

De aangekondigde release van Visual Studio 2005 en het .NET platform 2.0 (samen bekend onder de codenaam Whidbey) valt samen met de introductie van SQL Server 2005 (codenaam Yukon) waar Visual Studio 2005 in belangrijke mate mee samenwerkt. In dit artikel beschrijft Willem Koppenol aan de hand van de bèta 1-release (build februari 2005) een aantal van de voornaamste nieuwe mogelijkheden. Het is een persoonlijke selectie en pretendeert geen volledigheid. Bovendien kunnen zaken nog wijzigen voor de definitieve release.

bespreking

Visual Studio 2005 komt eraan

Evaluatie van bètaversie

De aangekondigde nieuwe release van de Microsoft .NET ontwikkelomgeving, Visual Studio 2005 en het .NET platform 2.0, is een major release. Het .NET platform en de ontwikkelomgeving worden op veel fronten verbeterd en uitgebreid. Talen als C# en VB.NET krijgen meer mogelijkheden. Via het Team System komt er in Visual Studio geïntegreerde ondersteuning voor het gehele traject van en alle disciplines in de systeemontwikkeling.

Visual Studio 2005	Visual Studio Orcas
IDE productivity enhancements	Part of Longhorn platform wave
Language innovation	Support for extensive managed interfaces in Longhorn
.NET Framework enhancements	New UI tools and designers
Office solution development	Improved security model
SQL Server 2005 integration	Support for new data storage model
Enterprise lifecycle improvements	

TABEL 1. De geplande Roadmap van Visual Studio releases

.NET FRAMEWORK 2.0 Veel nieuwe functionaliteit is terug te vinden in het .NET Framework, waarbij we een onderscheid kunnen maken tussen verbeteringen in de Common Language Runtime (CLR) en de Framework Class Library (FCL). De CLR is in versie 2.0 voorbereid op de 64-bits architectuur, kent perfor-

manceverbeteringen en ondersteunt generics. De FCL is uitgebreid met vele nieuwe classes en controls waaronder nieuwe generic collection classes en data access controls. Generics en een nieuw patroon voor asynchrone aanroepen zal ik hieronder nader toelichten.

Generics maken het mogelijk dat classes en methoden geparametriseerd worden met een type. Ontwikkelaars kunnen er heel generieke code mee schrijven en een hoog niveau van hergebruik mee realiseren. Een generieke class kan worden geïnstantieerd met een concreet type en werkt dan voor dat type. Generics werden in de vorm van template classes en functies voor het eerst toegepast in de STL (Standard Template Library) van C++ en zijn ook in Java 5 is opgenomen. In .NET 2.0 zal de CLR een implementatie van generics bevatten die type safe is en tijdens compilatietijd wordt gecheckt. De syntax van generics verschilt in de diverse .NET talen.

Generics worden toegepast in de nieuwe System.Collections.Generic namespace die onder meer de classes `Collection<T>`, `Queue<T>` en `Stack<T>` bevat. Het gebruik van generics vermindert bij het gebruik van collections de noodzaak tot typecasting omdat het type expliciet is gemaakt. Een voorbeeld van de definitie en de instantiatie van een generieke type in C# is:

```
public class SimpleGeneric<T> {
    private T m;
    public T member {
        get { return m; }
        set { m = value;}
    }
}
```

```
SimpleGeneric<int> simGen = new
SimpleGeneric<int>();
```

Het .NET Framework 2.0 komt met een nieuw patroon voor asynchrone aanroepen, dat minder complexiteit met zich meebrengt. Asynchroon aanroepen van methoden kon ook in eerdere versies van .NET al via `Begin<MethodName>` en `End<MethodName>`, maar had als nadeel dat de thread die een de asynchrone aanroep deed, niet dezelfde was als de thread die het asynchrone resultaat binnen kreeg. Met name bij het benaderen van GUI componenten gaf dit problemen en was een thread switch nodig met de `Invoke` methode van de betreffende GUI component. In .NET 2.0 is deze threading complexiteit niet langer de zorg van de ontwikkelaar en zijn asynchrone aanroepen event based. Voor een asynchrone aanroep met `<MethodName>Async` moet de ontwikkelaar slechts een `EventHandler` registreren voor het `<MethodName>Completed` event. Bij voltooiing van de methode wordt de `EventHandler` automatisch en op dezelfde thread aangeroepen zodat het benaderen van GUI componenten geen problemen geeft.

C# In de taal C# zijn in versie 2.0 constructies toegevoegd die het mogelijk maken de uitbreidingen in de CLR, zoals generics, te gebruiken. De nieuwe features zijn generics, iterators, anonymous methods en `Nullable` en `partial` types. De C# syntax van generics is zojuist al geschetst.

Iterators maken het mogelijk herhaaldelijk voorkomende enumerator patronen te vereenvoudigen. In C# kun je met een `foreach` constructie over een collectie itereren mits die collectie het `IEnumerable` interface implementeert. Voorheen moest daartoe een class worden gecreëerd die het `IEnumerator` interface implementeert. Met iterators kun je in C# 2.0 de compiler de implementatie van `IEnumerator` laten genereren. De iterator herken je als een statement block waarin één of meer `yield` statements voorkomen:

```
foreach (string s in strings) {
    yield return s;
}
```

Anonieme methoden zijn een taal constructie waarmee ontwikkelaars blokken code kunnen definiëren die in een delegate worden ingekapseld om op een later tijdstip te worden uitgevoerd. Voorheen kon alleen een reeds in een class gedefinieerde methode aan een delegate worden toegevoegd. Anonieme methoden zijn bijvoorbeeld zinvol wanneer de methode tijdens runtime moet wijzigen :

```
button.Click += new EventHandler(sender,
e){
    listBox.Items.Add(textBox.Text);
};
```

`Partial` types maken het voor C# ontwikkelaars mogelijk types met veel source-code te verdelen over verschillende bestanden. `Partial` types kunnen bijvoorbeeld gebruikt worden om de code die door het tool wordt gegenereerd te scheiden van zelf geschreven code en zorgen zo voor een gemakkelijker ontwikkeling en onderhoud.

`Nullable` types tenslotte zijn geïmplementeerd met de nieuwe generic class `System.Nullable<T>`. Ze kunnen aan een willekeurige class de eigenschap `Nullable` toevoegen. Dit kan nuttig zijn voor optionele parameters en het kunnen werken met ongeïnitieerde data.

VB.NET VB.NET 2005 heeft net als C# taaluitbreidingen voor generics. Ook `partial` en `Nullable` types zijn in VB.NET te gebruiken. Vooralsnog laat het zich aanzien dat er voor iterators en anonieme methoden geen ondersteuning is in VB.NET.

In VB.NET 2005 is verder veel aandacht besteed aan het terugbrengen van de hoeveelheid code voor veel voorkomende operaties. Visual Basic language extensions, waaronder de `My` classes, geven met nieuwe objecten en methoden toegang tot de meest gebruikte functionaliteit in het .NET Framework, waaronder het benaderen van forms, data sources, webservices, het file system en de registry. Tabel 2 bevat een voorbeeld dat het verschil in benodigde code in VB.NET tussen versie 2003 en 2005 illustreert.

Visual Basic .NET 2003	<pre>Const GreetingName As String = "Greeting" Dim sDisplay As Object Dim ResMgr As ResourceManager ResMgr = New ResourceManager("Res ourcesSample.MyStrings", _ Me.GetType.Assembly) sDisplay = ResMgr. GetString(GreetingName)</pre>
Visual Basic 2005	<pre>My.Resources.MyStrings.Greeting</pre>

TABEL 2. Het verschil in benodigde code in VB.NET tussen versie 2003 en 2005

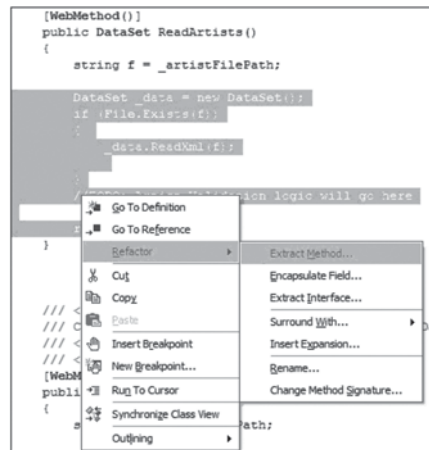
Verder is `Edit & Continue` weer terug van weggeveest in Visual Basic. Bij fouten in de code kun je met de `Edit & Continue` optie de fout herstellen, de code opnieuw laten uitvoeren en doorgaan. Deze optie is niet

bij alle fouten mogelijk. Het werkt bijvoorbeeld niet bij iteraties en procedures die conditioneel worden uitgevoerd.

VISUAL STUDIO IDE De Visual Studio IDE heeft er in versie 2005 veel handige features bij gekregen. In de Code Editors zijn aanzienlijke verbeteringen aangebracht. Intellisense is nu beschikbaar in elk project en in elke taal. Niet alleen in .NET talen, maar ook in HTML en ASP.NET. En ook voor attributes en generics.

Ook zijn er verschillende invulbare templates, expansions genaamd, voor veel voorkomende codeer constructies, zoals voor loops en try/catch blocks. Ontwikkelaars kunnen deze zich herhalende stukjes code snel toevoegen en invullen. Je gebruikt ze door een afkorting in te tikken en met Tab de tekst te laten expanderen. De expansions bevatten velden die met eigen waarden kunnen worden ingevuld. Het is ook mogelijk je eigen expansions te definiëren.

De formattering van code is sterk bepaald door persoonlijke voorkeuren. Men kan nu in Visual Studio kiezen uit een tiental veel gebruikte coderingsstijlen. Bij het kopiëren van een stukje code van het internet zal dit automatisch worden geformatteerd volgens de ingestelde coding stijl. Een interessante optie is verder refactoring dat voornamelijk alleen beschikbaar is in C#. Door refactoring kan het ontwerp en de efficiëntie van bestaande code worden verbeterd. Visual Studio 2005 kent de volgende Refactoring types: Rename, Encapsulate Field, Change Method Signature, Extract Method en Extract Interface.

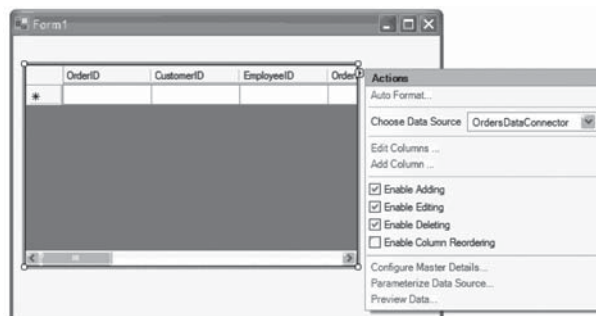


FIGUUR 1. Refactoring opties in C#

WINDOWS FORMS 2.0 Met Windows Forms maak je .NET GUI client applicaties met een rijker geschakeerd user interface en meer mogelijkheden dan webapplicaties. Versie 2.0 biedt diverse nieuwe controls, zoals het DataGridView (een nieuw data bound control), het MaskedTextBox edit control, een Sound Player control, een Web Browser control, een SplitContainer control en een Active Document Host control. Een aan-

tal van deze controls herintroduceren functionaliteit die verloren was gegaan in het migratiepad van VB6 naar .NET. Je krijgt de mogelijkheid een rendering style voor een Windows Form control te kiezen zodat een applicatie meer gaat lijken op Windows of Office. De default renderers kunnen desgewenst vervangen worden door eigen renderers.

Verder worden er in versie 2.0 een aantal zaken toegevoegd die de productiviteit verhogen door het benodigde aantal klikken voor veel terugkomende taken te verminderen, zoals Smart Tags, Document Outline view en SnapLines. Als je aan een control werkt, wordt de Smart Tag van dit control in de Windows Form Designer getoond. Via deze Smart Tag kun je snel gebruikelijke taken uitvoeren of eigenschappen van dat specifieke control wijzigen. Een voorbeeld hiervan is de DataSource property van het DataGridView. Met de Smart Tag kun je een data source selecteren zonder de designer te verlaten.



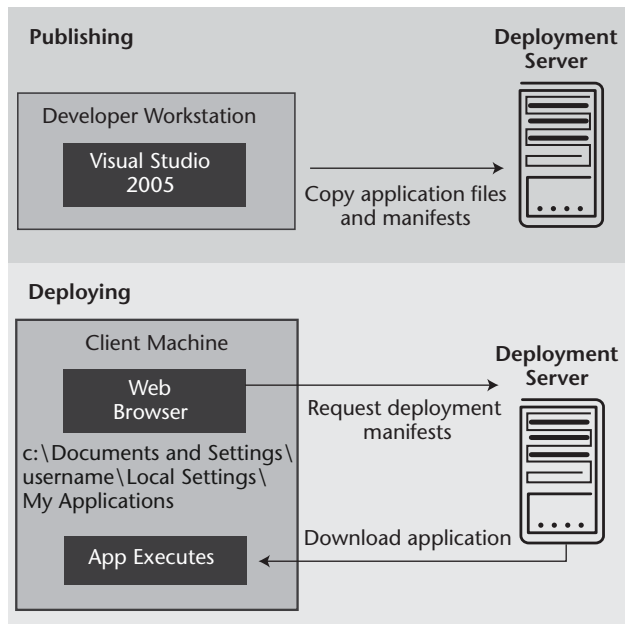
FIGUUR 2. Smart Tag voor een DataGridView Control

Het Document Outline venster geeft een hiërarchisch beeld van alle controls, zodat je eenvoudig door de controls kunt navigeren en de Z-order van een control in zijn container kunt aanpassen. Ook Snaplines is nieuw en stelt je in staat controls snel met elkaar en met het Form uit te lijnen.

CLICKONCE Het paradigma van Windows Forms 2.0 is de ClickOnce-technologie waarmee in belangrijke mate het uitrollen van Window Forms client applicaties wordt vereenvoudigd. Het probleem van Windows Forms applicaties is het uitrolproces bij een toenemend aantal clients. Voor duizenden clients wordt dit kostbaar en ondoenlijk. Met de ClickOnce-technologie van Windows Forms 2.0 wordt het uitrolproces door de gebruikers zelf uitgevoerd die daartoe op een installatiebestand klikken en een Wizard doorlopen.

Als riskante zaken zoals het installeren van drivers nodig zijn, is ClickOnce niet toepasbaar. De ClickOnce applicaties draaien in principe in een beveiligde zandbak, maar meer rechten kunnen desgewenst worden toegekend. ClickOnce applicaties kunnen worden geconfigureerd om automatisch te checken voor

updates. Visual Studio biedt veel ondersteuning voor de publicatie van ClickOnce applicaties.



FIGUUR 3. Het ClickOnce deployment proces

ASP.NET 2.0 De release van ASP.NET 1.0 zorgde voor een geheel nieuwe manier van webapplicatie ontwikkeling en stelde het *event driven*-programmeermodel beschikbaar voor webapplicaties. ASP.NET 2.0 bouwt dit model verder uit en brengt uitbreidingen als de Membership API, Personalisatie API, Master Pages, Themes, en Web Parts. ASP.NET 2.0. levert met direct in te zetten functieblokken voor vaak terugkerende code een hoop productiviteitsverbetering.

MEMBERSHIP Het opzetten van Forms-beveiliging in een webapplicatie wordt in ASP.NET 2.0 sterk vereenvoudigd met de Membership API. De Membership API regelt de toegang tot een applicatie. Deze API heeft functies zoals `CreateUser`, `ValidateUser` en `FindUsersByEmail`.

Het Provider Design Pattern koppelt een database aan de Membership API. Standaard komt ASP.NET met providers voor Access, SQL Server en Active Directory, maar het definiëren van een eigen Provider voor opslag in bijvoorbeeld XML is heel goed mogelijk. Het soort provider wordt in het configuratie bestand `web.config` van de webapplicatie gespecificeerd. Aan welke database en provider de Membership API is gekoppeld is geheel transparant. Wijzigingen in de provider hebben geen invloed op de code die de API gebruikt.

Terwijl Forms-beveiliging in een ASP.NET 1.x applicatie nogal werk kostte omdat je code moest schrijven om een database te openen en gebruikers op te slaan, en moest zorgen voor een login procedure en een login formulier, is dit met de Membership API in enkele aanroepen geregeld.

Bovendien komt ASP.NET 2.0 met een reeks beveiligingscontrols, die intern gebruik maken van de Membership API en die op allerlei manieren aanpasbaar zijn. Hierdoor is het in het geheel niet meer nodig code te schrijven om de toegang tot een webapplicatie te regelen. De controls hebben voor zichzelf sprekende namen als `Login`, `LoginName`, `LoginStatus`, `ChangePassword` en `PasswordRecovery`.

PERSONALISATIE Personalisatie van een webapplicatie kan in ASP.NET 2.0 worden geregeld via een Profile object en de `<profile>` sectie van het configuratiebestand `web.config`. Het profiel van een gebruiker leg je vast door eigenschappen met een naam en een type in de `<profile>` sectie op te nemen:

```
<configuration>
  <system.web>
    <profile>
      <property name="Name" type="System.
String "/>
      <property name="Age" type="System.
Integer "/>
    </profile>
  </system.web>
</configuration>
```

De op deze manier gedefinieerde eigenschappen zijn dan via het Profile object met naam en type in code te benaderen:

```
Profile.Name = "Pjotter Favolution";
```

De toegevoegde eigenschappen zijn via IntelliSense in auto completion-dropdownboxen meteen zichtbaar in Visual Studio. Op grond van de actuele waarden kun je personalisatie toepassen. Verder is het net als bij de Membership API nodig dat er een data provider is geconfigureerd. Dan worden namelijk de Profiles automatisch opgeslagen en aan de gebruiker gekoppeld zodra deze bij de webapplicatie inlogt.

LAY-OUT EN LOOK AND FEEL Nieuw in ASP.NET 2.0 is ook het Master Page concept. Met Master Pages kun je voorzien in een consistente lay-out voor alle pagina's in een webapplicatie. Master Pages fungeren als template voor de pagina's en pagina's kunnen zich baseren op een Master Page. Ze bevatten behalve de definitie van de look-and-feel elementen, ook de basiselementen van een HTML pagina zoals `<HTML>`, `<HEAD>`, `<BODY>` en `<FORM>`. In Master Pages definieer je ContentPlaceHolders, plekken waar pagina's gebaseerd op de Master Page hun inhoud neerzetten.

Elementen waarmee je de look-and-feel van de webapplicatie kunt bepalen zijn Themes en Skins. Met



FIGUUR 4. Master Page met ContentPlaceHolder

een Theme definieer je de totale look-and-feel van de applicatie, zowel kleuren en fonts als ook iconen en afbeeldingen. Deze Themes waren al beschikbaar in andere applicaties, maar zijn nu ook in Visual Studio opgenomen. ASP.NET 2.0 komt met een aantal voorgedefinieerde Themes en het is ook mogelijk eigen Themes toe te voegen. Daarnaast zijn er skin templates waarmee je voor een control een opmaak definieert. Er kunnen per control verschillende opmaken zijn gedefinieerd die worden geïdentificeerd met een SkinID. Binnen Themes kun je aan Skins opnemen. Ook skins zijn bekend in andere applicaties zoals bijvoorbeeld Windows Media Player.

COMPONENTEN EN PARTS Het maken van herbruikbare componenten is erg eenvoudig in ASP.NET 2.0. Door de component in een voorgedefinieerde Code subdirectory te zetten, wordt hij automatisch tijdens runtime naar een assembly vertaald. Aan deze assembly wordt vervolgens automatisch gerefereerd en hij is beschikbaar in alle pagina's van de website.

Ook nieuw is dat het Web Parts Framework (WPF) in ASP.NET 2.0 is opgenomen. Voorheen was het noodzakelijk hiervoor apart Windows SharePoint Portal Server of Services te installeren en de Web Parts SDK te installeren. Web Parts zijn nuttig als de applicatie op een portal moet lijken. Voor een echte portal-applicatie heb je toch SharePoint portal server nodig want de WPF is het front-end van een portal-framework, niet het portal-framework zelf.

CACHING VERBETERINGEN Met ASP.NET 2.0 wordt een nieuw nauwkeuriger cache-invalidatiemechanisme geïntroduceerd dat werkt met in combinatie met SQL Server. Een item in het Cache object kan nu ongeldig worden verklaard zodra de data in de SQL Server-database wijzigt. De SQL Server 2005-database ondersteunt

row-level cache invalidatie en genereert een event richting ASP.NET om een specifiek item in de cache ongeldig te verklaren.

WEBSERVICES Visual Studio 2002 en 2003 stonden al bekend voor hun ruime ondersteuning voor het maken, ontdekken, installeren en gebruiken van webservices. In Visual Studio 2005 wordt deze ondersteuning verder uitgebreid naar de nieuwste webservice-standaards. Zo voldoen de webservices die met Visual Studio 2005 worden ontwikkeld standaard aan de eisen voor interoperabiliteit volgens het Basic Profile 1.0 van de Web Service Interoperability Organisation (WS-I). Het Basic Profile bevat een hele set van regels waaraan webservices met het oog op interoperabiliteit moeten voldoen. Het afzien van het gebruik van SOAP Encoding voor het serialiseren van parameters is daar één van. Deze conformance wordt afgedwongen door een nieuwe property `ConformanceClaims` van het attribuut `WebServiceBinding`.

```
[WebServiceBinding(ConformanceClaims=WsiC1
aims.BP10)]

public class InteroperableService : System.
Web.Services.WebService
{
    [WebMethod]
    public string GetName()
    {
        return "Interoperability";
    }
}
```

Verder biedt Visual Studio 2005 ondersteuning voor zowel SOAP 1.1 als SOAP 1.2. Beide protocollen worden standaard aangeboden. Dit kan echter gewijzigd worden door het configuratiebestand `web.config` aan te passen. Bij het genereren van een proxy voor de client kan het gewenste protocol worden opgegeven en ook programmatisch kunnen webservice-clients de SOAP-versie bepalen door het zetten van de `SOAP-version` property van de proxy class.

WEBSERVICE ENHANCEMENTS De huidige webservices zijn gebaseerd om de simpele uitwisseling van SOAP-berichten en bieden geen ondersteuning voor zaken als security en transactions. Voor geavanceerdere webservices worden momenteel WS.* standaards ontwikkeld die als ze volwassen worden en breed geaccepteerd worden in het .NET platform zullen worden opgenomen. De voorlopige versies van deze standaards zijn voor early adapters nu al te gebruiken door de Web Services Enhancements (WSE) library's als Service Pack in Visual Studio te installeren. Toekomstige releases van

WSE zullen de nieuwste versies van deze specificaties bevatten. Daarnaast zal WSE het hosten van een web-service buiten IIS om ondersteunen (als executable of Windows Service) en zal zowel HTTP als TCP communicatie ondersteunen.

MSBUILD Tot nu toe was het niet mogelijk nauwgezette controle uit te oefenen over het bouwproces van .NET applicaties in Visual Studio. Het was nooit geheel duidelijk wat Visual Studio precies deed wanneer een project in de ontwikkelomgeving werd gebouwd en je kon het bouwproces slechts minimaal beïnvloeden. Met de introductie van Visual Studio 2005 zal dit sterk verbeteren omdat een nieuwe build engine, MSBuild, wordt meegeleverd. MSBuild wordt volledig geïntegreerd in de Visual Studio IDE, waardoor ontwikkelaars kunnen de voordelen genieten van deze omgeving maar toch het bouwproces naar hun hand kunnen zetten. MSBuild heeft een sterke verwantschap met de Open Source Ant build engine waarmee in de Java-wereld al langere tijd projecten worden gebouwd. Een simpel voorbeeld MSBuild script waarmee `HelloWorld.cs` wordt gebouwd is:

```
<Project>
  <Item
    Type="FilesToCompile"
    Include="HelloWorld.cs"
  />

  <Target Name="BuildHelloWorldExecutable"
">
    <Task
      Name="csc"
      Sources="@ (FilesToCompile)"
      OutputAssembly="HelloWorld.exe"
      TargetType="exe"
    />
  </Target>
</Project>
```

MSBuild maakt gebruik van een nieuw XML-bestandsformaat dat gebaseerd is op een XML-schema en eenvoudig is te gebruiken en uit te breiden. Dit MSBuild bestandsformaat stelt ontwikkelaars in staat precies te beschrijven welke artefacten gebouwd moeten worden, als ook hoe ze gebouwd moeten worden in verschillende configuraties. Alle stappen worden expliciet uitgeschreven in het XML-projectbestand. Het doet er daarbij niet toe of het MSBuild-bestand met de hand is geschreven of gegenereerd door de Visual Studio IDE. Dit betekent ook dat Visual Studio het "F5" bouwcommando niet langer opvat als een *black box*. Een ieder kan de precieze stappen van het bouwproces volgen en desgewenst aanpassen of uitbreiden. Het is ook mogelijk

herbruikbare regels uit een bouwbestand naar andere bestanden uit te splitsen, daaraan vervolgens te refereren en zo zorg te dragen voor een consistent bouwproces over verschillende projecten.

Verder wordt de MSBuild-engine ook een integraal deel van de .NET Framework 2.0-distributie. Ontwikkelaars kunnen dus vertrouwen op de MSBuild-infrastructuur los van de aanwezigheid van de IDE en eventuele licentievraagstukken. Tenslotte worden de MSBuild class library's opgenomen in het .NET Framework, zodat ontwikkelaars een aangepast bouwproces desgewenst ook kunnen debuggen in de taal van hun keuze.

DATA ACCESS In het .NET Framework 2.0 worden de data access-mogelijkheden verbeterd door de nauwe integratie met SQL Server 2005 en doordat er data access-controls en services bij komen. Ook worden er verbeteringen aangebracht in de ADO.NET class library's. Met name wordt de performance van DataSets sterk verbeterd doordat nu snelle binaire serialisatie wordt ondersteund. Verder krijgt ADO.NET een provider-onafhankelijk objectmodel, waardoor het mogelijk wordt provider-onafhankelijke code te schrijven. De ondersteuning van tracing wordt verder verbeterd.

Jammer is het dat Microsoft de ObjectSpaces technologie uit deze release heeft geschrapt. Dit veelbelovende Object Relational Mapping Framework kan vergeleken worden met Java Data Objects (JDO) in de Java-wereld en maakt het mogelijk het domeinmodel van de applicatie direct, zonder een expliciete vertaalslag naar een relationeel datamodel, te persisteren in een willekeurige datastore. Naar verluidt heeft Microsoft de ObjectSpaces technologie doorgeschoven naar een volgende release maar voorlopig zullen we het dus zonder persistentie-framework in .NET moeten stellen.

SQL SERVER 2005 INTEGRATIE Visual Studio 2005 wordt gelijktijdig met SQL Server 2005 gelanceerd en biedt een sterke mate van integratie met en ondersteuning voor SQL Server 2005. Zo zal de .NET Common Language Runtime (CLR) in SQL Server 2005 zijn opgenomen waardoor database-objecten zoals functies, stored procedures en triggers voortaan in C# of VB.NET of een andere .NET taal, kunnen worden geschreven. Deze .NET-talen bieden over het algemeen een rijker programmeermodel met constructies en operaties die niet aanwezig zijn in Transact SQL. Hoewel Transact SQL voor veel tabel-georiënteerde acties nog wel in zwang zal blijven, hoeven .NET-ontwikkelaars zich voor SQL Server gerelateerde berekeningen en complexe logica dus niet noodzakelijk in Transact SQL te verdiepen.

De rijke functionaliteit van de .NET Base Class Library (BCL) komt ter beschikking van database objecten, waardoor het bijvoorbeeld mogelijk wordt data te

valideren met de reguliere expressies uit `System.Text.RegularExpressions`. Ontwikkelaars zullen ook managed code kunnen gebruiken om de nieuwe SQL Server 2005 objecten zoals User Defined Types en Aggregates te bouwen. Database-objecten zullen verder een grotere mate van beveiliging kennen omdat ze draaien in de context van het fijnmazige CLR-security-model. Databasebeheerders krijgen meer mogelijkheden om een toegestane set acties vast te leggen als code resources binnen of buiten SQL Server 2005 wil benaderen.

Tenslotte zullen stored procedures en database objecten in .NET talen binnen Visual Studio gedebugged kunnen worden en zal de performance vanwege de optimalisaties in de compilers voor .NET talen aanzienlijk toenemen ten opzichte van Transact SQL.

TEAM SYSTEM Voor veel projectmanagers, architecten, ontwikkelaars en testers die in teams samen werken aan grootschalige .NET projecten, is het Visual Studio 2005 Team System waarschijnlijk de belangrijkste innovatie. Visual Studio ontstijgt hiermee haar oorspronkelijke focus als tool voor pure softwarecodering. Het Team System biedt een in Visual Studio geïntegreerde ondersteuning voor de hele levenscyclus van een softwareproject. Van requirementsanalyse, via ontwerp naar implementatie, testen en uitrollen.

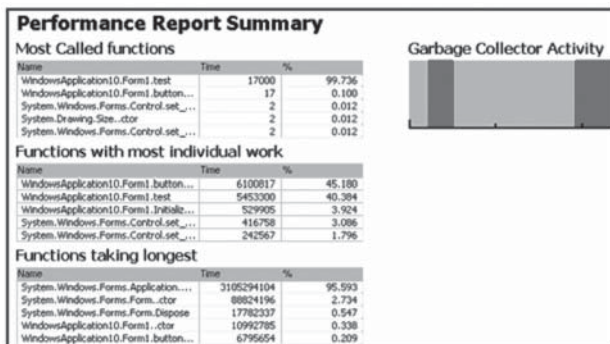
Het Team System voorziet in tools voor alle disciplines die bij een software ontwikkelproces een rol spelen en maakt goede communicatie tussen deze disciplines mogelijk. De Team Architect-editie voorziet in geïntegreerde tools voor het visueel ontwerpen van applicaties. De Developer-editie geeft de ontwikkelaar tools voor code profiling, code coverage en unit testing. De Test-editie levert testers de nodige tools voor het uitvoeren en managen van diverse testscenario's. Projectmanagers kunnen Visual Studio 2005 Team Foundation gebruiken en op een geïntegreerde manier met behulp van management-tools als Excel en Project de status van een project bekijken. Deze laatste editie levert ook ondersteunende tools om het ontwikkelteam als geheel te laten samenwerken.

In de aanstaande Visual Studio 2005 Bèta 2-release wordt het Team System voor het eerst meegeleverd. Je kunt ermee kennis maken door je op te geven voor het Visual Studio Beta 2-programma op www.microsoft.nl/betaexperience en krijgt dan bovendien een DVD met allerlei achtergrondinformatie. Software Release Magazine zal in een volgend nummer, als meer details bekend zijn, uitgebreid aandacht aan het Team System besteden.

SLOTWOORD De Visual Studio 2005 met .NET 2.0 release betekent een flinke stap voorwaarts. De release bevat een scala aan verbeteringen en vernieuwingen en verhoogt de productiviteit bij ontwikkeling voor .NET aanzienlijk. Met name de introductie van het Team

Het Team System voorziet in tools voor alle disciplines binnen het ontwikkelproces

System betekent een grote verbreding van de scope van Visual Studio. Als we Microsoft mogen geloven is dit echter nog niets vergeleken bij wat ons te wachten staat met de volgende versie van Visual Studio en .NET, met de code naam Orcas. Deze release valt samen met Longhorn, de nieuwe Windows-versie en zal het Indigo-platform voor gedistribueerde applicaties bevatten. De voorspelling van Microsoft is dat dit een grote technologische omwenteling wordt die een revolutie in de software industrie teweeg brengt.



FIGUUR 5. Performance rapport uit de Team System Developer Editie

drs. Willem Koppenol (e-mail: wkoppenol@twice.nl) is senior trainer en productspecialist software development bij Twice IT Training.