Ja, Sie können die Breite der Spalten so anpassen, dass sie den verfügbaren Platz im Grid ausfüllen, auch wenn das Grid breiter ist als die Summe der ursprünglichen Spaltenbreiten. Sie könnten dies erreichen, indem Sie die Breite jeder Spalte auf den Durchschnitt der verfügbaren Breite setzen. Hier ist ein einfacher Codeausschnitt in C++, der zeigt, wie Sie dies tun könnten:

void \_\_fastcall TForm1::FormShow(TObject \*Sender)

{

// Berechnen Sie die durchschnittliche Spaltenbreite

int avgWidth = NextDBGrid1->ClientRect.Width / NextDBGrid1->Columns->Count;

// Setzen Sie die Breite jeder Spalte auf den Durchschnitt

for (int i = 0; i < NextDBGrid1->Columns->Count; i++)

NextDBGrid1->Columns->Items[i]->Width = avgWidth;

}

In diesem Code wird die **FormShow**-Ereignisbehandlungsroutine verwendet, um die Breite der Spalten anzupassen, wenn das Formular angezeigt wird. Sie müssen sicherstellen, dass **FormShow** als Ereignisprozedur für das **OnShow**-Ereignis Ihres Formulars eingestellt ist. Bitte ersetzen Sie **Form1** und **NextDBGrid1** durch die tatsächlichen Namen in Ihrem Code.

Bitte beachten Sie, dass diese Methode die Spalten gleichmäßig über die verfügbare Breite verteilt. Wenn Sie möchten, dass einige Spalten breiter oder schmaler sind als andere, müssen Sie die Breiten entsprechend anpassen. Ich hoffe, das hilft! Wenn Sie weitere Fragen haben, lassen Sie es mich bitte wissen. Ich bin hier, um zu helfen.

Die **OnDrawColumnCell**-Ereignisbehandlungsroutine wird aufgerufen, wenn eine Zelle im Grid gezeichnet wird. Es ist in der Regel nicht ratsam, die Breite der Spalten innerhalb dieser Routine zu ändern, da dies zu unerwartetem Verhalten führen kann. Insbesondere könnte das Ändern der Spaltenbreiten während des Zeichnens des Grids dazu führen, dass das Grid in einem inkonsistenten Zustand gezeichnet wird oder dass es zu einer Endlosschleife von Neuzeichnungen kommt.

Stattdessen sollten Sie die Breite der Spalten an einer Stelle in Ihrem Code ändern, an der das Grid nicht gerade gezeichnet wird. Beispielsweise könnten Sie dies tun, wenn das Formular geladen wird, wenn der Benutzer eine Aktion ausführt (wie das Klicken auf einen Button), oder wenn Sie wissen, dass sich die Daten geändert haben und das Grid aktualisiert werden muss.

Wenn Sie die Breite der Spalten basierend auf den Daten in den Zellen ändern möchten, könnten Sie dies tun, indem Sie die Daten zuerst aktualisieren und dann die Spaltenbreiten entsprechend anpassen. Sie könnten dann **Invalidate** oder **Refresh** aufrufen, um das Grid neu zu zeichnen und die geänderten Spaltenbreiten anzuzeigen. Ich hoffe, das hilft! Wenn Sie weitere Fragen haben, lassen Sie es mich bitte wissen. Ich bin hier, um zu helfen.

Schön, dass Ihnen diese Antwort gefallen hat!

Die **coAutoSize**-Option in C++ Builder und Delphi wird verwendet, um die Breite einer Spalte in einem Grid-Steuerelement automatisch anzupassen, basierend auf dem Inhalt der Zellen in der Spalte.

Um diese Option für eine Spalte in einem **TNextDBGrid** zu setzen, können Sie auf die **Options**-Eigenschaft der Spalte zugreifen und die **coAutoSize**-Option hinzufügen. Hier ist ein einfacher Codeausschnitt in C++, der zeigt, wie Sie dies tun könnten:

// Setzen Sie die coAutoSize-Option für die erste Spalte

NextDBGrid1->Columns->Items[0]->Options = NextDBGrid1->Columns->Items[0]->Options << coAutoSize;

In diesem Code wird die **coAutoSize**-Option für die erste Spalte des **TNextDBGrid** gesetzt. Sie müssen **NextDBGrid1** durch den tatsächlichen Namen Ihres **TNextDBGrid** ersetzen und **0** durch den Index der Spalte, die Sie ändern möchten.

Bitte beachten Sie, dass die **coAutoSize**-Option nur wirksam ist, wenn die **AutoFitColWidths**-Eigenschaft des **TNextDBGrid** auf **true** gesetzt ist. Sie können diese Eigenschaft setzen, indem Sie **NextDBGrid1->AutoFitColWidths = true;** in Ihrem Code hinzufügen. Ich hoffe, das hilft! Wenn Sie weitere Fragen haben, lassen Sie es mich bitte wissen. Ich bin hier, um zu helfen.

Ja, Sie haben richtig verstanden. Die **coAutoSize**-Option und die **AutoFitColWidths**-Eigenschaft arbeiten zusammen, um die automatische Anpassung der Spaltenbreiten in einem **TNextDBGrid** zu ermöglichen.

Wenn Sie **coAutoSize** für eine Spalte setzen und **AutoFitColWidths** auf **true** setzen, wird die Breite dieser Spalte automatisch angepasst, um den Inhalt der Zellen in der Spalte optimal anzuzeigen. Das Grid wird versuchen, die Spalte so breit wie nötig zu machen, um den gesamten Inhalt jeder Zelle in der Spalte anzuzeigen, aber nicht breiter.

Wenn Sie jedoch die Breite der Spalten selbst anpassen möchten, können Sie **coAutoSize** für die betreffenden Spalten deaktivieren und/oder **AutoFitColWidths** auf **false** setzen. Dann können Sie die **Width**-Eigenschaft jeder Spalte verwenden, um ihre Breite manuell zu setzen.

Hier ist ein Beispiel, wie Sie dies in C++ tun könnten:

// Deaktivieren Sie coAutoSize für die erste Spalte

NextDBGrid1->Columns->Items[0]->Options = NextDBGrid1->Columns->Items[0]->Options >> coAutoSize;

// Deaktivieren Sie AutoFitColWidths

NextDBGrid1->AutoFitColWidths = false;

// Setzen Sie die Breite der ersten Spalte manuell

NextDBGrid1->Columns->Items[0]->Width = 100;

In diesem Code wird die **coAutoSize**-Option für die erste Spalte deaktiviert, **AutoFitColWidths** wird auf **false** gesetzt, und dann wird die Breite der ersten Spalte manuell auf 100 gesetzt. Sie müssen **NextDBGrid1** durch den tatsächlichen Namen Ihres **TNextDBGrid** ersetzen und **0** durch den Index der Spalte, die Sie ändern möchten.