Delphi OplossingsCourant

Vol 2. No. 5. Een publicatie van het TAS Advanced Technologies Delphi OplossingsCentrum- http://www.tas-at.com/doc

Best, 7 december 2000,

Op het moment dat ik dit welkomstwoord schrijf is de rust betreffende de aangekondigde overname van de TAS Groep door Pink Roccade weer een beetje wedergekeerd. TAS Advanced Technologies zal in de nieuwe organisatiestructuur opgaan in Everest uit Den Bosch. Voor het Delphi OplossingsCentrum (DOC) blijft het kantoor Best de thuishaven. En afgezien van het feit dat het vanaf 1 januari 2001 het "Everest DOC" zal zijn (in plaats van het TAS-AT DOC), zal er verder niet zoveel veranderen. We blijven ons intern en extern richten op Delphi en gerelateerde technieken en oplossingen. Als er iets verandert is het wel het feit dat Pink Roccade een vijf keer zo grote "interne" markt is dan de TAS Groep was. En dan heb ik het nog niet eens over de nieuwe groep klanten van Pink Roccade die nu ook van de kennis en ervaring van het DOC kunnen profiteren.
Als Inprise Enterprise Solutions Partner doet het Delphi OplossingsCentrum mee met de SDGN Developers Conference op 8 december. Ook hier zullen Micha Somers en ikzelf weer een Kylix presentatie en demonstratie verzorgen. Zodra Kylix officieel beschikbaar is zal dit opgevolgd worden met Kylix Clinics (zie http://www.drbob42.nl/training voor details).

Ook deze Delphi OplossingsCourant, geeft aan waar onze mensen zich mee bezighouden. De Delphi OplossingsCourant verschijnt vanaf nu vier keer per jaar. Mocht u als lezer geen (toekomstig) nummer willen missen, dan kunt u zich vrijblijvend aanmelden door een e-mailtje te sturen naar doc@tas-at.com onder vermelding van "abonnement DOC".

Inhoudsopgave

De **TAS-AT Delphi OplossingsCourant** is een productie van het TAS Advanced Technologies Delphi OplossingsCentrum, met medewerking van Arjan Jansen, Arnim Mulder en Micha Somers.

Eindredactie: Bob Swart - e-mail: doc@tas-at.com

Kylix = Delphi voor Linux

Auteur: **Bob Swart**

Opmerking: Kylix is nog niet beschikbaar, en is slechts een codenaam voor "Delphi voor Linux". Dit artikel is gebaseerd op voorlopige informatie en mijn persoonlijke inschattingen daarbij, en kan niet als definitief beschouwd worden. Dat is ook een van de redenen waarom ik in dit artikel nog helemaal geen code voorbeelden kan geven (maar geloof me: die komen er zodra Kylix officieel beschikbaar is).

Linux

De eerste release van Kylix die we kunnen verwachten zal te vergelijken zijn met Delphi Professional, maar dan voor Linux in plaats van voor Windows. Dat houdt dus in dat we geen Enterprise zaken vergelijkbaar met SQL Links, CORBA (VisiBroker for Delphi) of MIDAS terug zullen vinden in deze eerste release van Kylix. Dat is overigens niet helemaal waar, want een klein stukje van MIDAS (de ClientDataSet component) zal vanaf versie 6 ook in de Professional versies van Delphi en C++Builder zitten, maar dat is het onderwerp van een ander verhaal.

Kylix zal werken onder de meest recente editie van Linux; dus bijvoorbeeld SuSE, Mandrake of Red Hat 7.0 (zoals in mijn geval), en zal daarbij dan zowel de GNOME als de KDE desktops ondersteunen. Wat betreft de zogenaamde desktop thema's van GNOME en KDE is het op dit moment nog niet duidelijk of er 100% support in de eerste release van Kylix zal zitten, maar men is hier wel stevig mee bezig. De Kylix demonstraties laten in ieder geval zien dat de Kylix ontwikkelomgeving zélf zonder problemen draait onder GNOME of KDE met de desktop thema's.

Kylix = Delphi voor Linux

Kylix is een codenaam voor een heel project, dat als één van de resultaten een native ontwikkelomgeving voor Linux heeft. Deze ontwikkelomgeving zal in de eerste release te vergelijken zijn met Delphi Professional (zoals ik eerder al aangaf).

De tweede release van Kylix zal de C++Builder Professional voor Linux zijn, terwijl de derde release een soort combinatie van deze twee is (dus zowel C++Builder als Delphi in één omgeving), die bovendien de Enterprise zaken (zoals MIDAS en CORBA enzo) zal bevatten.

Deze Kylix Enterprise Studio verwacht ik zelf pas over een jaar, dus richt ik me voorlopig maar op de eerste release, die overigens ook een native command-line compiler DCC zal bevatten, en native Linux x86 ELF-machine code zal genereren (met Linux x86 bedoel ik Linux draaiend op Intel compatible hardware, dus niet bijvoorbeeld op een SUN Sparc, of DEC machine die Linux draait).

Windows en Linux

Naast de Kylix ontwikkelomgeving zelf, is een van de andere doelstellingen van *Project Kylix* (het mogelijk maken van) het porteren van toepassingen van Windows naar Linux en weer terug. De huidige Visual Component Library (VCL) van Delphi is bovenop de Windows API gebouwd, zowel wat routines betreft (kijk maar eens naar de Windows en SysUtils units) als wat de visuele componenten betreft (de VCL componenten zijn in feite bovenop de Windows "controls" gebouwd). En ook de Borland Database Engine (BDE) is alleen maar voor Windows gebouwd. Voor het ombouwen van de volledige VCL en BDE van Windows naar een Linux editie zou ontzettend veel tijd en moeite nodig zijn geweest. In plaats daarvan is besloten om een geheel nieuwe benadering te kiezen onder de naam CLX (spreek uit als "clicks"), wat staat voor Component Library X-platform. CLX bestaat uit een aantal onderdelen - daar kom ik zo op terug - en zal zowel onder Linux als onder Windows beschikbaar komen. Dus zowel met Kylix (de Delphi voor Linux) als met de toekomstige versie van Delphi voor Windows (die dan waarschijnlijk zowel met VCL als met CLX zal kunnen werken).

CLX

CLX kan gezien worden als een cross-platform VCL. Maar het is veel breder dan de VCL, want CLX bevat niet alleen maar "visuele" componenten, maar het hele scala aan cross-platform ontwikkelbouwstenen. De onderdelen van CLX heten BaseCLX, VisualCLX, DataCLX en NetCLX. In **BaseCLX** zitten de standaard zaken, zoals conversieroutines, systeemroutines (schijf en bestandsoperaties), en non-visuele componenten zoals TStringList.

VisualCLX bevat alle visuele componenten. Dit onderdeel van CLX is voor een deel gebaseerd op Qt, een cross-platform library van Trolltech die door Borland in licentie is genomen.

Qt is beschikbaar voor Windows en Linux (en nog andere platforms, waardoor VisualCLX potentieel ook op andere platforms beschikbaar zou kunnen komen in de toekomst). Je moet bij VisualCLX overigens niet alleen denken aan de TButton en TEdit, maar ook aan de data-aware componenten zoals TDBEdit en TDBGrid. Alle visuele componenten zitten in VisualCLX.

Waar VisualCLX onder andere de data-aware componenten bevat, zal **DataCLX** juist de data-access componenten bevatten. Niet door gebruik te maken van de BDE, want die zal niet naar Linux geport worden, maar door gebruik te maken van een nieuwe data access laag die dbExpress zal heten. Via dbExpress kunnen we straks bijvoorbeeld praten met InterBase en mySQL (en wellicht ook met Oracle 8i onder Linux, maar daar heb ik nog niks definitiefs over gehoord).

Het vierde en laatste onderdeel van CLX heet **NetCLX** en zal in feite alle internet en web-gerelateerde componenten bevatten. Dus zowel de FTP, POP3, SMTP en andere internet protocollen (in de vorm van de Indy componenten die de NetManage internet componenten op het component palette van Delphi zal vervangen), maar ook de WebBroker componenten zoals de PageProducer en TableProducers. Voeg daarbij de specifieke mogelijkheid om met zowel Kylix (onder Linux) als Delphi (voor Windows) straks web server toepassingen te kunnen maken voor de veelgebruikte Apache web server, en je begrijpt misschien dat daar erg naar wordt uitgekeken (in ieder geval door mijzelf).

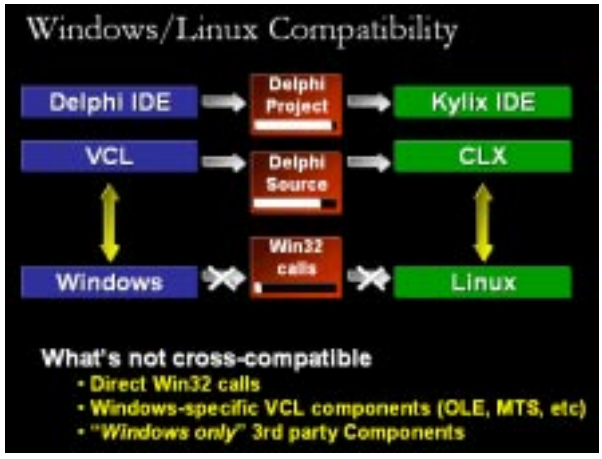
Cross-platform

Wie straks met Kylix gaat werken zal daarmee meteen ook CLX gebruiken. En als gevolg daarvan cross-platform toepassingen bouwen die zowel onder Linux als in de toekomstige versie van Delphi onder Windows zullen compileren en draaien. En da's fijn, want daarmee kun je plotseling voor een veel breder publiek je toepassingen beschikbaar maken (uiteraard zul je moeten wachten tot Delphi 6 uit is voordat je je CLX toepassing onder Delphi 6 kunt compileren, maar mijn persoonlijke verwachting is dat Delphi 6 reeds korte tijd na Kylix beschikbaar zou kunnen komen).

Van VCL naar CLX

Stel je wilt niet wachten tot Kylix uit is, of je hebt een bestaande toepassing en wilt die straks omzetten van Windows naar Linux. *Wat voor narigheid kun je dan allemaal verwachten?*

Vooral de mensen die ooit wel eens Borland Pascal source code naar Delphi hebben geport, of later van 16-bit Delphi naar 32-bits moesten weten dat het porten van toepassingen niet altijd een even fijne bezigheid is. En geloof me, porten van het ene besturingssysteem (Windows) naar het andere (Linux) maakt de zaak er niet eenvoudiger op, al is het maar omdat je moeite zult hebben om beide versies (Windows en Linux) naast elkaar te draaien. Onderstaand plaatje geeft wat zaken weer:



Als we uitgaan van het porten van een Delphi VCL toepassing, dan is een belangrijke bron van informatie de lijst van zaken die niet ondersteund zullen worden door Kylix onder Linux.

Deze zaken zullen dus op een andere manier geïmplementeerd moeten worden. Het betreft alle native Windows code, zoals aanroepen van Window APIs (MessageBox, TimeGetTime, etc.). Hetzelfde geldt voor alle hoger niveau Windows specifieke zaken zoals DDE, OLE, COM, DCOM, COM+, MTS, ActiveX controls, ActiveForms, Active Server Pages, ISAPI DLLs en ADO. Wow, dat is nogal een lijst. *Wat blijft er dan over, want we hadden al gezien dat ook de BDE niet onder Linux beschikbaar zal zijn?* Gelukkig valt het allemaal wel mee. InterBase is bijvoorbeeld zowel onder Windows als Linux aanwezig, en ook de meeste "normale" VCL componenten zullen ook als een vergelijkbare CLX component beschikbaar komen. Meestal met vergelijkbare properties, events en methods zodat de huidige VCL gebruikers (zowel de programmeurs als de toepassingen) met weinig aanpassingen van VCL naar CLX kunnen overstappen - mits de betreffende VCL component inderdaad in CLX voorkomt.

VCL vs. CLX

Moeten straks alle VCL toepassingen omgezet worden naar CLX? Is dit het einde van de VCL? Nee, natuurlijk niet. De VCL zal nog steeds verder ontwikkeld worden voor Delphi onder Windows, en zal daarbij juist alle nieuwe native Windows zaken gaan bevatten (in Delphi 6 zullen waarschijnlijk heel wat nieuwe Windows 2000 features in de VCL terug te vinden zijn, zoals COM+ support en dat soort nieuwe zaken). Wil je dus een echte Windows toepassing schrijven, en daarbij gebruik maken van alle toeters en bellen, dan is CLX niet geschikt, maar moet je gewoon de VCL blijven gebruiken. CLX is met name geschikt voor cross-platform toepassingen, waar de VCL voor de echte Windows toepassingen de beste keuze blijft.

Een vraag die nu natuurlijk naar boven komt is of CLX voldoende "native" ondersteuning biedt voor zowel Linux als Windows om de toepassingen er toch "echt" uit te laten zien, en niet maar halfbakken. Die laatste angst zal waarschijnlijk eenvoudig de wereld uit geholpen kunnen worden, aangezien met name de visuele componenten uit VisualCLX voor een deel gebaseerd zijn op de Qt bibliotheek van Trolltech die al enkele jaren beschikbaar is als cross-platform omgeving.

Cross-platform Tips

Wie zich nu al wil voorbereiden op het gebruik van Kylix en CLX moet nog even geduld hebben. Aan de andere kant zijn er wel al bepaalde zaken betreffende de Windows en Linux besturingssystemen waar nu al vast rekening mee gehouden kan worden (en dan heb ik het nog niet eens over de componenten die straks onder Linux niet aanwezig zullen zijn). Zo is een van de grootste verschillen tussen Linux en Windows de case-sensitive bestandsnamen. Dat wil zeggen dat "file.txt" en "file.TXT" onder Windows naar hetzelfde wijst, maar onder Linux niet. En ik durf niet met zekerheid te zeggen dat alle toepassingen die ik in het verleden heb geschreven hier goed mee om gaan. En hetzelfde geldt straks als iemand onder Kylix een toepassing maakt die van het bestand "file.txt" een backup maakt in "file.TXT" (wat onder Linux immers een ander bestand is). Als CLX toepassing zal het onder Windows meteen compileren en opnieuw werken, maar een ander (waarschijnlijk ongewenst) resultaat hebben. En als je nu denkt "dit zal mij niet overkomen", denk dan eens aan de namen en de corresponderende bestandnamen van je Units. Als je unit "MyUnit" heet dan zal hij ook echt in "MyUnit.pas" moeten staan, en niet in "myunit.pas", want anders zal de compiler onder Linux hem niet vinden. Tja, en voldoen dan al je oude Delphi projecten nog aan deze regels? Ik zal er toch even extra naar moeten kijken (gelukkig is dit laatste probleem iets makkelijker te constateren; je programma zal immers niet compileren en je precies vertellen welke units er niet gevonden kunnen worden). Uiteraard blijft de ObjectPascal taal zelf onveranderd, en kun je daar rustig een variabele met naam "waarde" declareren en als "Waarde" of zelfs "WaarDe" gebruiken.

Verder zijn er verschillen in het gebruik van drive letters onder Windows (dat kent Linux niet), en wordt de datum en tijd intern op een andere manier opgeslagen. Dit heeft ook gevolgen voor de bestands datum/tijd velden die je bijvoorbeeld met FindFirst en FindNext terug krijgt. Tot slot nog een belangrijk verschil is het feit dat we onder Windows altijd een \ gebruiken (een back-slash), terwijl het onder Linux een / is (een slash). Om hier toch portable

code voor te schrijven is het de bedoeling dat iedereen gebruik gaat maken van de constante "PathSeparator" die backslash zal zijn onder Windows en slash onder Linux. Daarnaast is het aan te raden om nu al gebruik te maken van de IsPathDelimiter, IncludeTrailingBackslash, ExcludeTrailingBackslash en ExtractFilePath uit de SysUtils unit, omdat die onder Linux ook goed werken (maar dan met een slash).

En verder...

Kylix is meer dan alleen maar Delphi voor Linux. Het is een project dat als doel heeft om cross-platform toepassingen te ontwikkelen voor zowel Windows als Linux, en daarmee een hele nieuwe wereld (en markt) aan te boren. Ook de Linux wereld zelf zal profiteren van Kylix - direct of indirect. In directe zin zal dit zijn door duizenden ervaren en minder ervaren Delphi ontwikkelaars die in Kylix een ontwikkelomgeving onder Linux zullen aantreffen waarmee ze snel en efficient native toepassingen voor Linux kunnen maken. Dat doen ze echter niet alleen met Kylix, en dat is het indirecte nut voor de Linux markt: al sinds begin maart 2000 is een Kylix Kickstart initiatief door Borland bezig, met als doel om alle externe partijen die componenten, hulpmiddelen, boeken, cursussen en wat dan ook op het gebied van Delphi (voor Linux) willen bouwen zoveel mogelijk te ondersteunen met informatie en technische support zodat zij hun produkten (vrijwel) klaar kunnen hebben op het moment dat Kylix zelf beschikbaar komt. Op dat moment komen er dus naar verwachting een paar honderd componenten, hulpmiddelen, boeken en andere produkten beschikbaar voor Linux. Wat alleen maar een goede zaak is, of je nu Kylix wilt gebruiken of niet.
Uiteraard kom ik hier uitgebreid op terug zodra Kylix daadwerkelijk beschikbaar is.

Meer Informatie

Voor meer informatie over Kylix heeft Borland een officiële Kylix pagina ingericht te http://www.borland.com/kylix (met artikelen en FAQs over Kylix). Daarnaast loont het altijd de moeite om een kijkje op Dr.Bob's Kylix Kicks te nemen op http://www.drbob42.com/kylix (met het laatste nieuws en een verslag van Kylix presentaties en demonstraties).

Delphi OplossingsConsult

Auteur: **Arnim Mulder**

Vraag: Ik heb een component geschreven, maar wanneer ik deze gebruik zie ik dat mijn component gaat flikkeren.
Antwoord: De property DoubleBuffered (van TWinControl) kan hier uitkomst bieden. Een TRUE value van dit property zorgt ervoor dat er niet direct op het scherm wordt getekend, maar eerst in het geheugen.

Vraag: Ik wil vanuit mijn applicatie graag geluidsbestanden afspelen, maar deze niet los mee distribueren. Is dit mogelijk?
Antwoord: Ja, je moet de geluidsbestanden dan als resource meecompileren in je Delphi project. Hiervoor moet je eerst een bestand aanmaken met de .rc extensie. Hierin geef je aan welke bestanden je mee wilt compileren en van welk type ze zijn.
De syntax van een regel in het rc bestand is "NAAMINTERN TYPE BESTANDSNAAM (bijv. "PING WAVE c:\ping.wav"). Hierna moet het rc bestand worden gecompileerd met BRCC32.exe (uit de Delphi\Bin directory). Er wordt dan een nieuw bestand aangemaakt met dezelfde naam, maar dan met de .res extensie. Nu is het zaak om het gecompileerde resource-bestand mee te compileren in het Delphi project. Hiervoor moet je {$R BESTAND.RES} toevoegen aan je project. Om dan nog het geluidsbestand af te spelen kan de volgende (voorbeeld)code gebruikt worden:

```
uses MMSystem;

procedure TForm1.BClick(Sender: TObject);
var
  FindHandle, ResHandle: THandle;
  ResPtr: Pointer;
begin
 FindHandle :=
  FindResource(HInstance, 'PING', 'WAVE');
 if FindHandle<>0 then
 begin
  ResHandle :=
   LoadResource(HInstance, FindHandle);
  if ResHandle<>0 then
  begin
   ResPtr:=LockResource(ResHandle);
   if ResPtr<>Nil then
    SndPlaySound(PChar(ResPtr),
      snd_ASync or snd_Memory);
   UnlockResource(ResHandle)
  end;
  FreeResource(FindHandle)
 end
end;
```

Book Review: XML Pocket Reference

Author: Robert Eckstein
Publisher: O'Reilly
ISBN: 1-56592-709-5
Price: US$ 8.95

This 107-page book is the size of two floppy disks. And it will probably only take you an hour or two to read it from cover to cover, but the information in it is densely packed, and selected for usefulness. No unnecessary introduction parts, no jokes, no stories, just a plain down-to-earth reference of XML.
The book consists of five sections (they are too small to call chapters or even parts). The first one lists XML terminology (the syntax), including a list of "bad" habits (left over from the HTML age) like the use of quotation marks for attribute values. I missed the fact that XML tags must be lower case now, but I guess that's more a rule than a bad habit (too bad I'm used to write tags in upper case myself). This first section of the book also contains a short overview of an XML document, a Document Type Definition (DTD) and a simple XSL stylesheet.
By the time you've finished with that one, and start with the second section of the book, you're still only at page 14. The XML Reference starts with the definition of well-formed XML (this is the first time I see a few parts of sentences that have been mentioned before in the "bad habits" section), followed by the XML Instruction reference (4 pages), Element and Attribute Rules, a list of XML Reserved Attributes and finally the Entity References. All very concise, but still useful.

See also: http://www.amazon.com/exec/obidos/ASIN/1565927095/drbobsdelphiclin

On page 22, the Document Type Definitions (DTD) section starts. This, too, is only 15 pages in length, but covers Element Declarations, Entities, Attribute Declarations and Subsets. By the time you've finished reading the book so far (page 38), you should be able to write your first XML documents - with or without a required DTD - and know what you're talking about.
The fourth section of the book is a bit more advanced, and takes a whopping 34 pages to cover the Extensible Stylesheet Language (XSL). In this section you read about Formatting Objects and General Formatting, followed by two relative large sections (more than 10 pages each) on Pattern Matching and XSL Elements. Unfortunately, this part of the book was written about a year ago, and is partly outdated by now (even at the time the author wrote it, the XSL specification was moving, so he suggests the homepage for the W3C XSL working group for the most up-to-date reference on this topic.
The final section of the book - again over 30 pages - is about XLink and XPointer. I must admit that when I started to read that section, I had no idea what the use of XLink and XPointer could be for the average developer. Both fall under the Extensible Linking Language (ELL), which is a subset of XML that specifically works with XML links. So, using XLink and XPointer you can define relationships between XML documents. This was an interesting section to read, although it ended rather abruptly - leaving only a 9 page index to conclude the book. Although no effort was wasted to make this book as condense as possible, a final word or list of resources in the end would have been nice.

Having finished the book, I can hardly believe it was only 107 small pages in size. And the typeface isn't even that tiny to begin with (the same size as the font on my 17" monitor). But due to the fact that I know it contains a lot of information (that I don't all know or have in my head), and mainly due to its handy size, this book is one that I carry around with me a lot - and I have been doing so since I got it, which was almost three months ago by the time you read this. Add the absolute low price to this information, and I promise you this is a book you won't be sorry or replace soon. Until O'Reilly publishes the second edition, that is...

WAPPEN met Delphi (1)

Auteur: **Arnim Mulder**

Wie in Nederland heeft er tegenwoordig nog geen mobieltje? Naast het bellen kun je met je mobieltje ook SMS berichten ontvangen en versturen. Helaas is SMS beperkt tot eenrichtingsverkeer door het verzenden of ontvangen van één tekstbericht. WAP is tweerichtingsverkeer en biedt mogelijkheden om met je mobiele telefoon (beperkt) te internetten. Zelfs graphics zijn mogelijk (al moet je hier niet al te veel van voorstellen op zo'n klein display)! Om te kunnen WAPpen moet je naast een WAP-toestel ook een WAP-Operator (ISP voor WAP) hebben. Een mobiele telefoon heeft over het algemeen niet veel intern geheugen (enkele kilobytes), dus de grootte van een wapsite moet minimaal blijven.

WAP staat voor Wireless Application Protocol en werkt via het HTTP protocol. Dus een bestaande webserver kan zonder problemen met WAP pagina's overweg... als je tenminste de webserver informeert welke bestands-extensies met WAP te maken hebben. Er dienen twee mime-typen toegevoegd te worden: "text/vnd.wap.wml" voor de extensie ".wml" en "image/vnd.wap.wbmp" voor de extensie ".wbmp" (bitmaps voor een wap telefoon). Het toevoegen van deze instellingen verschilt per server.

Als dan eenmaal een wapsite is ontwikkeld moet het nog getest worden. Het zou te kostbaar zijn om iedere keer wanneer we onze WAP site willen testen een verbinding moeten leggen via onze mobiele telefoon. Om dan toch WAP pagina's te kunnen bezoeken gebruiken we een WAP Emulator. De Nokia WAP Toolkit is een hele goede. Volgens de website is dit een trial versie, maar bij mij werkt het nog steeds, ondanks dat ik al lang over de 30 dagen trial periode heen ben. Het voordeel van de Nokia WAP Toolkit is dat gemakkelijk bestaande graphics kunnen worden geconverteerd naar WBMP, het graphics-formaat voor WAP. Een andere emulator is UpDev Phone (te downloaden van updev.phone.com) en is geen trial versie.

Een gewone website op het internet bestaat uit HTML pagina's. WAP maakt daarentegen gebruik van WML, een uitgeklede vorm van HTML. Graphics zoals JPG en BMP's kunnen niet worden getoond op een WAP telefoon. Deze bestanden moeten eerst worden geconverteerd naar WBMP, een speciaal voor WAP aanwezig bestandsformaat. Verder is WML beperkt omdat op een klein display nu eenmaal niet veel informatie getoond kan worden.

Tot zover de theorie. In de volgende Delphi OplossingsCourant vervolg ik mijn verhaal door het maken van een echte WAP applicatie in Delphi.

Dr.Bob's Delphi Clinic

Bob Swart organiseert begin 2001 weer **Dr.Bob's Delphi Clinic**, een curriculum voor Delphi programmeurs die over (gaan) stappen naar de laatste versie van Delphi (onder Windows of Linux), en specifieke interesse hebben in *advanced technologies* zoals Intelligente Internet Oplossingen.
De dagen hebben ieder een eigen thema. Alhoewel elke trainingsdag zonder probleem afzonderlijk te volgen is, vormen zij samen een logisch opvolgend geheel. De onderwerpen van 2001 zijn als volgt:

19 januari 2001: Delphi Intranet Development (ActiveX)

Keywords: HTML, Interfaces, Type Library, ActiveX, ActiveForms, Active Server Pages/Objects
Op deze dag zullen we met alle technieken werken die met name ingezet kunnen worden bij intranet development. We beginnen met een korte introductie in Delphi (COM) Interfaces. Direkt daarna gaan we met ActiveX aan de slag, in de vorm van ActiveForms (client side) en Active Server Object (server side). In beide gevallen zullen we met interfaces en de Type Library gaan werken.

9 februari 2001: Delphi Internet Development (e-commerce)

Keywords: HTML, CGI, WinCGI, ISAPI, WebBroker, InternetExpress, XML
Deze dag kan gezien worden als vervolg op de Intranet Clinic van 19 januari. Echter, op deze dag besteden we juist aandacht aan technieken voor internet development, dus geen ActiveX meer, maar juist pure server-side ontwikkelingen met WebBroker (CGI en ISAPI), InternetExpress (XML). Gerelateerde e-commerce onderwerpen die aan de orde komen zijn security en gedistribueerde toepassingen (voorloper op de MIDAS Clinic van 2 maart).

2 maart 2001: Delphi MIDAS 3 Development (N-Tier)

Keywords: DataSetProvider, ClientDataSet, XMLBroker, MidasPageProducer, XML
Op deze dag gaan we multi-tier toepassingen ontwikkelen met behulp van Borland's MIDAS (Multi-tIer Distributed Application Services Suite) versie 3 - een grote verbetering t.o.v. voorgaande versies van MIDAS. Zaken als DataSetProvider, ClientDataSet, XMLBroker en verschillende Connection componenten (DCOM, TCP/IP sockets, CORBA, HTTP) komen aan de orde, maar ook interne zaken als XML, het briefcase model, reconcile errors, en de nieuwe stateless remote datamodules.

23 maart 2001: CORBA Programming Techniques (VisiBroker)

Keywords: CORBA, ORB, IDL, Client Stubs, Server Skeletons, IDL-2-PAS, Exceptions
Op deze dag zullen we leren "communiceren" met CORBA (VisiBroker). In Delphi Enterprise met VisiBroker for Delphi zullen we CORBA objecten, clients en servers maken die alle met elkaar communiceren. We beginnen met een Delphi Server met de Type Library (makkelijk) en vervolgen met Delphi Clients middels VisiBroker for Delphi (iets moeilijker). Tevens laten we de nieuwe IDL-2-PAS Wizard zien, en zaken als structured parameters en CORBA exceptions.

De lokatie van de clinic is een cursusruimte bij Landis-IM in Eindhoven waar de clinics op vrijdag worden gegeven van 9.00 tot 17.00 uur. De clinic is praktisch van aard (dus niet alleen maar een slideshow), doch niet hands-on (voor de deelnemers), alhoewel u natuurlijk wel een eigen laptop kunt meenemen en gebruiken gedurende de dag. Tijdens de clinic worden de onderwerpen door de trainer direkt toegepast met Delphi op een Windows NT machine. Door de kleine groep en het "live" karakter van de clinic bestaat hierbij altijd de mogelijkheid tot het samen onderzoeken van zaken die wellicht net buiten het cursusmateriaal vallen. Voor cursisten is de trainer (ook na de clinics) altijd per e-mail te bereiken voor vragen of nadere uitleg!

De prijs per afzonderlijke clinic is **hfl. 895,=** (te betalen aan Bob Swart), inclusief koffie/thee, lunch en (engelstalige) syllabus geschreven door Bob Swart. Indien meer personen van een bedrijf deelnemen aan dezelfde trainingsdag geldt een korting van 100 gulden per persoon.
Voor meer informatie, zie http://www.drbob42.nl/training of stuur een mailtje naar **b.swart@chello.nl**