

Delphi OplossingsCourant

Vol. 11. No. 2, Een gratis elektronische publicatie van Bob Swart Training & Consultancy (eBob42) - <http://www.eBob42.com>



Helmond, 21 september 2009,

Welkom bij het nieuwe nummer van de Delphi OplossingsCourant – de tweede editie van dit jaar. Dit nummer staat in het teken van Delphi 2010 en een aantal nieuwe mogelijkheden van deze .nieuwe versie van Delphi. Wie nog Delphi 2005 of ouder gebruikt en overweegt om over te stappen naar Delphi 2010 moet niet te lang wachten overigens: na 1 januari 2010 mag je alleen een upgrade kopen als je hooguit *drie versies* achterloopt (dus om Delphi 2010 te kopen moet je dan minimaal Delphi 2006 bezitten). Wie nog niet klaar is voor Unicode, zou dus eigenlijk nog voor het eind van het jaar naar Delphi 2007 moeten overstappen – de laatste niet-Unicode versie van Delphi. Op pagina 4 ga ik wat dieper in op deze verandering (waar ik als reseller ook niet echt blij mee ben).

Een andere verandering treed op 1 oktober in werking in de wet op de telecom: om spam tegen te gaan, mag geen ongevraagde e-mail meer verstuurd worden. Omdat ik voor de aankondiging van de DOC alleen maar e-mail adressen van de abonnees heb bewaard, zonder naam, mag ik officieel geen gebruik meer maken van deze lijst (na 1 oktober). En dus moet ik iedereen vragen om zich weer opnieuw aan te melden door een mailtje te sturen naar Bob@eBob42.com, waarna ik ook de naam zal bewaren (en de aankondigingen vanaf nu met een persoonlijke groet zal beginnen).

Tot slot nog excuses voor de vertraging: deze Delphi OplossingsCourant zou eigenlijk veel eerder na mijn Delphi 2010 seminar op 11 september verschenen zijn, is helaas een week vertraagd. Toch nog veel leesplezier gewenst, en tot de volgende keer. Groetjes,

Bob Swart Training & Consultancy

Met mijn bedrijf Bob Swart Training & Consultancy hou ik me met name bezig met de ondersteuning van Delphi (Prism) ontwikkelaars door middel van consultancy, trainingen en workshops, coaching en support (ook per e-mail), en het leveren van de benodigde licenties, subscriptions en cursusboeken voor zelfstudie (al dan niet onder begeleiding).

Trainingen en Workshops

Mijn Delphi (Prism) en RAD Studio trainingen zijn inclusief dik cursusboek en e-mail ondersteuning (ook achteraf als de training is afgelopen). Vanaf begin 2007 worden de trainingen gegeven in de cursusruimte in Helmond Brandevoort (ideaal met de trein te bereiken – 150 meter van NS station Helmond Brandevoort), zie ook www.eBob42.org.



Alle Win32 en .NET onderwerpen zijn ook als maatwerk training mogelijk in de vorm van een Delphi Workshop in Helmond Brandevoort, of Custom Clinic (op locatie).

Zie <http://www.eBob42.com/workshop> voor details.

Reseller

Bob Swart Training & Consultancy is reseller van Embarcadero voor de BeNeLux, en levert zowel Delphi (Prism) als CodeGear RAD Studio inclusief persoonlijke ondersteuning (en nieuwsgroepen). Ook All-Access en subscriptions zijn verkrijgbaar. Zie <http://www.eBob42.com/CodeGear> voor details.

Delphi 2010 nieuwe features

Eind augustus 2009 is Delphi 2010 uitgekomen. Voor wie niet bij een van de Delphi 2010 Launch events was, volgt hier een korte samenvatting van de meest in het oog springende nieuwe features. Verderop in dit nummer van de Delphi OplossingsCourant volgt nog een meer gedetailleerd artikel over de Touch en Gesture ondersteuning, alsmede een artikel over de DataSnap 2010 uitbreidingen.

Alle edities van Delphi 2010 genieten van de algemene uitbreidingen aan de IDE en code editor. Zo is er de nieuwe IDE Insight feature, waarbij je via de F6 toets een soort zoekwindow krijgt waarin je alle opties en mogelijkheden ziet die zinvol zijn in de huidige context (zowel in de code editor of forms designer bijvoorbeeld). Zelf gebruik ik F6 niet, want ik gebruik de Classic Keyboard Layout, en daar wordt F6 gebruikt om naar de volgende tab van de code editor te springen. Voor wie net als ik deze layout nog in zijn vingers heeft, moet de Ctrl+. combinatie gebruiken (control en punt).

Zelfs de debugger is uitgebreid met nieuwe features. De meest in het oog springende zijn de debug visualizers, om bijvoorbeeld een date niet langer als floating point getal maar als een echte datum te laten zien. Ook voor een TStringList of TString is het nu fijner om in de debugger de inhoud te zien. Het leuke is dat we ook zelf eigen debug visualizers kunnen schrijven voor speciale structuren die je (beter) wilt kunnen zien tijdens het debuggen.

Een andere nieuwe feature van de Delphi 2010 debugger is ondersteuning voor multi-threaded debuggen, met de mogelijkheid om individuele threads tijdelijk te bevriezen (freeze) en later ook weer te ontdooien (thaw).

Delphi 2010 ondersteunt touch en gestures, en – onder Windows 7, met speciale hardware – ook de interactive gestures en multi-touch events. Dit lijkt op het eerste gezicht misschien alleen maar “leuk” en niet noodzakelijk, maar als straks Windows 7 uitkomt en multi-touch hardware (zoals netbooks) goedkoper wordt, is het fijn als je ontwikkelomgeving daar al ondersteuning voor biedt. En ook voor bestaande toepassingen kun je touch en gestures al toevoegen (en dat werkt niet alleen onder Windows 7, maar ook onder Windows 2000, XP en Vista).

Ook bevat Delphi 2010 ondersteuning voor een speciaal Direct2D canvas, waarmee anti-aliased fonts, vectors en andere grafische grappen zijn uit te halen die momenteel nog alleen onder Windows 7 ondersteund worden (dat wil zeggen dat het Direct2D canvas ook alleen met Windows 7 zal werken – Microsoft overweegt om Direct2D ook naar Windows Vista te back-porten, maar ik zou daar niet al te zeer op hopen als Windows 7 eenmaal uit is).

QA Audits en Metrics waren tot voor kort alleen beschikbaar voor de Enterprise (of hogere) edities van Delphi, maar zijn vanaf Delphi 2010 ook beschikbaar in de Professional editie. Niet alle QA Audits en Metrics, maar voldoende om een begin te maken en een indruk te krijgen van de mogelijkheden.



De Enterprise editie van Delphi 2010 bevat daarnaast ook nog een aantal uitbreidingen. Zo is er nu een Firebird DBX4 driver, met ondersteuning voor Firebird 2.1.1 en 1.5. Dit is een andere driver dan de InterBase driver die in het verleden soms ook gebruikt kon worden voor Firebird databases. Ik vind het persoonlijk jammer dat de Firebird driver niet in de Professional versie van Delphi zit. Aan de andere kant is het ook een beetje een compliment voor Firebird, omdat deze nu op hetzelfde niveau wordt gezet als de andere “Enterprise”-level DBMS drivers zoals Oracle, SQL Server, DB2, Informix, Sybase, etc.

RAD Studio 2010 bestaat uit Delphi 2010 en C++Builder 2010 voor Win32, en Delphi Prism voor .NET (de plug-in voor de Visual Studio IDE). Delphi Prism is ook uitgebreid met nieuwe mogelijkheden, waaronder Aspect Oriented Programming onder de naam Cirrus. In een later nummer van de Delphi OplossingsCourant zal ik hier nader op ingaan – en natuurlijk ook in mijn Delphi Prism cursusboek dat (zodra het eindelijk af is) gratis naar al mijn klanten zal worden verstuurd die RAD Studio of Delphi Prism bij mij hebben afgenomen.


Delphi's toekomstige features

Tijdens mijn Delphi 2010 Launch seminar heb ik ook een tweetal slides laten zien die betrekking hebben op toekomstige versies van Delphi, namelijk Project Delphi "X" en Project "Commodore".

Project Delphi "X"





- **Cross-platform Windows, MacOS, and Linux.**
 - emphasis on creating GUI applications with dbExpress and DataSnapX connectivity.
- **Other features under consideration for Delphi "X" include:**
- **Cross-platform component library**
 - Limited backward compatibility
- **DataSnapX the ability to create both Server and Clients on Windows, Mac, Linux, and Web**




7

Tijdens CodeRage 4 liet Michael Rozlog (Product Manager Delphi) weten dat op dit moment de focus met name naar Project Delphi "X" gaat, en in mindere mate naar Project "Commodore".

Project "Commodore"



- **Commodore project is bringing 64-bit native development to Delphi, C++Builder, and the VCL.**
- **Features under consideration for Commodore:**
- **Full compiler, RTL, and VCL support for 64-bit native compilation**
 - An option in the IDE to compile an application either as a 32-bit or a 64-bit application
- **Multi-core/Multi-threaded applications**
- **Support for parallelization in the RTL**



8

Delphi upgrades na 1-1-2010

Er gaat per 1 januari 2010 iets veranderen bij het kopen van een upgrade van Delphi, C++Builder of RAD Studio. Op dit moment zijn alle oude versies van Delphi voldoende om als uitgangspunt te dienen voor de aanschaf van een "Upgrade" (die goedkoper is dan een "New User" licentie). Per 1 januari 2010 zijn echter alleen nog de laatste drie versies geldig om een upgrade van te kopen, en oudere versies kunnen dan niet meer gebruikt worden als uitgangspunt voor de upgrade!

Dit betekent dat straks alleen Delphi 2006, Delphi 2007 en Delphi 2009 nog gebruikt kunnen worden om een upgrade naar Delphi 2010 te verkrijgen. Dat is slecht nieuws voor wie nog op Delphi 2005 zit en wellicht ook over wil straks. En alhoewel ik niet verwacht dat er heel veel mensen Delphi 8 for .NET nog gebruiken, vind ik het met name vervelend voor de grote groep Delphi 7 ontwikkelaars. Als die na 1 januari 2010 een nieuwe versie van Delphi willen aanschaffen, zullen zij de New User prijs moeten betalen.

De Upgrade en New User prijzen (ex.BTW) van Delphi en RAD Studio zijn als volgt:

	Delphi 2010		RAD Studio 2010	
	<i>Upgrade</i>	<i>New User</i>	<i>Upgrade</i>	<i>New User</i>
Architect	€ 2.299	€ 3.499	€ 2.799	€ 4.299
Enterprise	€ 1.299	€ 1.999	€ 1.799	€ 2.799
Professional	€ 399	€ 899	€ 649	€ 1.399

Zoals te zien is, is de New User prijs voor Delphi 2010 zelfs hoger dan de Upgrade prijs voor een vergelijkbare RAD Studio 2010 met daarin zowel Delphi 2010, C++Builder 2010 als Delphi Prism 2010!

Mijn advies voor Delphi 7 gebruikers: upgrade nog voor eind dit jaar naar Delphi 2010. Daar zit Unicode in, maar daarmee zit je de komende jaren goed als je nog eens echt wilt overstappen naar een Delphi 2011, 2012 of 2013 bijvoorbeeld. Een upgrade van Delphi 7 naar 2007 lijkt technisch gezien misschien interessanter, want dan hoeft je de Unicode migratie niet mee te maken, maar Delphi 2007 kan alleen gebruikt worden als upgrade pad naar Delphi 2009, Delphi 2010 en straks Delphi 2011. Maar als Delphi 2012 uitkomt dan is Delphi 2007 niet meer voldoende. Moeilijke beslissing, maar één ding is zeker: wie nu op Delphi 7 blijft zitten, moet straks een New User licentie kopen. Of voor altijd Delphi 7 blijven gebruiken, of overstappen op een andere ontwikkelomgeving – maar dat is het risico dat Embarcadero hiermee neemt. Zakelijk gezien kan ik ze geen ongelijk geven: de groep Delphi 7 gebruikers brengt natuurlijk geen (licentie) geld meer in het laatje. Maar aan de andere kant zijn het wel hele trouwe Delphi gebruikers die je niet zomaar wilt afsnijden van de groep die nog wel Upgrades kunnen krijgen. Nat als de Delphi 6, 5, 4, 3, 2 en 1 gebruikers natuurlijk, alleen is die groep in mijn perceptie kleiner dan die van Delphi 7.

Wie nog niet "klaar" is om de overstap te maken van Delphi 1 t/m 7 naar Delphi 2010, hoeft in ieder geval niet bang te zijn voor de nieuwe embedded forms designer in de IDE: het is in Delphi 2010 mogelijk om de embedded forms designer uit te zetten en zelfs het oude Component palette terug te krijgen, zoals ik tijdens de Delphi 2010 Launch liet zien.

Wat gaat de toekomst van Delphi brengen? Is het wat dat betreft zinvol om nu nog een upgrade aan te schaffen? Er zijn in ieder geval twee grote projecten aan de gang: Delphi Project "X", de crossplatform compiler voor Windows, Linux en Mac OS X, en daarnaast Delphi Commodore, de langverwachte 64-bits versie van Delphi. Allebei met (een subset van?) de VCL en straks alleen toegankelijk tegen een upgrade prijs voor wie een licentie van hooguit drie versies terug bezit. Betekent voor mij als reseller ook meer werk als het gaat om het checken van de licenties (van nieuwe klanten).

Nog een laatste tip: tot 25 september geldt een actie van Embarcadero: bij aanschaf van een product krijg je een coupon voor het downloaden van een tweede product (van maximaal dezelfde prijs, en het mag niet hetzelfde product zijn). Dat zou nog een klein voordeel zijn voor wie van Delphi 7 of 2005 nu nog wil overstappen naar Delphi 2010 en dan gratis Delphi Prism 2010 of C++Builder 2010 of een ander Embarcadero product erbij kan krijgen. Is niet veel, maar toch wat, en het is nog geen 25 september...

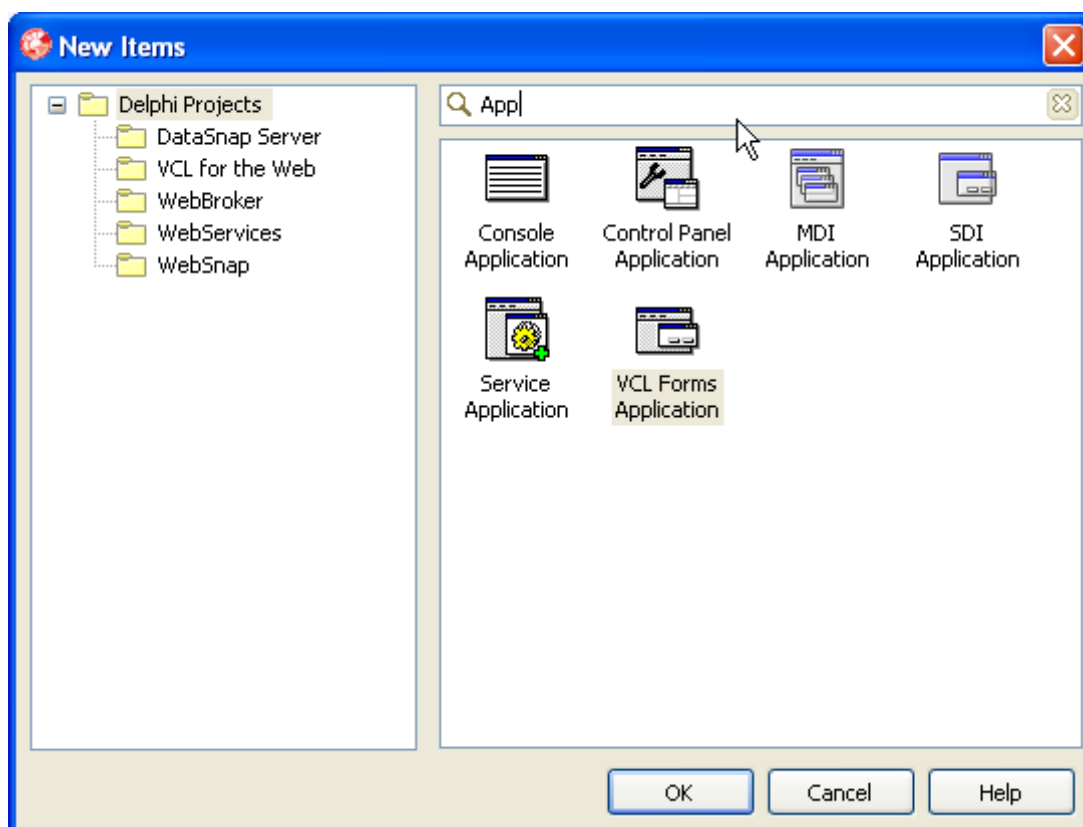
Delphi 2010 op gevoel

Op het moment van schrijven is Delphi 2010 nog niet officieel uitgekomen of aangekondigd. Bepaalde details die in dit artikel worden beschreven kunnen derhalve afwijken van de uiteindelijke versie. Zie <http://www.eBob42.com> of <http://www.bobswart.nl> voor meer informatie en bestelmogelijkheden van Delphi 2010 en RAD Studio 2010.

Touch en Gestures

Een van de nieuwe features van Delphi 2010 is de directe ondersteuning voor touch in de vorm van gestures en multi-touch (dat door Windows 7 wordt ondersteund). Nu was het altijd al mogelijk om met Delphi toepassingen te maken voor touch schermen, omdat daarbij het scherm zich als een muis voordoet (dus je kan zonder muis en toetsenbord op buttons klikken bijvoorbeeld), maar niet eerder was het zonder meer mogelijk om Delphi op speciale gestures te laten reageren. Dat kan nu wel, en daar wil ik het in dit artikel over hebben.

Allereerst start in Delphi 2010, en begin met een nieuwe application. Via File | New – Other kom ik in de Object Repository, en die is uitgebreid met een handige filter optie: ik kan nu een paar letters intikken van datgene wat ik zoek (zoals “App”), en dan laat de Object Inspector vervolgens alleen nog maar die icons zien die matchen. Een fijne manier om de juiste boom te vinden zonder te verdwalen in het bos.



Een ander fijn iets is dat de Object Inspector nu ook icoontjes laat zien die er wel zijn, maar nog niet gebruikt kunnen worden (zoals in de ActiveX categorie, waarbij een hoop wizards pas gestart kunnen worden als er al een project is). Uiteraard werkt het filter hier ook op, maar dat is in het bovenstaande screenshot niet te zien.

Als ik een nieuwe VCL Forms Application heb, dan ga ik eerst een normale kleine database toepassing bouwen. Gebruik een TClientDataSet om tot een executable te komen die geen database drivers nodig heeft (en bijvoorbeeld als stand-alone .exe op een USB stick gezet kan worden). Als data wil ik biolife

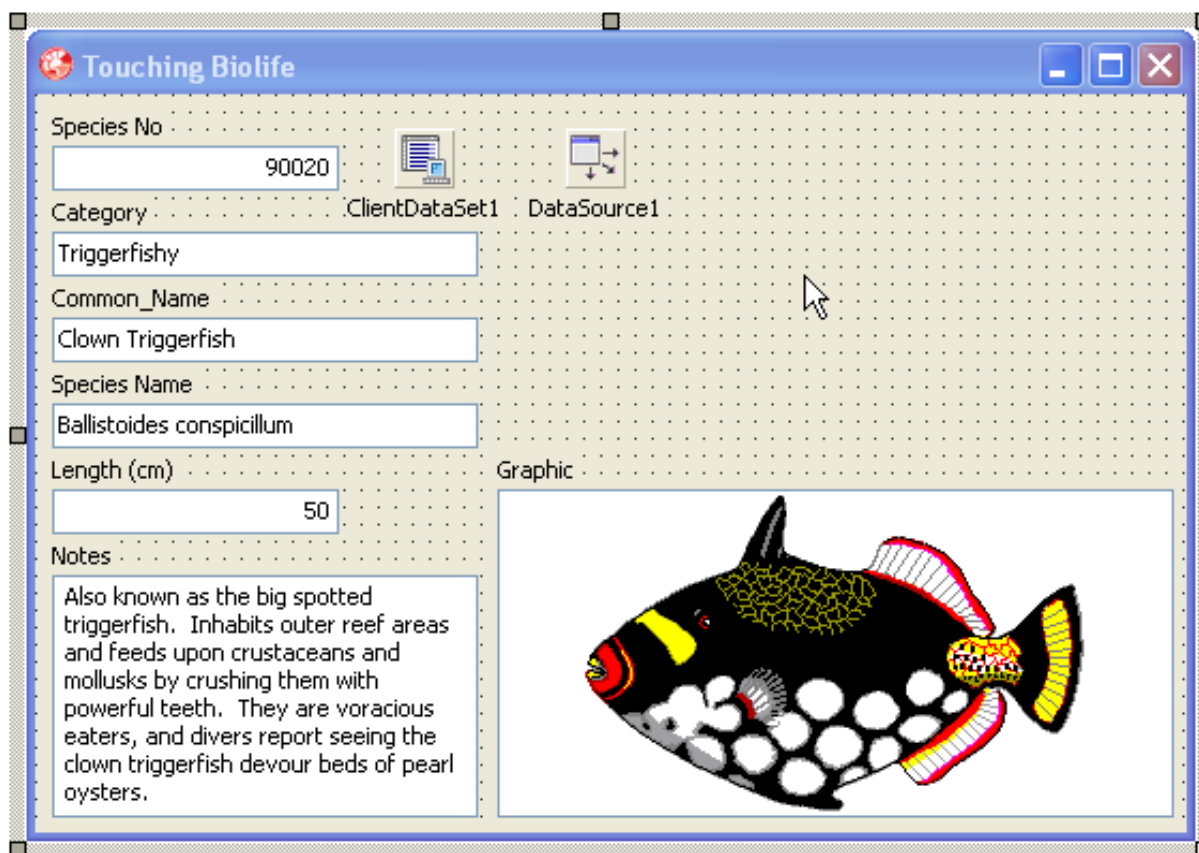
laten zien, en die kunnen we vinden in de biolife.xml file in de Common Files\CodeGear Shared\Data directory (wellicht dat dit in de toekomst de Common Files\Embarcadero Shared\Data directory moet worden).

Als de FileName property van de TClientDataSet component naar het biolife.xml bestand wijst, kunnen we via de Object Inspector de Active property van de TClientDataSet op True zetten om alle data tijdens design-time al te zien. Door vervolgens de filename weer leeg te maken, blijft wel de data zichtbaar, maar is die in de DFM file opgeslagen, en niet meer buiten de executable. Dat is een speciale truck die ik af en toe gebruik, en die ervoor kan zorgen dat de data zelf in de executable wordt meegelinkt. Natuurlijk zijn er wel consequenties aan deze truck: ten eerste wordt de .DFM file ongeveer 2 MB groot (dus het duurt even als je wilt save, of als je het project gaat compileren – vooral het linken duurt nu langer dan normaal). Lastiger is echter dat de data in de .exe zelf zit, en dus niet meer gewijzigd kan worden. Dat is meestal vervelend, behalve als het om een brochure gaat die je toch niet wilt laten wijzigen (een programma dat fietsroutes laat zien met plaatjes bijvoorbeeld). In ons voorbeeld ga ik er even vanuit dat we de data uit biolife wel willen zien, maar niet hoeven aan te passen, en ben ik nog steeds op weg om een stand-alone executable te maken.

Biolife Velden

Nu de data in de TClientDataSet zit, kunnen we erop dubbelklikken om de Fields Editor te krijgen. Via de rechtermuisknop kunnen we dan kiezen voor Add All Fields, zodat we alle velden uit de biolife tabel in de Fields Editor zien. Ik wil ze bijna allemaal zien: alleen Length_In vind ik niet nodig. Dus ik selecteer ze op Length_In na allemaal, en sleep ze van de Fields Editor naar het Form, waar ze netjes als label en data-aware control neergezet worden. Plus een TDataSource als verbinding tussen de TClientDataSet en de data-aware controls die zojuist automatisch zijn neergezet voor ons.

De meeste velden komen in een TDBEdit control, met uitzondering van Notes (die komt in een TDBMemo) en Graphic (die komt in een TDBImage). Na het verplaatsen van de TDBImage, ziet mijn form er als volgt uit:



Let op dat ik geen TDBNavigator neerzet, want daar wil ik mijn touchscherm voor gebruiken.

Touch en Action

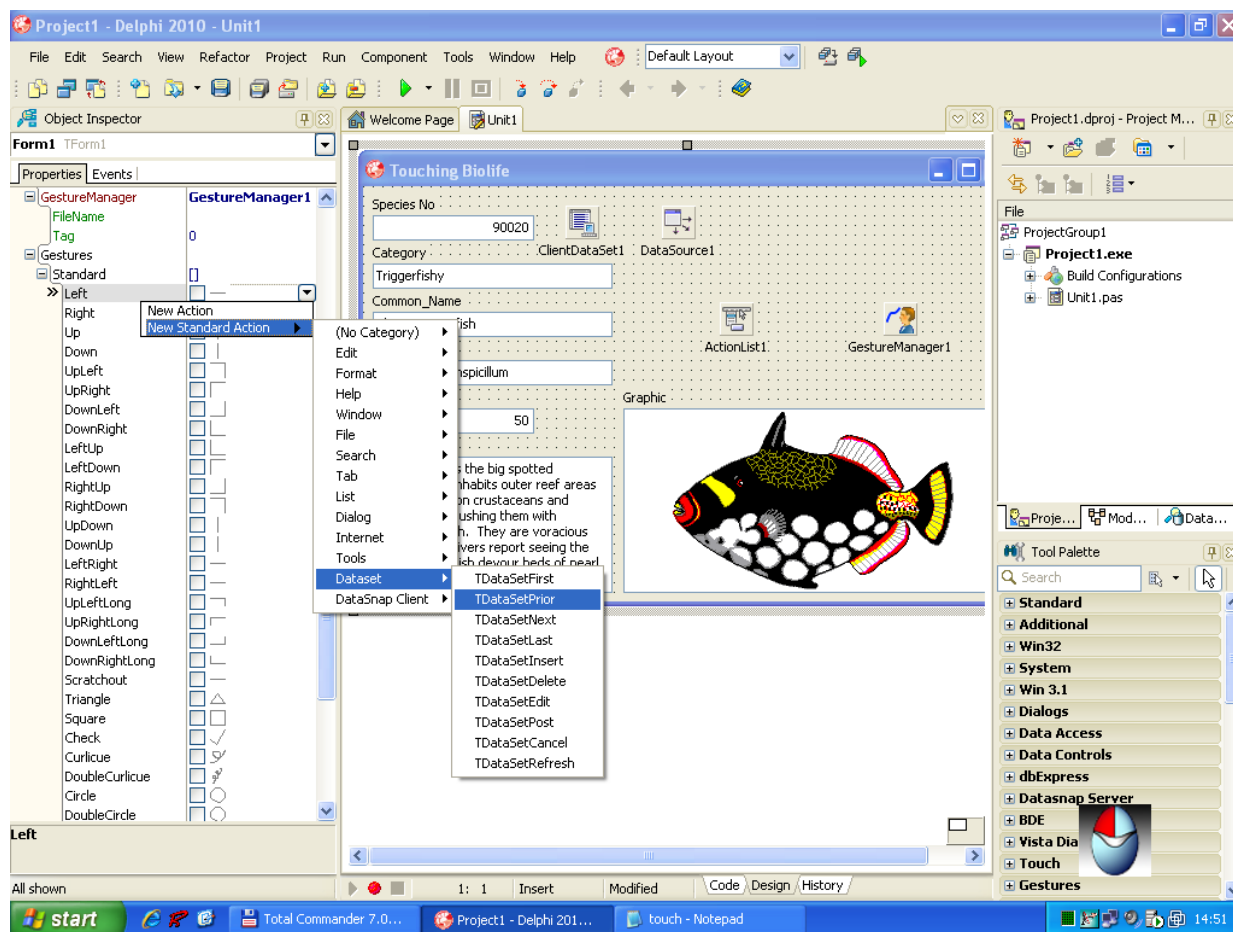
We komen nu bij het interessante gedeelte: de touch en gesture support in Delphi 2010. Allereerst een voorbereidende stap: zet een TActionList component op het form. Delphi 2010 onderkent al een flink aantal standaard “gestures”, en die kunnen we direct aan actions koppelen. Dat scheelt weer een hoop event handlers en code schrijven, want het is juist fijn om standaard gedrag (een action) aan een voorbedachte veegbeweging over het scherm (gesture) te kunnen koppelen.

De TActionList hoeft niet te worden ingevuld, maar het is wel nodig dat hij op het form staat. Vervolgens moeten we er nog een TGestureManager component naast zetten (die ook het volgende screenshot). Als derde stap volstaat het om deze TGestureManager te koppelen aan de GestureManager subproperty van het form. Deze property is een subproperty van de Touch property van het form overigens, dus je moet eerst even Touch openklappen in de Object Inspector om alle gesture zaken te zien.

Gestures

Onder Gesture is een verzameling Standard gestures te vinden. Die begint met de gesture “Left”, dan “Right”, etc. Per Gesture staat er een checkbox (of de gesture gekoppeld is aan een actie op event) en een kleine afbeelding die de gesture beweging moet voorstellen: een horizontale lijn voor Left en Right (bij Left gaat de veegbeweging van rechts naar links, en bij Right net andersom natuurlijk).

Voor iedere gesture kunnen we vanuit de Object Inspector een New Action aanmaken (waar we dan de OnExecute event handler van moeten implementeren), of we kunnen een bestaande standaard action kiezen (en daar was de TActionList voor nodig). Voor ons voorbeeld wil ik bijvoorbeeld de Left gesture koppelen aan een Standard Action uit de Dataset categorie, namelijk TDataSetPrior (om naar het vorige record te gaan).



Na het koppelen van de Left gesture, is het daarna even zo makkelijk om de Right gesture te koppelen aan de TDataSetNext action, de ChevronLeft aan de TDataSetFirst en de ChevronRight aan de

TDataSetLast action. De twee Chevron gestures zijn te vergelijken met de groter dan en kleiner dan tekens, maar dan met je vinger over het scherm.

Actie!

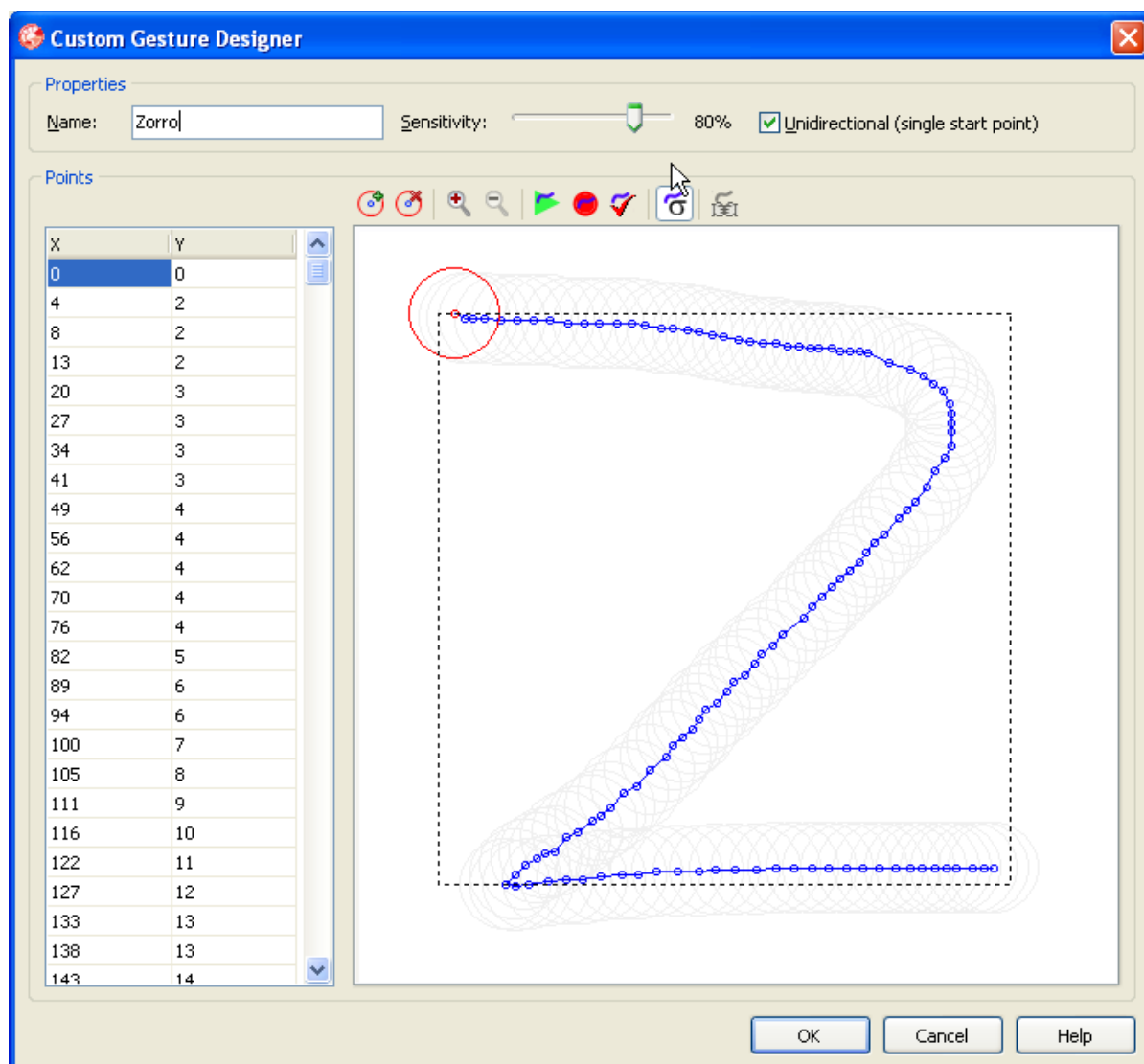
Wat nog rest is het compileren van het project. Dan kunnen we het draaien en uitproberen. Wie een TouchScreen heeft kan nu met vingerbewegingen over het scherm de Left, Right, ChevronLeft of ChevronRight bewegingen maken, om zodoende door de verschillende records van de biolife tabel te lopen. Geen toetsenbord of TDBNavigator meer nodig.

En wie geen TouchScreen heeft, kan ook zonder gelukkig: want je kan de veegbewegingen ook met de muis nadoen (dan moet je de linkermuisknop ingedrukt houden terwijl je de veegbeweging over het scherm maakt). Maar dat is natuurlijk niet zo leuk als de ervaring met een echt touch scherm zoals in een Kiosk op locatie waar geen muis of toetsenbord is.

Eigen Bewegingen

Behalve de ingebouwde gestures, is het met Delphi 2010 ook mogelijk om zelf nieuwe gebaren toe te voegen. Dat kunnen we met de GestureManager zelf doen: even met de rechtermuisknop erop klikken en de Custom Gestures kiezen. Er volgt dan een dialoog waarin we op Create kunnen klikken om een nieuwe gesture te maken, waarbij we met de vinger (of de muis) het "pad" van de gesture kunnen tekenen, en vervolgens kunnen bijstellen en checken (door het na te spelen en te zien of het herkend wordt).

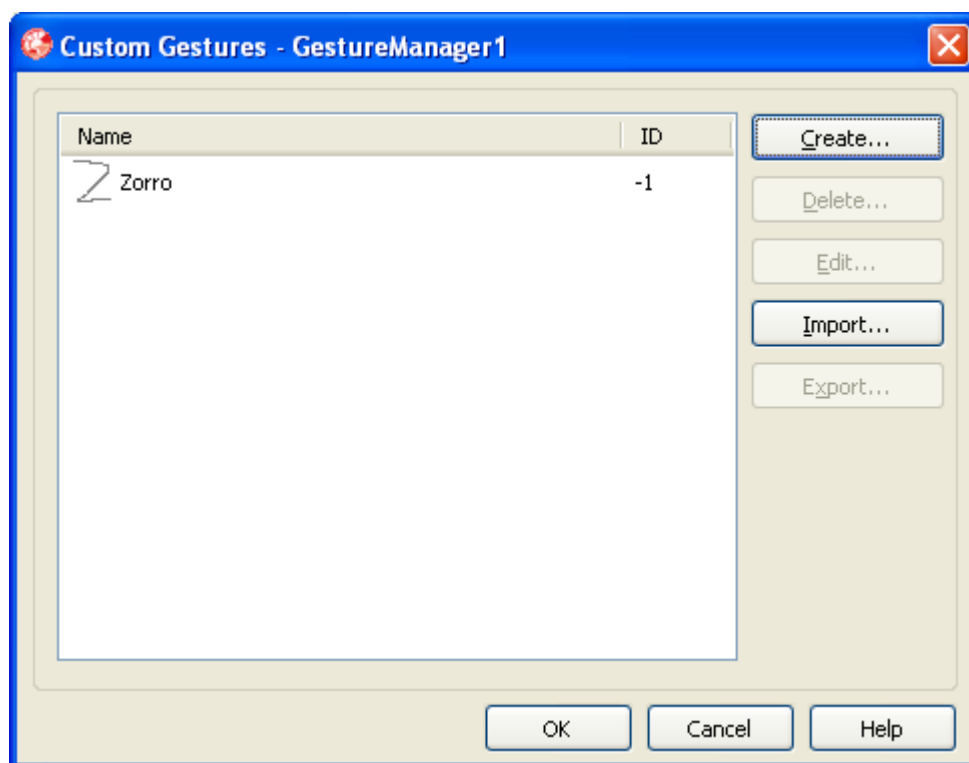
Als test heb ik de vorm van een Z nageveegd, en die gesture de naam Zorro gegeven, zoals op de volgende screenshot te zien is.



Het is wel belangrijk om te beseffen van gestures uit één vloeiende beweging moeten bestaan. Je kunt dus geen X maken ofzo, want dat zijn twee losse strepen en niet één. Voor de Z die ik gemaakt heb, moest ik nog wel de sensitivity aanpassen (standaard staat die op 80%), om te zorgen dat we de Z ook daadwerkelijk kunnen herkennen zonder dat je precies over de blauwe bolletjes gaat bij het naspelen.

In de Custom Gesture dialoog kun je tevens punten uit de gesture halen, nieuwe punten toevoegen, eventueel in- of uitzoomen, een simulatie van de gesture afspelen en deze tenslotte proberen na te spelen om te zien of hij herkend wordt (waarbij je dan de sensitivity moet aanpassen, zoals ik net al zei). Het kost wat moeite, maar uiteindelijk heb je wel je custom gesture die je kan gebruiken in je toepassing.

Als je op OK klikt, kom je terug in de Custom Gesture dialoog van de GestureManager, en zie je de vorm van de gesture, plus de naam en het ID (dat begint bij -1 te tellen, en gaat dan verder omlaag).



Gelukkig hoef je niet voor iedere toepassing steeds opnieuw je custom gestures na te spelen, want je kan custom gestures opslaan (exporteren naar een .dgp bestand – Delphi Gesture File), en dan in een andere toepassing weer importeren in de TGestureManager.

Nu we een custom gesture hebben, kunnen we daar op twee manieren mee omgaan. Ten eerste kunnen we de OnGesture event van het form gebruiken. In deze event handler krijgen we de EventInfo mee van type TGestureEventInfo, en een var parameter genaamd Handled om aan te geven of we zelf deze gesture hebben afgehandeld. Overigens komen we alleen in de OnGesture event handler als de binnenkomende gesture niet al is afgehandeld door één van de gestures die we aan een action hebben gekoppeld (zoals de Left, Right, ChevronLeft en ChevronRight).

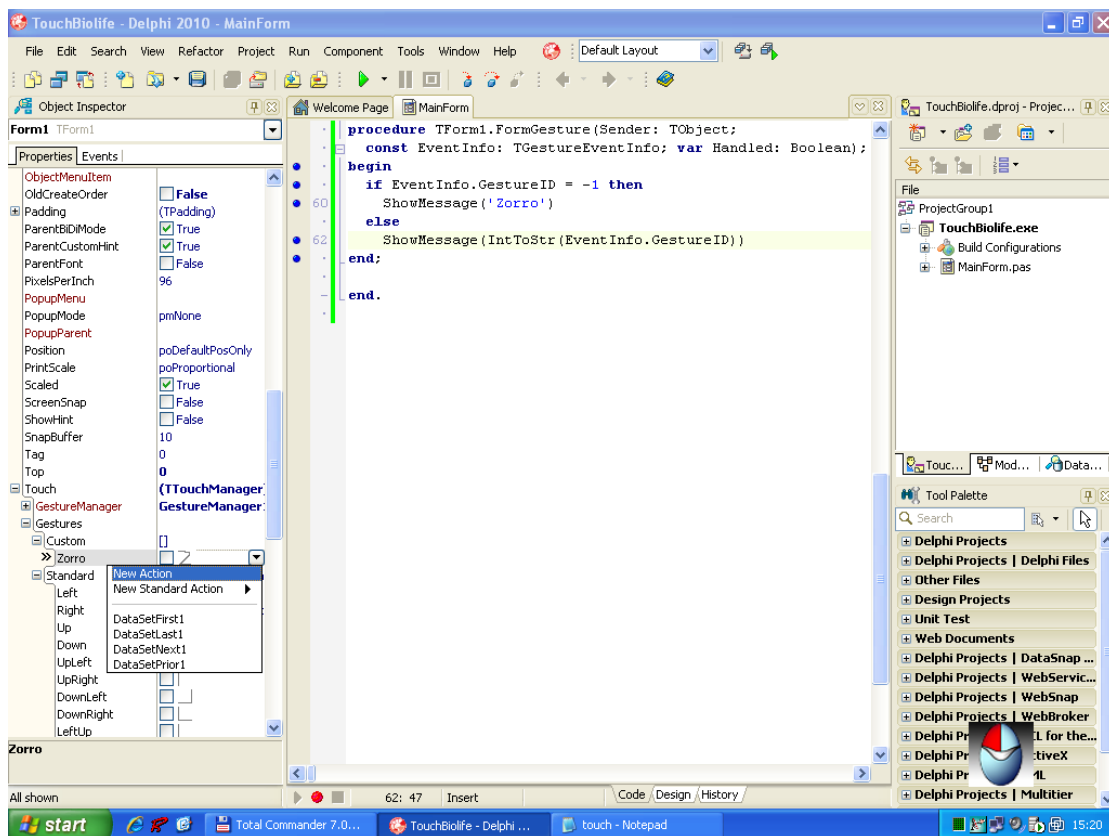
Om onze custom gesture met GestureID -1 te onderscheppen en erop te reageren heb ik de volgende code geschreven:

```
procedure TForm1.FormGesture(Sender: TObject;
  const EventInfo: TGestureEventInfo; var Handled: Boolean);
begin
  if EventInfo.GestureID = -1 then
    ShowMessage('Zorro')
  else ShowMessage(IntToStr(EventInfo.GestureID))
end;
```

Als er een gesture wordt gemaakt die niet herkend is, dan is het GestureID gelijk aan 0. En -1 is onze eerste custom gesture.

We moeten op deze manier wel bijhouden welke ID's onze custom gestures hebben, en gelukkig is er ook nog een andere manier om toegekende custom gestures in te stellen en te koppelen aan actions zoals we dat eerder ook met Standard gestures hebben gedaan.

Kijk nog eens terug naar de derde screenshot. Daarbij zien we onder de Touch property de Gesture subproperty met meteen daaronder de Standard lijst met gestures. Dat was het geval toen er nog geen custom gestures waren toegevoegd aan de TGestureManager. Nu is dat wel het geval, en laat de Object Inspector voor de lijst met Standard gestures nog een andere lijst zien: met Custom gestures, die in dit geval bestaan uit één custom gesture: de Zorro. Je ziet zelfs het plaatje van het pad van deze custom gesture, in de vorm van een ruwe Z.



We kunnen nu net als bij de standard gestures een action koppelen, in dit geval maak ik een nieuwe action aan, en selecteer die vervolgens in de Object Inspector waarna ik de OnExecute event van de nieuwe action implementeer, bijvoorbeeld als volgt:

```
procedure TForm1.Action1Execute(Sender: TObject);
begin
  ShowMessage('Zorro also likes to touch Delphi 2010');
end;
```

Nu deze geïmplementeerd is, zal de OnGesture event handler niet meer reageren op de Zorro gesture (probeer maar eens).

Het draaien van de toepassing, die ik Touching Biolife heb genoemd, doet in voorkomen niet onder voor een normale toepassing. Aan de screenshot is (nog) niet te zien dat het hier om een touch- en gesture-enabled toepassing gaat. Je zou hooguit door het ontbreken van de TDBNavigator kunnen afvragen hoe je naar het volgende record gaat.

Touching Biolife

Species No

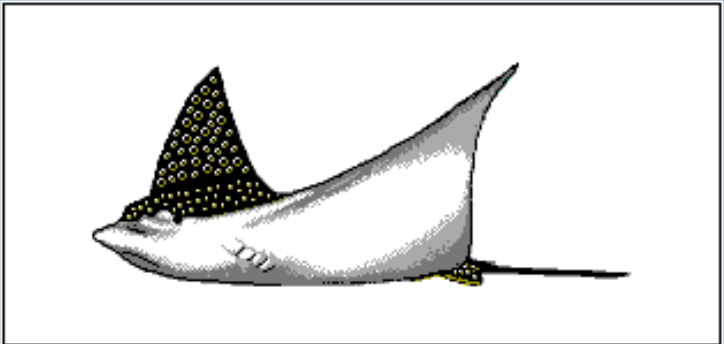
Category

Common_Name

Species Name

Length (cm)

Notes

Graphic


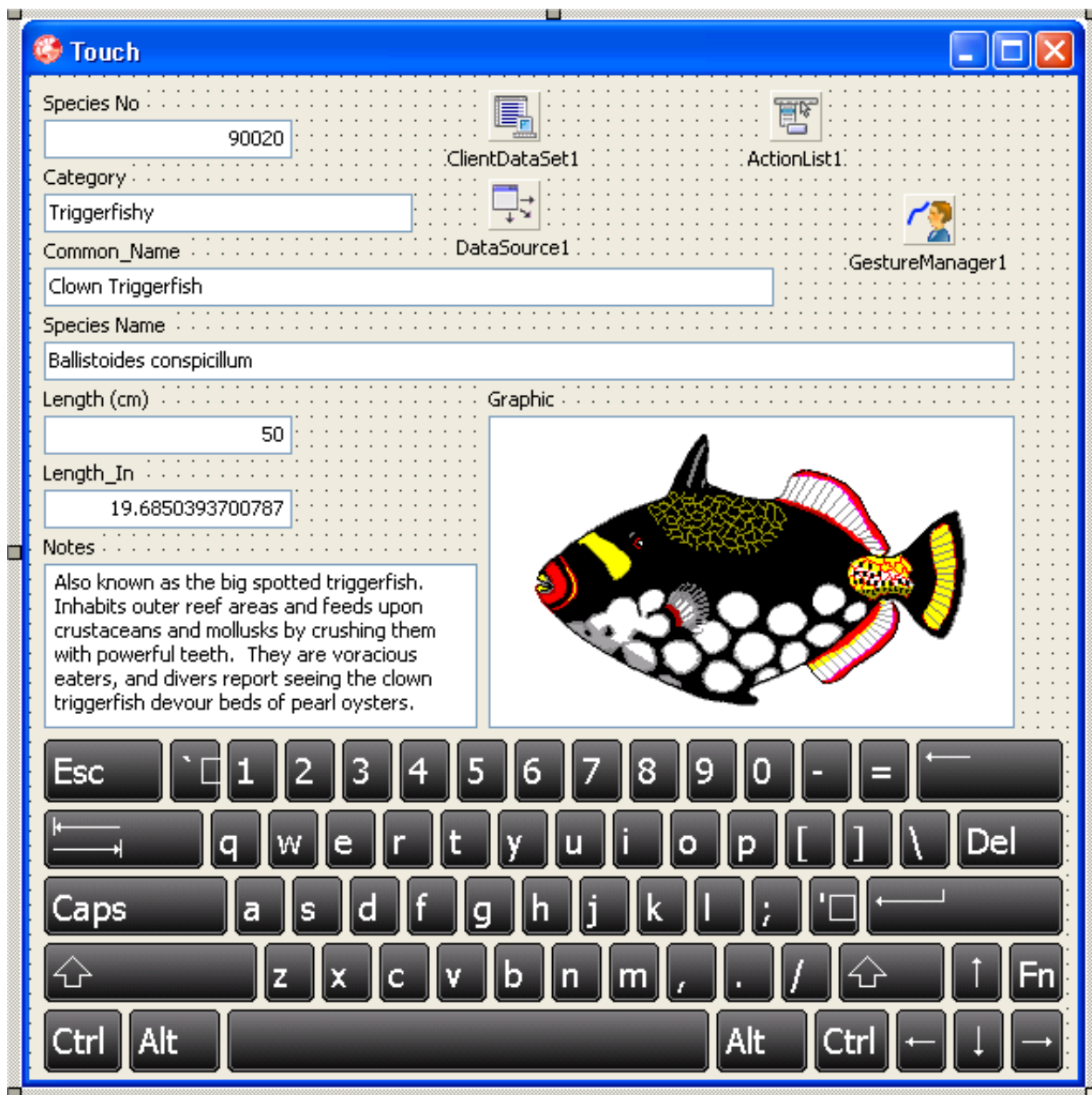
Het vegen met je vinger over het scherm van links naar rechts (voor het volgende record) of van rechts naar links (voor de vorige) is in het begin van onwennig, maar gaat al snel heel natuurlijk. Dit effect is mooier als het bijvoorbeeld een landkaart betreft waarbij je met je vinger de kaart een stuk kunt "verschuiven" naar links, rechts, boven of beneden..

Dat kan ook met deze gestures, en zelfs nog beter met Windows 7 specifieke Multi-touch en interactive gestures waarbij de EventInfo structuur van de OnGesture event handler is ingevuld met velden voor de inertia (de snelheid waarmee je vinger over het scherm bewoog) en de start en stop coördinaten van de gesture. Deze mogelijkheden worden alleen door Windows 7 ondersteund, in combinatie met hardware die ook compatible is met Multi-/interactive gestures (niet een normaal TouchScreen, maar bijvoorbeeld een smart netbook).

Touch Keyboard

Wat ik nog wel kan laten zien is de laatste stap om de toepassing volledig muis en toetsenbord onafhankelijk te maken: voor deployment op een touch screen in een kiosk bijvoorbeeld. Omdat er soms toch tekst ingevoerd moet worden, heeft Delphi 2010 nu een speciaal TTouchKeyboard component. Eentje die voor de huidige locale de keyboard layout kan laten zien (zodat gebruikers uit Frankrijk ook makkelijk de accenten kunnen invoeren). In de screenshot op de volgende pagina staat de US-International versie aan de onderkant van het Touching Biolife form:

Behalve een volledige keyboard, kan de TTouchKeyboard zich ook als numeriek keyboard voordoen, door de layout property van Standard naar NumPad te zetten. In beide gevallen kunnen we via de KeyCaptions property de caption van enkele speciale toetsen zetten, zoals de Esc, Backspace, Tab, Del, Caps, Enter, LeftShift, RightShift, LeftCtrl, RightCtrl, LeftAlt en RightAlt. Helaas is het verder niet mogelijk om de layout van het toetsenbord aan te passen – net zo min als het mogelijk is om het toetsenbord de Franse toetsen te laten zien bijvoorbeeld, zonder dat de Windows locale daarop is ingesteld.



Dit is natuurlijk niet een effectieve manier om de input te verzorgen: handiger is het om het TTouchKeyboard component in een kleine pop-up dialoog te voorschijn te toveren op het moment dat iemand een stukje tekst moet of wil invoeren (en dus niet plompverloren onderaan het scherm). Maar als het keyboard er eenmaal staat, dan kun je met je vinger op de "toetsen" drukken voor het gewenste resultaat. En bij gebrek aan TouchScreen kun je hetzelfde doen met de muis. Dus als software ontwikkelaar hoef je wat dat betreft niet zelf over een TouchScreen te beschikken: gestures en het virtuele keyboard werken ook met de muis. Alleen voor de Multi-touch of interactive touch heb je zowel Windows 7 als de specifieke hardware nodig (ook voor het testen), en vandaar dat ik die bewaar voor een volgende keer.

Deployment

Om er een echte stand-alone executable van te maken (met de data van de TClientDataSet in de DFM file), moeten we ook nog even MidasLib aan de uses clause van het project toevoegen, anders moeten we de MIDAS.DLL meeleveren.

Overigens bevat Delphi 2010 Enterprise (of hoger) de source code van de MIDAS.DLL, te vinden in de C:\Program Files\Embarcadero\RAD Studio\7.0\source\db\midas directory, in de vorm van een C++Builder project (dus met name bruikbaar voor wie RAD Studio heeft, met C++Builder erin).

DataSnap 2010

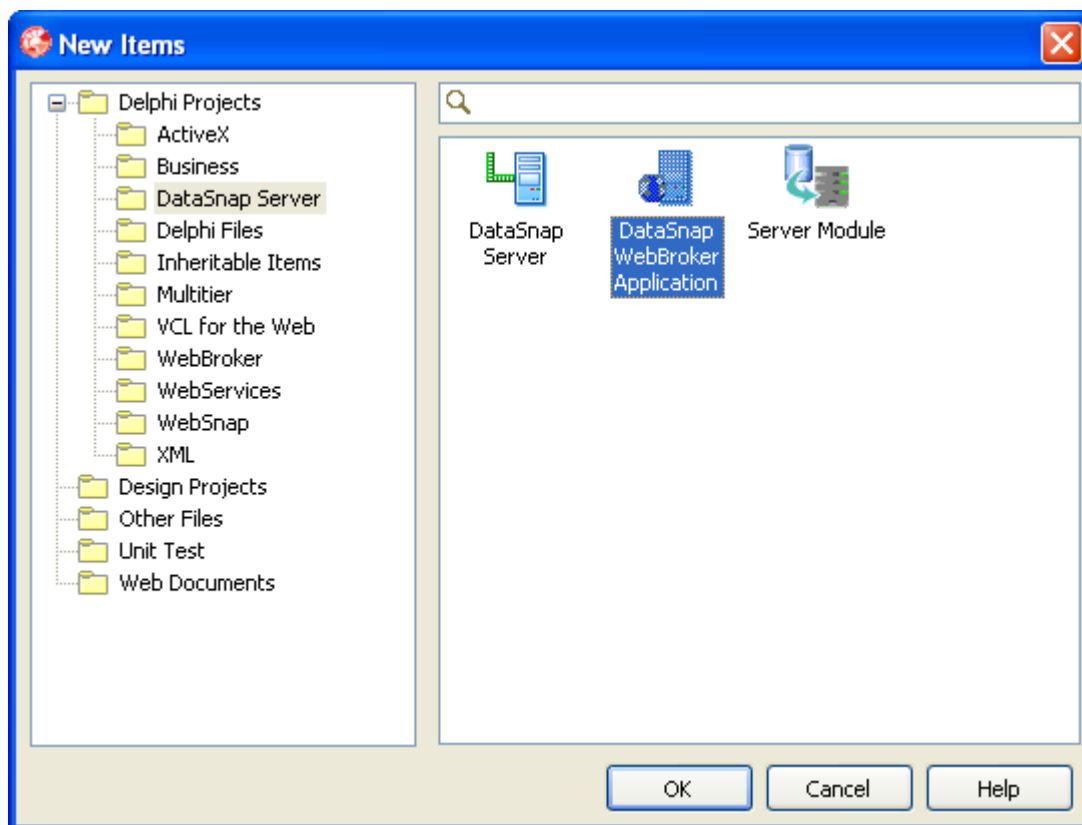
Zoals ik bij mijn Delphi 2010 seminar van 11 september al liet zien, zijn er in DataSnap 2010 een flink aantal uitbreidingen en verbeteringen gekomen t.o.v. DataSnap 2009. En DataSnap 2009 was al de start van de nieuwe architectuur die niet langer op COM gebaseerd was. Dus nu meer dan ooit is DataSnap in Delphi 2010 Enterprise (of Architect) de moeite waard om nader te bekijken.

Tijdens de Software Development Conference (SDC 2009) geef ik op dinsdag 20 oktober een sessie over DataSnap 2010 waarin ik diep zal ingaan op de nieuwe features, waaronder de DataSnap targets, de beschikbare transport protocol, de filters die we erop kunnen loslaten (voor compressie maar ook zelfgemaakte encryptie), de callbacks, en natuurlijk deployment issues.

DataSnap Targets

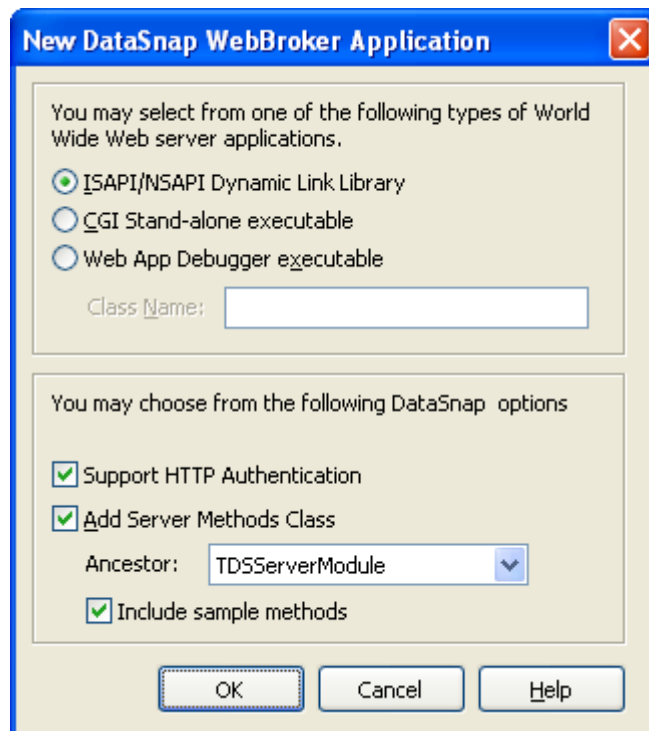
Delphi 2009 introduceerde de nieuwe DataSnap architectuur – niet langer op COM gebaseerd – maar hierbij moesten we zelf de server executable bouwen met de drie DataSnap componenten (DSServer, DSServerClass en de TDSTCPServerTransport component). Dat is nu niet meer nodig: Delphi 2010 bevat Wizards om het skelet van de DataSnap server toepassing al voor te bakken en de Server Module te maken met de benodigde DataSnap componenten erop.

En het beste nieuws is dat we niet één, maar twee wizards hebben voor twee soorten DataSnap server toepassingen: een “normale” DataSnap Server wizard (met keuze uit een Windows GUI, Windows Service of Console toepassing), alsmede een DataSnap WebBroker Application wizard.



Voordeel van de DataSnap WebBroker Application wizard is dat je hiermee een CGI, ISAPI of Web App Debugger toepassing kunt maken die onder IIS te deployen is (CGI en ISAPI) of vanuit de IDE te debuggen is (de Web App Debugger target). In het geval van CGI of ISAPI zorgt IIS dus voor het afhandelen van het inkomende en uitgaande data verkeer, en kun je je wat de server betreft richten op de server methods. CGI is overigens niet echt een optie als het gaat om DataSnap toepassingen: want dan kan wordt de toepassing voor ieder request gestart en weer gestopt, wat relatief veel tijd kost, en dat kan niet echt de bedoeling zijn.

Behalve de keuze voor de target kun je zowel in de DataSnap WebBroker als in de normale DataSnap wizard kiezen voor HTTP Authentication support (altijd fijn), de voorvader class van het server method object (minstens TPersistent, maar je kan ook voor data module of TDSServerModule kiezen) en een voorbeeld EchoString server method.



De nieuwe wizards zorgen ervoor dat je in ieder geval al meteen een DataSnap Server toepassing hebt met alle verbindingen tussen de verschillende DS componenten meteen goed in orde.

DataSnap Transport

Een andere uitbreiding in DataSnap 2010 zijn de mogelijke transport protocollen die gebruikt kunnen worden. Waar DataSnap 2009 alleen de TDSTCPServer Transport component had, kent Delphi 2010 ook het TDSHTTPServerTransport component, of beter nog de daarvan afgeleide TDSHTTPService component. Deze ondersteunt HTTP, zodat we nu ook in de stand-alone DataSnap server toepassingen (dus zonder IIS te gebruiken) ook voor HTTP kunnen kiezen..

Tip: waar TCP/IP standaard over poort 211 werkt – en die moet je dus aanpassen, want poort 211 is zo langzamerhand wel bekend bij “sniffers” – daar werken de HTTP service componenten standaard met poort 80. Die moet je dus ook even aanpassen als je IIS op je machine hebt staan, want je kan niet een twee proces starten (je DataSnap server toepassing) dat poort 80 gebruikt. Dan krijg je een fatale fout, en de vraag of je je toepassing wilt (post-mortem) debuggen. Kan even duren voor je door hebt dat dat dan aan die poort 80 ligt (ik spreek helaas uit ervaring).

HTTP Authentication

Als je HTTP of HTTPS (met IIS) gebruikt, dan is het ook fijn om de optie HTTP Authentication aan te zetten; Om dit aan de DataSnap server kant eenvoudig af te handelen moeten we vervolgens een TDSHTTPServiceAuthenticationManager component gebruiken, en daar de HTTPAuthenticate event van implementeren. Als input argumenten krijgen we de waarde van Protocol, Context, User en Password, en we kunnen op basis daarvan de var parameter valid een waarde geven.

```
procedure TServerContainer1.DSHTTPServiceAuthenticationManager1HTTPAuthenticate  
  (Sender: TObject; const Protocol, Context, User, Password: string;  
   var valid: Boolean);
```

REST en JSON

Het voordeel van HTTP Authentication is dat je ook clients die niet in Delphi zijn geschreven kan koppelen aan de DataSnap server, vooral omdat er nu ook een REST interface is, waar je JSON data mee terugkrijgt. Wie de demo DataSnap server maakt moet maar eens de volgende URL aanroepen van het voorbeeld method EchoString:

<http://localhost/datasnap/rest/TServerMethods1/EchoString/Delphi>

Het resultaat is een JSON object met de inhoud "Delphi" (het laatste deel van de URL is de string die we willen echo-en). Dit is uiteraard nog maar een eenvoudig voorbeeld, maar je kan op deze manier ook een hele tabel teruggeven of records, en daar dan bijvoorbeeld met PHP een web pagina mee vullen.

Callbacks

De laatste nieuwe feature in DataSnap 2010 die ik even wil noemen is de ondersteuning voor callbacks. Dat wil zeggen dat je vanaf de server een callback naar de client kan doen, maar alleen gedurende de aanroep van een server method door de client. En de callback method zelf moet de client dan als argument meegeven aan de server method (dus die moet bij voorbaat al geschikt zijn om een callback method als argument te kunnen verwerken).

Als voorbeeld kunnen we de server method van EchoString aanpassen met een Execute callback method. Aan de client kant moeten we eerst de callback method aan de client kan definiëren door een afgeleide class van TDBXCallback te maken en de Execute method ervan te implementeren:

```
type
  TCallbackClient = class(TDBXCallback)
    function Execute(const Arg: TJSONValue): TJSONValue; override;
  end;

function TCallbackClient.Execute(const Arg: TJSONValue): TJSONValue;
var
  Data: TJSONValue;
begin
  Data := TJSONValue(Arg.Clone);
  ShowMessage(TJSONObject(Data).Get(0).JsonValue.Value);
  Result := Data;
end;
```

Vervolgens kunnen we aan de server kant de implementatie van de EchoString aanpassen zodat we de callback aanroepen

```
function TServerMethods2.EchoString(Value: string;
  const callback: TDBXCallback): string;
var
  msg: TJSONObject;
  pair: TJSONPair;
begin
  Result := Value;
  msg := TJSONObject.Create;
  pair := TJSONPair.Create('ECHO', Value);
  pair.Owned := True;
  msg.AddPair(pair); // new echo value

  callback.Execute(msg);
end;
```

Natuurlijk is ook dit maar een eenvoudig voorbeeld, maar het laat wel zien hoe we tijdens het uitvoeren van de EchoString method aan de server kant, de callback van de client kunnen aanroepen. In de praktijk kan dit nuttig zijn om tijdens een update ook meteen de server in staat te stellen andere informatie naar de client te sturen.

Tijdens mijn sessie bij de SDC 2009 zal ik op deze en andere DataSnap 2010 voorbeelden dieper ingaan, gevolgd later dit jaar door DataSnap 2010 trainingen in Helmond Brandevoort.

Software Developer Conference 2009

De Software Development the Netherlands (SDN) organiseert op 19 en 20 oktober weer de jaarlijkse SDN conferentie, ook wel bekend onder de naam SDC. Zie <http://www.sdc.nl/> voor de website met meer details. Er zullen niet minder dan 16 Delphi sessies zijn, door bekende sprekers zoals Cary Jensen, Marco Cantù, Chad Hower, Hadi Hariri, van Embarcadero Pawel Glowacki en Barry Kelly, en tot slot ikzelf met een sessie over DataSnap.

Marco Cantù

- Creating Windows 7 Applications in Delphi
- Multi-Threading in Delphi
- Domain Specific Languages in Delphi

Pawel Glowacki

- Delphi Natural Input

Hadi Hariri

- ASP.NET MVC in Delphi Prism
- Advanced ASP.NET MVC in Delphi Prism

Chad Hower

- Connecting to .NET with CrossTalk
- IntraWeb: Then, Now, and Future

Cary Jensen

- Introduction to WPF Development with Delphi Prism
- An Introduction to LINQ in Delphi Prism
- 9 Thread Synchronization Options in Delphi Compared
- Advantage Database Server and Delphi: A Perfect Match

Barry Kelly

- Delphi Compiler RTTI Enhancements
- Design of a combined garbage-collection and serialization mechanism

Bob Swart

- Delphi 2010 DataSnap Enhancements

Of je nu Delphi of Delphi Prism gebruikt, met deze sessies krijg je zowel informatie over de laatste versie(s) als over bestaande mogelijkheden. De SDC is het grootste Delphi event in de BeNeLux, en is dan ook niet te missen!

Zie <http://www.sdc.nl/Registration/VisitorRegistration/tabid/77/language/en-US/Default.aspx> voor wie zich nog niet heeft ingeschreven. SDN-leden krijgen 100 Euro korting op de prijs van 695 Euro. Niet leden kan ik een korting van 50 Euro aanbieden als ze de referentie code "eBob42" vermelden op het inschrijfformulier.

Ik hoop veel Delphi ontwikkelaars te zien bij de SDC.

De eerste bugjes in Delphi 2010

Delphi 2010 was nog maar net uit, of de eerste publieke bugjes waren al gevonden – en gemeld in Quality Central. Hieronder volgt een kleine samenvatting van een aantal van deze bugs. Wie zelf een bug vindt, zou die eigenlijk ook moeten aanmelden in Quality Central, of melden aan zijn/haar reseller, zodat deze bug opgelost kan worden. Belangrijk daarbij is om te zorgen voor een klein code voorbeeld waarmee de bug gereproduceerd kan worden. Alleen maar de melding dat feature X niet goed werkt zal helaas niet snel leiden tot een oplossing.

TStopwatch

De diagnostics.pas unit bevat een handig record TStopwatch, dat gebruikt kan worden om de tijd te meten die nodig is om bepaalde stukken code uit te voeren. Helaas zit in de rapportage via de stopwatch in de class function GetTimeStamp, die de TickCount met 1000 in plaats van 10000 vermenigvuldigt, waardoor in lage resolutie (als de highresolution counter niet wordt gebruikt) de TimeStamp verkeerd berekend wordt uit de GetTickCount, en een factor 10 te klein is.

Zie <http://qc.embarcadero.com/wc/qcmain.aspx?d=77628> voor meer details.

DataSnap en BLOB

Er zitten een flink aantal uitbreidingen in DataSnap 2010, maar ook een nieuwe bug: het blijkt dat DataSnap 2010 soms een probleem heeft met BLOB velden. Als je twee BLOB velden in één SELECT statement ophaalt, en je doet dat niet in de juiste volgorde, dan wordt het tweede BLOB veld niet goed doorgegeven. Doet een beetje denken aan het verhaal van uni-directional datareaders waarbij je ook de velden in de juiste volgorde moet uitlezen.

Zie <http://qc.embarcadero.com/wc/qcmain.aspx?d=77295> voor meer details.

TComboBox BevelKind bkFlat

Er zit ook iets gek in de TComboBox controls: als je de BevelKind property op bkFlat zet (of een andere waarde dan bkNone), dan gaat deze control zich een beetje raar gedragen. Tijdens design-time wordt hij steeds groter als je hem probeert ergens anders neer te zetten. Een beetje vreemd resultaat, maar als je BevelKind weer op bkNone zet is alles weer gewoon.

Andere controls, zoals de TEdit, hebben overigens geen probleem met de BevelKind property.

Zie <http://qc.embarcadero.com/wc/qcmain.aspx?d=77265> voor meer details.

Uninstall Delphi 2010

Tot slot nog een bug bij het de-installeren van Delphi 2010: daar kunnen Delphi 2007 en 2009 last van hebben, omdat na het verwijderen van Delphi 2010 de bestanden CC3280MT.DLL en CC3290MT.DLL verdwenen zijn uit de Windows\System32 directory, terwijl die ook nodig zijn voor Delphi 2007 en 2009, aldus het error report te <http://qc.embarcadero.com/wc/qcmain.aspx?d=77157>

Ik moet zeggen dat ik zelf op mijn systeem met RAD Studio 2007 en Delphi 2010 alleen een CC3260MT.DLL in mijn Windows\System32 directory heb, en geen 80 of 90, maar misschien geldt dat niet voor iedereen.

Deze en andere Delphi en RAD Studio 2010 bugs zijn terug te vinden op Quality Central via de volgende link: <http://qc.embarcadero.com/wc/qcmain.aspx?search=1&proj=10&vers=14.0>

Mijn support nieuwsgroepen zijn ook een goede plek om bugs te melden (of deze en andere bugs te bespreken). Wie Delphi, Delphi Prism, RAD Studio of C++Builder bij mij afneemt krijgt automatisch toegang tot deze support nieuwsgroepen.

Delphi bestellen bij Bob Swart

Onderstaande prijzen zijn exclusief 19% BTW en gelden voor de Electronic Software Delivery (ESD) edities, die u kunt downloaden en kunt installeren met een bijbehorend serienummer dat u ontvangt. Een ESD licentie betreft een download – u kunt zelf indien nodig een ISO image op DVD branden.

Delphi Prism 2010 Enterprise

New User	€ 700,=
Subscription (verplicht 1 ^e jaar)	€ 699,= per jaar



Delphi Prism 2010 Professional

New User	€ 250,=
Subscription (verplicht 1e jaar)	€ 249,= per jaar

Delphi 2010 Architect (Win32)

New User	€ 3.499,=
Upgrade	€ 2.299,=
Subscription Delphi for Win32	€ 1.050,= per jaar



Delphi 2010 Enterprise (Win32)

New User	€ 1.999,=
Upgrade	€ 1.299,=
Subscription Delphi for Win32	€ 600,= per jaar

Delphi 2010 Professional (Win32)

New User	€ 899,=
Upgrade	€ 399,=
Subscription Delphi for Win32	€ 270,= per jaar

CodeGear RAD Studio 2010 Architect (ESD)

New User	€ 4.299,=
Upgrade	€ 2.799,=
Subscription CodeGear RAD Studio	€ 1.290,= per jaar



CodeGear RAD Studio 2010 Enterprise (ESD)

New User	€ 2.799,=
Upgrade	€ 1.799,=
Subscription CodeGear RAD Studio	€ 840,= per jaar

CodeGear RAD Studio 2010 Professional (ESD)

New User	€ 1.399,=
Upgrade	€ 649,=
Subscription CodeGear RAD Studio	€ 420,= per jaar

Special Aanbieding: bij de aanschaf van een Embarcadero tool voor 24 september 2009, krijgt u een coupon voor een tweede Embarcadero product gratis.

Zie <http://www.eBob42.com/CodeGear> of stuur e-mail naar Bob@eBob42.com voor meer informatie of om een bestelling te plaatsen