OrangeUI控件使用基础

## 一． the most basic and also important property of OrangeUI

1. Properties

Every control has its own properties,

You can find them in Properties,

Here are some properties of button control

Icon

PushedIcon

IsPushed:means whether pushed

AutoSize:means adjust size automatically

Like this:

2.SelfOwnMaterial

Material: it means pictures, color, draw parameters of text,

SelfOwnMaterial:it means control’s own skin or draw parameters etc.

The two basic properties of SelfOwnMaterial are

BackColor:TDrawRectParam(background color)

IsTransparent:Boolean(whether background is transparent),

If IsTransparent is False,

Then we will use Backcolor to draw background,

If transparent is True,

Then the control will be transparent,

For example ,there are some materials of button :

NormalPicture: means normal state of background picture

DownPicture: means background picture when you click mouse

DrawCaptionParam: means draw parameters of button’s caption

DrawPictureParam: means draw parameters of background picture

Like this:

3. MaterialUseKind etc

//usekind of component type

ComponentTypeUseKind:TComponentTypeUseKind;

ctukDefault: means use default component type

 ctukName: means use assigned name of component type

//name of component-type,for example “Default”, “normal”

ComponentTypeName:String;

//usekind of material

mukSelfOwn: means use SelfOwnMaterial

mukDefault: means use default material of skin

mukName: means use a named material of skin

mukRef: means use property RefMaterial

//the material name in skinpackage

MaterialName:String;

//use other materials

RefMaterial:TSkinMaterial;

//material from control itself

SelfOwnMaterial:TSkinMaterial;

//whether store SelfOwnMaterial

KeepSelfOwnMaterial:Boolean;

 If it is false,then the control don’t use SelfOwnMaterial,

 And SelfOwnMaterial is nil.

//current material which the control is using.It depends on MaterialUseKind

CurrentUseMaterial:TSkinMaterial;

## 二、Draw of OrangeUI

We use TDrawCanvas in unit uDrawCanvas to draw OrangeUI,

On VCL platform ,we use GDI+ library to implement it

On FMX platform,we use TCanavs from OrangeUI to implement it.

These are four basic draw methods:

DrawText this is used for draw text

DrawSkinPicture this is used for draw picture of skin

DrawRect this is used for draw rectangle

DrawLine this is used for draw line

Next we will introduce draw parameters

1. TDrawParam: means base class of all draw parameters

//transparency: a number from 0 to 255

Alpha:Byte

//which state the control is currently in

EffectStates:TDPStates

 dpstMouseDown: state when mouse click

 dpstMouseOver: state when mouse hover

 dpstPushed: state when pressed

//you can use it to set dynamic effect

DrawEffectSetting:TDrawEffectSetting

 MouseDownEffect: effect when mouse click

 MouseOverEffect: effect when mouse hover

 PushedEffect: effect when pressed

//you can use it to costomizely draw rectangle or its position

DrawRectSetting:TDrawRectSetting

//whether to use self-defined rectangle

Enabled:Boolean

//you can set the unit as percent or px

SizeType: TDPSizeType

//position

Left,Top:Double

//right margin

Right:Double

//bottom margin

Bottom:Double

//width

Width:Double

//height

Height:Double

//type of horizontal position

PositionHorzType:TDPPositionHorzType

 dpphtNone: null

 dpphtLeft: horizontal left

 dpphtCenter: horizontal center

 dpphtRight: horizontal right

//type of vertical position

PositionVertType:TDPPositionVertType

 dppvtNone: null

 dppvtTop: vertical top

 dppvtCenter: vertical center

 dppvtBottom: vertical bottom

2.TDrawRectParam: parameters while drawing rectangle

//whether fill this rectangle

IsFill:Boolean

//color of this filled rectangle

FillColor: TDrawColor

//whether the angle of rectangle is round

IsRound:Boolean

//width and height of round angle

RoundWidth and RoundHeight:Double

//color of border

BorderColor:TDrawColor

//width of border

BorderWidth:Double

//border of rectangle

BorderEadges:TDRPBorderEadges

 beLeft: left border

 beTop: top border

 beRight: right border

 beBottom: bottom border

//corner of rectangle

RectCorners:TDRPRectCorners

 rcTopLeft: top left corner of rectangle

 rcTopRight: top right corner of rectangle

 rcBottomLeft: bottom left corner of rectangle

 rcBottomRight: bottom right corner of rectangle

//whether it is line

IsLine:Boolean

//position of line

LinePosition: TDRPLinePosition

 lpLeft: the line is on the left

 lpTop: the line is on the top

 lpRight: the line is on the right

 lpBottom: the line is on the bottom

3.TDrawTextParam: parameters while drawing text

//you can use it to set font

DrawFont: TDrawFont

//when words are too long, whether to wrap it

IsWordWrap:Boolean

//when the screen can’t show whole font,type of trimming

FontTrimming:TFontTrimmingType

 fttNone: no trimming

 fttCharacter: trim by characters

 fftWord: trim by words

//style of horizontal align

FontHorzAlign:TFontHorzAlign

 fhaLeft: horizontal left

 fhaCenter: horizontal center

 fhaRight: horizontal right

//style of vertical align

FontVertAlign:TFontVertAlign

 fvaTop: vertical top

 fvaCenter: vertical center

 fvaBottom: vertical bottom

//whether to draw vertically

IsDrawVert:Boolean

4.TDrawPictureParam: parameters while drawing pictures

//when drawing pictures, whether automatically fit its size

IsAutoFit:Boolean

//when drawing pictures, whether stretch

IsStretch:Boolean

//style of stretch

StretchStyle:TPictureStrechStyle

 issTensile: most normal pattern of filled stretch

 issSquare: pattern of square stretch

//style of horizontal align

PictureHorzAlign:TPictureHorzAlign

phaLeft

phaCenter

phaRight

//style of vertical align

PictureVertAlign:TPictureVertAlign

pvaTop

pvaCenter

pvaBottom

5.TDrawLineParam: parameters while drawing line

//color of line

Color:TDrawColor

//width of line

PenWidth:Double

## 三.Picture class of OrangeUI

In OrangeUI we use TDrawPicture as picture class,

Properties of TDrawPicture:

//binding SkinImageList

SkinImageList:

//Index of drawn picture in SkinImagelist

ImageIndex:Integer

//Name of drawn picture in SkinImagelist(more readable than using index)

ImageName:String

//draw itself, draw sub-picture ,draw picture list

PictureDrawType:TPictureDrawType

 pdtAuto: judge automatically

 pdtPicture: when drawing, use pictures as its own setting

 pdtImageList: when drawing, use pictures from ImageIndex

 pdtImageName: when drawing, use picture lists from ImageName

 pdtReference: when drawing, use picture from RefPicture

 pdtFile: when drawing, use file

 pdtResource: when drawing, use resource picture

 pdtUrl: when drawing, use url

//path of image file

FileName:String

//resource name of image

ResourceName:String

//url of image

Url:String