TNCComponent

Propertys Methods

This component is the base component for all Non Client Components which are intended to be compatable with TNComponentForm.

TNCComponent provides one abstract method, RePaint which MUST be overriden in any decendant. The purpose of this method is to paint the component as appropriate.

The are several methods used as the interface between TNCComponentForm and TNCComponent. These would normally never be overriden as incorrect operation of these methods would prevent the correct operation of the NCComponent.

RePaint Method

Applies to
All NCComponents

Declaration
procedure RePaint; virtual; abstract

Description
Repaint is called by TNCForm every time an NCComponent needs to repaint itself. It is the responsibity of the component writer to ensure that repainting is done.

All decendent NCComponents MUST override ths method.

See Also
TNCComponent
Painting NCComponents

MouseMove Method

Applies to
TNCComponent

Declaration
procedure MouseMove(MouseX, MouseY : integer); virtual;

Description
Once the left mouse button is pressed when the mouse pointer is over an NCComponent, this method is called every time the mouse is moved until the left mouse button is released.
MouseX and MouseY contain the screen co-ordinates of the pointer.

The default method does nothing

See Also
TNCComponent

LButton Method

Applies to
TNCComponent

Declaration
procedure LButton(ButtonState : TNCClickState); virtual;

Description
Every time the left mouse button is pressed or released this method is called. ButtonState is csUp if the button is now up (released) or csDown if now down (pressed).

The default method does nothing.

See Also
TNCComponent

TNCButton

Propertys Methods Events

TNCButton is a decendent of TNCComponent and provides additional buttons on the Caption Bar. These buttons can be placed either left or right and used the same as any of the standard Delphi button components.

TNCButton Propertys

ButtonLock Glyph

TNCButton Methods

LButton MouseMove
RePaint

TNCButton Events

OnClicked OnUnClicked

Non Client Components

NCComponents

Non-Client Components are Non Visual components which creates a control or some other component on the Caption Bar of a the window.

In order to use a Non Client component, it's parent form MUST be TNCComponentForm or a decendent. Note that if you use TForm the components won't work, but no error will be generated. To use TNCComponent form, you need to add **TNCForm** to the **uses** clause in your form then change

TMyForm = class (TForm)
to
TMyForm = class(TNCComponentForm)

At run time the NCComponent has access to the Caption Bar and can use it for whatever purpose it wishes, within the constraints of TNCComponent. All Non Client Components MUST decend from TNCComponent.

TNCComponentForm

Declaration
TNCComponentForm = class(TForm)

Description
TNCComponentForm is the heart of TNCComponents. The standards TForm provides no support for the actions required to perform the painting on the Caption Bar, and indeed anywhere in the Non Client Area.

TNCComponentForm works by using the Create method to scan it's components property for any TNCComponent. It then calls the ParentRegister method for each one. Once this is done, by intercepting several messages and taking appropriate action the NCComponent behave (I hope) as required. The relevent messages are;

WM_NCPAINT; WM_NCACTIVATE; WM_NCLBUTTONDOWN; WM_LBUTTONUP; WM_NCLBUTTONDBLCLK; WM_GETMINMAXINFO; WM_MOUSEMOVE; WM_ACTIVATE;

In addition to this, TNCComponentForm does to other things.

(1) Automatically adjusts it's minimum size so all NCComponents are always shown.
(2) Override the Caption property of NCForm so that it is displayed in centre of the caption bar remaining AFTER the NCComponents have been added. (This has no effect on the user as they still read and write the Caption Property, however , TNCComponentForm writes the caption to the title bar, not the default window procedure)

TNCComponent Propertys

VerticalOffset
CaptionHt
ParentHandle
RelativePaintPos
PosRight
PosLeft
BorderValid
Position
Width
DragBy
ParentActive

TNCComponent Methods

RePaint	LButton
RButton	LButtonDblClk
RButtonDblClk	MouseMove
GetPos	ParentRegister
IsCovered	ParentState

Non Client Components

TNCComponent

TNCBlank TNCButton
TNCLabel TNCClock

ParentRegister Method

Applies to
All NCComponents

Declaration
function ParentRegister(Wnd : HWnd; PaintPos, VOffSet,CaptionHt : integer): integer;

Description
This method is critical to the correct operation of all Non Client Components.

TNCComponentForm calls this method for each NCComponent in it's Components list. The parameters passed are :

Wnd : Hwd	Handle to the Parent Window. It is accessable though the ParentHandle Property.
PaintPos : integer	This is the position reletive to the Left or Right of the Window that the object is painted. This value is obtained from the RelativePaintPos Property.
VOffSet : integer	Vertical width of the Window's border. Obtained from VerticalOffSet Property.
CaptionHt : integer	Height of Parent's caption bar. Obtained from the CaptionHeight Property.

The return value is the width of the object however this is not currently used by TNCComponentForm.

ParentHandle Property

Applies to
All NCComponents

Declaration
property ParentHandle : HWnd

Property Type
Protected, Read only

Description
ParentHandle hold the Handle of the Parent Window. It may be used whenever a window handle is required , for example when drawing or painting on the window.

RelativePaintPos Property

Applies to
All NCComponents

Declaration
property RelativePaintPos : integer;

Property Type
Protected, Read/Write

Description
This property holds the relative horizontal position of the visible element of the NCComponent.
The actual horizontal position that an NCComponent paints itself if found from calling the GetPos method.

See also
Painting NCComponents

VerticalOffset Property

Applies to
All NCComponents

Declaration
property VerticalOffSet : integer;

Property Type
Protected, Read only

Description
This property holds the vertical position of the visible element of the NCComponent. This is the height of the window border

See also
Painting NCComponents

CaptionHeight Property

Applies to
All NCComponents

Declaration
property CaptionHeight : integer;

Property Type
Protected, Read only

Description
CaptionHeight is set to the height of the parent window's caption bar during the run-time initialisation.

PosRight Property

Applies to
All NCComponents

Declaration
property PosRight : integer;

Property Type
Protected, Read/Write

Description
This property is used to store the actual leftmost position of the object. When the NCComponent is RePainted, this value must be updated to reflect the new position. It is for use, along with PosLeft, by the MouseMove and IsCovered Methods.

See also
Painting NCComponents

ParentActive Property

Applies to
All NCComponents

Declaration
property ParentActive : boolean

Property Type
protected, Read Only

Description
If the parent window is the current active window then ParentActive is True, otherwise, ParentActive is False.This would be used when an NCComponent need to paint itself differently depaending on when the window is active or not.
e.g. If the component's colour needed to match the Caption Bar colour.

PosLeft Property

Applies to
All NCComponents

Declaration
property PosLeft : integer;

Property Type
Protected

Description
This property is used to store the actual leftmost position of the object. When the NCComponent is RePainted, this value must be updated to reflect the new position. It is for use, along with PosRight, by the MouseMove and IsCovered Methods.

See also
Painting NCComponents

BorderValid Property

Applies to
All NCComponents

Declaration
property BorderValid : boolean;

Property Type
Protected, Read only

Description
TNCComponent only supports TNCComponentForm BorderStyles of bsSizeable or bsSingle. If the parent border is either of these then BorderValid is True otherwise it is False. TNCComponents should NOT attempt to paint themselves if BorderValid is False.

GetPos Method

Applies to
All NCComponents

Declaration
function GetPos : integer;

Description
Return the leftmost position at which the object must be painted. The position is in screen co-ordinates.

Width Property

Applies to
All NCComponents

Declaration
property Width : integer;

Property Type
Published

Description
This property stores the width of the object. It is set at design time, and should never be altered at run time.

See Also
Painting NCComponents

Position Property

Applies to
All NCComponents

Declaration
property Position : TNCComponentPosition;

Property Type
Published

Description
Position stores the position of the object, either to the Left or Right of the screen. The default is rIght. Note that although it is accessable at run time it should NEVER be altered at run time.

See TNCComponentPosition

TNCComponentPosition

Declaration
TNCComponentPosition = (bpLeft,bpRight);

Description
If the position of an object is set to bpLeft then it is positioned as far left as possible on the caption bar. Similarly, if it is bpRight it is positioned to the right.
Components are positioned in order of creation, and that positioning takes into account the Parents BorderIcons property.

See Also
Position Property

IsCovered Method

Applies to
All NCComponents

Declaration
function IsCovered(MouseX, MouseY : integer) : boolean;

Description
When called with the position of the mouse pointer in MouseX and MouseY, the function returns True if the mouse in covering the object otherwise, False.

MouseX and MouseY are is screen co-ordinates.

DragBy Property

Applies to
All NCComponents

Declaration
property DragBy : boolean;

Property Type
Published

Description
DragBy is used by TNCComponentForm to decide whether a form can be dragged by the component. Normally a window can be dragged by the caption bar, except the buttons. DragBy defaults to False,

Generally Buttons or other controls which require the user to interact with them using the mouse must have DragBy set False, whereas a contol to display text or information could be dragged. When deciding whether to allow dragging using the control, the normal test would be the use of the LButton Method. If this actually does something, then DragBy must be False otherwise it can be True.

Painting NCComponents

Each NCComponent is responsible for painting itself. Each time it needs to do this, the TNCComponentForm will call the RePaint Method. Exactly what you choose your component to look like is entirely up to you, however, it must be painted within the bounds it was designed for. You cannot changed the size or relative position at run time.

The RePaint Method is also responsible for updating PosLeft and PosRight each time it repaints a component. These two values are found from

```
  PosLeft := GetPos;
  PosRight := PosLeft+Width;
```

NB : Failure to do this will prevent IsCovered, and therefore the Component, from working properly.

The area in which the component must be painted is

Top Left
```
        X := GetPos;
        Y:=   VerticalOffSet;
```

Botton Right
```
        X := GetPos + Width
        Y := VerticalOffSet + CaptionHeight;
```

In addition, the first line in the RePaint method should set the a protected TRect type variable ParentRect to the currect WindowRect by calling GetWindowRect(ParentHandle,ParentRect)

The following is a suggested template for the RePaint Method

```
procedure TMyNCComponent.RePaint;
var
  Pos : integer;
  WndDC : hDC;
begin
  GetWindowRect(ParentHandle,ParentRect);
  Pos := GetPos;
  WndDC := GetWindowDC(ParentHandle);
 ---------------------------------------------
  Component specific code here
  ---------------------------------------------
  ReleaseDC(ParentHandle,WndDC);
  PosLeft := Pos;
  PosRight := Pos+Width;
end;
```

WndDC provides the Device context on which the component can be painted and Pos gives the X co-ordinated of the left of the component.

For example the following code is the Paint method TNCButton (This method is called by RePaint with a flag to indicate whether the button is to be painted up or down)

```
begin
  GetWindowRect(ParentHandle,ParentRect);
```

```
  Pos := GetPos;
  WndDC := GetWindowDC(ParentHandle);
  BmDC := CreateCompatibleDC(WndDC);
  OldBmp := SelectObject(BmDC,FGlyph.Handle);
  if Up then
    BitBlt(WndDC,Pos,VerticalOffSet,Width,CaptionHeight,bmDC,0,0,SRCCOPY)
  else
    BitBlt(WndDC,Pos,VerticalOffSet,Width,CaptionHeight,bmDC,Width,0,SRCCOPY);
  SelectObject(bmDC,OldBmp);
  DeleteDC(bmDC);
  ReleaseDC(ParentHandle,WndDC);
  PosLeft := Pos;
  PosRight := Pos+Width;
end;
```

TNCClickState

Declaration
TNCClickState = (csUp,csDown);

Description
Calls to LButton and RButton pass the state of the button concerned as either csUp or csDown.

RButton Method

Applies to
All NCComponents

Declaration
procedure RButton(ButtonState : TNCClickState); virtual;

Description
Every time the right mouse button is pressed or released this method is called. ButtonState is csUp if the button is now up (released) or csDown if now down (pressed).

The default method does nothing

See Also
TNCComponent

 TNCBlank

TNCBlank adds a spacer of a selected Width to the caption bar. The spacer matches the colour of the title bar.
It is a decendant of TNCComponent and overides the Create and RePaint methods of that class. Other than this it does nothing.

TNCBlank can be used as a spacer, or new classes can be derived from it. (e.g TNCLabel)

TNCLabel

Propertys Methods

TNCLabel is a text display component which allows messages to be placed on the Caption Bar.

A display area is created in which the Caption is displayed. The alignment of the Caption within this window depends on the value of the TextAlign property.

TNCClock

Propertys Methods

TNCClock is a decendant of TNCLabel. It creates a timer which it uses to periodically read the current Date and Time. This is then displayed on the caption bar.

The timer and therefore the display can be turned on or off by use of the EnableTimer and DisableTimer methods. In addition, you can choose whether date, time or both are displayed by changing the Display Property.

ParentState Method

Applies to
All NCComponents

Declaration
procedure ParentState(Active : boolean); virtual;

Description
The ParentState method is called by TNCForm each time the parent window changes activation state. The default procedure sets the ParentActive property as appropriate.
You would not normally override this method, however it may be if necessary if processing was required when a window changed activation. (although TForm's OnActivate and OnDeactivate would normally be more appropriate).

LButtonDblClk Method

Applies to
TNCComponent

Declaration
procedure LButtonDblClk; virtual;

Description
Every time the left mouse button is double clicked this method is called.

The default method does nothing.

See Also
TNCComponent

RButtonDblClk Method

Applies to
All NCComponents

Declaration
procedure RButton; virtual;

Description
Every time the right mouse button is double clicked this method is called.

The default method does nothing.

See Also
TNCComponent

OnClick Event

Applies to

TNCButton

Description

The TNCButton is a push button control. When you click the button an OnClick event is generated. If ButtonLock is True, the OnClick Event is generated when the button is locked down. When it is clicked again to release the lock, the OnUnClick Event is generated.

OnUnClick Event

Applies to

TNCButton

Description

This event is ONLY generated when the ButtonLock is True. When the button is locked down an OnClicked event is generated. When the button is clicked again to unlock it, the OnUnClick Event is generated.

ButtonLock Property

Applies to
TNCButton

Declaration
property ButtonLock : boolean;

Property Type
published

Description
TNCButton can operate in two modes. With ButtonLock set to false it acts like a normal Push Button, i.e click the button to cause an OnClick event which the program can act upon. If ButtonLock is set to True however, clicking causes the button the be locked down. Again, an OnClick event is generated. When the button is clicked again it now returns to the up position and an OnUnClick event is generated.

This function would typically be used as some form of on/off switch.

Glyph Property

Applies to
TNCButton

Declaration
property Glyph : TBitmap

Property Type
Published

Description
The Glyph property hold the bitmap which is used to draw the button in the up and down positions. It must be the height is the Caption Bar and have a width twice the width property. The buttons must be stored next to each other, as shown below

The buttons can be designed in any way you like, e.g. On/Off switch, but must confirm to standard the TNCButton expects. Although the bitmap can be set at design time, if you were writing for a wide user base you may well wish to set the bitmap at run time, after determining the users system metrics.

See Also
Painting NCComponents

LButton Method

Applies to
TNCButton

Declaration
procedure LButton(ButtonState : TNCClickState); virtual;

Description
Every time the left mouse button is pressed or released this method is called. ButtonState is csUp if the button is now up (released) or csDown if now down (pressed).

LButton causes the Button to be repainted in it's new state then calls the OnClick or OnUnClick events as appropriate.

MouseMove Method

Applies to
TNCButton

Declaration
procedure MouseMove(MouseX, MouseY : integer); virtual;

Description
Once the left mouse button is pressed when the mouse pointer is over an NCComponent, this method is called every time the mouse is moved until the left mouse button is released.
MouseX and MouseY contain the screen co-ordinates of the pointer.

While the left mouse button is down, MouseMove checks to see whether the mouse has been move on to or off of the button and if so the button is repainted as appropriate.

RePaint Method

Applies to
TNCButton

Declaration
procedure RePaint; override;

Description
RePaint check whether the button is up or down, and whether ButtonLock is True or False, then paints the button as appropriate.

See Also
Painting NCComponents

TNCLabel Propertys

Caption TextColor
TextAlign BackgroundColor
ForceCaptionColor

Caption Property

Applies to
TNCLabel, TNCClock

Declaration
property Caption : TCaption;

Property Type
published

Description
Caption hold the text which is displayed in the by TNCLabel. The alignment of the text within the display area depends on the TextAlign property. In the TNCClock component, the caption is rewritten before display by the RePaint Method and is effectively, Read Only.

TTextAlign

Declaration
TTextAlign = (taLeft, taCenter, taRight);

Description
TTextAlign is used by TNCLabel to define where the text is displayed within the Components drawing area.

TextAlign Property

Applies to
TNCLabel

Declaration
property TextAlign : TTextAlign;

Property Type
published

Description
TextAlign defines how the Caption is displayed within the display area.

TextColor Property

Applies to
TNCLabel

Declaration
TextColor : TColor;

Property Type
published

Description
TextColor holds the color which the text of the TNCLabel will be drawn. The color will be the same regardless of whether the window is active or not. You can set the text colour to the normal caption colors by setting ForceCaptionColor to true. If you do this, the TextColor property is ignored at run time.

BackgroundColor Property

Applies to
TNCLabel

Declaration
BackgroundColor : TColor;

Property Type
published

Description
BackgroundColor holds the color which the background of the TNCLabel will be painted. The color will be the same regardless of whether the window is active or not. You can set the Background colour to the normal caption colors by setting ForceCaptionColor to true. If you do this, the BackgroundColor property is ignored at run time.

ForceCaptionColor Property

Applies to
TNCLabel

Declaration
ForceCaptionColor : boolean;

Property Type
published

Description
Normally the Text and Background of the NCComponent will be TextColor and BackgroundColor repectively. However, if ForceCaptionColor is true, the colours are the same as the normal caption bar i.e. clCaptionText and clActiveCaption. In addition, when the window becomes inactive, the colors change to clInactiveCaptionText and clInactiveCaption.

TNCLabel Methods

RePaint

RePaint Method

Applies to
TNClabel

Declaration
procedure RePaint; override;

Description
Repaint is called by TNCForm every time an NCComponent needs to repaint itself. It is the responsibity of the component writer to ensure that repainting is done.

A TRect structure(the display area) is created in which the text is drawn. This area is CaptionHeight x Width in size. The position of the text within this Rect is given by the TextAlign Property. The text and background colours are given by TextColor and BackgroundColor respectively unless ForceCaptionColor is set to true, in which case the colours will match those of the caption bar, including any color changes when the window changes activation state.

See Also
Painting NCComponents
ParentActive Property

TNCClock Propertys

Display

TNCClock Methods

EnableTimer DisableTimer
RePaint

Display Property

Applies to
TNCClock

Declaration
property Display : DisplaySet

Property Type
published

Description
Sets the display to either date, time or date and time.

DisplaySet , DisplayType

Declaration
DisplaySet = set of DisplayType
DisplayType = [dtDate, dtTime];

Description
DisplaySet is used by the TNCClock component to control what is displayed by the component.

EnableTimer Method

Applies to
TNCClock

Declaration
procedure EnableTimer;

Description
When called the Date/Time timer is enabled, this switches the display of the date and/or time on.

DisableTimer Method

Applies to
TNCClock

Declaration
procedure DisableTimer;

Description
When called the Date/Time timer is Disabled, this switches the display of the date and/or time off

RePaint Method

Applies to
TNCClock

Declaration
procedure RePaint; override;

Description
Repaint is called by TNCForm every time an NCComponent needs to repaint itself. It is the responsibity of the component writer to ensure that repainting is done.

As the TNCClock component is a decendant of TNCLabel it uses the inherited RePaint Method Before calling this method however, TNCClock sets the Caption to the desired Date and/or combination.

A TRect structure(the display area) is created in which the text is drawn. This area is CaptionHeight x Width in size. The position of the text within this Rect is given by the TextAlign Property. The text and background colours are given by TextColor and BackgroundColor respectively unless ForceCaptionColor is set to true, in which case the colours will match those of the caption bar, including any color changes when the window changes activation state.

See Also
Painting NCComponents
ParentActive Property

© Copyright 1995 by Stephen J Dibble

All Rights Reserved

First, the bad news. !

This package is supplied 'as is'. i.e. I do not accept any responsibilty for subsequencial loss if it lays waste to your hard disc or does anything undesirable and if you make use of this package you do so entirely at your own risk.

Now, the good news !

You are free to use the contents of this package as you like, **except** the contents may **NOT** be distributed as part of a library or other collection of components for **profit**. You may distribute as part of a free library providing the whole of the package is included .You may make use of these components in your software, and distribute that software for whatever profit you see fit.

If you have any problems, suggestions or bugs to report then please contact

Steve Dibble

CompuServe ID: 70630,2340.

or
72 Holmwood Road
Freezywater
Enfield
Middlesex
EN3 6QH
England