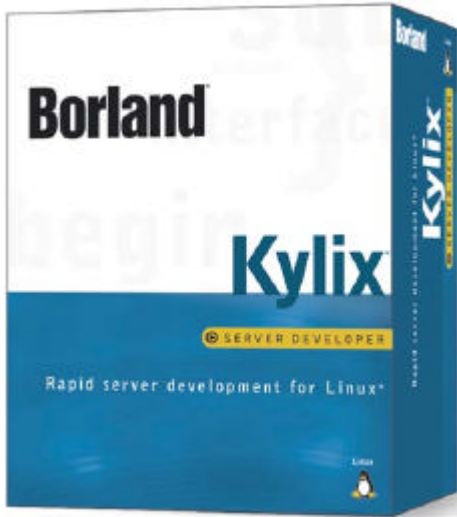


Delphi OplossingsCourant

Vol 3. No. 0. Een speciale uitgave en publicatie van het Delphi OplossingsCentrum- <http://www.drbob42.nl/doc>

Borland Kylix Gelanceerd

Auteur: **Bob Swart**



Laatste Nieuws: Op woensdag 31 januari werd Kylix officieel gelanceerd tijdens de Linux World Expo in New York. Vandaag, woensdag 14 februari (de zesde verjaardag van Delphi) wordt Kylix gelanceerd in Nederland.

Kylix Desktop en Kylix Server

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. Er is een Kylix Desktop Developer en een Kylix Server Developer (met daarin de web server en client/server zaken).

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) zit in Kylix en zal vanaf versie 6 ook in de Professional versies van Delphi en C++Builder zitten.

Kylix zal werken onder de meest recente editie van Linux; dus bijvoorbeeld SuSE, Mandrake of Red Hat 7.0, en zal daarbij dan zowel de GNOME als de KDE desktops ondersteunen. De Kylix demonstraties laten 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).

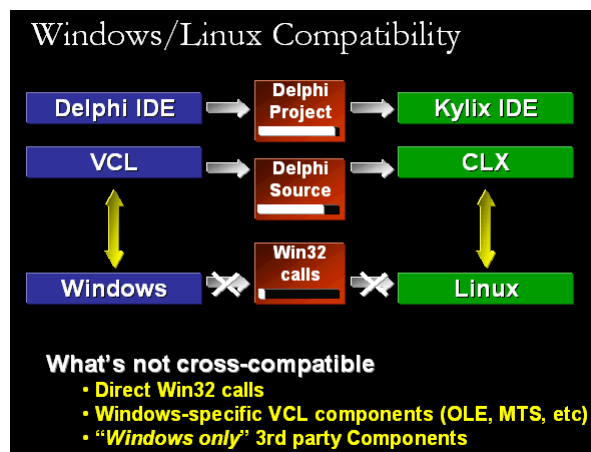
Cross-platform

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?*

Voor 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 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 omgaan. 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 bestandsnamen 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 efficiënt 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, hulp-middelen, 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 producten (vrijwel) klaar kunnen hebben op het moment dat Kylix zelf beschikbaar komt. Op dat moment komen er naar verwachting een paar honderd componenten, boeken en andere producten beschikbaar voor Linux. Wat alleen maar een goede zaak is, of je nu Kylix wilt gebruiken of niet.

Meer Informatie

Voor meer over Kylix heeft Borland een Kylix pagina te <http://www.borland.com/kylix>, en daarnaast loont het altijd de moeite om een kijkje op Dr.Bob's Kylix Kicks te nemen te <http://www.drbob42.com/kylix>.

Dr.Bob's Delphi/Kylix Clinic

Bob Swart organiseert begin 2001 weer **Dr.Bob's Delphi Clinic**, een curriculum voor Delphi programmeurs die over (willen/gaan) stappen naar de laatste versie van Delphi, met op vrijdag 16 maart zelfs een **Kylix Clinic** samen met Micha Somers.

De data/onderwerpen van begin 2001 zijn:

2 maart 2001: MIDAS 3 Development

Keywords: DataSetProvider, ClientDataSet, XMLBroker, MidasPageProducer, XML

Op deze dag gaan we multi-tier toepassingen ontwikkelen met behulp van Borland MIDAS 3. 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.

16 maart 2001: Kylix Programming

Keywords: Kylix, BaseCLX, VisualCLX, DataCLX, dbExpress, NetCLX, Apache

Op deze dag zullen we leren programmeren met Kylix, en zien we de onderdelen van CLX in actie voor desktop, database en web server toepassingen onder Linux.

Als belangrijk onderdeel zullen we ook zien hoe bestaande kennis en toepassingen vanuit Delphi naar Kylix omgezet kunnen worden.

23 maart 2001: CORBA (VisiBroker)

Keywords: CORBA, ORB, IDL, Client Stubs, Server Skeletons, IDL-2-PAS, Exceptions

Op deze dag zullen we leren "communiceren" met CORBA met de nieuwe VisiBroker for Delphi. We zullen CORBA objecten, clients en servers maken die met elkaar communiceren. Tevens laten we de nieuwe IDL-2-PAS Wizard zien, en zaken als structured parameters en CORBA exceptions.

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.

Zie <http://www.drbob42.nl/training> voor meer info of stuur een mailtje naar **b.swart@chello.nl**