

Delphi OplossingsCourant

Vol. 8. No. 1, Een gratis elektronische publicatie van Bob Swart Training & Consultancy (eBob42) - <http://www.eBob42.com>



Helmond, 5 januari 2006,

Welkom bij het nieuwe nummer van de Delphi OplossingsCourant. Allereerst wil ik iedereen een gelukkig en vooral gezond 2006 wensen. Met minder spam, minder bugs en meer productiviteit. En of de nieuwe Delphi 2006 in mijn ogen daarbij nog een rol kan spelen is verderop in dit nummer te lezen.

Borland Developer Studio 2006 is eind vorig jaar uitgekomen, en bevat naast Delphi en Delphi for .NET nu ook C++Builder (alsmede C# natuurlijk). Behalve Win32 alleen nog .NET 1.1. ondersteuning overigens. Voor .NET 2.0 support moeten we wachten op **Delphi "Highlander"**, die pas medio volgend jaar (2006) zal verschijnen.

Voor geregistreerde Delphi 2006 gebruikers is ook al de **eerste update** van Borland Developer Studio 2006 te downloaden. Deze bevat met name verbeteringen voor de C++Builder persoonlijkheid.

Het grote artikel deze keer gaat helemaal over Borland Developer Studio 2006, en met name dan de nieuwe features in Delphi 2006.

Eind vorig jaar was ik al meer gaan samenwerken met de **Software Development Network (SDN)** op het gebied van trainingen en cursusboeken voor SDN-leden. Vanaf 2006 ben ik ook actief als SDN Delphi Medewerker, wat wil zeggen dat ik meer betrokken ben bij de Delphi gerelateerde zaken voor de website, het magazine, en de SDN events zoals de SDE en de SDC. Zie <http://www.sdn.nl> voor meer informatie.

Veel leesplezier, en schroom niet om op- of aanmerkingen te mailen of in mijn weblog achter te laten te <http://www.eBob42.com/weblog>.

Groetjes,

Nieuwe Delphi 2006 Clinics

In de eerste helft van dit jaar geef ik weer een achttal nieuwe trainingen over en met Delphi 2006 (maar ook gebruikers van Delphi 2005, 8, 7, 6 of lager zijn welkom natuurlijk).

19 januari

Delphi 2006 Win32/.NET Development Essentials

<http://www.eBob42.com/training/2006.htm>

2 februari

ADO.NET 1.1 Database Development

<http://www.eBob42.com/training/2006ado.htm>

16 maart

ASP.NET 1.1 Web Development

<http://www.eBob42.com/training/2006asp.htm>

30 maart

Enterprise Core Objects III Development

<http://www.eBob42.com/training/2006eco.htm>

13 april

XML, SOAP and Web Services Development

<http://www.eBob42.com/training/2006xml.htm>

27 april

.NET Compact Framework Development

<http://www.eBob42.com/training/2006cf.htm>

8 juni

Delphi Custom Component Development

<http://www.eBob42.com/training/2006comp.htm>

22 juni

Advanced ASP.NET Web Development

<http://www.eBob42.com/training/2006asp2.htm>

Losse cursusboeken zijn ook te koop, maar pas beschikbaar nadat de training voor het eerst is gegeven. Zie <http://www.eBob42.com/courseware>.

Wie liever alleen of met z'n collega komt is van harte welkom in Helmond voor een workshop. En wie liever zelf op kantoor blijft kan mij uitnodigen voor een custom clinic, dan vullen we de onderwerpen en agenda helemaal op maat in. Zie <http://www.eBob42.com/workshop> voor alle mogelijkheden, of stuur vrijblijvend een mailtje.

Borland Developer Studio 2006

Delphi en C++Builder voor Win32 samen met C# en Delphi voor .NET in één doos

Op 10 oktober 2005 werden Delphi 2006, C++Builder 2006 en C#Builder 2006 aangekondigd [zie http://www.borland.com/us/company/news/press_releases/2005/10_10_05_borland_delphi_and_cpp_builder_2006_deliver_key_new_features.html], als de Borland Developer Studio (BDS 4.0).

Op dinsdag 29 november 2005 werd Borland Developer Studio (BDE) 2006 in Nederland gelanceerd [zie http://info.borland.de/edm/EMEA/Invitation_Delphi_2006_Netherlands_Web.html]. Tijdens de lancering mocht ik weer een geslaagde presentatie doen (over nieuwe features in Enterprise Core Objects III. Ook was ik aanwezig bij de HCC Gebruikersdag op zaterdag 10 december, en heb daar toen al laten zien wat er mogelijk was.

Inmiddels zijn de eerste dozen met Borland Developer Studio 2006 ontvangen. Er is zelfs al een eerste update beschikbaar en voor geregistreerde gebruikers van Borland Developer Studio 2006 te downloaden van http://www.borland.com/downloads/registered/download_delphi.html. Deze update is met name bedoeld is om de C++Builder persoonlijkheid op te krikken van de "preview" status naar de officiële "release" status, en zal later nog gevolgd worden door een tweede update (dus wie nu een bug vindt in Delphi 2006 wordt verzocht dat via Quality Central <http://qc.borland.com> te melden, zodat het hopelijk al meegenomen kan worden in de volgende update).

Vergeleken met Delphi 2005, zit er nu een extra omgeving in de doos, namelijk C++Builder 2006. Borland Developer Studio 2006 is wat dat betreft een opvolger van C++Builder 6 (een update waar we sinds begin 2003 op hebben zitten wachten). Het is soms wel een beetje onduidelijk, want op de doos kan "Delphi 2006" of "C++Builder 2006" staan (of zelfs C#Builder 2006), maar uiteindelijk is het allemaal hetzelfde product: Borland Developer Studio 2006 met vier persoonlijkheden.

Als de IDE eenmaal draait dan vallen weer een hoop nieuwe dingen op ten opzichte van vorige nieuwe versies. De Object Repository laat uiteraard alle nieuwe projectmogelijkheden van de vier persoonlijkheden zien.

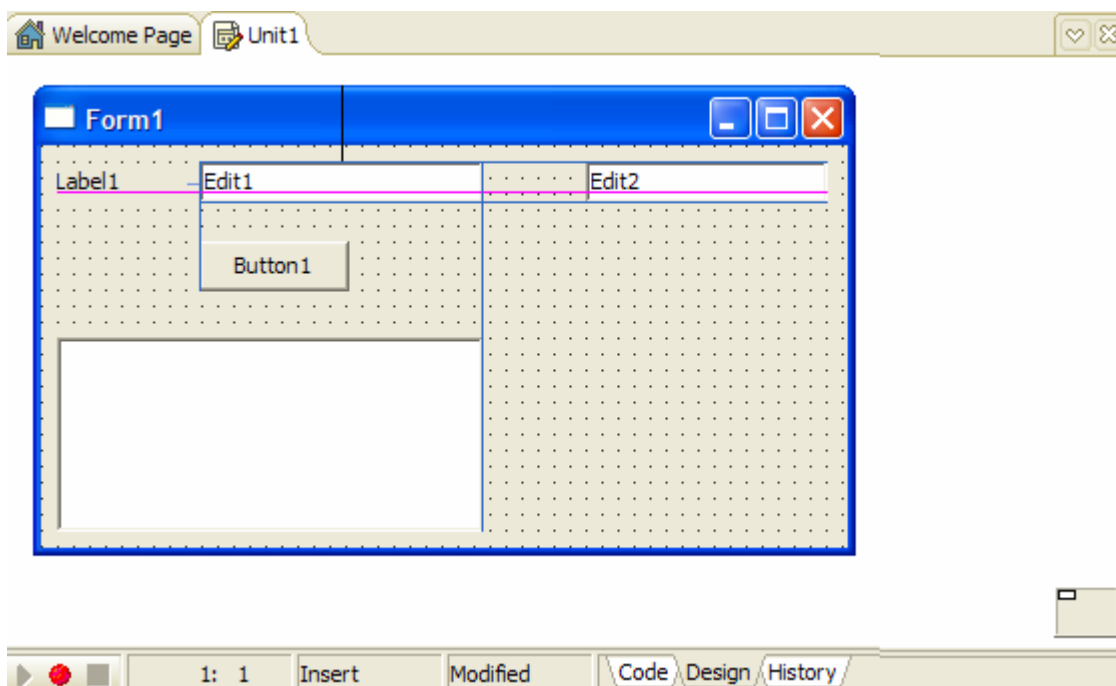
Vergeleken met Delphi 7 (of Delphi 8 for .NET) hebben we een enorme extra hoeveelheid ontwikkelmateriaal. En dat laat zich merken ook, vooral bij het opstarten. De Delphi 2005 IDE was vele malen trager met starten dan de Delphi 6 of Delphi 7 IDE. Bovendien had de Delphi 2005 IDE regelmatig last van geheugenlekken of andere geheugenproblemen, waardoor hij nog wel eens wou vastlopen (of erg traag werd). Voor Delphi 2006 (of BDS 2006 zoals het eigenlijk heet) zijn deze problemen voor het grootste deel opgelost door gebruik te maken van een nieuwe memory manager (de FastCode FastMM 4). Alleen daardoor al is de opstarttijd van BDS 2006 - inclusief C++Builder - bijna 30% sneller dan die van Delphi 2005. Behalve de gehele BDS eenheid, waarbij je voor alle vier de mogelijkheden projecten kunt openen en starten, worden er nu ook shortcuts geïnstalleerd voor de losse persoonlijkheden. Je kan dus ook alleen Delphi 2006 voor Win32 starten. En alhoewel die nog steeds langzamer start dan Delphi 7, is die weer sneller dan de gehele BDS 2006 omgeving.

Let op: ook al kun je ervoor kiezen om bijvoorbeeld alleen Delphi 2006 voor Win32 te starten, de IDE zal toch het .NET Framework nodig hebben om te kunnen draaien. Vandaar ook dat tijdens installatie wordt gecontroleerd of het .NET Framework versie 1.1 aanwezig is, en desgewenst wordt geïnstalleerd op je machine. Dat geldt uiteraard niet voor de Win32 toepassingen die je met Delphi 2006 voor Win32 maakt: die hebben alleen Win32 nodig. Wellicht is het hierbij nuttig op te merken dat de ondersteuning voor Windows 95, Windows 98, Windows NT en Windows ME officieel niet meer aanwezig is. Delphi 2006 Win32 toepassingen hebben officieel minimaal Windows 2000, XP of 2003 nodig voor deployment. Natuurlijk draait het vaak nog best onder NT, 98 of zelfs 95, maar dat is allemaal "op eigen risico", en wordt officieel niet meer ondersteund.

Voor .NET toepassing geldt dat Borland Developer Studio 2006 alleen het .NET Framework versie 1.1 ondersteunt, en niet de recent uitgekomen .NET Framework versie 2.0.

VCL Designer

De VCL Designer is uitgebreid met zogenaamde Live Guidelines, die je vertellen wanneer controls netjes op een lijn staan (zowel boven, onder, links of rechts), een redelijke afstand tot elkaar of de border van het form hebben, en dat de tekst van een label en edit op een lijn staan. De zwarte lijnen geven aan dat de componenten op een goede afstand van de border of het frame afstaan, de blauwe lijnen geven aan dat de componenten aan de boven, onder of zijkant netjes gelijk staan, terwijl een horizontale roze lijn aangeeft dat de tekst zelf (binnen de TLabel en TEdit etc.) op gelijke hoogte staat. Behalve de snap-to-grid functionaliteit die we al kende, hebben we nu ook de mogelijkheid om de componenten naar een blauwe of roze lijn te laten "snappen".



Ik moet zeggen dat deze wellicht kleine nieuwe feature in praktijk heel erg fijn werkt, en als ik af en toe terugschakel naar Delphi 7, dan mis is hem enorm. Het kost veel meer tijd om via de rechtermuisknop de Align properties goed te zetten.

Een andere fijne feature is rechtsonder de VCL Designer te vinden, en kunnen we gebruiken om de default positie van de VCL forms te specificeren. Het betreft een klein vierkantje dat de desktop representeert, en waarbinnen een nog kleiner vierkantje het VCL form aangeeft (zie ook de screenshot hierboven, waarin beide vierkantjes te zien zijn). Standaard staat het kleinstste vierkantje linksboven, maar door nu het kleinste vierkantje te verslepen naar het midden of ergens anders in het kleine vierkantje, kun je dus de default positie van je form bepalen. Het effect is dat de Top en Left properties van je VCL Form gezet worden (en dat had je ook handmatig kunnen doen, maar op deze manier krijg je er een beter gevoel bij wat het eindresultaat zal zijn).

Al deze opties en nieuwe features zijn overigens niet beschikbaar voor WinForms (en ook niet voor ASP.NET Web Forms uiteraard), maar alleen voor VCL en VCL for .NET Forms.

Code Editor

Als we naar de Code Editor kijken, zien we een aantal zaken die meteen opvallen. Zo heeft niet elke regel meer een zichtbaar regelnummer, maar wordt dat om de 10 regels gedaan, plus de huidige regel zelf, met om de vijf regels een streepje in plaats van een punt. Dit geeft een rustiger overzicht, waarbij je echter nog steeds snel kunt zien waar je bent (of welk regelnummer je ook zoekt).

```

unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls;

type
  TForm2 = class(TForm)
    Edit1: TEdit;
    Label1: TLabel;
    Button1: TButton;
    ListBox1: TListBox;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation
  {$R *.dfm}

  procedure TForm2.Button1Click(Sender: TObject);
  begin
  end;
end.

```

Daarnaast zien we soms gele en groene kleuren links van de code. Die geven aan wat de status is van de editor buffer. Groen wil zeggen dat het nieuwe code is, die inmiddels ook op disk is bewaard. Geel wil zeggen nieuwe code die echter nog niet naar disk is weggeschreven. Dus als op dat moment de stroom uitvalt (of de IDE zou vastlopen), dan weet je in ieder geval dat de groene code veilig is, maar de gele code weg is. Kun je altijd zelf inschatten hoe groot het risico is van het niet even save van je code.

Block Completion en Live Templates

De Delphi 2006 IDE is ook wat hulpvaardiger als het gaat om het genereren van code (een feature die ze Block Completion noemen). Als je "begin" intikt en daarna een return, dan krijg je de "end" kado. Dat is soms handig. Wat al wat leuker is, is het intikken van "try", waarna je een "finally" en "end" kado krijgt. Het leukste is echter het gebruik van het case statement. Als je case variabele daarbij een enumerated type is, dan is een druk op de tab toets voldoende om alle mogelijke waarden als case labels te krijgen. Neem bijvoorbeeld de Position property van de TForm. Als je "case" intikt gevolgd door een spatie, krijg je meteen het volgende te zien:

```

procedure TForm2.Button1Click(Sender: TObject);
begin
  case True of
  end;
end;

```

Als je op de plek van True dan "Position" intikt, gevolgd door een tab, krijg je de volgende code:

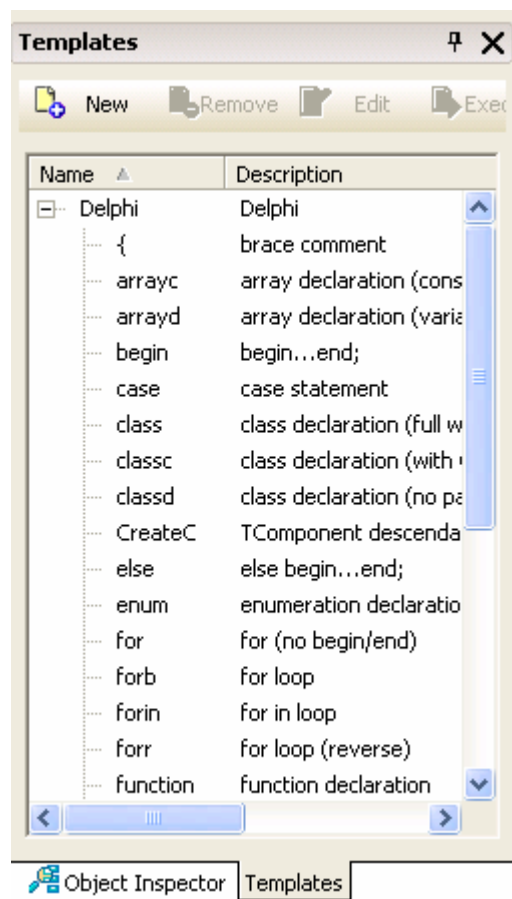
```

case Position of
  poDesigned: ;
  poDefault: ;
  poDefaultPosOnly: ;
  poDefaultSizeOnly: ;
  poScreenCenter: ;
  poDesktopCenter: ;
  poMainFormCenter: ;
  poOwnerFormCenter: ;
end;

```

Geloof me, als je dit een paar keer hebt gebruikt, dan merk je pas hoe prettig het is (nooit meer een gemist case label, of het in hoeven tikken van alle mogelijke waarden).

Een andere nieuwe feature is Live Templates genaamd. Een overzicht van de beschikbare Live Templates voor de actieve persoonlijkheid van BDS kun je zien met View | Templates. Let op: dit levert geen dialoog op, maar resulteert in een extra tab in de Object Inspector met daarin de aanwezige templates.



Live Templates start je door het betreffende keyword in te tikken, gevolgd door een spatie (zoals ik eerder al bij het case statement vertelde). Als een Live Template actief is, kun je met de tab toets door de verschillende items lopen en die desgewenst verder invullen. Met Escape beëindigt je deze toestand.

Je kan zelf ook Live Templates toevoegen aan deze lijst. Het voordeel van Live Templates ten opzichte van de "oude" Code Templates die je met Control+J kon activeren, is dat Live Templates een stuk code genereren waarin je met de tab toets nog doorheen kunt lopen om bepaalde elementen in te vullen. Je kan nog steeds de "oude" Code Templates krijgen voor Control+J te blijven gebruiken, maar nu krijg je na de naam van de template gevolgd door een spatie de nieuwe Live Template.

Bijvoorbeeld voor een for loop, levert dit het volgende resultaat na het intikken van "forb" en een spatie:

```
procedure TForm2.Button1Click(Sender: TObject);
begin
  for I := 0 to List.Count - 1 do
  begin
    loop counter
    <tab> or <shift + tab> to navigate
  end;
end;
```

Met de tab kun je nu van de "I" naar de "0" of de "List.Count" springen om die aan te passen. Met een druk op Escape kun je voortijdig uit de "live" mode komen, maar als je doortabt tot het eind, dan krijg je zelf de variabele declaratie voor de "I" kado, wat tot het volgende resultaat komt:

```
procedure TForm2.Button1Click(Sender: TObject);
var
  I: Integer;
begin
  for I := 0 to List.Count - 1 do
  begin
    |
  end;
end;
```

De cursor staat na afloop tussen de begin en end, klaar om verder te gaan. De automatische variabele declaratie kun je trouwens ook "los" krijgen als de "in-place variabele declaration" option. Tik ergens midden in je code gewoon "var" in gevolgd door een tab. Daarna krijg je het volgende beeld, en kun je de naam van de variabele en het type intikken.

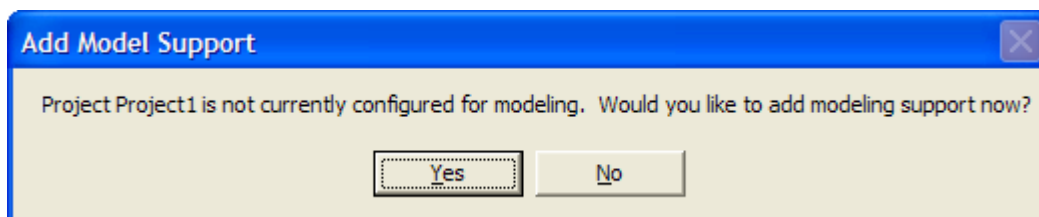
```
procedure TForm1.Button1Click(Sender: TObject);
begin
  LVar: Integer;
  en variable name to declare
  <tab> or <shift + tab> to navigate
```

Na nog een tik op de tab toets is de variabele declaratie naar de juiste plaats getransporteerd, Wederom zonder dat je je huidige plek in de source code verliest, zodat je meteen verder kunt gaan waar je mee bezig was. Al met al best handig.

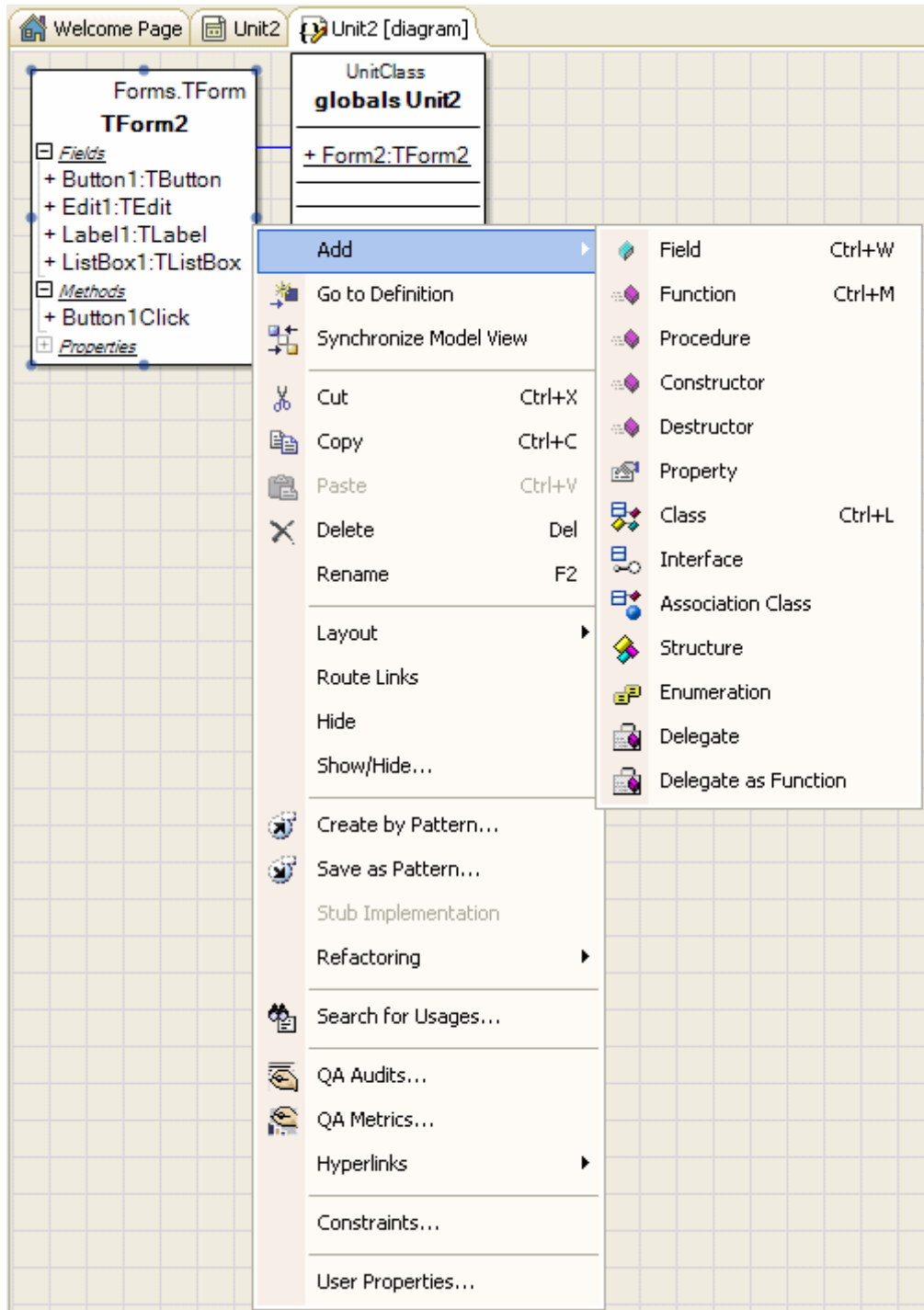
Together Designer en Design Patterns

In iedere editie van Delphi – en niet alleen voor .NET - zoals sommige mensen beweren - is er nu integratie van de code editor met de Together UML designer. Met behulp van deze designer kunnen we een UML schema zien van de classes in een project. En niet alleen zien, maar ook aanpassen.

Dit kunnen we in alle projecten doen – zelfs in Console toepassingen – met uitzondering van C++ projecten. Om het proces te starten moet je op de Model View tab klikken van de Project Manager. In voorgaande versies van Delphi was deze tab eigenlijk alleen zinvol bij ECO projecten, maar in Delphi 2006 krijg je bij een niet-ECO project nu de volgende vraag



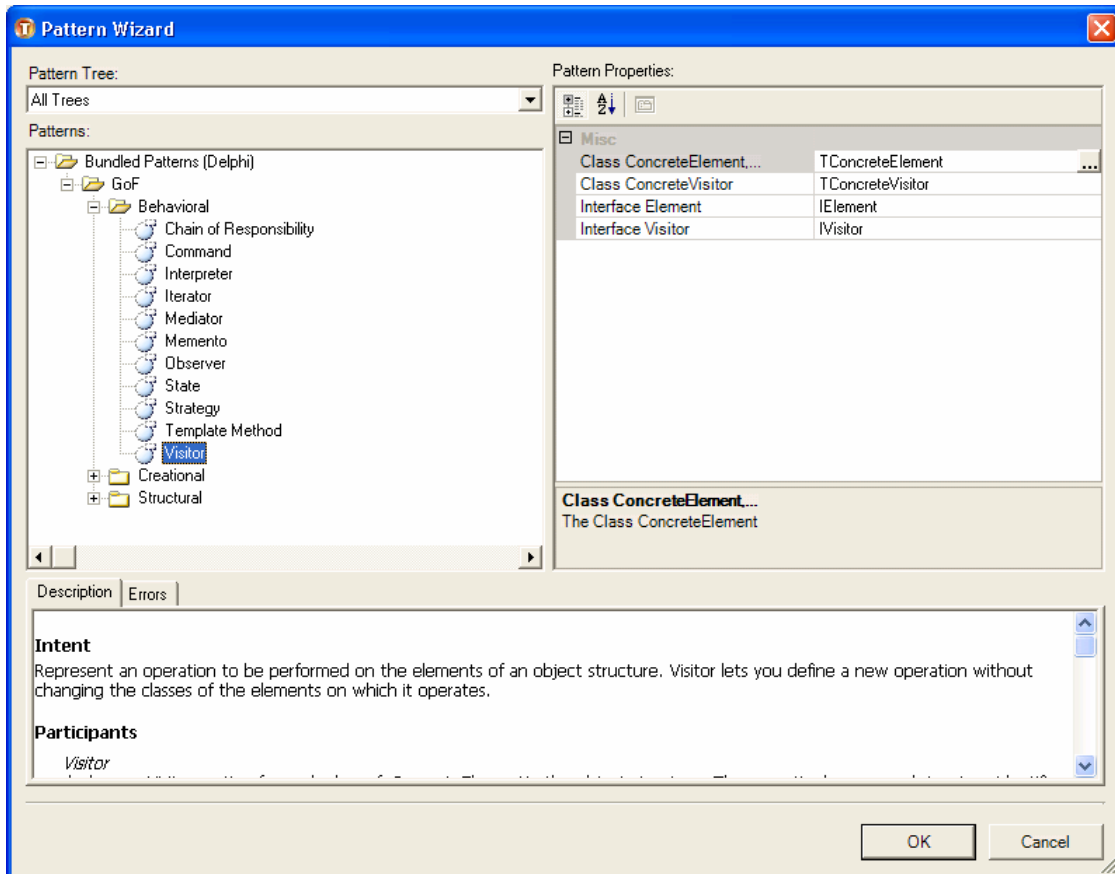
Als je op Yes klikt, dan krijg je (na enkele seconden) in de Model View een overzicht van de object structuren in je project. En deze kan je ook aanpassen via een UML diagram.



We kunnen aan een class bijvoorbeeld van alles toevoegen: een nieuw field, function, procedure, constructor, destructor, property, etc. (zie bovenstaand screenshot). Voor individuele elementen kunnen we bepalen of die strict private, private, protected, etc. zijn, en we kunnen associaties en afgeleide classes geheel in UML modelleren. Dit kan dus ook al met Delphi 2006 Professional.

Daarnaast kunnen we nieuwe code genereren door gebruik te maken van de "Node by Pattern" of "Link by Pattern" mogelijkheid, waarbij we uit een verzameling Design Patterns kunnen kiezen. Wie nog niet eerder van Design Patterns heeft gehoord, doet er goed aan het boek van de "Gang of Four" te lezen. Voor meer informatie, zie <http://www.drBob42.com/reviews/0201633612.htm> voor een review en mogelijkheden om het boek met korting via Amazon.com of Amazon.co.uk te bestellen.

Referenties naar de "Gang of Four" zul je trouwens ook binnen Delphi nog vaak terugzien. Onder andere in de dialoog waaruit je een pattern kunt kiezen:

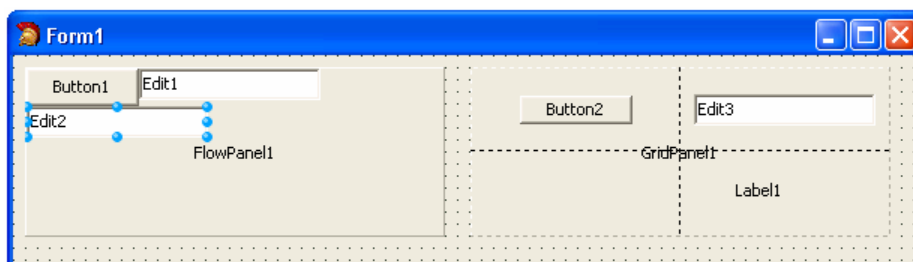


De "GoF" hier verwijst hierbij naar de *Gang of Four*. Voor ieder design pattern is er een beschrijving opgenomen waarin je als leek kunt zien wat de bedoeling is, waar het pattern het beste toepast kan worden, en wat de consequenties zijn. Voor Win32 en .NET zijn de Design Patterns onderverdeeld in Behavioral, Creational, en Structural categoriën. Voor .NET zijn er daarnaast ook nog een verzameling Standard patterns. Erg nuttig, en net als Refactoring en Unit Testing die bij de vorige versie van Delphi werden opgenomen, is het Delphi Team er wederom in geslaagd om een belangrijke ontwikkeltechniek als design patterns in Delphi te integreren.

Win32

Voor Win32 ontwikkelaars is het goed te weten dat er een stuk of 20 FastCode routines in de Win32 RTL zijn opgenomen, die de snelheid van bestaande Win32 Delphi toepassingen kan doen toenemen door ze alleen maar te hercompileren met Delphi 2006. Samen met de FastCode FastMM 4 memory manager die ook al in Delphi 2006 is terug te vinden, was dit reden voor het Delphi R&D Team om de mensen achter FastCode met de Spirit of Delphi 2006 award te belonen.

Er zijn een paar nieuwe VCL componenten, zoals de TFlowPanel en TGridPanel, die een beetje werken zoals de ASP.NET designer in flow layout of grid layout. Bij een FlowPanel kun je de FlowStyle zetten (bijvoorbeeld op LeftRightTopBottom), en vervolgens zullen de componenten van links boven naar rechts onder naast elkaar gezet worden. Je kan daarbij de positie van de componenten niet meer aanpassen door ze te verslepen: ze springen steeds terug naar de natuurlijke flow-positie.



Bij het TGridPanel kunnen we opgeven in rows en columns we het grid willen opdelen (default 2 rows en 2 columns), en daarna kunnen we in iedere "cell" precies één componen plaatsen.

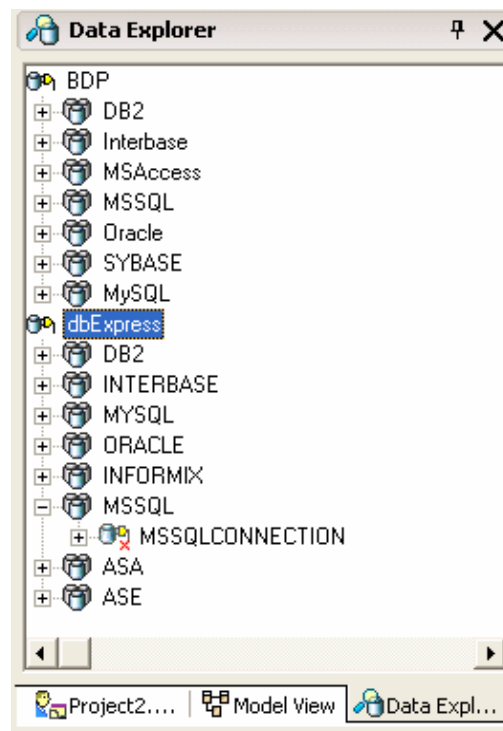
Wat handiger en zeker leuker is de TTrayIcon component, die de mogelijkheid biedt om een TrayIcon te laten zien, inclusief animatie, zonder één regel code te hoeven schrijven. Je kunt via een imagelist een lijst van icoontjes meegeven, en tijdens runtime het zichtbare icoontje bepalen. Ook animatie van de icoontjes is mogelijk, net als ballon helpteksten. Werkt prima in praktijk.

Er zijn ook een aantal nieuwe classes, zoals de TCustomTransparentControl die gebruikt kan worden als base class voor transparent controls, de TMargins class en de TPadding class.

dbExpress

Op database gebied zijn er weer een aantal bugs opgelost in dbExpress, en zijn de drivers uitgebreid met ondersteuning voor de nieuwste databases, waaronder SQL Server 2000/2005, Oracle 10g, 9.1.0 en 9.2.0, IBM DB2 UDB 8.x en 7.x, Informix 9.x, Adaptive Sybase Anywhere 9 en 8, Sybase 12.5, MySQL 4.0.24 en Interbase versies 7.5, 7.1, 7.0 en 6.5. Wie de nieuwste versie InterBase 7.5 installeert, moet overgens na installatie de InterBase Manager starten, "Run Interbase as a service" optie uitzetten, en de server opnieuw starten voor je een connectie met een lokale database kunt maken.

Het leukste dbExpress nieuws is dat de dbExpress drivers nu ook in de Data Explorer gebruikt worden.

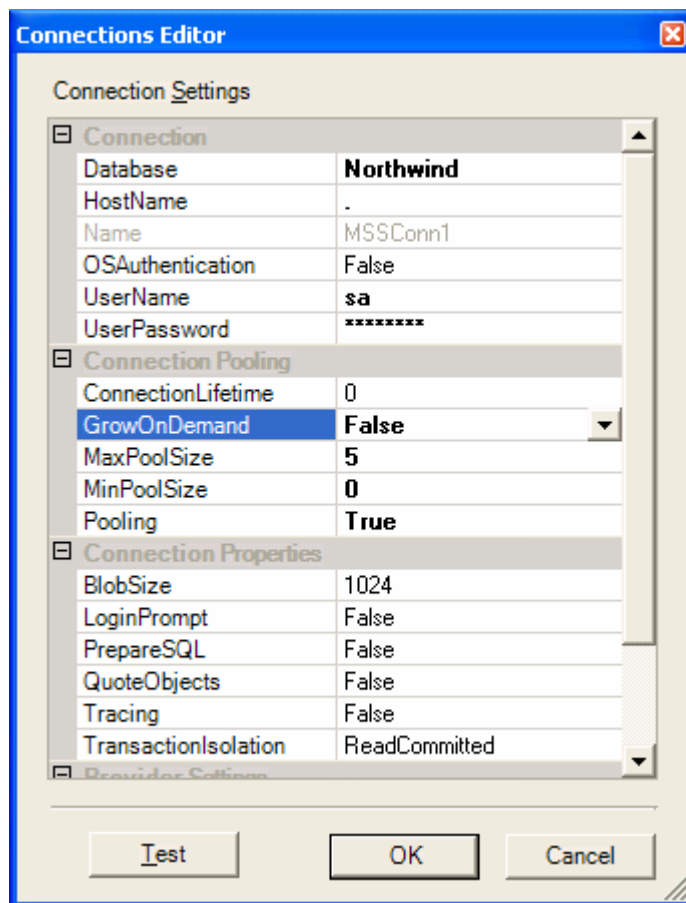


.NET

Op .NET gebied zijn meer uitbreidingen te vinden. De Borland Data Providers voor .NET ondersteunen dezelfde databases en versies als dbExpress, met uitzondering van Adaptive Sybase Anywhere 9, 8 die niet door BDP ondersteund wordt. Aan de andere kant wordt MS Access wel door BDP maar niet door dbExpress ondersteund (in Win32 kunnen we daar natuurlijk via dbGo for ADO mee werken).

Belangrijker nog dan de lijst met drivers, is het feit dat de Borland Data Providers (BDP) eindelijk zijn uitgebreid met Connection Pooling. Niet op database niveau, maar op BDP niveau, dus onafhankelijk van de database die wordt gebruikt. Er is hierbij wel iets waar je goed op moet letten: de MinPoolSize staat standaard op 0, en de MaxPoolSize op 100. Het gevolg is dat er default maximaal 100 connecties naar je database worden gemaakt. Als je echter MSDE 2000 gebruikt, dan zul je na de vijfde connectie merken dat de snelheid behoorlijk afneemt. Dat komt door de aard van MSDE, die vanaf de vijfde concurrent connectie de handrem erop zet. Om dat te voorkomen, kun je in dergelijke gevallen beter de MaxPoolSize op 5 zetten.

Dat is echter nog niet helemaal voldoende, want ook de `GrowOnDemand` property staat default op `True`, met als gevolg dat de pool zal groeien als de `MaxPoolSize` wordt bereikt. En zelfs als die op 5 staat, krijg je dan alsnog een groeiende pool, met de bijbehorende performance problemen. Je moet dus ook de `GrowOnDemand` op `False` zetten.



Hetzelfde geldt voor Local InterBase die een maximum van 10 open connecties heeft als ik het me goed kan herinneren - hier krijg je zelfs een foutmelding als je er teveel probeert te openen. Even opletten dus. Maar als het eenmaal werkt zul je zien dat de performance van ASP.NET toepassingen er bijvoorbeeld flink op vooruit kan gaan (vergeleken met een oplossing zonder gebruik te maken van connection pooling).

Behalve connection pooling zijn de BDP DataAdapters nu ook uitgebreid met de mogelijkheid tot zgn "error reconciliation", wat wil zeggen dat je nu op een prettige manier kan reageren op fouten in updates (zoals een record dat al door een andere gebruiker was gewijzigd bijvoorbeeld). Dergelijke mogelijkheden bestonden uiteraard al in Win32, maar in de .NET wereld was er nog geen fijne ondersteuning voor gebruikers van ADO.NET, wat nu dus door Borland in de BDP wordt goedge maakt, en ook heel fijn is voor ASP.NET toepassingen.

ASP.NET

Over ASP.NET gesproken, misschien wel het beste nieuws is dat de HTML editor niet langer meer de HTML code "indeelt" zoals de HTML editor denkt dat het hoort. Dat leverde in Delphi 2005 nog wel eens problemen op (als de end-tag toch echt op dezelfde regel moest staan bijvoorbeeld). We hebben nu weer de volledige zeggenschap over de HTML layout. De default layout van Web Forms is nu trouwens Flow Layout, wat wel zo prettig is, omdat de meeste mensen toch geen gebruik maken van Grid Layout.

De ASP.NET Deployment Manager is ook uitgebreid met de mogelijkheid om externe assemblies op te geven (zodat die niet eerst in de `\bin` directory van het huidige project geplaatst hoeven te worden, wat soms tot design-time problemen kon leiden, zoals ik bij mijn Advanced ASP.NET training laat zien).

Let wel: Borland Developer Studio 2006 biedt alleen ondersteuning voor .NET Framework versie 1.1, en dus voor ADO.NET 1.1 en ASP.NET 1.1 en nog niet voor ASP.NET 2.0. Voor .NET 2.0 (en ASP.NET 2.0) support moeten we wachten op de volgende versie van Delphi, met de codenaam Delphi Highlander (zie <http://www.drBob42.com/Highlander> voor meer details zodra die beschikbaar komen). Tijdens mijn Advanced ASP.NET training van 22 juni 2006 zal ik wel al enkele ASP.NET 2.0 features laten zien, maar meer ter voorbereiding van wat straks mogelijk is.

Enterprise Core Objects III

Op het gebied van ECO III zijn er verschillende uitbreidingen terug te vinden. Het belangrijkste nieuws is wel het feit dat ECO nu in iedere editie van Delphi zit. Maar dan wel de .NET smaak, want ECO is alleen beschikbaar onder .NET (en dus niet voor Win32).

In Delphi 2006 for .NET Professional kunnen we met de ECO Designer UML class models maken, met associates en inheritance tussen de classes, en attributes, operations, etc. Dit model kunnen we vervolgens in de Professional versie van Delphi for .NET persistent maken in een XML document. De Enterprise en Architect edities voegen hier de mogelijkheid aan toe om de EcoSpace in een echte database persistent te maken (waardoor je er ook met meerdere gebruikers in kan werken). Dit heeft tot gevolg dat je voor ASP.NET support van ECO ook de Enterprise of Architect versie van Delphi 2006 nodig hebt. Een nieuwe ECO III feature die alleen in de Architect editie van Delphi 2006 zit, is de ondersteuning voor State Machines. Hiermee kun je op eenvoudige wijze een workflow of process structuur in je model onderbrengen. De gebruikelijke demos (die ik ook bij de lancering van Delphi 2006 deed) betreffen meestal het aanvragen, goedkeuren dan/wel afkeuren van een bepaald item (een dienstreis of gadget bijvoorbeeld). Ik een toekomstig artikel zal ik hier dieper op ingaan.

Zie <http://www.drBob42.com/eco> voor een aantal artikelen over de vorige versies van Enterprise Core Objects, inclusief een serie die het ontwerp en de implementatie van mijn ASP.NET weblog in detail beschrijft. Op 30 maart 2006 zal ik overigens voor het eerst een gehele Delphi 2006 Enterprise Core Objects III trainingsdag verzorgen, zie <http://www.eBob42.com/training/2006eco.htm> voor details.

.NET Compact Framework

Tot slot wil ik nog melden dat Delphi 2006 ondersteuning biedt voor het .NET Compact Framework. Hiervoor moet je wel zelf het .NET Compact Framework op je machine installeren, zodat de compiler ze kan vinden. Het gaat makkelijk dan met de .NET Compact Framework Technology Preview voor Delphi 2005, want daar had je nog twee verschillende command-line compilers. Nu is er eentje, die zowel voor .NET Framework als voor het .NET Compact Framework targets kan opleveren. Er is echter nog steeds geen design-time ondersteuning voor .NET Compact Framework toepassingen - daarvoor moeten we wachten tot de volgende versie van Delphi, die dan ook .NET 2.0 zal ondersteunen. In deze versie van Delphi zal Borland waarschijnlijk Delphi uitbreiden met VCL voor .NET Compact Framework, aldus valt op te maken uit de recente Delphi Roadmap.

Op 27 april 2006 zal ik de eerste Delphi 2006 .NET Compact Framework Development trainingsdag verzorgen, zie <http://www.eBob42.com/training/2006cf.htm> voor details.

Upgrade?

Borland Developer Studio ondersteunt zowel Win32 als .NET. Het is zelfs na Delphi 8 en Delphi 2005 de derde versie van Delphi voor het .NET Framework versie 1.1. En daarmee ook de laatste versie voor .NET 1.1, want de volgende versie van Delphi zal .NET 2.0 ondersteunen. Het is ook in mijn ogen de meest stabiele versie van Delphi voor .NET 1.1, dus ik kan iedereen die met .NET wil werken aanraden om op Delphi 2006 over te stappen. Het is wat kosten betreft een zure appel voor wie eerst Delphi 8 kocht en daarna de overstap naar Delphi 2005 heeft gemaakt, dat ben ik met iedereen eens. Toch ben ik van mening dat Delphi 2006 de moeite waard is, en niet een versie van Delphi is die overgeslagen moet worden.

Wie nog vragen of opmerkingen heeft over dit artikel (of Delphi 2006 in het algemeen) kan me altijd een e-mailtje sturen of feedback achterlaten in mijn weblog te <http://www.eBob42.com/blog>