)
property [OnShapeAdded](#)
property [OnNewLayer](#)
property [OnDeleteLayer](#)

property [OnShapeDragging](#): TShapeDraggingSizing
property [OnShapeResizing](#): TShapeDraggingSizing
property [OnMouseEnter](#): TNotifyEvent
property [OnMouseLeave](#): TNotifyEvent
property [OnMouseEnterShape](#): TEnterLeaveShape
property [OnMouseLeaveShape](#): TEnterLeaveShape

5.1 OnActionToolToSelecting

TActionToolToSelecting = procedure() of object;

property OnActionToolToSelecting: TActionToolToSelecting

Description:

Use OnActionToolToSelecting to handle when the ShapeTool return to SpSelecting,

5.2 OnDeleteLayer

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

property OnDeleteLayer: TLayerop

Description:

Use OnDeleteLayer to handler a layer be deleted.

5.3 OnMouseEnter

property OnMouseEnter:TNotifyEvent

Description:

When the mouse enter TCAD, this event trigger.

example:

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

5.4 OnMouseEnterShape

TEnterLeaveShape = procedure(ShapeID:integer;LayId:integer;AShape:TMyShape) of object;

property OnMouseEnter:TNotifyEvent

Description:

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

example:

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

5.5 OnMouseLeave

property OnMouseLeave:TNotifyEvent

Description:

When the mouse leave TCAD, this event trigger.

example:

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

5.6 OnMouseLeaveShape

TEnterLeaveShape = procedure(ShapeID:integer;LayId:integer;AShape:TMyShape) of object;
property OnMouseLeave: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(ShapeID));  
end;
```

5.7 OnNewLayer

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

property OnNewLayer: TLayerOp ;

Description:

Use OnNewLayer to handle a new layer be added.

5.8 OnShapeAdded

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

property OnShapeAdded: TShapeAdded;

Description:

Use OnShapeAdded to handler a shape is added

Examples:

```
Memo1.Lines.Add('You Add a Shape,its id is : ' + inttostr(ShapeID) + ',in Layer ' +  
inttostr(LayerId) + ', it is a ' + AShape.ClassName + ',Now CAD has ' + inttostr(ShapeCount) +  
'Shapes')
```

5.9 OnShapeDragging

TShapeDraggingSizing = procedure (Shapeld:integer;LayId:integer;AShape:TMyShape; FromPoint:TPoint;ToPoint:TPoint) of object;

property OnShapeDragging:TShapeDraggingSizing

Description:

When a Shape is dragging, this event be trigger.

Example:

```

procedure TForm1.MyCAD1ShapeDragging(Shapeld, LayId: Integer;
  AShape: TMyShape; FromPoint, ToPoint: TPoint);
begin
  Memo1.Lines.Add('-----Drag Event-----');
  Memo1.Lines.Add('Shapeld is: '+Inttostr(Shapeld)+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](#)

5.10 OnShapeResizing

TShapeDraggingSizing = procedure (Shapeld:integer;LayId:integer;AShape:TMyShape; FromPoint:TPoint;ToPoint:TPoint) of object;

property OnShapeResizing:TShapeDraggingSizing

Description:

When a Shape is Resizing, this event be trigger.

Example:

```

procedure TForm1.MyCAD1ShapeResizing(Shapeld, LayId: Integer;
  AShape: TMyShape; FromPoint, ToPoint: TPoint);
begin
  Memo1.Lines.Add('-----Resize Event-----');
  Memo1.Lines.Add('Shapeld is: '+Inttostr(Shapeld)+AShape.Name +' resized ','+
  'From x:'+ inttostr(FromPoint.x)+' y:' +inttostr(FromPoint.y)+
  'To x:'+ inttostr(ToPoint.x)+' y:' +inttostr(ToPoint.y));

end;

```

5.11 OnShapeSelected

TShapeSelected = procedure(LastSelectedShapelD: Integer; LastSelectedShape: TMyShape; LastLayId: integer; SelectedShapelD: Integer; SelectedShape: TMyShape; LayerId: integer) of object;

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

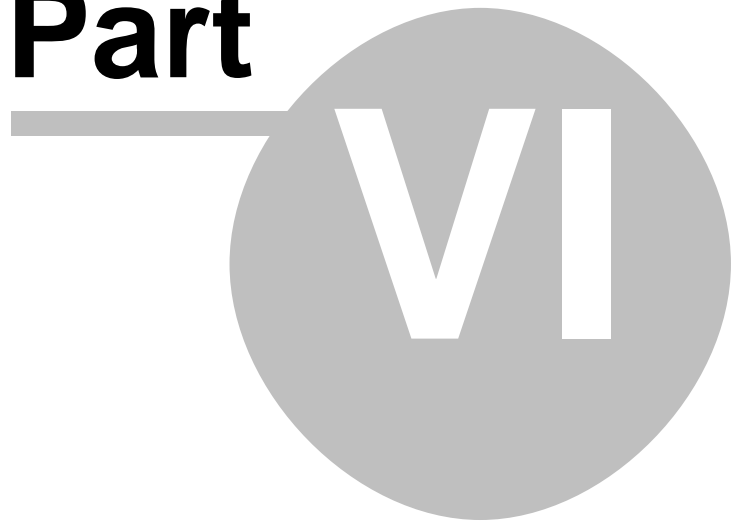
Examples:

```
procedure TForm1.MyCAD1ShapeSelected(LastSelectedShapeID: Integer;
  LastSelectedShape: TMyShape; LastLayerId, SelectedShapelD: Integer;
  SelectedShape: TMyShape; LayerId: Integer);
begin
  Memo1.Lines.Add('-----Select Event-----');
  if lastSelectedShapelD = -1 then Memo1.Lines.Add('No last shape selected')
  else

    Memo1.Lines.Add(' Last shape id is ' + inttostr(LastSelectedShapeID)
      + ',in Layer ' + inttostr(LastLayerId) + ', it is a ' + LastSelectedShape.ClassName);

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

Part



6 About Crystal Component

Our goal is to 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 about our products from our web.

users outside of china,please visit:

<http://www.codeidea.com>

<http://cad.codeidea.com>

<http://usa.codeidea.com>

china users:

<http://china.codeidea.com>

E-Mail:

hongbin.fei

webmaster@codeidea.com

yuefen.yao

support@codeidea.com

Telephone:

+86 572 7281888 (Office)

+86 572 2050545 (Fax)

+86 (0)13906722760 (Mobile)

Address:

Room 307#,TiaoXi Road Commerce Building

HuZhou, ZheJiang

China

Postcode:313000

Index

- A -

About Crystal Component 49
 AddShapeByCode 22
 AlignBottom 22
 AlignLeft 22
 AlignRight 23
 AlignTop 23

- B -

BkBitmapMode 8
 BringToFront 23
 Brush 9

- C -

ChangeAllTextFontSize 24
 ColorOfBackground 9
 ColorOfHot 10
 Create 24
 CurrentLayerId 10

- D -

defines 5
 DeleteAllLayers 24
 DeleteAllShapes 25
 DeleteLayerById 25
 DeleteLayerByName 25
 DeleteSelectedShape 25
 DeleteShape 26
 Destroy 26
 DrawAllShape 26

- E -

Events 43

- G -

GetLayerIdByName 27
 GetLayerIdByNo 27
 GetLayerNameById 28
 GetLayerNoById 28

GetLayerNoByName 28
 GetLayersCount 29
 GetMaxLayerId 29
 GetSelectedShapeid 29
 GetShapesCount 30
 GetShapesCountInALayer 30
 GridHeight 11
 GridShow 11
 GridWidth 11
 GroupWorkingShape 31

- I -

InVisibleLayerById 31
 InVisibleLayerByName 31
 IsTCADFile 32
 IsVisibleLayerById 32

- L -

LabelXY 12
 LableValue 13
 LoadFromFile 33
 LoadFromStream 33

- M -

Methods 20

- N -

NewLayer 34

- P -

PageHeight 13
 PageOrientation 14
 PageStyle 14
 PageWidth 14
 Pen 15
 Print 35
 PrintABorder 15
 PrintBackground 15
 PrintLayers 35
 PrintPreview 35
 PrintPreviewLayers 20, 36
 Properties 7

- R -

Ratio 17
ReturnToSelecting 15

- S -

SaveToFile 37
SaveToStream 37
SendtoBack 37
SetBkBitmap 38
SetCurrentLayerId 38
SetLayerNameByid 39
SetLayerNameByName 39
SetMyImage 39
ShapeTool 16
Snap 16
SnapPixels 17

- T -

TheUnit 17

- U -

UnGroupShape 40

- V -

VisibleAllLayer 40, 41
VisibleLayerByID 41

- Z -

Zoom 18

www.codeidea.com

cad.codeidea.com