



Komponente HexEdit für Delphi 5.0

INHALTSVERZEICHNIS

Komponente HexEdit	4
Private - Funktion	5
Protected - Funktion.....	6
Public - Funktion	7
Published - Funktion	7
property Maxlength;Komponenten - Fehler	7
Komponenten - Fehler.....	8
Beispiel - Anwendung.....	8
Beispiel - Quellcode Teil 1.....	9



Microsoft, . Windows, Windows NT, Windows 2000, Windows XP sind Marken oder eingetragene Marken der Microsoft Corporation.

Delphi 5.0 sind Produkte der Inprise Corporation.

Die Nennung weitere Namen erfolgt in diesem Handbuch in der Regel ohne Erwähnung bestehender Patente, Gebrauchsmuster oder Warenzeichen. Das Fehlen eines entsprechenden Vermerks begründet nicht die Annahme, die Namen seien frei nutzbar. Alle Warenzeichen werden anerkannt.



Komponente HexEdit

Die Komponente HexEdit ist von mir auf den Delphi 5.0 umgesetzt worden. Diese Komponente ist aus der NumEdit - Komponente entstanden. Sie können nur die Zahlen 0 - 9 und die Buchstaben A - F eingeben. Alle anderen Buchstaben und Sonderzeichen werden unterdrückt. Ich war es leid, immer wieder den korrekten Dezimalwert berechnen zu lassen. Die Komponente wird auch in dem Eprombrenner andauernd benötigt. Die schon erwähnt, Sie können Hexwerte eingeben (z.B. 00 ... FF) und erhalten dann den korrekten Dezimalwert.

Da es hierbei um eine Ableitung von einer Standard - Komponente handelt, sollte diese ohne Problem bei Ihnen funktionieren. Es sind aber weitere Zusatzfunktionen hinzu gekommen die in dem Handbuch beschreiben werden. Alles andere wird dabei von mir ausgeklammert, da dieses in der Hilfe von Delphi ja vorhanden ist.

Die meisten Funktionen sind in der Protected Struktur versteckt und sollten auch nicht sichtbar gemacht werden durch eine Ableitung. Man muss hier eigentlich nichts ändern.

Die Aufschlüsselung der Komponente wird in den nachfolgenden Seiten erklärt.



Private - Funktion

Wie das Wort Private schon ausdrückt, sind hier alle Variablen hinterlegt, die man tunlichst nicht direkt ändern sollte.

FAbout	: THexEditAboutInfo;	
FNumRETURN	: boolean;	Ture = Kommastelle prüfen, sowie setzen.
FHexValue	: integer;	Hex-Wert von HexEdit
FCursor_Pos	: word;	Position vom Cursor
FCursor_MaxPos	: word;	Max Position vom Cursor

Die Methode Set_PointString setzt die Kommastelle und die nötigen Stellen hinter dem Komma.
Je nach Einstellung von PointShifting

```
procedure SetFHexvalue( value : integer );
```

Die Methode SetNumRETURN überprüft ob eine andere Komponente HexEdit mitteilt, das ein Windows-Fenster gelöscht wird und jetzt alle aktuellen Werte überprüft werden müssen.

ACHTUNG !!! Das muss jeder HexEdit - Komponente einzeln mitgeteilt werden.

```
procedure SetNumRETURN( Value : boolean );
```

Konvertierung der Hexwerte in Dezimalwerte.

```
procedure SetHextoint( Value : string );
```



Protected - Funktion

Die Methode erstellt eine Instanz von THexEdit.

constructor Create(AOwner:TComponent); override;

Botschaftsbehandlung für Ihre Komponenten modifizieren, müssen Sie sich über die gewünschte Wirkung genau im klaren sein.

procedure WndProc(var Message: TMessage); override;

Das Ereignis KeyPress wird ausgelöst, sobald der Benutzer eine einzelne Zeichentaste drückt. Auch hier muss man sich klaren was erreicht werden soll.

procedure KeyPress (var Key: char); override;

Das Ereignis KeyDown wird ausgelöst, sobald der Benutzer eine Taste drückt. Diese wurde auch verändert für NumEdit.

procedure KeyDown (var Key: Word; Shift: TShiftState); override;



Public - Funktion

Hier sind alle Funktionen und Variablen die für die Öffentlichkeit zugänglich sind.

Die anderen Methode sind so geblieben wie im Original.

Published - Funktion

Endlich kommt der Objektinspektor zum Zug. Alles was sichtbar sein soll, in der Komponenten wird hier angezeigt.

Info über die Komponente

property About : THexEditAboutInfo read FAbout write FAbout;

Die Eigenschaft NumRETURN überprüft ob eine andere Komponente HexEdit mitteilt, das ein Windows-Fenster gelöscht wird und jetzt alle aktuellen Werte überprüft werden müssen. ACHTUNG das muss jedem HexEdit einzeln mitgeteilt werden.

property NumRETURN : boolean read FNumRETURN write SetNumRETURN;

Mit der Eigenschaft PointShifting, wird die Kommastelle festgelegt.

property PointShifting : integer read FPointShifting write SetPointShifting;

Mit der Eigenschaft Hexvalue wird der ermittelte Hexwert als Dezimalwert an die Komponente übergeben.

property Hexvalue : integer read Fhexvalue write setFhexvalue;

Mit der Eigenschaft Maxlength legt die maximale Eingabelänge fest.

property Maxlength;

Komponenten - Fehler

Standard Fehlerarten können bei der Verwendung der Komponente auftreten.

Beispiel - Anwendung

Eine Beispielanwendung ist der Komponente beigelegt und sollte die grundsätzlichen Funktionen erklären.

Es wurde eine HexEdit - Komponente und ein Button verwendet.



Das von mir geschriebene Beispiel, ist eine klassische Komponenten - Testumgebung. Wenn die Komponente in der Komponentenleiste eingebunden ist, so muss man die SELF - Variante nicht machen.

Sie auf den folgenden Seiten das Beispiel.

Und viel Spaß beim ausprobieren.

Ich übernehme keine Haftung bei der Fehlerhaften Bedienung der Komponente. Sollte es durch die Komponente, einen gravieren Schaden auf Ihren PC geben, so kann ich auch hier nicht Haften, da nicht ausgeschlossen werden kann, das andere Programme die Sie verwenden, hierfür die Ursache sind.



Beispiel - Quellcode Teil 1

```
unit Testkomp_Unit;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  HexEdit, StdCtrls, Buttons, ExtCtrls;

type
  TForm1 = class(TForm)
    Label15: TLabel;
    Button2: TButton;
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  Form1: TForm1;
  xNum : THexEdit;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  xNum := THexEdit.create(self);
  XNum.parent := self;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  label15.caption := inttostr( xnum.Hexvalue );
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  xNum.free;
end;

end.
```