

# CnPack 平滑字体组件帮助文档

作者：周劲羽、刘啸、e-  
部门：CnPack 开发组 界面组  
类别：帮助文档  
版本：V0.8.2.0  
创建：2007.08.15  
修改：

## 一、 引 言

### 1、 组件包概述

CnPack 平滑特效字体组件包是一个基于平滑（Anti-Aliasing）和特效（Effect）字体技术的组件包，支持透明、半透明、阴影、模糊、渐变色、字体纹理、噪声、轮廓字等特效。该组件包包含以下组件：

- TCnAALabel: 平滑特效字体标签组件，它类似于普通的 TLabel 组件，提供阴影、渐变色、透明、半透明等功能。
- TCnAALinkLabel: 平滑特效超链接标签组件，用于显示超链接的标签，支持淡入淡出切换。
- TCnAAText: 平滑特效文本组件，它是支持多行文本的平滑字体组件，通过使用标签，允许为每一行文本指定特定的对齐方式和字体特效。
- TCnAAScrollText: 平滑滚动文本组件，它是动态滚动的多行文本组件，支持标签功能。
- TCnAAFadeText: 平滑特效渐隐文本组件。
- TCnAAFontDialog: 平滑特效字体对话框。

此外，还有部分支持平滑特效的类定义，本文中一并说明。

## 二、 TCnAALabel 组件

### 1、 组件功能概述

平滑特效文本组件，用于显示多行文本，通过使用标签，允许每行文本使用不同的对齐方式和字体特效。它继承于基类 TCnAAGraphicControl。

### 2、 所在文件

CnAACtrls.pas，平滑特效字体组件单元。

### 3、 属性说明

**property** AutoSize;

是否自动设置控件尺寸。

**property** Border;

控件边界保留宽度。

**property** Font;

控件字体。

**property** Width **default** 46;

控件宽度。

**property** Height **default** 12;

控件高度。

**property** Effect: TCnAAFontEffect;

平滑特效字体属性。

以下属性来源于基类 TCnAAGraphicControl:

**property** Border: TBorderWidth;

控件边界保留宽度。

**property** Canvas;

控件画布。

**property** Drag: TCnDrag;

拖动的相关属性。

**property** ParentEffect: TCnParentEffect;

父控件影响的相关属性。

### 4、 方法说明

以下方法来源于基类 TCnAAGraphicControl:

**procedure** BeginUpdate;

功能: 开始更新, 调用该方法后, 对控件属性的更改不会导致控件重绘, 建议在批量修改控件时使用。

注意该方法必须与 EndUpate 成对使用。

---

**procedure** EndUpdate;

功能：结束更新，见 **BeginUpdate**。用户结束更新后通常还应调用 **Changed** 方法通知控件重绘。

**procedure** Changed;

功能：通知控件属性已变更，要求控件重绘。

## 5、 事件说明

以下事件来源于基类 **TCnAAGraphicControl**：

**property** OnMouseEnter: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

鼠标移入控件内部时触发的事件。

参数：

1. Sender: TObject; 该组件实例。

**property** OnMouseLeave: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

鼠标移出控件内部时触发的事件。

参数：

1. Sender: TObject; 该组件实例。

## 三、 TCnAALinkLabel 组件

### 1、 组件功能概述

平滑特效超链接标签组件，用于显示超链接的标签，支持淡入淡出切换。它继承于 **TCnAALabel**。

### 2、 所在文件

CnAACtrls.pas, 平滑特效字体组件单元。

### 3、 属性说明

**property** HotLink: TCnHotLink;

超链接属性。

### 4、 方法说明

**procedure** Click;

模拟用户点击该控件，调用 **HotLink** 的 **URL** 属性。

## 5、 事件说明

同 TCnAALabel。

## 四、 TCnAAText 组件

### 1、 组件功能概述

平滑特效文本控件，用于显示多行文本。通过使用标签，允许每行文本使用不同的对齐方式和字体特效。它继承于基类 TCnCustomAAText 和 TCnAAGraphicControl。

### 2、 所在文件

CnAACtrls.pas, 平滑特效字体组件单元。

### 3、 属性说明

**property** AutoSize;  
是否自动设置控件尺寸。

**property** Border;  
控件边界保留宽度。

**property** Font;  
控件字体。

**property** Width **default** 46;  
控件宽度。

**property** Height **default** 12;  
控件高度。

**property** Text: TCnTextParam;  
控件文本内容及显示参数。

以下属性来源于基类 TCnCustomAAText:

**property** Fonts: TCnFontLabels;  
字体标签属性。

**property** Labels: TCnUserLabels;  
用户标签属性。

以下属性来源于基类 TCnAAGraphicControl:

**property** Border: TBorderWidth;

控件边界保留宽度。

**property** Canvas;

控件画布。

**property** Drag: TCnDrag;

拖动的相关属性。

**property** ParentEffect: TCnParentEffect;

父控件影响的相关属性。

## 4、 方法说明

以下方法来源于基类 TCnAAGraphicControl:

**procedure** BeginUpdate;

功能: 开始更新, 调用该方法后, 对控件属性的更改不会导致控件重绘, 建议在批量修改控件时使用。

注意该方法必须与 **EndUpate** 成对使用。

**procedure** EndUpdate;

功能: 结束更新, 见 **BeginUpdate**。用户结束更新后通常还应调用 **Changed** 方法通知控件重绘。

**procedure** Changed;

功能: 通知控件属性已变更, 要求控件重绘。

## 5、 事件说明

以下事件来源于基类 TCnAAGraphicControl:

**property** OnMouseEnter: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

鼠标移入控件内部时触发的事件。

参数:

1. Sender: TObject; 该组件实例。

**property** OnMouseLeave: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

鼠标移出控件内部时触发的事件。

参数:

1. Sender: TObject; 该组件实例。

## 五、 TCnAAScrollText 组件

### 1、 组件功能概述

平滑滚动文本控件，用于多行文本的动态滚动显示。它继承于基类 TCnCustomAAText 和 TCnAAGraphicControl。

### 2、 所在文件

CnAACtrls.pas, 平滑特效字体组件单元。

### 3、 属性说明

**property** RepeatedCount: Integer;

已循环滚动次数，运行期只读属性。

**property** CurrPos: Integer;

当前显示的内容在整个图像中的位置，用户可用它来手动控制控件滚动。

**property** BmpHeight: Integer;

整个图像的高度。

**property** AutoUpdate;

是否允许控件参数变更时自动重新创建滚动内容。如果有很多参数需要在运行时设置，可将该属性设为 **False**，待设定完参数后调用 **Reset** 方法。

**property** Active: Boolean;

是否允许文本滚动。

**property** Height **default** 280;

控件高度。

**property** Width **default** 240;

控件宽度。

**property** ScrollDelay: Word;

滚动时的延时，单位为毫秒。

**property** ScrollStep: Integer;

一次滚动的象素数，如果设定为负数将向下滚动。

**property** RepeatCount: TBorderWidth;

允许循环次数，指定次数的循环结束将自动停止滚动，并产生 **OnComplete** 事件。该值设为 0 将无限循环。

**property** RepeatDelay: Word;

完成一次滚动循环后的延时，如果不需要延时，可设为 0。

**property** Text: TScrollTextParam;

滚动文本内容和参数属性。

以下属性来源于基类 TCnCustomAAText:

**property** Fonts: TCnFontLabels;

字体标签属性。

**property** Labels: TCnUserLabels;

用户标签属性。

以下属性来源于基类 TCnAAGraphicControl:

**property** Border: TBorderWidth;

控件边界保留宽度。

**property** Canvas;

控件画布。

**property** Drag: TCnDrag;

拖动的相关属性。

**property** ParentEffect: TCnParentEffect;

父控件影响的相关属性。

## 4、 方法说明

**procedure** Reset; override;

重新创建滚动内容，用于 AutoUpdate 为假时，在运行期动态修改控件参数后初始化控件滚动。

**procedure** ReStart;

重新开始滚动，清滚动计数器，文本从头开始滚动。

以下方法来源于基类 TCnAAGraphicControl:

**procedure** BeginUpdate;

功能：开始更新，调用该方法后，对控件属性的更改不会导致控件重绘，建议在批量修改控件时使用。

---

注意该方法必须与 **EndUpdate** 成对使用。

**procedure** EndUpdate;

功能：结束更新，见 **BeginUpdate**。用户结束更新后通常还应调用 **Changed** 方法通知控件重绘。

**procedure** Changed;

功能：通知控件属性已变更，要求控件重绘。

## 5、 事件说明

**property** OnComplete: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

指定次数的滚动循环结束事件，见 **RepeatCount**

参数：

1. Sender: TObject; 该组件实例。

**property** OnTextReady: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

参数：

1. Sender: TObject; 该组件实例。

滚动内容已初始化事件。

**property** OnPainted: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

控件重绘事件。

参数：

1. Sender: TObject; 该组件实例。

以下事件来源于基类 **TCnAAGraphicControl**：

**property** OnMouseEnter: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

鼠标移入控件内部时触发的事件。

参数：

2. Sender: TObject; 该组件实例。

**property** OnMouseLeave: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

鼠标移出控件内部时触发的事件。

参数：

1. Sender: TObject; 该组件实例。

## 六、 TCnAAFadeText 组件

### 1、 组件功能概述

平滑特效渐隐文本控件，用于多行文本的淡入淡出切换显示。它继承于基类 TCnCustomAAText 和 TCnAAGraphicControl。

### 2、 所在文件

CnAACtrls.pas, 平滑特效字体组件单元。

### 3、 属性说明

**property** LineIndex: Integer;

当前显示的行索引号，用户可手动设置。

**property** RepeatedCount: Integer;

已循环滚动次数，运行期只读属性。

**property** Active: Boolean;

是否允许文本淡入淡出切换。

**property** Height **default** 34;

控件高度。

**property** Width **default** 240;

控件宽度。

**property** Font;

控件字体。

**property** RepeatCount: TBorderWidth;

允许循环次数，指定次数的循环结束将自动停止滚动，并产生 OnComplete 事件。该值设为 0 将无限循环。

**property** Text: TFadeTextParam;

控件文本内容和参数属性。

以下属性来源于基类 TCnCustomAAText:

**property** Fonts: TCnFontLabels;

字体标签属性。

**property** Labels: TCnUserLabels;

用户标签属性。

以下属性来源于基类 TCnAAGraphicControl:

**property** Border: TBorderWidth;

控件边界保留宽度。

**property** Canvas;

控件画布。

**property** Drag: TCnDrag;

拖动的相关属性。

**property** ParentEffect: TCnParentEffect;

父控件影响的相关属性。

#### 4、 方法说明

**procedure** FadeTo(Line: Integer);

淡入淡出切换到指定行。

**procedure** FadeToNext;

淡入淡出切换到下一行。

**procedure** FadeToStr(AText: string);

淡入淡出切换到指定文本。

以下方法来源于基类 TCnAAGraphicControl:

**procedure** BeginUpdate;

功能: 开始更新, 调用该方法后, 对控件属性的更改不会导致控件重绘, 建议在批量修改控件时使用。

注意该方法必须与 **EndUpdate** 成对使用。

**procedure** EndUpdate;

功能: 结束更新, 见 **BeginUpdate**。用户结束更新后通常还应调用 **Changed** 方法通知控件重绘。

**procedure** Changed;

功能: 通知控件属性已变更, 要求控件重绘。

#### 5、 事件说明

**property** OnComplete: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

指定次数的滚动循环结束事件，见 RepeatCount。

参数：

1. Sender: TObject; 该组件实例。

**property** OnPainted: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

控件重绘事件。

参数：

1. Sender: TObject; 该组件实例。

以下事件来源于基类 TCnAAGraphicControl:

**property** OnMouseEnter: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

鼠标移入控件内部时触发的事件。

参数：

1. Sender: TObject; 该组件实例。

**property** OnMouseLeave: TNotifyEvent;

TNotifyEvent = **procedure**(Sender: TObject) of object;

鼠标移出控件内部时触发的事件。

参数：

1. Sender: TObject; 该组件实例。

## 七、 TCnAAFontDialog 组件

### 1、 组件功能概述

平滑特效字体对话框组件，使用方法同普通对话框。

### 2、 所在文件

CnAAFontDialog.pas, 平滑特效字体对话框组件单元。

### 3、 属性说明

**property** Effect: TCnAAEffect;

平滑特效参数属性。

**property** Font: TFont;

字体属性。

**property** AllowChangeFont: Boolean;

是否允许更改字体，如果为 True，将允许用户更改 Font 属性。如果为 False，则不

显示“字体”按钮。

#### 4、 方法说明

**procedure** Assign(Source: TPersistent); **override**;  
对象赋值方法。

**function** Execute: Boolean;  
打开对话框，返回执行结果。

#### 5、 事件说明

**property** OnClose: TNotifyEvent;  
TNotifyEvent = **procedure**(Sender: TObject) **of** object;  
对话框关闭事件。

参数:

1. Sender: TObject; 该组件实例。

**property** OnShow: TNotifyEvent;  
TNotifyEvent = **procedure**(Sender: TObject) **of** object;  
对话框显示事件。

参数:

1. Sender: TObject; 该组件实例。

## 八、 TCnCustomParam 类

可定制的平滑字体控件参数基类，一般用于属性定义，不需要用户直接创建。它在 CnAAFont.pas 平滑特效字体单元中定义。

**property** Font: TFont;  
控件字体。

**property** Quality: TAAQuality;  
平滑字体显示精度。

**type** TAAQuality = ( aqHigh, aqNormal, aqLow, aqNone );  
平滑字体显示精度类型。

- aqHigh - 4X4 采样的最高显示精度，速度较慢。
- aqNormal - 3X3 采样的普通显示精度，最佳性能速度比。
- aqLow - 2X2 采样的低显示精度，速度较快。
- aqNone - 无平滑效果。

**property** Alignment: TAlignment;  
文本对齐方式。

**property** Layout: TTextLayout;

文本垂直方向对齐方式。

**property** WordWrap: Boolean;

文本是否自动换行。

**property** Transparent: Boolean;

控件是否透明。

**property** FontEffect: TCnAAEffect;

平滑特效字体属性。

**property** BackGround: TPicture;

控件背景图像。

**property** BackGroundMode: TBackGroundMode;

控件背景图像显示模式。

**type** TBackGroundMode =(bmTiled, bmStretched, bmCenter, bmNormal);

背景图像显示模式。

- bmTiled                   - 平铺显示。
- bmStretched             - 自动缩放显示。
- bmCenter                - 在中心位置显示。
- bmNormal                - 在左上角显示。

**property** BackColor: TColor;

控件背景颜色。

## 九、 TCnCustomTextParam 类

可定制的平滑字体文本类控件参数基类，继承于 TCnCustomParam，一般用于属性定义，不需要用户直接创建。

**property** Lines: TStrings;

控件文本内容。

**property** RowPitch: TRowPitch;

文本行间距。

**property** LabelEffect: TLabelEffect;

文本中字体、控制标签的作用范围。

---

```
type TLabelEffect =(leOnlyALine, leUntilNextLabel);
```

标签作用范围类型。

**leOnlyALine**, 字体、控制标签仅在当前行有效, 无标签的行字体、对齐等由默认值决。

**leUntilNextLabel**, 标签所对应的参数影响当前行和后续行, 直到遇到下一标签为止。

```
property FontEffect: TCnAAEffect;
```

平滑特效字体属性。

## 十、 TCnAAEffect 类

平滑特效字体效果参数类, 一般用于属性定义, 不需要用户直接创建。

```
property Shadow: TCnAAShadow;
```

字体阴影参数。

```
property Gradual: TCnAAGradual;
```

渐变显示参数。

```
property Texture: TCnAATexture;
```

字体纹理参数。

```
property Alpha: TAlpha;
```

不透明度, 参数范围为 0..100, 0 为完全透明, 100 为不透明。

```
property Blur: TBlurStrength;
```

模糊度, 参数范围为 0..100, 0 为不模糊。

```
property Angle: TAngle;
```

字体旋转角度, 参数范围为 -360..360, 单位为度。

```
property Noise: Byte;
```

噪声效果, 参数范围为 0..255。

```
property Spray: TSprayRange;
```

喷溅效果, 参数范围为 0..100。

```
property Outline: Boolean;
```

是否显示轮廓效果。

```
property HorzMirror: Boolean;
```

是否水平镜像。

**property** VertMirror: Boolean;

是否垂直镜像。

**constructor** Create(ChangedProc: TNotifyEvent); **override**;

类构造器，参数为通知事件。

**procedure** SaveToIni(Ini: TCustomIniFile; **const** Section: **string**); **virtual**;

保存到 INI 文件中。

## 十一、TCnAAShadow 类

平滑特效字体阴影显示参数类，一般用于属性定义，不需要用户直接创建。

**property** Blur: TBlurStrength;

阴影模糊度，参数范围为 0..100，0 为不模糊。

**property** Alpha: TAlpha;

阴影不透明度，参数范围为 0..100，0 为全透明，100 为不透明。

**property** Color: TColor;

阴影颜色

**property** OffsetX: TOffset;

阴影水平方向偏移量，为负值时向左偏移。

**property** OffsetY: TOffset;

阴影垂直方向偏移量，为负值时向上偏移。

**property** Enabled: Boolean;

是否允许使用。

## 十二、TCnAAGradual 类

平滑特效字体渐变显示参数类，一般用于属性定义，不需要用户直接创建。

**property** Style: TGradualStyle;

渐变显示风格。

**type** TGradualStyle = (gsLeftToRight, gsRightToLeft, gsTopToBottom, gsBottomToTop, gsCenterToLR, gsCenterToTB);

渐变方式类型。

- gsLeftToRight - 从左向右渐变
- gsRightToLeft - 从右向左渐变
- gsTopToBottom - 从上向下渐变
- gsBottomToTop - 从下向上渐变
- gsCenterToLR - 从中间向左右渐变
- gsCenterToTB - 从中间向上下渐变

**property** StartColor: TColor;  
渐变起始色。

**property** EndColor: TColor;  
渐变结束色。

**property** Enabled: Boolean;  
是否允许使用。

### 十三、TCnAATexture 类

平滑特效字体纹理显示参数类，一般用于属性定义，不需要用户直接创建。

**property** Mode: TTextureMode;  
纹理图像显示模式。

**type** TTextureMode =(tmTiled, tmStretched, tmCenter, tmNormal);  
纹理图像显示模式。

- tmTiled - 平铺显示
- tmStretched - 自动缩放显示
- tmCenter - 在中心位置显示
- tmNormal - 在左上角显示

**property** Picture: TPicture;  
纹理图像。

**property** Enabled: Boolean;  
是否允许使用。

### 十四、TCnDrag 类

拖放相关属性类，一般用于属性定义，不需要用户直接创建。

**property** DragKind: TDragKind;  
拖放类型，同 TControl 中定义。

**property** DragCursor: TCursor;  
拖放光标，同 TControl 中定义。

**property** DragMode: TDragMode;  
拖放模式，同 TControl 中定义。

## 十五、TCnParentEffect 类

父控件影响相关属性类，一般用于属性定义，不需要用户直接创建。

**property** ParentColor: Boolean;  
使用父控件的颜色，同 TControl 中定义。

**property** ParentFont: Boolean;  
使用父控件的字体，同 TControl 中定义。

**property** ParentShowHint: Boolean;  
使用父控件的提示显示设置，同 TControl 中定义。

## 十六、TCnFontLabels 类

字体标签列表类，继承于 TOwnedCollection，定义了可在文本类控件中使用的字体标签集，一般不需要用户直接创建。

**property** Items[ Index: Integer ]: TCnFontLabel; **default**;  
字体标签项数组属性。

**function** AddItem( AName: **string**; AFontName: **string**; AFontSize: Integer;  
AFontColor: TColor; AFontEffect: TFontStyles; Shadow: Boolean; OffsetX, OffsetY:  
Integer ): TCnFontLabel;  
增加一个新的字体标签。

**function** IndexOf( **const** Name: **string** ): Integer;  
根据标签名查找子项索引号。

**procedure** Check( **var** AText: **string**; AFont: TFont; AEffect: TCnAAEffect );  
检查可能带字体标签的字符串。如果找到相应的标签，删除字符串中的标签（包含 <> 标记），并用该标签定义的 Font 和 Effect 属性设置参数中的对应属性。

## 十七、TCnFontLabel 类

字体标签列表项类，TCnFontLabels 的子项，继承于 TCollectionItem，一般不需要用户直接创建。

**property** FontLabels: TCnFontLabels;

所有者。

**property** Name: **string**;

字体标签名。

**property** Font: TFont;

字体属性。

**property** Effect: TCnAAEffect;

平滑字体特效显示属性。

## 十八、TCnUserLabels 类

用户标签列表类，继承于 TOwnedCollection，定义了在本类控件中可使用的用户标签集，一般不需要用户直接创建。

**property** Items[Index: Integer]: TCnUserLabel; **default**;

用户标签项数组属性。

**function** AddItem(AName: **string**; AText: **string**; AStyle: TLabelStyle): TCnUserLabel;

增加一个新的用户标签。

**type** TLabelStyle = (lsLeftJustify, lsCenter, lsRightJustify, lsRegOwner, lsRegOrganization, lsAppTitle, lsDate, lsTime, lsCustom);

用户标签类型，分对齐控制标签和文本标签，文本标签在运行时被指定的文本取代：

- |                   |                             |
|-------------------|-----------------------------|
| lsLeftJustify     | - 左对齐标签，不可见控制标签，决定文本对齐方式。   |
| lsCenter          | - 中心对齐标签，不可见控制标签，决定文本对齐方式。  |
| lsRightJustify    | - 右对齐标签，不可见控制标签，决定文本对齐方式。   |
| lsRegOwner        | - 注册表中用户名标签，系统定义标签。         |
| lsRegOrganization | - 注册表中组织名标签，系统定义标签（NT 下无效）。 |
| lsAppTitle        | - 应用程序标题标签，系统定义标签。          |
| lsDate            | - 当前日期标签，系统定义标签。            |
| lsTime            | - 当前时间标签，系统定义标签。            |
| lsCustom          | - 用户自定义标签类型。                |

```
function IndexOf(const Name: string): Integer;
```

根据标签名查找子项索引。

```
procedure Check(var AText: string; var Align: TAlignment);
```

检查可能带文本标签的字符串。如果找到相应的标签，将文本中的标签（包含<>符号）用标签的 **Text** 属性取代，同时，还将调用标签的 **OnGetText** 事件。如果是对齐标签，将设置参数中的 **Align** 属性。

## 十九、TCnUserLabel 类

用户文本标签列表项类，TCnUserLabels 的子项，继承于 TCollectionItem，一般不需要用户直接创建。

```
property UserLabels: TCnUserLabels;
```

所有者。

```
property Name: string;
```

用户标签名。

```
property Text: string;
```

标签所对应的文本，运行期显示文本中的标签用该值代替。

```
property Style: TLabelStyle;
```

标签类型。

```
type TLabelStyle = (lsLeftJustify, lsCenter, lsRightJustify, lsRegOwner,  
lsRegOrganization, lsAppTitle, lsDate, lsTime, lsCustom);
```

用户标签类型，分对齐控制标签和文本标签，文本标签在运行时被指定的文本取代：

- |                   |                             |
|-------------------|-----------------------------|
| lsLeftJustify     | - 左对齐标签，不可见控制标签，决定文本对齐方式。   |
| lsCenter          | - 中心对齐标签，不可见控制标签，决定文本对齐方式。  |
| lsRightJustify    | - 右对齐标签，不可见控制标签，决定文本对齐方式。   |
| lsRegOwner        | - 注册表中用户名标签，系统定义标签。         |
| lsRegOrganization | - 注册表中组织名标签，系统定义标签（NT 下无效）。 |
| lsAppTitle        | - 应用程序标题标签，系统定义标签。          |
| lsDate            | - 当前日期标签，系统定义标签。            |
| lsTime            | - 当前时间标签，系统定义标签。            |
| lsCustom          | - 用户自定义标签类型。                |

```
property OnGetText: TGetTextEvent;
```

取用户标签所对应的文本事件，对系统标签也有效。

## 二十、TCnAAMask 类

平滑特效字体蒙板处理类，供绘制平滑字体时内部使用。

**property** Height: Integer;

蒙板图的高度。

**property** Width: Integer;

蒙板图的宽度。

**property** Quality: TAAQuality;

平滑字体绘制精度。

**procedure** Assign(Source: TPersistent); **override**;

对象赋值方法。

**procedure** DrawMask(Text: **string**);

绘制文本蒙板图。

**procedure** DrawMaskEx(Text: **string**; Extend: TSize; Point: TPoint );

绘制文本蒙板图增强版。

**procedure** Blur(Blur: TBlurStrength);

对蒙板图进行模糊处理。

**procedure** Outline;

对蒙板图进行轮廓处理。

**procedure** Spray(Amount: Integer);

对蒙板图进行喷溅处理。

**procedure** HorzMirror;

对蒙板图进行水平镜象处理。

**procedure** VertMirror;

对蒙板图进行垂直镜象处理。

**function** TextExtent(s: **string**): TSize;

返回文本高、宽。

**function** TextHeight(s: **string**): Integer;

返回文本高度。

**function** TextWidth(s: **string**): Integer;

返回文本宽度。

**function** ScanLine(Line: Integer): Pointer; **overload**;

返回蒙板图扫描线地址。

**function** Pixel(x, y: Integer): Byte;

返回蒙板图指定像素灰度值

**function** PixelAddr(x, y: Integer): Pointer;

返回蒙板图指定像素地址。

## 二十一、 TCnAABlend 类

平滑特效字体图像混合处理类，供绘制平滑字体时内部使用。

**property** ForeBmp: TBitmap;

字体前景图。

**procedure** Assign(Source: TPersistent); **override**;

对象赋值方法。

**procedure** Blend(x, y: Integer; AColor: TColor; Alpha: TAlpha; Mask: TCnAAMask);

按指定颜色进行混合。

**procedure** BlendEx(x, y: Integer; Alpha: TAlpha; Mask: TCnAAMask);

使用前景图 ForeBmp 进行混合。

## 二十二、 TCnAAFont 类

平滑特效字体绘制类，封装了基本的平滑字体绘制方法，用户可手动使用。

**property** Quality: TAAQuality;

平滑字体绘制精度。

**property** Canvas: TCanvas;

用于平滑字体绘制输出和文本尺寸检测的画布。

**constructor** Create( ACanvas: TCanvas ); **virtual**;

类构造器，参数为绘制平滑字体文本和计算文本大小时使用的画布，允许为 nil，如果为 nil，请在调用文本方法前对 Canvas 属性赋值。

```
procedure TextOut(x, y: Integer; s: string; Alpha: TAlpha = 100; Blur: TBlurStrength = 0);
```

输出平滑字体文本到当前设置的 **Canvas** 中，使用它的字体属性和画刷设置。如果要输出背景透明的文本，需要将 **Canvas.Brush.Style** 设为 **bsClear**。

注：该方法不支持多行文本。

**x, y: Integer** - 文本输出位置。  
**s: string** - 要绘制的字符串。  
**Alpha: TAlpha** - 文本的不透明度，默认为完全不透明。  
**Blur: TBlurStrength** - 文本的模糊度，默认为不进行模糊处理。

```
function TextExtent( s: string ): TSize; virtual;  
返回文本高、宽。
```

```
function TextHeight( s: string ): Integer; virtual;  
返回指定文本的显示高度，使用当前的 Canvas 属性。
```

```
function TextWidth( s: string ): Integer; virtual;  
返回指定文本的显示宽度，使用当前的 Canvas 属性。
```

## 二十三、 TCnAAFontEx 类

扩展的平滑特效字体绘制类，继承于 **TCnAAFont**，实现了阴影、渐变、纹理等特效。用户可手动 **TCnAAFontEx** 来绘制带特效的平滑字体文本，使用方法类似于 **TCnAAFont**。

```
property Effect: TCnAAEffect;  
平滑字体绘制时的特效参数。
```

```
property Quality: TAAQuality;  
平滑字体绘制精度。
```

```
property Canvas: TCanvas;  
用于平滑字体绘制输出和文本尺寸检测的画布。
```

```
constructor Create(ACanvas: TCanvas); override;
```

类构造器，参数为绘制平滑字体文本和计算文本大小时使用的画布。允许为 **nil**，如果为 **nil**，请在调用文本方法前对 **Canvas** 属性赋值。

```
function TextExtent(s: string): TSize; override;  
返回文本高、宽。
```

注：**Effect** 参数中的阴影、旋转角度等设置将影响返回结果。

---

```
procedure TextOut(x, y: Integer; s: string);
```

使用 **Effect** 设置的字体特效，输出平滑字体文本到当前设置的 **Canvas** 中，使用它的字体属性和画刷设置。如果要输出背景透明的文本，需要将 **Canvas.Brush.Style** 设为 **bsClear**。

注：该方法不支持多行文本。

**x, y: Integer** - 文本输出位置。  
**s: string** - 要绘制的字符串。

## 二十四、 TCnHotLink 类

平滑特效字体超链接标签控件超链接参数类，继承自 **TCnCustomParam**。

```
property Fade: Boolean;
```

是否允许淡入淡出显示。

```
property FadeDelay: Cardinal;
```

淡入淡出显示延时。

```
property Color: TColor;
```

高亮时的高亮时的字体颜色。

```
property BackColor: TColor;
```

高亮时的背景颜色。

```
property FontEffect: TCnAAEffect;
```

高亮时的字体特效参数。

```
property URL: string;
```

超链接内容或文件名，例如：

<http://www.cnpack.org> - 网页。

<mailto:master@cnpack.org> - 邮件地址。

<mailto:master@cnpack.org?subject=你好> - 带邮件标题的邮件地址链接。

<c:\tools\anyexe.exe> - 可执行文件。

<d:\cnaafont\readme.txt> - 文本文件等其它文件。

其它有效的超链接地址或文件名，相当于“开始”菜单中的“运行”命令。

```
property UnderLine: Boolean;
```

高亮时是否显示下划线。

```
property Transparent;
```

高亮时的透明设置。

**property** BackGround;

高亮时的背景图像。

**property** BackGroundMode;

高亮时的背景图像显示模式。

## 二十五、 示例与常见问题解答

### 1、 怎样在控件中使用标签？

支持多行文本的控件，允许使用标签来控制文本的对齐方式、字体特效和设置文本内容。这些标签的定义通常在控件的 **Fonts**、**Labels** 属性中。

**Fonts** 属性定义了字体标签集，**Labels** 属性定义了对齐标签和用户标签，字体标签和对齐标签为控制标签，不显示在正文中，根据标签作用范围对当前行或后继行生效，其它的标签为变量标签，文本显示时标签用对应的文本代替。在 **Labels** 中，系统定义了一些特殊类型的标签，用户可也根据需要更改或创建自己的标签。

使用标签时需要用一对尖括号 '<' '>' 将标签名括起来，标签名即 **Fonts**、**Labels** 属性子项的 **Name** 属性。

### 2、 怎样设置才可让 **TCnAAScrollText** 控件的内容向下滚动？

只需要将 **TCnAAScrollText** 控件的 **ScrollStep** 属性设置为负数就可以了。

### 3、 怎样使用 **TAAScrollText** 控件？

使用方法：

- 在窗体上放置一个控件。
- 确定你需要用的字体，修改 **Fonts** 属性，该属性允许你自定义字体及对应的标签。对每一种字体，**Font** 属性为显示字体，**Name** 为标签名，在 **Text** 下的 **Lines** 中使用，**Style** 为字体显示风格，可以定义透明度、阴影、模糊、渐变色等。你可以定义任意多的字体，也可以修改系统自带的字体。
- 确定你需要使用的变量，修改 **Labels** 属性，该属性允许你自定义显示变量。在文本属性中引用了该标签的地方将被替换为相应的内容。对每一个标签，**Name** 为标签名，**Style** 为标签种类，其中 **IsLeftJustify**、**IsCenter** 和 **IsRightJustify** 种类的标签用于控制文本对齐方式。只有 **IsCustom** 为用户自定义标签，其它均为特殊标签，可显示应用程序名、注册表中的用户名、单位名、时间、日期等信息。系统为每一种特定风格都带有一种标签，你也可以定义自己的标签。你需要将 **Style** 设定为 **IsCustom**，并在 **Text** 中指定它的值或在该标签的 **OnGetText** 事件中指定 **Text** 值。
- 修改控件的 **Text** 下的 **Lines** 属性，写入自己的内容。要控制显示字体和对齐方式或使用变量，请使用标签，标签用尖括号括起来。如 **<Left>** 为一个表示文本左对齐的标签。控制标签（包括字体和对齐）对当前行及后续行有效（受 **LabelEffect** 参数影响），可放在一行中任意位置。而变量标签则放在需要放置的地方。你可以参考控件默认的文本内容。

- 修改文本的其它属性，如淡入淡出、默认对齐方式、字体显示精度、背景色等。
- 修改滚动属性，如滚动速度、循环次数、循环等待时间、高度宽度等。
- 设置 **Active** 属性，决定是否滚动，**AutoUpdate** 决定是否每次更新内容都重新显示。
- 编写 **OnComplete**、**OnTextReady** 等事件，完成其它功能（可选）。

使用技巧：

- 该控件的最佳用途是作关于窗口。只要设置好循环次数，并在 **OnComplete** 事件中写上 **Close**，就可以实现关于窗口的自关闭。
- 如果文本中部分内容需在运行时确定，可以将其定义为标签，并在标签 **OnGetText** 事件中返回文本。
- 默认情况下修改控件属性将自动绘制显示文本，如果需要在运行时修改显示，可以将 **AutoUpdate** 属性置 **False**，在窗口的 **FormActivate** 或类似事件中进行设置，并调用 **Reset** 方法。也可以使用 **BeginUpdate** 通知控件暂停设置，更新完属性后再调用 **EndUpdate** 通知控件进行设置（推荐）。

#### 4、 使用 **TCnAAFont** 进行平滑输出的例子

```
var
  AAFont: TCnAAFont;
  W, H: Integer;
  S: string;
begin
  // 创建TCnAAFont 实例，并指定它使用PaintBox1 的画布进行绘制
  AAFont := TCnAAFont.Create(PaintBox1.Canvas);
  try
    with PaintBox1.Canvas do
      begin
        Font.Name := '宋体'; // 设置字体
        Font.Size := 24;
        Font.Color := clBlue;
        Brush.Style := bsClear; // 设置透明绘制
      end;
      S := '这是一个使用平滑字体的例子';
      W := AAFont.TextWidth(S);
      H := AAFont.TextHeight(S);
      with PaintBox1 do // 在控件中心输出文本
        AAFont.TextOut((Width - W) div 2, (Height - H) div 2, S, 80, 0);
      AAFont.Canvas := Image1.Canvas; // 也可以切换到另一画布
      AAFont.TextOut(10, 10, S); // 绘制时将使用Image1.Canvas 的字体属性
    finally
      AAFont.Free;
    end;
  end;
```