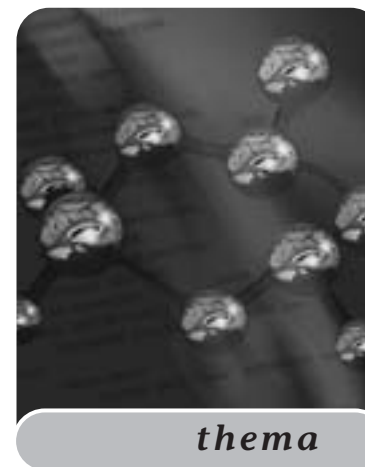


Zo'n 10 jaar geleden leidde ik op een conferentie over COBOL een journaliste rond langs de stands. Overall zag je monitoren met tekstgebaseerde schermindelingen. Plotseling stopte ze, wees naar een scherm en vroeg me wat dat met COBOL te maken had. Het scherm toonde een volledig grafische administratieve toepassing onder Windows. Toen ik haar vertelde dat deze programmatuur ook volledig in COBOL geschreven was, wilde ze me eerst niet geloven. COBOL, dat werd toch alleen nog gebruikt om oude, haast prehistorische applicaties draaiende te houden? Sindsdien is er aan het imago van COBOL niet veel veranderd.



thema

COBOL .NET

Producten voor modernisering COBOL-software

Voor de vaste lezers van Software Release Magazine is .NET een bekende technologie die eigenlijk geen nadere toelichting behoeft. Softwaregigant Microsoft heeft sinds een jaar of vier al haar kaarten wat betreft softwareontwikkeling op het .NET platform gezet en concurreert hiermee rechtstreeks met het J2EE platform. Het meest kenmerkende verschil tussen beide technologieën is dat J2EE je dwingt om kennis te nemen van Java, of je het nu leuk vindt of niet. In deze context is het mooie van .NET dat het taalonafhankelijk is. Microsoft zelf levert onder andere de programmeertalen C#, C++, J# en Visual Basic .NET. De compilers vertalen de code naar een tussencode die officieel Microsoft Intermediate Language (MSIL) heet. Deze code op haar beurt wordt weer uitgevoerd door de Common Language Runtime (CLR) het runtime systeem van het .NET platform. De ontwikkelomgeving voor .NET wordt gevormd door Visual Studio.

BREDE ONDERSTEUNING COBOL heeft waarschijnlijk heel wat meer toelichting nodig. Het is een van de oudste nog in gebruik zijnde programmeertalen, waarvan de oorsprong aan het einde van de jaren vijftig ligt. De opzet van de taal was om in eenvoudig beschrijvend Engels administratieve problemen op te lossen. In de jaren zeventig beleefde de taal haar hoogtepunt. Elke programmeur in opleiding leerde COBOL, geen AMBI-diploma werd uitgereikt zonder dat men examen had afgelegd voor de module T2 (COBOL!) en er kwam geen nieuwe computer op de markt zonder ondersteuning voor deze programmeertaal. Veel programmeertalen zijn gekomen en gegaan sinds COBOL op de markt ver-

scheen, maar COBOL bleef brede ondersteuning genieten. Hier zijn vele redenen voor. De drie voornaamste zijn:

- 1) Vele grote, complexe systemen (bijvoorbeeld die van banken en verzekeringsmaatschappijen) die slechts moeilijk te vervangen zijn, zijn geschreven in COBOL.
- 2) Geen enkele taal is zo zelfdocumenterend en verbaal als COBOL.

Voorbeeld Hallo-programma in COBOL 2002

```
Program-id. Hallo.  
Data Division.  
Working-storage Section.  
77 Toets Pic X.  
Procedure Division.  
    Display "Hallo van COBOL"  
    Accept Toets.  
End Program Hallo.
```

Voorbeeld Hallo-programma in C#

```
using System;  
namespace Hallo  
{  
    class Hallo  
    {  
        [STAThread]  
        static void Main(string[] args)
```

```

    {
        Console.WriteLine("Hallo van C#");
        Console.ReadLine();
    }
}

```

Voorbeeld Hallo-programma in Java

```

package Hallo;
import java.io.*;
public class Hallo
{
    public static void main(String[] args){
        BufferedReader keyboard;
        String line;
        try {
            System.out.print("Hallo van Java\n");
            keyboard = new BufferedReader (new
            InputStreamReader(System.in));
            line = keyboard.readLine();
        }
        catch (IOException e){ }
    }
}

```

FIGUUR 1. Een vergelijking van de codes voor een heel simpel Hallo-programmaatje in COBOL 2002, C# en Java.

Bovendien werd er tijdens de opleidingen ook altijd ingehamerd dat je duidelijk moest programmeren, niet slim (was dat op de C-opleidingen niet juist andersom?). Hierdoor is het heel normaal dat COBOL-systemen 15 tot 20 jaar meegaan. Systemen in andere programmeertalen worden veel sneller afgeschreven!

3) COBOL is compleet. Wat bij vele talen optioneel is of waarbij je aangewezen bent op aanvullende functionaliteit van derden, zit standaard in de taal. Denk bijvoorbeeld aan een geïndexeerd bestandssysteem en geavanceerde schermaansturing.

Ondanks dat COBOL al vele jaren geen populariteitspool meer wint, zijn er wereldwijd nog steeds een tiental leveranciers van deze programmeertaal, waaronder Computer Associates, Hitachi en IBM, toch ook niet de minsten. COBOL anno 2004 lijkt slechts op het eerste gezicht nog op de allereerste ANSI-standaard uit 1968. Eind 2002 is nog een nieuwe ISO-standaard vastgesteld, waardoor in COBOL 2002 objectgeoriënteerd geprogrammeerd kan worden. COBOL van vandaag opent voor oude toepassingen de deur naar moderne en populaire technologie, zoals bijvoorbeeld relationele databases, grafische interfaces inclusief mobiele telefoons en handcomputers, XML, enNET.

FUJITSU SOFTWARE Fujitsu Software Corporation (FSC) was de eerste COBOL-leverancier die aankondigde met deze klassieker onder de programmeertalen het .NET raamwerk te gaan ondersteunen. FSC is een volledige dochter van het Japanse Fujitsu en levert al COBOL compilers sinds het midden van de zeventiger jaren. In eerste instantie alleen op mainframes en in Japan, later ook op Unix systemen en vanaf 1995 ook onder Windows. In dat jaar opent men ook een kantoor in San Jose, vanwaar men de verkoop en ondersteuning in de VS, Canada en West-Europa ter hand nam. Fujitsu heeft al vele jaren een wereldwijde samenwerking met Microsoft. Dit heeft er onder andere toe geleid dat Fujitsu in staat was om in 2000 op Microsoft's Professional Developers Conference, waar .NET werd geannonceerd, een vroeg prototype van haar NetCOBOL onder Visual Studio .NET te laten zien. Inmiddels is NetCOBOL for .NET toe aan de derde versie; na versie 1.0 en 1.1 wordt sinds juli 2003 versie 2.0 geleverd.

MICRO FOCUS Het duurde enige jaren voor de gevestigde COBOL-leveranciers met een antwoord kwamen op de innovatieve stap van Fujitsu. In het rijtje van gevestigde leveranciers hoort zeker Micro Focus thuis. Deze onderneming werd in 1976 opgericht in Engeland en wist in de loop der tijd marktleider te worden in de COBOL-markt voor pc's en Unix-omgevingen. In de zomer van 1998 fuseerde het Engelse Micro Focus met het Amerikaanse Intersolv, hetgeen resulteerde in de nieuwe onderneming Merant. Schijnbaar bracht dit niet wat de partijen er van hoopten, want sinds najaar 2001 staat Micro Focus weer op eigen benen. Sindsdien is er duidelijk sprake van een nieuw elan met meer nadruk op productontwikkeling. In oktober 2003 werd voor het reeds langer bestaande Net Express 4.0 een speciale uitbreiding uitgebracht, genaamd Net Express with .NET.

INSTALLATIE Van de Duitse distributeur van Fujitsu ontving ik een evaluatiesetje met 6 cd's slechts in cellofaan ingepakt. Voordat men met de installatie van NetCOBOL for .NET kan beginnen, moet natuurlijk Microsoft's Visual Studio 2003 al geïnstalleerd zijn; Visual Studio 2002 wordt niet ondersteund!

De installatie van NetCOBOL for .NET valt uiteen in vier stappen: Visual Studio .NET Prerequisites (1 cd), Visual Studio .NET Integrated Development Environment (1 cd), Product Documentation (3 cd's) en NetCOBOL for .NET (1 cd).

De eerste twee stappen garanderen dat Visual Studio over alle vereiste opties en instellingen beschikt. De derde stap installeert de complete Microsoft MSDN Library, voor het geval u dit nog niet gedaan had. Als u uw systeem al compleet hebt ingericht, is eigenlijk



FIGUUR 2. Het installatiescherm van NetCOBOL for .NET met de vier stappen

alleen de vierde en laatste stap vereist, waarvoor u maar 1 cd nodig heeft. Na de installatie heeft u van de leverancier nog een licentiecode nodig om de software te activeren en u kunt in de meeste gevallen van start.

Het was jammer dat de installatieroutine niet 100% voorzag in het correct configureren van het systeem. Zo moest er bij het gebruik van ASP.NET en webservices in de beheeromgeving van IIS (Microsoft's Internet Information Services) een virtuele COBOL website geconfigureerd worden en ook moest er handmatig een aanpassing gemaakt worden in het "machine.config"-bestand in de .NET Framework configuratiemap. Deze laatste stap wordt in de documentatie slechts summier, haast terloops aangegeven. Omdat dit soort aanpassingen geen dagelijkse routinewerkzaamheden zijn, had dit wel wat duidelijker aangegeven mogen worden, of nog beter, automatisch geregeld worden door de installatieroutine!

NET EXPRESS WITH .NET De Micro Focus vestiging in Hoofddorp stuurde voor de evaluatie ook een pakket met zes cd's. Deze zaten in een keurige cassette samen met een beknopte installatiehandleiding en een kaart met productinformatie. Daarnaast wordt een "Getting Started" handboek van 270 pagina's meegeleverd waarin beknopt de meest elementaire zaken omtrent Net Express with .NET uiteengezet. Het geheel zit verpakt in een keurige kartonnen doos.

Ook Net Express with .NET ondersteunt geen Visual Studio 2002. Alhoewel de installatie hier eenvoudiger lijkt en in eerste instantie alleen vraagt om te kiezen uit de installatie van Net Express with .NET met of zonder

Net Express 4.0, heeft de set cd's niet alleen hetzelfde aantal, maar ook dezelfde opbouw als bij NetCOBOL. Ook hier heeft men een activatiecode nodig voordat men met de software aan de slag kan. Vervelender is dat na de installatie, die volgens een melding correct is verlopen, er verschillende problemen optreden (zoals het spontaan afsluiten van de Visual Studio omgeving) die er op wijzen dat er toch iets fout is gegaan. Daarom moest er enkele malen gedeïnstalleerd en opnieuw geïnstalleerd worden. Met de nodige ondersteuning van Micro Focus is dit probleem uiteindelijk opgelost. Wellicht is dit probleem mede veroorzaakt door een conflict met de installatie van een eerdere bètaversie van Net Express with .NET. Volgens Micro Focus verloopt de installatie doorgaans zonder problemen.

DIALECTEN Het zal duidelijk zijn dat twee COBOL-dialecten die grotendeels de gebruikersinterface met Visual Studio moeten delen en die allebei ontworpen zijn voor dezelfde omgeving, veel gemeen hebben.

PROGRAMMEERINTERFACE In de allereerste plaats is er voor de leek nauwelijks verschil in het op deze wijze ontwikkelen in NetCOBOL of in Net Express. De verschillen in de gebruikersinterface beperken zich tot details; zo voegt NetCOBOL aan het 'Tools'-menu een submenu toe genaamd 'NetCOBOL for .NET Utilities', terwijl Net Express hieraan een submenu 'Micro Focus COBOL Tools' toevoegt.

Afgezien van deze kleine nuanceverschillen is de interface exact hetzelfde als van de VB- of C#-programmeur. Dat betekent dat alle technische grappen en grullen die ter beschikking staan van een VB, C++ of C#-programmeur, ook ter beschikking staan van de COBOL-programmeur. De editor geeft bijvoorbeeld met verschillende kleuren aan tot wat voor soort taalelement elk woord in de code behoort, de inhoud van het helpenvenster wordt automatisch aangepast aan de cur-



FIGUUR 3. Visual Studio liep in eerste instantie regelmatig stuk bij het gebruik van Net Express with .NET



FIGUUR 4. Een overzicht van de Visual Studio 2003 interface zoals die er uit zou kunnen zien voor een Net Express COBOL-ontwikkelaar

sorpositie in de code en ook is de debugger geïntegreerd, zodat geconstateerde fouten al tijdens de debugsessie gecorrigeerd kunnen worden. Voor de gemiddelde COBOL-programmeur, zeker degenen die altijd in mainframeomgevingen hebben moeten werken, gaat er een wereld van mogelijkheden open. Een beetje irritant is wel dat de generieke Visual Studio editor geen rekening houdt met de specifieke tabstops die vele COBOL-programmeurs gewend zijn om te gebruiken.

TAALSYNTAX COBOL is altijd een sterk gestandaardiseerde programmeertaal geweest. Dit heeft enorm bijgedragen aan het succes van de taal. Organisaties waar vandaag nog steeds in COBOL geprogrammeerd wordt, zijn gemiddeld vaak al aan hun derde COBOL-dialect toe. Vaak is men ooit begonnen op een mainframe of minicomputeromgeving (bijvoorbeeld S36/38, AS/400, VAX VMS), daarna doorgestroomd naar een Unix-omgeving en tegenwoordig wordt vervolgens vaak gebruikt gemaakt van Windows. Op de oude mainframes en mini's was de leverancier van de COBOL-taal altijd de computerleverancier die hun eigen variant van de taal meeleverden. Onder Unix kwam dat ook nog wel voor, maar vaker kwam de COBOL-omgeving dan van een leverancier als Acucorp (toen Acucobol), Liant (toen Ryan McFarland) of Micro Focus. Het overstappen van de ene leverancier naar de andere gaf altijd wel wat problemen, maar was zelden complex omdat alle COBOL-talen aan dezelfde standaard voldeden, waardoor 90-95% van de code bij alle leveranciers identiek was.

De laatste COBOL-standaard is vrij recent tot stand gekomen, namelijk in december 2002. Voor zover bekend zijn hiervoor nog geen producten gecertificeerd. In het verleden duurde het ook twee à drie jaar voordat

de leveranciers hun producten aangepast hadden en vervolgens de certificatieprocedures hadden doorlopen. Dat zal nu wel niet anders zijn.

De huidige versie 2.0 van NetCOBOL for .NET dateert van 2003 en kan dus ook nog niet voldoen aan de ISO 2002 standaard. Fujitsu stelt zelf in haar documentatie dat NetCOBOL voldoet aan de ANSI 1985 en de ISO 1985-1989 standaard met een aantal optionele functies zoals de Report Writer functie. Daarnaast heeft Fujitsu nog haar eigen set van uitbreidingen toegevoegd. Het gaat te ver om al die uitbreidingen hier te behandelen, maar het is wel grappig dat op die lijst van uitbreidingen onder andere de 'Micro Focus native functions' genoemd wordt.

Ook Net Express with .NET kan nog niet aan de 2002-standaard voldoen en voldoet dan ook aan dezelfde oudere standaarden als NetCOBOL. Wel vermeldt Micro Focus in haar documentatie dat men ook voldoet aan de ANSI 1989/ISO 1993 uitbreiding op de eerdere standaard. Het betreft hier de zogenaamde 'intrinsic functions' een set van standaard functies die vanuit COBOL aangeroepen kunnen worden. (Deze vermelding ontbrak in de NetCOBOL-documentatie en bij het controleren van vijf willekeurige functies bleken deze ook niet bekend te zijn in de programmeerhandleiding.) Ook Net Express with .NET heeft zijn eigen set van uitbreidingen op de standaard. Volgens de documentatie bevinden die zich voornamelijk op het terrein van IBM-uitbreidingen en mainframe-ondersteuning. Dit heeft natuurlijk te maken met het feit dat Micro Focus veel gebruikers telt in de financiële sector die op grote schaal gebruik maken van mainframes.

OBJECTORIËNTATIE Een onderdeel van de nieuwe 2002 standaard is dat COBOL nu gebruikt kan worden om objectgeoriënteerd te ontwikkelen. COBOL 2002 beschikt hiermee over OO-kenmerken als klassen, methoden, constructors, objecten, overerving, polymorfisme en het inkapselen van gegevens. Zowel NetCOBOL for .NET als Net Express with .NET ondersteunen OO, maar nergens wordt aangegeven hoe dicht men de 2002-standaard benaderd. Om hier een idee van te krijgen, is een COBOL 2002 voorbeeldprogramma dat in de standaard genoemd wordt (de bankapplicatie) overgezet naar de twee .NET omgevingen. Het voorbeeldprogramma beslaat maar zo'n 125 regels code.

Hoofdprogramma:

```
PROGRAM-ID. BANKMAIN.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS Account.
DATA DIVISION.
```



```

WORKING-STORAGE SECTION.
01 an-object          USAGE OBJECT
REFERENCE Account.
PROCEDURE DIVISION.
go-now.
    INVOKE Account "newAccount"
RETURNING an-object.
    INVOKE an-object "displayUI".
    SET an-object to NULL.
    GOBACK.
END PROGRAM BANKMAIN.

Klasse:

CLASS-ID. Account INHERITS Base.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
REPOSITORY.
    CLASS Base.
FACTORY.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 number-of-accounts      PIC 9(5)
VALUE ZERO.
PROCEDURE DIVISION.

METHOD-ID. newAccount.
DATA DIVISION.
LOCAL-STORAGE SECTION.
LINKAGE SECTION.
01 an-object          USAGE IS OBJECT
REFERENCE ACTIVE-CLASS.
PROCEDURE DIVISION RETURNING an-
object.
begin-here.
    INVOKE SELF "new" RETURNING an-
object.
    INVOKE an-object
"initializeAccount" USING BY CONTENT number-
of-accounts.
    EXIT METHOD.
END METHOD newAccount.

METHOD-ID. addAccount.
PROCEDURE DIVISION.
method-start.
    ADD 1 TO number-of-accounts.
    EXIT METHOD.
END METHOD addAccount.

METHOD-ID. removeAccount.
PROCEDURE DIVISION.
main-entry.
    SUBTRACT 1 FROM number-of-
accounts.
    EXIT METHOD.
END METHOD removeAccount.
END FACTORY.

OBJECT.
DATA DIVISION.

```

```

WORKING-STORAGE SECTION.
01 account-balance      PIC S9(9)V99.
01 account-number      PIC X(9).
01 the-date            PIC 9(8).
PROCEDURE DIVISION.

METHOD-ID. displayUI.
DATA DIVISION.
LOCAL-STORAGE SECTION.
01 in-data.
    03 action-type      PIC X.
    03 in-amount        PIC
S9(9)V99.
    03 in-wrk          PIC X(12).
PROCEDURE DIVISION.
method-start.
    DISPLAY "Enter D for Deposit, B
for Balance or W for Withdrawal"
ACCEPT in-data
EVALUATE action-type
    WHEN "D"
        PERFORM get-amount
        INVOKE SELF "deposit" USING
in-amount
    WHEN "W"
        PERFORM get-amount
        INVOKE SELF "withdraw"
USING in-amount
    WHEN "B"
        INVOKE SELF "balance"
    WHEN OTHER
        DISPLAY "Enter valid trans-
action type."
    EXIT METHOD
END-EVALUATE
EXIT METHOD
.
get-amount.
    DISPLAY "Enter amount 9(9).99"
ACCEPT in-wrk
COMPUTE in-amount = FUNCTION NUM-
VAL (in-wrk)
.
END METHOD displayUI.

METHOD-ID. balance.
DATA DIVISION.
LOCAL-STORAGE SECTION.
01 display-balance      PIC
$ZZZ,ZZZ,ZZ9.99B-.
PROCEDURE DIVISION.
disp-balance.
    MOVE account-balance to display-
balance
    DISPLAY "Your Account Balance
is:" display-balance
    EXIT METHOD.
END METHOD balance.

METHOD-ID. deposit.
DATA DIVISION.

```

```

LINKAGE SECTION.
01 in-deposit PIC      S9(9)V99.
PROCEDURE DIVISION USING in-deposit.
make-deposit.
    ADD in-deposit TO account-balance
    EXIT METHOD.
END METHOD deposit.

METHOD-ID. withdraw.
DATA DIVISION.
LINKAGE SECTION.
01 in-withdraw          PIC
S9(9)V99.
PROCEDURE DIVISION USING in-withdraw.
withdraw-start.
    IF account-balance >= in-withdraw
        SUBTRACT in-withdraw FROM
account-balance
    ELSE
        DISPLAY "Your Balance is
Inadequate"
    END-IF
    EXIT METHOD.
END METHOD withdraw.

METHOD-ID. initializeAccount.
DATA DIVISION.
LINKAGE SECTION.
01 new-account-number PIC 9(5).
PROCEDURE DIVISION USING new-account-
number.
Begin-initialization.
    MOVE ZERO TO account-balance
    MOVE new-account-number TO
account-number
    MOVE FUNCTION CURRENT-DATE (1: 8)
TO the-date
    EXIT METHOD.
END METHOD initializeAccount.
END OBJECT.
END CLASS Account.

```

FIGUUR 5. Het OO-COBOL-voorbeeldprogramma uit het officiële COBOL 2002 standaardisatiedocument van de ISO

Bij NetCOBOL moeten de meeste coderegels worden aangepast voordat het programma probleemloos door de compiler gaat. De wijzigingen variëren van een klein detail waarbij een klassenaam die in de 2002-standaard "Base" heet en die Fujitsu "FJBase" noemt, tot een structureel verschil wanneer de LOCAL-STORAGE sectie bij NetCOBOL niet blijkt te bestaan. Het is duidelijk dat Net Express dicht tegen de standaard aan zit en daar waarschijnlijk op korte termijn volledig aan zal voldoen.

.NET Het .NET Framework bestaat uit letterlijk duizenden functies die aan te roepen zijn vanuit de verschillende programmeertalen die .NET ondersteunen. Het leuke is dat tussen deze programmeertalen de wijze

van aanroepen in detail verschilt, maar dat de oplossingsstappen in de diverse functies aardig op elkaar lijken. Je kunt hierdoor een Visual Basic en een COBOL-programma naast elkaar leggen en instructie na instructie met elkaar vergelijken. Tussen de twee .NET COBOL-implementaties zijn er slechts de minieme syntax-verschillen, want de .NET-functies zijn gelijk.

VOOR- EN NADELEN: NETCOBOL Natuurlijk hebben beide .NET COBOL-producten hun sterke en zwakke punten. Om met NetCOBOL te beginnen, het grootste voordeel van dit product is dat het het langste op de .NET markt aanwezig is en dat Fujitsu al vroeg een samenwerking met Microsoft had. Het gevolg is dat er behoorlijk wat publicaties over NetCOBOL for .NET te vinden zijn. Het lijkt ook wat meer toegesneden op wat gebruikelijk is in de Visual Studio .NET wereld. Zo kun je met NetCOBOL binnen Visual Studio gebruik maken van de 'Forms Painter', terwijl dit met Net Express niet kan. De compatibiliteit met andere COBOL-dialecten is bij NetCOBOL zeer beperkt. Bij het overstappen vanuit een andere COBOL-omgeving dient u dan ook rekening te houden met ingrijpende codeaanpassingen. NetCOBOL for .NET lijkt zich meer te richten op het ontwikkelen van nieuwe applicaties, dan op het moderniseren van reeds bestaande. Als je net kennismak met een vreemde ontwikkelomgeving, dan knallen regelmatig de compilerfoutmeldingen om je oren. Het is dan erg vervelend als Fujitsu je nergens die foutmelding terug laat vinden met een duidelijke verklaring wat nu de oorzaak is.

Er is – voor zover bekend – geen Nederlandse distributeur van NetCOBOL; de dichtstbijzijnde zit in Duitsland. Ondersteuning is dan ook wat lastig te krijgen, te meer daar e-mailberichten door Fujitsu standaard niet beantwoord schijnen te worden. Wel kan met een ondersteuningscontract toegang verkregen worden tot een afgeschermd gedeelte van de website, speciaal voor technische ondersteuning. Op deze wijze konden we een vervelend probleem zelf oplossen. De NetCOBOL for .NET prijsstelling ligt op het niveau van Visual Studio of wat daarboven. Hierdoor bevindt deze COBOL-omgeving zich binnen het bereik van kleinere ontwikkelaars.

VOOR- EN NADELEN: NET EXPRESS WITH .NET Net Express with .NET 4.0 is de allereerste uitgave voor het .NET platform van de al veel langer bekende Net Express COBOL-omgeving. De 'Form Painters' ontbreken inderdaad, maar de rest is af. Net Express biedt een brede compatibiliteit ten aanzien van COBOL-dialecten en voldoet ook bijna aan de COBOL 2002 standaard. Slordig is wel dat oude tekstgebaseerde COBOL-instructies voor scherm aansturing probleemloos door de compiler verwerkt worden, maar dat ze vervolgens niet cor-

NetCOBOL en Net Express with .NET getest met COBOL-programma in Unix-omgeving

Om een indruk te krijgen of de nieuwe .NET COBOL-producten nog steeds zonder problemen oude programmatuur kunnen verwerken, is een gedateerd COBOL-programma van circa 700 regels genomen, afkomstig uit een Unix-omgeving van vijftien jaar geleden. Voor de kenners, het maakt gebruik van de zogenaamde RM/COBOL-85 syntax. Het doel van het programma is om een indruk te krijgen van de snelheid van het bestandssysteem dat gebruikt wordt. Hiertoe toont het eerst een verklarende tekst op het scherm, wacht tot er een toets wordt ingedrukt en leest en schrijft in enkele stappen vervolgens enkele duizenden records waarbij per stap informatie naar het scherm geschreven wordt. Aan het eind van de verwerking wacht het programma op het indrukken van een toets. Kortom, een vrij simpel programma, dat erg ver van de complexe praktijk met ingewikkelde orderprocedures en rekening-courant boekingen afstaat. Des te groter is dus de verbazing bij de constatering dat de twee .NET producten het niet mogelijk maken om met wat configuratie-instellingen deze programma's te vertalen en onder Windows te laten lopen.

De NetCOBOL-compiler struikelt al aan het begin van het programma over de SELECT-instructies waarmee de bestanden gedefinieerd worden. Hier één of twee woordjes weghalen is vaak alles wat nodig is om dit toch correct te vertalen. In de "working-storage section", het deel van het programma waar je het werkgeheugen van het programma definieert, doen zich ook enkele problemen voor; twee zogenaamde 77-level velden mogen niet gedefinieerd worden (opgelost door er maar een 01-level van te maken) en een indexveld met de definitie "PIC 9(4) COMP-1" wordt in eerste instantie niet vertaald omdat de picture-clausule hier niet toegestaan zou zijn, maar later in de procedure divisie levert het veldtype weer een probleem op omdat het niet toegestaan zou zijn als index voor een tabel. Dit probleem is opgelost door van dit veld maar een standaard numeriek type te maken en door met de INDEXED BY-clausule

een index te benoemen in de betreffende tabel. Vervolgens leveren de vele ACCEPT en DISPLAY-instructies hopen problemen op doordat de RM-uitbreidingen op de standaard niet geaccepteerd worden. Hieronder vallen dus onder andere LINE, POSITION, HIGH en REVERSE. Om dit programma toch te kunnen vertalen zijn deze opties op de ACCEPT en DISPLAY maar weggehaald, waardoor het programma nu in een soort ouderwetse tty-modus gaat werken.

De verwachtingen ten aanzien van Net Express with .NET waren veel hoger gespannen omdat Micro Focus een ruime ervaring heeft in compatibiliteit met andere producten. Zo kenden de oudere producten een configuratie-instelling om de compiler RM/COBOL compatibel te maken en ook het nieuwste .NET product blijkt zo'n instelling te hebben. Bij het vertalen struikelt de compiler slechts over één woordje in de definitie van een sorteerbestand en ook de working-storage sectie en de procedure divisie worden zonder problemen vertaald. Helaas gaat het hier bij de uitvoering van het programma mis; het programma werkt wel, maar er is helemaal niets zichtbaar op het scherm en ook de toetsenbord invoer werkt niet. Navraag bij Micro Focus levert na wat interne contacten de informatie op dat .NET Framework 1.1 de benodigde functionaliteit mist in de aansturing van het zogenaamde consolevenster. Dit betekent dat bepaalde functionaliteit, die wat betreft syntax correct door de compiler en debugger verwerkt worden, dus nooit correct zal werken. Informatie hierover stond in het "Start Here!" document dat normaal gesproken na installatie geraadpleegd wordt. Maar deze informatie stond niet in de reguliere helpteksten die in Visual Studio geraadpleegd worden. Het zou prettig zijn als de compiler op zijn minst een waarschuwing zou genereren als dit soort functionaliteit gebruikt wordt. De ontwikkelaar die zich een maand na installatie nog weet te herinneren welke beperkingen in dit 'Start Here'-document staan, verdient met Koninginnedag een lintje!

rect werken. De documentatie biedt hier wel vaak een aanknopingspunt voor een foutmelding van de compiler, maar het lijkt net alsof de nieuwste meldingen ontbreken. Vaak vind je ze wel, soms niet. Net Express blijkt met de standaard instellingen stukken sneller en efficiënter te werken dan NetCOBOL met de standaardinstellingen. Ons oude RM/COBOL-programma (zie kader) draaide onder Net Express maar liefst drie keer zo snel als onder NetCOBOL, terwijl het objectprogramma 44% kleiner was. (Beide producten bieden diverse optimalisatiemogelijkheden, maar die hebben we niet getest!) Micro Focus heeft in Europa diverse lokale vestigingen, waaronder in Nederland. De website biedt standaard weinig ondersteuning, maar met een gebruikerscode en het juiste wachtwoord, beiden te verkrijgen via een onderhoudscontract, gaat er een wereld van

informatie voor u open. U kunt dan bijvoorbeeld ook ondersteuning per e-mail krijgen. Dat is maar goed ook, want er is maar weinig lectuur in de handel verkrijgbaar speciaal voor Net Express with .NET. Praktisch alle informatie zult u hier dus bij de leverancier moeten halen. Dit heeft natuurlijk zijn prijs. Micro Focus richt zich op grotere kapitaalkrachtige organisaties die bereid zijn te betalen voor kwaliteit. Maar ja, niet iedereen kan zich een Mercedes veroorloven.

TOEKOMSTIGE ONTWIKKELINGEN Beide leveranciers zijn nauw betrokken bij het vaststellen van de COBOL-standaard. Ze hebben allebei hun permanente vertegenwoordigers in de standaardisatiecommissie. Het kan dan ook niet anders als dat Fujitsu en Micro Focus op korte termijn hun producten COBOL 2002

compatible zullen maken. Beide leveranciers zullen ook zeker Microsoft volgen in haar plannen met Visual Studio 2005 en Whidbey. Micro Focus werkt achter de schermen hard aan een Forms Painter, omdat alle andere programmeertalen die beschikbaar zijn onder .NET, inclusief NetCOBOL, daarmee inmiddels zijn toegerust.

CONCLUSIES Het ontwikkelen in COBOL is met de komst van NetCOBOL for .NET en Net Express with .NET een stuk leuker geworden. Met zo'n ontwikkelomgeving hoef je je niet meer te schamen als je op een verjaardag vertelt dat je COBOL-programmeur bent. Alles wat de VB en C# freaks kunnen, kun jij nu ook. Maar wat betekenen deze producten nu concreet voor een organisatie? Voor organisaties is het een sterk punt dat ze hun bestaande COBOL-software stapje voor stapje kunnen moderniseren en voor de gebruikers langzaam de deur naar Windows, .NET en het web kunnen openen. We noemen dit de evolutionaire aanpak. Voordelen van deze aanpak zijn het behoud van bestaande investeringen en vooruitgang door kleine stapjes met lage risico's en met relatief geringe investeringen. Voor organisaties die momenteel geen COBOL in huis hebben, is er geen groot voordeel te behalen, en dat is jammer. Natuurlijk kun je voor .NET prima in COBOL ontwikkelen, maar dat kun je ook in alle andere programmeertalen die .NET ondersteuning bieden. De COBOL-taal is over het algemeen beter leesbaar en zelfdocumenterend als andere programmeertalen, maar probeer tussen al die .NET-ontwikkelaars maar eens een

COBOL-programmeur te vinden. Twintig jaar geleden was dat geen probleem. Je was als automatiseerder in opleiding haast verplicht om COBOL te leren, maar tegenwoordig worden er in Nederland nog nauwelijks mensen opgeleid met deze taal. De COBOL .NET producten kunnen dus heel interessant zijn voor een kleinere groep van organisaties die al grote investeringen gedaan hebben in COBOL en deze investeringen verder willen uitbuiten door deze te integreren met nieuwe technologie. Maar nieuwe toepassingen en nieuwe gebruikers zullen voor deze producten nauwelijks te vinden zijn, omdat de COBOL .NET producten te weinig voor hebben op de alternatieven. Terwijl de expertise voor die alternatieven voldoende aanwezig is, vergrijsst die voor COBOL in snel tempo.

Bronvermeldingen

Website Fujitsu Software - NetCOBOL: www.netcobol.com
Website Micro Focus: www.microfocus.com
COBOL 2002 Standard, uitgegeven door ISO december 2002
COBOL and Visual Basic on .NET, door Chris Richardson
Microsoft .NET for COBOL Programmers, door Howard E. Hinman

*René van Oevelen, freelance COBOL consultant
(e-mail: rene@oevelen.com)*

PATCHES

Patches

PATCHES

Patches

PATCHES

Patches

PATCHES

Magic Hands maakt met Mag4 conversie mogelijk naar 4GL

Magic Hands biedt in samenwerking met partners een oplossing voor de conversie van RPG en Cobol applicaties naar Magic applicaties. Met deze oplossing kunnen ondernemingen gemakkelijk de overstap maken van een 3GL naar een moderne kostenbesparende post 4GL, zodat direct van de allernieuwste technologieën gebruik gemaakt kan worden. Alle functionaliteit blijft behouden waardoor de opgebouwde business intelligentie en gedane investeringen niet verloren gaan. Magic Hands heeft al een eerste pilot gedraaid waarbij een COBOL mainframe-applica-

tie succesvol "vertaald" is in een Magic-applicatie die geschikt is voor de IBM eServer i5.

Magic Hands is een Magic Consulting & Solutions partner, die zich ten doel stelt haar opdrachtgevers te ondersteunen met kennis en diensten tijdens de gehele levenscyclus van hun applicaties. Om dit te bewerkstelligen biedt Magic Hands Support Level Agreements aan waarin alle diensten, voor de ontwikkeling en het beheer van op Magic Software gebaseerde applicaties, ondergebracht kunnen worden. Tevens biedt Magic Hands detachering en projectdiensten. Hierbij worden binnen de projectmanagement diensten de Prince II en DSDM methodieken

gehanteerd. Magic Hands werkt uitsluitend met door Magic Software Enterprises gecertificeerde Magicians, dit garandeert een optimale kennis van alle Magic producten. Magic Hands maakt onderdeel uit van de Hands Group. Meer informatie is te vinden op www.hands-group.com

Samenwerking Oracle en Microsoft

Oracle is Premier-level partner geworden in Microsoft's Visual Studio Industry Partner programma. Door aan het VSIP programma deel te nemen continueert Oracle haar commitment aan software ontwikkelaars. Door betere integratie van

de Oracle Database met Microsoft Visual Studio .NET 2003 kunnen ontwikkelaars hun productiviteit verhogen en eenvoudiger op Oracle gebaseerde applicaties op het Microsoft Windows platform inzetten. Oracle's deelname aan het VSIP programma onderstreept het toenemende belang van samenwerking op het gebied van Microsoft .NET technologie. Momenteel nemen ruim 200 partijen deel en zijn er meer dan 400 beschikbare add-on tools voor de Visual Studio .NET IDE. Oracle zal binnenkort een download voor de integratie met Visual Studio .NET 2003 beschikbaar stellen op de OTN-website <http://otn.oracle.com>.