

Visual Basic .NET 2005

NOG HOGERE PRODUCTIVITEIT EN NIEUWE FEATURES OP GEBIED VAN TAAL

Visual Basic .NET 2005 bevat nieuwe features in de taal en op het gebied van productiviteit voor ontwikkelaars. Het VB-team in Redmond heeft zichzelf voorgenomen om bij te blijven met ontwikkelingen in C# en voorop te lopen op het gebied van productiviteit. Generics, operator overloading en XML-commentaar getuigen van deze strategie. Productiviteit is nog steeds het paradepaardje van VB. Het is op dit gebied dat versie 2005 grote vooruitgang boekt. Met de nieuwe My namespace kan veelgebruikte broncode, zoals je die in versie 2003 nodig had, worden gereduceerd. Bovendien keert edit & continue terug. Dit nieuws werd op de Professional Developer Conference met luid applaus verwelkomd.

In tabel 1 staat een overzicht van de nieuwe mogelijkheden van Visual Basic .NET 2005. De vetgedrukte features in de tabel komen hieronder aan bod, waarbij de meeste aandacht uitgaat naar de productiviteitsverbeteringen.

Productiviteitsverbeteringen

Om toegang te krijgen tot beschikbare namespaces in VB is er een nieuw object beschikbaar: Global. Global.System ontsluit de .NET Framework namespaces. Door gebruik te maken van dit object worden de project classes zichtbaar. Via Global.Microsoft wordt toegang verkregen tot VisualBasic, C#, Win32 en Internal namespaces. Global.Microsoft.VisualBasic legt Microsoft.VisualBasic.dll bloot. Onderdeel van Microsoft.VisualBasic is het nieuwe MyServices, voor VB-developers benaderbaar met My. My groepeer vaakgebruikte code. Zo biedt My toegang tot onder andere de registry, settings en resources. De vele resultaten van My zijn ook bereikbaar via de System namespaces en Win32-calls. Met

My heb je echter minder regels code nodig. Bovendien is My vaak intuïtiever in gebruik. In tabel 2 staat een overzicht.

MyServices
Application
Computer
User
Forms
Web Services
Settings
Resources

Tabel 2. MyServices

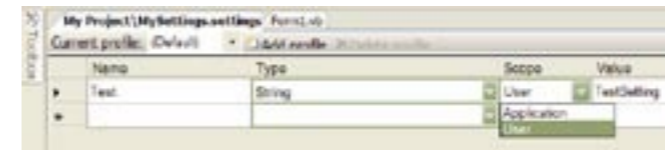
Hierna komen de belangrijke aspecten aan de orde van MyServices.

My.Application

My.Application verschaft toegang tot het clickonce deployment-

Productiviteit	Taal
Microsoft.VisualBasic.dll	Fundamenteel
Global	Generics
MyServices	Partial classes
Worker object; asynchroon programmeren	Operator overloading; declareer en gebruik operators voor je eigen classes, bijvoorbeeld & om een bepaalde waarde van twee classes te samenvoegen
	Custom events; meer controle over hoe een call-back wordt uitgevoerd, bijvoorbeeld asynchroon
Ontwikkelomgeving	Keywords
Code snippets	Using
Edit & continue	Continue; springen naar de volgende iteratie in een loop
Auto correction	IsNot; IsNot Nothing in plaats van Not Is Nothing
Exception helper	
Intellisense filtering; slimmere intellisense	
Rename symbol; rechtermuisklik op object biedt hernoemfaciliteit van dit object, werkt beter dan 'replace'	
Overig	Overig
Data source configuration wizard	Unsigned types; types die alleen positieve waardes kunnen bevatten, nodig voor sommige API-calls.
XML code comments; standaard toelichting op methoden en classes. Voor genereren documenten en intellisense	
Nieuwe Windows forms features; betere uitlijning, 'in place property editing', nieuwe controls	

Tabel 1. Overzicht nieuwe mogelijkheden VB.NET 2005



Afbeelding 1. Settings-editor.

object. Hiermee kun je controleren of nieuwe updates beschikbaar zijn en, als dat het geval is, kun je de opdracht geven deze updates op te halen en te installeren, synchroon dan wel asynchroon. Via de application namespace is het eenvoudig om commandline-argumenten, waarmee de applicatie is opgestart, uit te lezen. Het wegschrijven van log-informatie is nog nooit zo makkelijk geweest. De mogelijkheid om het gehele exception-object inclusief stack trace weg te schrijven in de log werkt prima; zie codevoorbeeld 1. In App.Config kun je aangeven hoe de log wordt weggeschreven. Mogelijkheden zijn Filelog, Eventlog, Console en XML. Naast methoden en properties ondersteunt My.Application events. Er zijn events die afgaan bij het starten en stoppen van de applicatie. Er is een ander event dat afgaat als de executable al is opgestart en nog een keer wordt geactiveerd. Het is mogelijk om events af te vangen als een fout optreedt die nog niet op een andere plek is afgevangen. Maar met stip op nummer één staat wat mij betreft My.Application.NetworkAvailabilityChanged. Dit event gaat af als de status van de netwerkverbinding verandert. Dit lijkt vanzelfsprekend, maar probeer het maar eens met de bestaande versies van VB of met C#.

My.Computer

My.Computer is de meest uitgebreide van de My namespaces. Deze namespace verstrekt toegang tot onder andere audio en het clipboard. Via My.Computer.Info is het eenvoudig om te kijken hoeveel geheugen de computer heeft of hoeveel nog beschikbaar is. Ook is het mogelijk informatie te krijgen over onder andere de naam van de machine, de versie van het OS en cultuurinstellingen zoals het datum/tijd-formaat. Letterlijk 'information at your fingertips!' Ieder-

```
Try
    'Wegschrijven informatie in log
    My.Application.Log.WriteEntry("Application is doing well", _
        TraceEventType.Information)
    'Afsluiten applicatie
Catch ex As Exception
    My.Application.Log.WriteException(ex, True)
Finally
    My.Application.Exit()
End Try
```

Codevoorbeeld 1.

Wegschrijven gehele exception-object inclusief stack trace in de log

```
'File systeem
appDataDir = My.Computer.FileSystem.SpecialDirectories.CurrentUserApplicationData
If Not My.Computer.FileSystem.DirectoryExists(appDataDir & "\Demo") Then
    My.Computer.FileSystem.CreateDirectory(appDataDir & "\Demo")
End If
appData = My.Computer.FileSystem.ReadAllText(appDataDir & "\Demo\" & _
    "appdata.xml")
'Netwerk
If My.Computer.Network.IsAvailable Then
    If My.Computer.Network.Ping("www.qliner.com") Then
        MessageBox.Show("Website qliner.com is available.")
    End If
End If
'Printer
My.Computer.Printers.DefaultPrinter.WriteTextFile(appDataDir & _
    "\Demo\" & "appdata.xml")
```

Codevoorbeeld 2.

Mogelijkheden van My.Computer



Afbeelding 2. Code snippets.

een die heeft gewerkt met System.IO weet hoe lastig het kan zijn om te werken met Streams. Met My.Computer.FileSystem bespaar je niet alleen regels code, maar het gebruik ervan is ook makkelijk. Met My.Computer.FileSystem kun je bijvoorbeeld de inhoud van een folder kopiëren en als je dat wilt kun je hierbij een voortgangsbalk laten tonen. Via de Computer namespace krijg je toegang tot het toetsenbord, de muis, het netwerk, de seriële poorten, de printer en het scherm. Uitvragen of het netwerk beschikbaar is, pingen, files downloaden en uploaden is allemaal mogelijk: vraag het My.Computer.Network. Communiceren met seriële poorten is een complexe aanpak, met My.Computer.Ports niet meer. Met My.Computer.Printers kun je bestanden printen, maar ook eigen rapporten maken en printen. Tot slot ligt met My.Computer de registry aan je voeten; zie codevoorbeeld 2 voor de mogelijkheden van My.Computer.

My.User

De mogelijkheden van de user namespace zijn beperkt. Ik heb het vermoeden dat Microsoft hier nog aan gaat werken. Je kunt met de user namespace bekijken welke gebruiker is aangemeld en tot welke groep deze behoort; zie codevoorbeeld 3.

Forms, WebServices, Settings, Resources

De My.Forms, My.WebServices, My.Settings en My.Resources namespaces hebben met elkaar gemeen dat ze automatisch, design-time, worden gevuld. Als een applicatie geen Forms bevat dan is My.Forms leeg. Als een applicatie Form1 bevat kun je deze via My.Forms benaderen zonder dat je dat formulier expliciet hoeft te creëren; precies zoals je dat van Visual Basic 6 gewend was. Hetzelfde geldt voor Webservices, als webservices. Als er een Web-service-referentie is gelegd, dan biedt de My namespace strongly typed toegang; zie codevoorbeeld 4. De My namespace biedt 'static' toegang tot veelgebruikte classes. Het voordeel is dat het niet nodig is om steeds onnodig te declareren. Settings kun je opslaan in config-files. VB biedt een settings-editor om deze gegevens te manipuleren; zie afbeelding 1. Met de scope-instelling uit het voorbeeld is het mogelijk te bepalen of deze setting moet worden bewaard voor iedereen, of slechts voor de actieve gebruiker. Settings met de gebruikersscope kunnen niet alleen strongly typed worden uitgelezen met de My.Settings namespace, maar ze kunnen ook runtime worden gewijzigd en opgeslagen; zie codevoorbeeld 5.

```
If My.User.IsInRole("Users") Then
    userName = My.User.Identity.Name
End If
```

Codevoorbeeld 3.

Welke gebruiker is aangemeld?

```
My.Forms.Form1.Show()
My.WebServices.TestWebService.TestMethod()
```

Codevoorbeeld 4.

Strongly typed toegang

```
My.Settings.Test = "edwinj@avanade.com"
My.Settings.Save()
```

Codevoorbeeld 5.

Settings runtime wijzigen en opslaan

```
Dim generator As New Random
Dim randomValue As Integer
' Generates numbers between 1 and 5, inclusive.
randomValue = generator.Next(1, 6)
```

Codevoorbeeld 6.
Markeren van te vervangen delen

```
Dim employees As New NorthwindDataSet.EmployeesDataTable
Dim employeePump As New EmployeesTableAdapter

employeePump.Fill(employees)
For Each employeeRecord As NorthwindDataSet.EmployeesRow In employees
    Console.WriteLine(employeeRecord.FirstName)
Next
```

Codevoorbeeld 7.
Koppeling met Northwind Employee-tabel

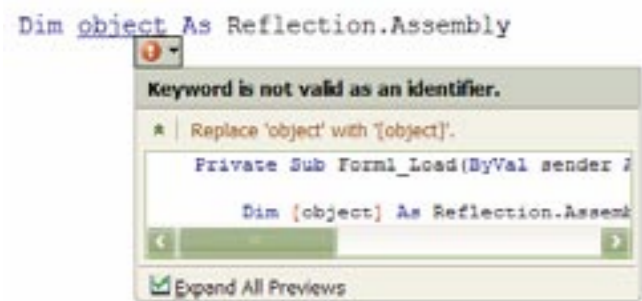
De settings-editor vult niet alleen de config-file (app.config of user.config), maar genereert ook broncode. Om deze broncode te bekijken, moet je eerst 'show all files' selecteren in de solution explorer. Voor Resources is er een editor vergelijkbaar met die van settings. Je kunt zelf een resource-file aanmaken en wijzigen met de editor. Ook kan je op een Windows Form de property 'localizable = true' instellen. Nu kun je door de language property te wijzigen voor elk land nieuwe waarden opgeven voor de objecten op het formulier. Per land wordt dan een resource-file gegenereerd die je strongly typed kunt benaderen met MyResources en die je kunt wijzigen met de editor. In de editor kun je de volgende types opslaan: string, icon, image, audio en text-files.

Code snippets

Als je in de sourcecode-editor met de rechtermuis klikt, zijn er nieuwe opties: 'insert snippet' en 'create snippet'. Als je regelmatig stukken code herhaalt die ongeveer gelijk zijn, dan kan het handig zijn er een snippet van te maken. In afbeelding 2 is duidelijk te zien hoe snippets kunnen worden gebruikt. Een snippet is meer dan het simpelweg plakken van een blok code. Het is mogelijk om snippets te gebruiken als code-templates waar bepaalde delen dienen te worden vervangen. De te vervangen delen worden geel gemarkeerd; zie codevoorbeeld 6. Bij het aanmaken van een snippet kun je zelf bepalen welke onderdelen vervangbaar zijn. Voor het gebruikersgemak kun je aan deze onderdelen ook een tooltip meegeven. Zelf gemaakte snippets worden opgeslagen in een aparte directory in XML-formaat. Je hoeft hiervoor de VB-editor dus niet te gebruiken. Met de code snippet-manager uit het tools-menu kun je zelf bepalen waar in de code snippets-hiërarchie hergebruikte code wordt opgeslagen.

Edit & continue

Edit & continue is terug! Als je een foutje hebt gemaakt dan kun je met edit & continue dit foutje verbeteren, de code opnieuw laten uitvoeren en doorgaan. Er zijn boze tongen die beweren dat je er lui van gaat programmeren, maar als je edit & continue op de juiste manier toepast, kan dit de productiviteit alleen maar ten goede komen. Compileren is namelijk tijdrovend; zeker als je een groot project hebt. Edit & continue werkt trouwens niet voor alle



Afbeelding 3. Auto correction

wijzigingen. Je kunt geen iteraties of conditionele procedures doorbreken of aangemaakte variabelen verwijderen.

Autocorrect, exception helper

Speciaal voor mensen die vaak typfouten maken is er autocorrect. Naast hulp bij het corrigeren van typfouten zal autocorrect waar mogelijk oplossingen aandragen; zie afbeelding 3. Door te klikken op 'replace object with [object]' wordt deze verbetering automatisch doorgevoerd. Als er meer oplossingsrichtingen zijn, worden deze onder elkaar weergegeven. Als de compiler geen meldingen meer geeft, kunnen er nog runtime-fouten in de applicatie zitten. Voor deze fouten is er de exception helper. Voordat gebruik wordt gemaakt van [object] zal deze moeten worden gecreëerd. Als dat niet gebeurt, resulteert dat in een runtime NullReferenceException. De exception helper geeft behalve de foutmelding ook tips om de fout op te lossen. In afbeelding 4 is dit proces duidelijk te zien.

Datasource configuration wizard

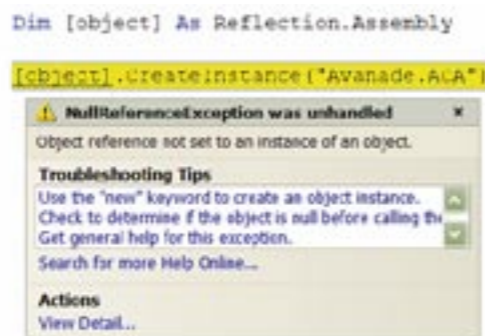
Om het gemak van het programmeren rond databases aan te tonen, wordt in demo's vaak gebruik gemaakt van de drag en drop-mogelijkheid van tabellen uit de SQL Server explorer. Velen die dit trucje na de demo zelf hebben willen toepassen, zijn tot de conclusie gekomen dat het weliswaar heel makkelijk is, maar dat de verbinding met de database is gekoppeld aan een Windows-formulier of ASPX-pagina, terwijl je de dataset liever projectbreed beschikbaar hebt of in een datalayer-class zonder GUI-designer. Met de nieuwe datasource configuration wizard, bereikbaar via datasources naast de solution explorer, kun je projectbrede verbindingen met tabellen, views en of stored procedures realiseren. De aangemaakte DataTable is vervolgens strongly typed benaderbaar. Zie codevoorbeeld 7 voor een koppeling met de Northwind Employee-tabel. Naast de wizard is er een datasource designer. Met deze designer kun je bijvoorbeeld de SQL query aanpassen met behulp van een query-builder; zie afbeelding 5. Als een datagrid-control op een rich client-formulier wordt opgenomen, verschijnt er een klein driehoekje. Door op dit driehoekje te klikken, verschijnen de acties die je met het control kunt ondernemen. In afbeelding 6 is te zien hoe de datagrid wordt gekoppeld aan de Employee-tabel uit de Northwind-database.

Taal-features

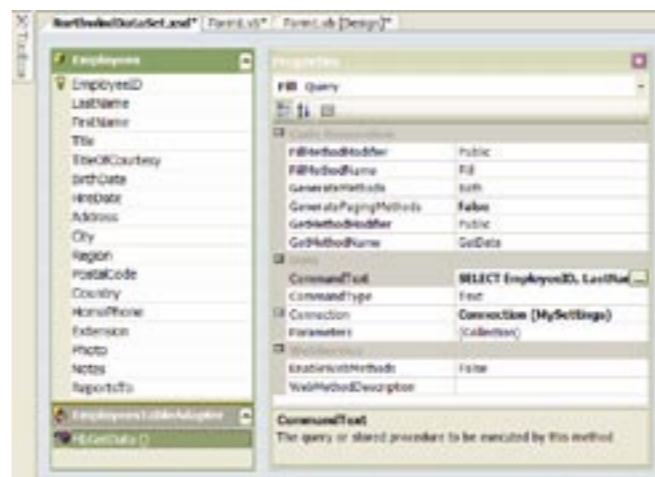
Naast productiviteitsverbeteringen is er een aantal features aan VB toegevoegd op het gebied van de taal zelf.

Partial classes

Het is met versie 2005 mogelijk om een class op te delen in verscheidene files. Twee classes met dezelfde naam in twee verschillende files worden door de compiler weer samengevoegd. Dit is mogelijk door de class als partial te declareren. VB maakt er zelf uitgebreid gebruik van, bijvoorbeeld voor Windows Forms. De door de VB IDE gegenereerde code wordt nu opgenomen in een partial class, verborgen voor de ontwikkelaar. Dat was in versie 2003 ook het geval, maar daar werd de code verborgen in een dichtgeklapte region. Nu is de gegenereerde code opgenomen in een aparte file, die pas zichtbaar



Afbeelding 4. Exception helper



Afbeelding 5. Datasource designer

wordt als je 'show all files' selecteert. Naast het voordeel van partial classes voor codegeneratoren, kun je met partial classes ook twee ontwikkelaars aan één class laten werken.

Using

Ook al maakt de Common Language Runtime gebruik van garbage collection voor het opruimen van geheugen, dat is geen reden om zelf troep te laten liggen. Het is goed programmeergebruik om objecten die je niet meer nodig hebt zo snel mogelijk op te ruimen ('dispose'). Op die manier weet de garbage collector dat het object ook echt verwijderd mag worden. Als je werkt met resources waar je zuinig mee om moet gaan zoals een databaseconnectie of GDI-

```
'Visual Basic 2003
Dim myFont As Font
Try
    myFont = New Font("Tahoma", 12)
Finally
    If Not myFont Is Nothing Then
        myFont.Dispose()
    End If
End Try
```

Codevoorbeeld 8.
Automatisch verwijderen na gebruik

```
' Visual Basic 2003
Dim ferrariList As New ArrayList

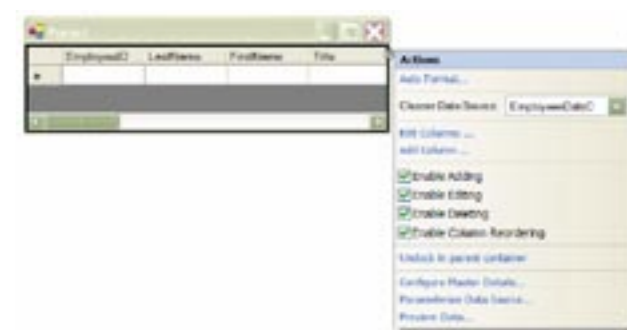
ferrariList.Add(308)
ferrariList.Add("Testarossa")
For Each ferrariCode As Integer In ferrariList
    MsgBox(ferrariCode)
Next
```

Codevoorbeeld 9.
ArrayList met een integer en een string

```
' Visual Basic 2005
Dim ferrariGeneric As New Generic.List(Of Integer)

ferrariGeneric.Add(308)
ferrariGeneric.Add("Testarossa")
For Each ferrariCode As Integer In ferrariGeneric
    MsgBox(ferrariCode)
Next
```

Codevoorbeeld 10.
Opslaan in een generic list van Ferrari ArrayList



Afbeelding 6. Actions

objecten dan is dat belangrijk. In versie 2003 roep je de 'dispose' van een object meestal aan in het Finally-onderdeel van de Try. Als je een object in gebruik neemt met using wordt dit automatisch verwijderd na gebruik; zie codevoorbeeld 8.

Generics

In VB6 gebruik je array's om objecten te groeperen. Als je de grootte van array's runtime wilt veranderen moet je gebruik maken van redim. Met de komst van .NET kun je nog steeds array's gebruiken, maar het is handiger om te werken met een ArrayList uit de System.Collections namespace. Met ArrayList kun je eenvoudig objecten toevoegen, tussenvoegen en bepalen of een bepaald object aanwezig is. Nadeel van ArrayList is echter dat je er objecten van alle types in kunt stoppen. Zie codevoorbeeld 9 waar een integer en een string in de ArrayList worden opgenomen. System.Collections.Generic is een nieuwe namespace die de voordelen biedt van ArrayList maar ook strongly typed is. Met Generics kun je het type bepalen dat je opneemt. De betekenis van generiek volgens de Van Dale is: "waarvan het voorwerp enkel naar de soort bepaald is". De voordelen van Generics zijn performance en veiligheid. Generic collections bieden betere performance, omdat de compiler vooraf weet uit welke elementen de collectie bestaat. Het gebruik van Generic collections is veiliger omdat tijdens het compileren al wordt gecontroleerd of de juiste objecttypes worden gebruikt. Naast de list class bevat de Generic namespace ook nog andere classes, waaronder hashtable. Het is mogelijk je eigen Generic collection te definiëren. In codevoorbeeld 10 staat hoe de ArrayList van Ferrari's in een generic list kan worden opgeslagen. Let op het gebruik van het 'Of' keyword

Reputatie en refactoring

Op het gebied van productiviteit bevestigt Visual Basic 2005 haar reputatie. De nieuwe My namespace, edit & continue en andere zaken zorgen ervoor dat VB weer vooroploopt op dit gebied. Gelukkig gaat dat niet ten koste van belangrijke taalontwikkelingen zoals Generics. Een belangrijk verschil met C# is nu nog de afwezigheid van refactoring. Op de PDC heb ik gehoord dat het VB-team heeft gekozen voor Edit & continue, terwijl C# heeft gekozen voor refactoring. Beide talen kunnen profiteren van deze ontwikkelingen in de versie na 2005. Voor diegene die in VB gebruik wil maken van refactoring zijn er add-on's op de markt. Toen ik op internet zocht naar Visual Basic .NET en refactoring vond ik het volgende plaatje (zie afbeelding 7) dat het refactoring-menu van VB zou zijn. Zelfspot is gezond.

Nuttige internetadressen
http://lab.msdn.microsoft.com/express/vbasic/default.aspx
http://msdn.microsoft.com/vbasic/default.aspx
http://blogs.msdn.com/vbteam
http://msdn.microsoft.com/vbasic/rss.xml

Edwin Jongsma is .NET Solutions Lead bij Avanade (www.avanade.com), een samenwerkingsverband van Microsoft en Accenture. Hij is bij Avanade verantwoordelijk voor de afdeling Software Development. Het e-mailadres van Edwin is edwinj@avanade.com