

Socket-Komponenten ab Delphi 7

Nachinstallation und dynamische Verwendung

von Narses

Version 1.00 vom 13.09.2006

Zielsystem: Delphi 7 Win32VCL

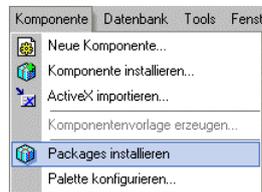
1 Voraussetzungen

Seit der Delphi-Version 7 sind die Socket-Komponenten (TClientSocket und TServerSocket) nicht mehr standardmäßig in der Komponentenpalette verfügbar, aber noch im Lieferumfang enthalten. Es handelt sich dabei um die Datei `dclsockets70.bpl` bzw. eine entsprechende Datei, falls es eine neuere Delphi-Version ist.

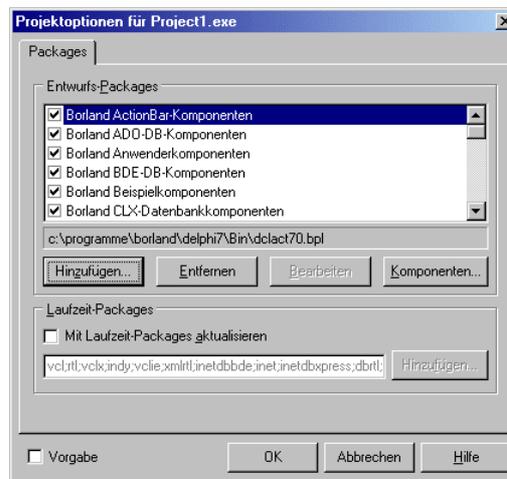
Leider ist bei den Delphi Personal Editions diese Datei nicht mit im Lieferumfang enthalten, Borland wünscht also offensichtlich nicht, dass PE-Benutzer die IDE-Integration verwenden dürfen. Die Verwendung dieser Datei ist demnach in einer PE-Version illegal. Das ist aber kein Grund, auf die Socket-Komponenten verzichten zu müssen, denn es ist statt dessen die Datei `ScktComp.dcu` mit dabei, so dass eine dynamische Verwendung der Komponenten möglich und auch legal ist. Ein Beispiel dazu ist in Kapitel 3 angegeben.

2 Die Socket-Komponenten nachinstallieren (nicht in PE-Versionen)

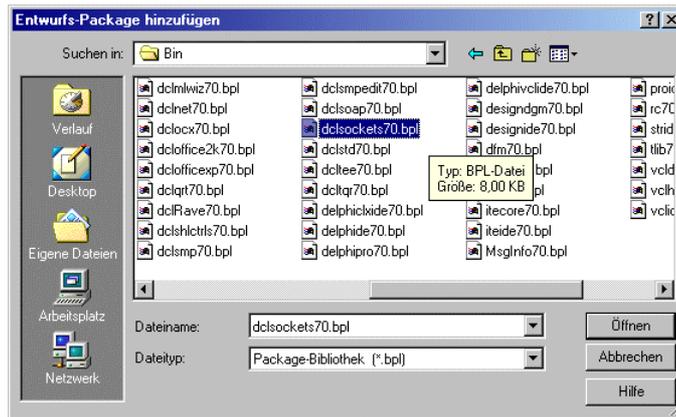
Wir klicken im Menü „Komponente“ auf den Befehl „Packages installieren“:



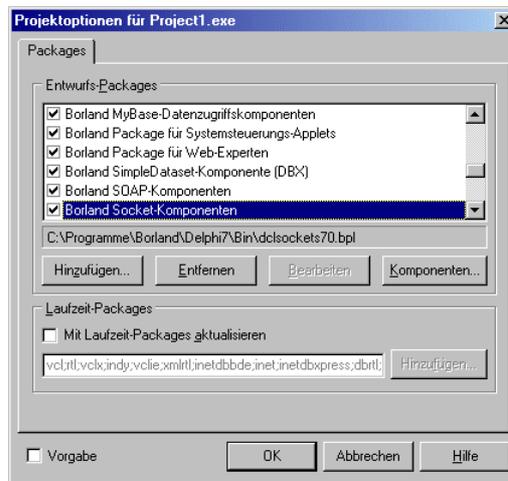
Im folgenden Dialog wählen wir den Button „Hinzufügen...“:



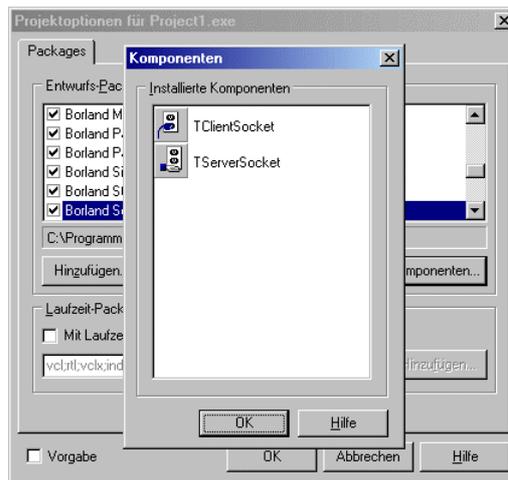
Dann suchen wir im `\bin`-Verzeichnis der Delphi-Installation (normalerweise `C:\Programme\Borland\Delphi7\Bin`) die Datei `dclsockets70.bpl` bzw. eine entsprechende Datei, falls es eine neuere Delphi-Version ist und klicken auf „Öffnen“:



Das Package mit den Socketkomponenten wird jetzt neu in der Liste angezeigt:

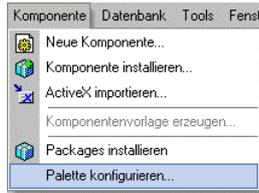


Wir klicken auf den Button „Komponenten...“, um zu prüfen, ob die richtigen Komponenten installiert wurden. Es sollte etwa folgendes Bild angezeigt werden:

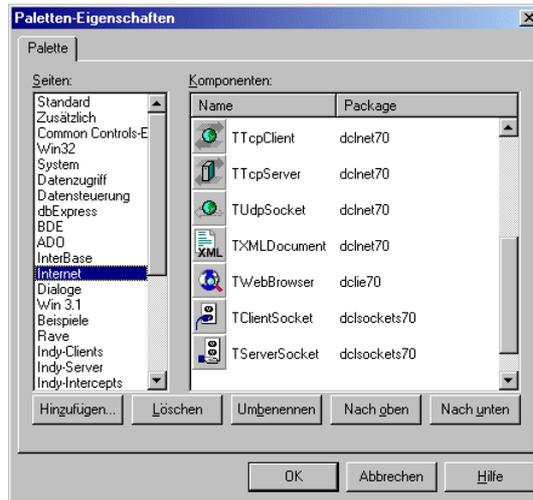


Wir schließen beide Dialog mit „OK“.

Falls die Komponenten jetzt noch nicht im Register „Internet“ angezeigt werden, können wir die Komponentenpalette noch anpassen. Dazu klicken wir im Menü „Komponente“ auf den Befehl „Palette konfigurieren...“:



Im folgenden Dialog können wir nun links das Register „Internet“ auswählen und rechts die darin befindlichen Komponenten gegebenenfalls anordnen:



Wenn die Socketkomponenten erfolgreich hinzugefügt wurden, sollte das Register „Internet“ etwa so aussehen:



Die erste Komponente ist der TClientSocket, die zweite der TServerSocket.

3 Socket-Komponenten dynamisch verwenden (auch PE-Versionen)

Die Socket-Komponenten lassen sich natürlich auch dynamisch anlegen und verwenden. Es fehlt dann lediglich die IDE-Integration, funktionale Einschränkungen gibt es nicht. Hier ein Beispiel mit dem TClientSocket:

```

uses
  ..., ScktComp;

type
  TForm1 = class(TForm)
  procedure DynConnectEvent(Sender: TObject; Socket: TCustomWinSocket);
  public
    ClientSocket1: TClientSocket; // dynamisch erzeugter TClientSocket
  end;

// bei Programmstart ausführen
procedure TForm1.FormCreate(Sender: TObject);
begin
  // Objekt erstellen; Eigentümer ist das Formular -> so wird das Objekt beim
  // Beenden automatisch mit freigegeben
  ClientSocket1 := TClientSocket.Create(Form1);
  // Ereignisse eintragen
  ClientSocket1.OnConnect := DynConnectEvent;
  // fertig, ClientSocket dynamisch erstellt
  ClientSocket1.Port := 12345; // Port eintragen
  ClientSocket1.Active := TRUE; // und aktivieren
end;
    
```