void \_\_fastcall TForm1::grdMainCellColoring(TObject \*Sender, int ACol, int ARow, TColor &CellColor, TColor &GridColor, TCellState CellState)

{

TRect R;

TField\* Field;

if (ACol == 1)

{

Field = NextDBGrid1->Columns->Items[ACol]->Field;

if (Field != NULL && !Field->IsNull)

{

R = NextDBGrid1->CellRect(ACol, ARow);

InflateRect(&R, -3, -2);

NextDBGrid1->Canvas->Pen->Color = clGrayText;

NextDBGrid1->Canvas->Brush->Color = (TColor)Field->AsInteger;

NextDBGrid1->Canvas->Rectangle(R);

}

}

}

In diesem Code:

* **NextDBGrid1** ist eine Instanz von **TNextDBGrid**, die Sie in Ihrem Formular erstellt haben.
* **TField** ist eine Klasse, die ein Feld in einer Datenbanktabelle repräsentiert.
* **CellRect** ist eine Methode, die das Rechteck einer Zelle in Pixelkoordinaten zurückgibt.
* **InflateRect** ist eine Funktion, die die Größe eines Rechtecks ändert.
* **clGrayText** ist eine Konstante, die eine Graufarbe darstellt.
* **Canvas** ist eine Eigenschaft, die eine Zeichenfläche darstellt, auf der Sie zeichnen können.
* **Field->AsInteger** konvertiert den Wert des Felds in einen Integer. Sie müssen sicherstellen, dass das Feld einen Integer-Wert enthält, um Laufzeitfehler zu vermeiden.

\_\_

CheckBox zeichnen

void \_\_fastcall TNxDBCheckBoxColumnDisplay::Paint()

{

bool Value;

TNxDBCheckBoxColumn\* Column = dynamic\_cast<TNxDBCheckBoxColumn\*>(this->Column);

if (SameText(this->AsString, Column->ValueChecked.c\_str()))

Value = true;

else if (SameText(this->AsString, Column->ValueUnchecked.c\_str()))

Value = false;

else

return;

DrawCheckBoxState(this->ClientRect, Value, false, false);

}

**SameText** ist eine Funktion, die in der **System::Sysutils**-Einheit in C++ Builder definiert ist. Sie vergleicht zwei Zeichenketten unabhängig von Groß- und Kleinschreibung.

* **DrawCheckBoxState** ist eine Funktion, die Sie definieren müssen, um den Zustand des Kontrollkästchens zu zeichnen. Sie könnte so aussehen:

void DrawCheckBoxState(TRect Rect, bool Checked, bool Enabled, bool Focused)

{

UINT State = DFCS\_BUTTONCHECK;

if (Checked)

State |= DFCS\_CHECKED;

if (!Enabled)

State |= DFCS\_INACTIVE;

DrawFrameControl(Canvas->Handle, &Rect, DFC\_BUTTON, State);

if (Focused)

{

TRect FocusRect = Rect;

InflateRect(&FocusRect, -1, -1);

Canvas->DrawFocusRect(FocusRect);

}

}

In diesem Code:

* **DrawFrameControl** ist eine Funktion, die ein Kontrollkästchen auf einem Gerätekontext zeichnet.
* **DFCS\_BUTTONCHECK**, **DFCS\_CHECKED** und **DFCS\_INACTIVE** sind Konstanten, die den Zustand des Kontrollkästchens darstellen.
* **Canvas->Handle** ist der Gerätekontext, auf dem gezeichnet wird.
* **Canvas->DrawFocusRect** ist eine Funktion, die einen Fokusrechteck auf einem Gerätekontext zeichnet.

Bitte beachten Sie, dass Sie die richtigen Bibliotheken einbinden und die richtigen Namensräume verwenden müssen, um diesen Code zu kompilieren. Es könnte auch notwendig sein, einige Anpassungen vorzunehmen, um ihn in Ihrem spezifischen Kontext zu verwenden. Insbesondere müssen Sie sicherstellen, dass **Canvas** korrekt definiert ist und auf einen gültigen Gerätekontext verweist. Sie könnten **Canvas** durch **NextDBGrid1->Canvas** ersetzen, wenn Sie innerhalb einer Methode von **NextDBGrid1** arbeiten.