

WebSphere Studio Application Developer (WSAD) is de nieuwe Java programmeeromgeving van IBM die voortkomt uit het open source project Eclipse en functioneel gezien een opvolger is van de bekende programmeeromgeving Visual Age for Java. Zoals de naam WebSphere Studio Application Developer al aangeeft, zijn hierin ook de mogelijkheden van IBM's web development tool WebSphere Studio opgenomen. WSAD combineert een programmeeromgeving voor Java met een omgeving voor client-side web development: naast Java classes kunnen dus ook HTML- en JSP-pagina's geschreven worden.

*bespreking*

# Nieuwe generatie web development

## *IBM introduceert opvolger Visual Age for Java*

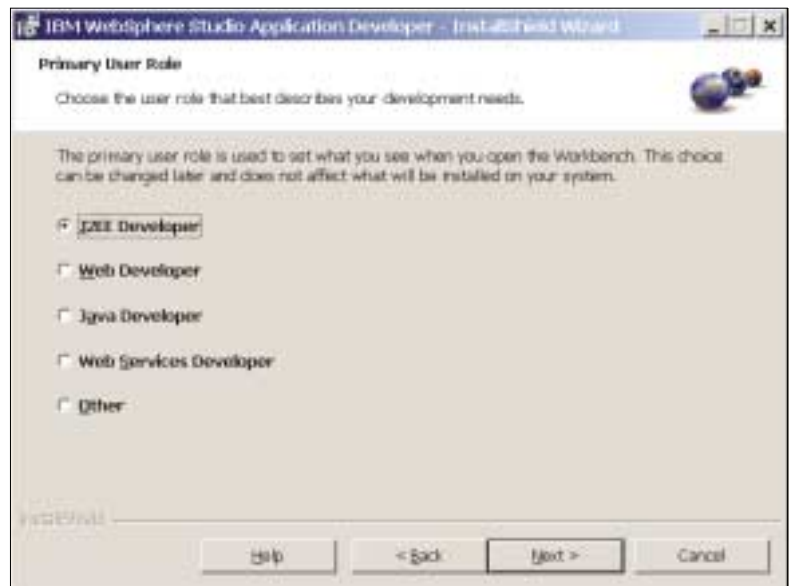
WSAD is meer dan een opvolger of een combinatie van eerdere tools van IBM. Het is bedoeld als integrale oplossing voor het bouwen van webapplicaties met Java. Het is in feite een reactie op de sterk veranderde wereld van internettoepassingen. Visual Age for Java en WebSphere Studio stammen uit een tijd dat Java met name werd gebruikt voor het schrijven van applets, en internetapplicaties nog vooral bestonden uit statische HTML-pagina's met daarnaast CGI-scripts voor het verzenden van gegevens.

De afgelopen vijf jaar is het internet, en daarmee ook Java als internettaal, sterk in ontwikkeling geweest. Met de komst van het J2EE-platform (Java Enterprise Edition) kent Java nu een groot aantal toepassingen binnen een webapplicatie, zoals EJB's en servlets. Java wordt ook aangetroffen aan het front-end van het internet. HTML-pagina's en stukken Javacode vormen samen zogenaamde JSP-pagina's (Java Server Pages).

Naast de opkomst van J2EE is vooral de wens tot integratie van omgevingen en platforms een cruciale ontwikkeling voor internettoepassingen. De wereld is immers groter dan alleen Java, bovendien willen bedrijven zich steeds vaker richten op meerdere omgevingen. Web services en XML zijn momenteel sleuteltechnologieën voor het bouwen van webapplicaties voor verschillende systemen. IBM heeft met WSAD in ieder geval een leidende

rol bij deze ontwikkelingen. Zo kan WSAD prima overweg met Java, HTML, WebSphere en J2EE en biedt het ook ondersteuning voor XML en Web Services.

**ROLLEN BINNEN WSAD** Bij de installatie van WSAD valt direct het hoge versienummer op: 4.02. Dit versienummer heeft betrekking op de versie van de applicatieserver WebSphere van IBM waaraan WSAD gekoppeld is. Bij het installeren van WSAD 4.02 blijkt verder



FIGUUR 1: Rollen binnen WSAD

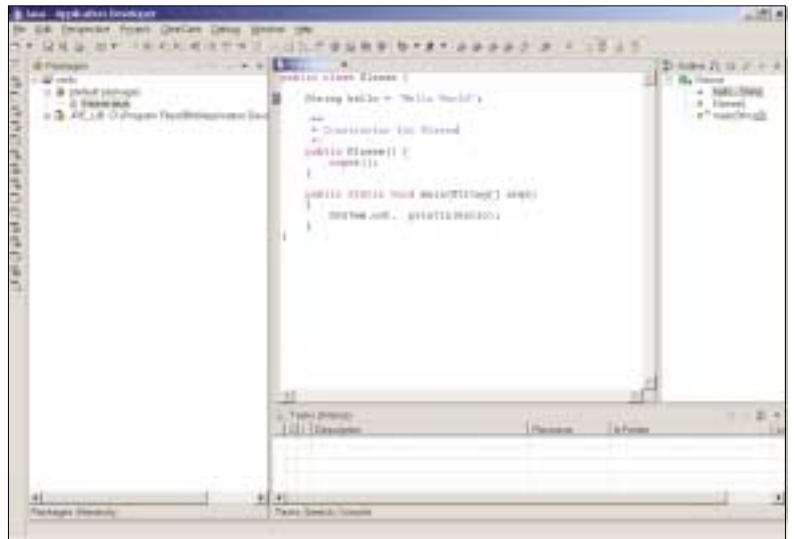
dat we met een veelzijdige omgeving voor internetontwikkelaars te maken hebben. We kunnen namelijk kiezen uit de rollen J2EE developer, web developer, Java developer en web services developer. De keuze van een rol bepaalt hoe de workbench van WSAD er direct na het opstarten uitziet. Dit kan overigens later naar eigen inzicht weer worden gewijzigd.

Na het selecteren van een rol worden we gevraagd een keuze te maken uit een aantal versiebeheerprogramma's: WSAD is standaard te koppelen aan het open source programma CVS (Concurrent Version Systems) of het veelgebruikte Rational ClearCase, maar het is ook mogelijk een ander programma te gebruiken.

Zorg wel bij het installeren dat u bent ingelogd als administrator om system services te starten. Als dit niet het geval is, verschijnt aan het einde van de installatie namelijk de melding "Error 1920. Service IBM Agent Controller failed to start" en moet de installatie opnieuw worden uitgevoerd. Het installeren van WSAD wordt dan een erg tijdrovende bezigheid. Verder is een behoorlijke hoeveel intern geheugen vereist. 128 MB is in theorie mogelijk, maar alleen aan te raden voor gebruikers met zeer veel tijd en geduld. IBM zelf raadt 256 MB aan, maar voor een lekker vlotte schermopbouw is 512 MB geen overbodige luxe.

**PERSPECTIVES EN VIEWS** Typerend voor de workbench van WSAD is het werken met perspectives en views. Door het selecteren van een perspective kun je de workbench optimaliseren voor een bepaalde rol. Een perspective bestaat uit een aantal views die de resources in een perspective weergeven. Als je bijvoorbeeld kiest voor de perspective J2EE (gebruik hiervoor de opties 'open', 'Java' in het menu 'perspective'), worden views weergegeven met informatie over de Java classes in een webapplicatie.

De views zijn in feite deelvensters die zich aan de randen van het venster bevinden. Het hoofdvenster in het midden van de workbench geeft de inhoud weer van het bestand dat geopend is. In de door ons geselecteerde Java perspective zien we de views 'outline' en 'packages'. Deze views tonen respectievelijk een overzicht van de methoden en variabelen weergegeven van de Java class die in het hoofdvenster is geopend en een overzicht van de packages waarvan in een project gebruik wordt gemaakt. Dit lijken absoluut zinvolle opties voor een Javaprogrammeur. Het hoofdvenster zelf valt een beetje tegen door het ontbreken van regelnummers. Tijdens het debuggen wordt wel aangegeven op welke regel een fout optreedt, maar dit regelnummer valt dan niet meer terug te vinden in de programmacode. Dit was ook een veelgehoorde klacht over Visual Age for Java en



**FIGUUR 2:** De Java perspective met aan de linkerkant koppelingen naar andere perspectives

we hadden verwacht dat IBM dit probleem reeds opgelost zou hebben. In de HTML- en XML-perspectives is regelnummering overigens wel aanwezig.

Voor het openen van een nieuwe perspective gebruiken we de snelkoppeling 'open perspective' in de linkerbovenhoek van het scherm. De interface wordt vervolgens opnieuw opgemaakt en we zien een leeg hoofdvenster met daaromheen onder meer de views 'breakpoints' en 'console'. In de eerste view staan alle breakpoints die in de Javacode zijn gezet, in de tweede wordt de standaard output afgebeeld. De view 'console' is een goed voorbeeld van de gebruikersvriendelijkheid van WSAD. Waar met de combinatie Visual Age for Java en WebSphere bij het debuggen van een applicatie een bestand met standaardoutput in de directorystructuur van WebSphere

## Voor een aantal activiteiten is nog steeds Visual Age for Java vereist

moet worden gezocht en geopend, wordt deze output in de debug perspective van WSAD standaard op het scherm getoond.

Naast de perspectives 'Java' en 'debug' zijn er nog talloze andere perspectives te selecteren. Alle facetten van een moderne webapplicatie kunnen apart worden bekeken. Zo kan er gekozen worden uit Data, Debug, Help, J2EE, Java, Java Type Hierarchy, Plug-in development, Profiling, Resource, Scripts, server, Team, Web en XML. Dit betekent overigens niet dat je als gebruiker verplicht bent van al deze perspectives gebruik te maken. Een perspective is niets meer en niets minder dan een manier

om tegen een webapplicatie aan te kijken. De applicatie zelf verandert niet door er op een andere manier tegenaan te kijken. In theorie is het mogelijk alle werkzaamheden in dezelfde perspective uit te voeren. Ook is het mogelijk een perspective te veranderen door er views aan toe te voegen of views te verwijderen. Zo kan een perspective aan de wensen van de individuele gebruiker worden aangepast. Voor de meeste gebruikers zullen drie tot vier perspectives voldoende zijn.

**EEN FILE-BASED JAVA-OMGEVING** De Java programmeeromgeving van WSAD is gebaseerd op de technologieën van het open source platform Eclipse en bevat een groot aantal features die we al kennen van Visual Age for Java. Voorbeelden hiervan zijn incrementeel compileren, automatisch herstellen van classes na een crash en het werken met een scrapbook. Ook zijn veel toetsencombinaties identiek aan die van Visual Age for Java. Zo kan met Ctrl + Spatiebalk een lijst met mogelijke methoden bij een variabele worden opgevraagd.

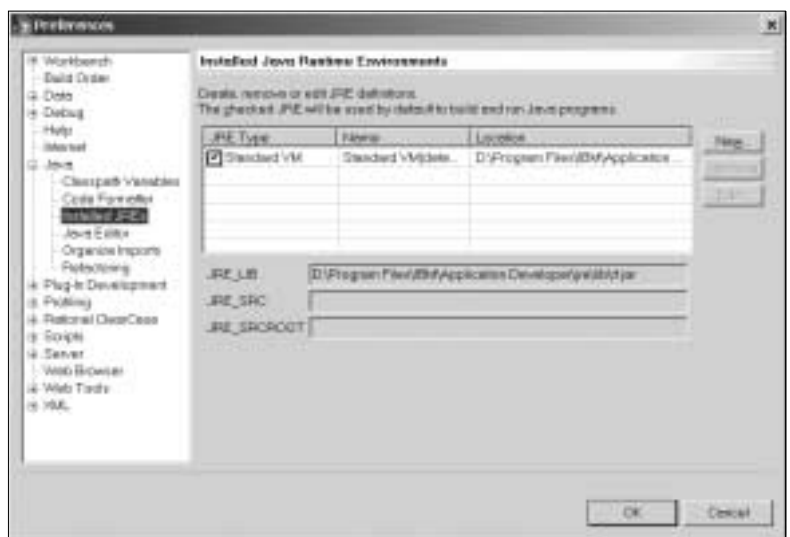
Een groot voordeel van WSAD ten opzichte van Visual Age for Java is dat de Java-omgeving file-based is. Waar alle Javacode in Visual Age for Java werd opgeslagen in één grote repository, wordt code in WSAD opgeslagen in Javabestanden. Naast een betere aansluiting bij andere programmeeromgevingen voor Java, zoals JDeveloper en JBuilder, biedt dit ook meer editmogelijkheden voor de gebruiker. Zo kunnen Linux-gebruikers een commando als grep loslaten op de Javabestanden en zijn ze niet langer beperkt tot de editmogelijkheden die de programmeeromgeving zelf aanbiedt. Bovendien sluit file-based code beter aan bij de realiteit van Java, waarin tenslotte verwacht wordt dat elke class zich in een bijbehorend Javabestand bevindt.

**PLUGGABLE JDK'S** Het compileren van Javacode gebeurt in WSAD, net als in Visual Age for Java, met een interne compiler. Een pluspunt ten opzichte van Visual Age for Java is hierbij dat het mogelijk is meerdere JDK's (Java Development Kits) aan een project te koppelen. Waar programmeurs in Visual Age for Java beperkt waren tot de JDK die met de omgeving werd geleverd, kan aan WSAD op eenvoudige wijze een nieuwe JDK worden toegevoegd. In het venster 'preferences' (te bereiken via het menu 'windows') vind je een overzicht van de JRE's (Java Runtime Environment) die reeds zijn geïnstalleerd. Door op 'new' te klikken, kan een andere JRE of JDK worden toegevoegd. Hiermee kan handig gebruik gemaakt worden van de nieuwste features van de taal Java, zoals bijvoorbeeld in de nieuwe JDK versie 1.4. Een JRE is een soort uitgekleepte JDK, die alleen de bestanden bevat die nodig zijn om Javacode uit te voeren. Een JDK bevat daarnaast nog andere features, zoals een debugger en documentatie bij classes.

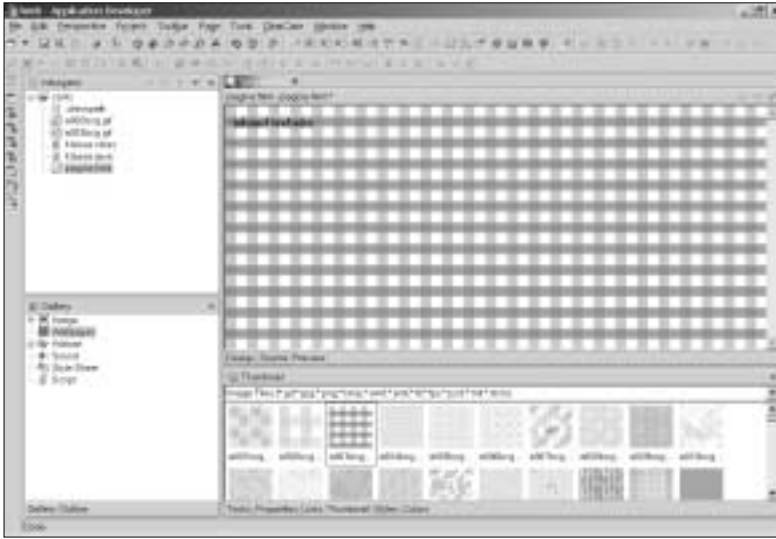
**WEBSITES ONTWERPEN** De omgeving voor het ontwerpen van website benaderen we via de web perspective van WSAD. Het hoofdvenster bestaat uit een 'source' tabblad die de code achter een HTML- of JSP-pagina weer geeft, een 'preview' tabblad om te kijken hoe een pagina in een browser wordt afgebeeld en een 'design' tabblad waarmee het mogelijk is een pagina te wijzigen zonder direct in de code te duiken. Verder is het mogelijk met JavaScript en stylesheets te werken en kunnen afbeeldingen en audiobestanden makkelijk via 'drag-and-drop' worden toegevoegd.

## De nieuwe programmeeromgeving van IBM is een toenadering richting Microsoft

**IBM VERSUS MICROSOFT** De nieuwe programmeeromgeving van IBM vormt in verschillende opzichten een toenadering richting Microsoft. Niet alleen is WSAD wat betreft gebruikersvriendelijkheid vergelijkbaar met de producten van Microsoft, het heeft inhoudelijk ook veel gemeen met het recentelijk geïntroduceerde Visual Studio .NET van Microsoft. Zo baseren zowel IBM als Microsoft zich op de open source technologieën UDDI, WSDL en SOAP die op dit moment de standaarden zijn voor web services. Deze standaardisering van technieken is een resultaat van de vorming van een alliantie tussen IBM en Microsoft, de "Web Services Interoperability Organization (WS-I)", die streeft naar compatibiliteit van web services en XML -technologieën. Ook de scope van WSAD valt te vergelijken met die van Microsoft's product. Het gaat hier in beide gevallen om een geïntegreerde programmeeromgeving waarin een volledige internetapplicatie kan worden ontworpen en getest. Het grote verschil



FIGUUR 3: JDK's toevoegen



FIGUUR 4: De Web perspective

tussen WSAD en Microsoft's .NET is uiteraard de keuze van de programmeertaal. WSAD is volledig gericht op Java, terwijl .NET vooral gebruik maakt van C#, Visual Basic .NET en C++.

**DOORBRAAK IN WEB DEVELOPMENT** Al met al kunnen we stellen dat WebSphere Studio Application Developer in vrijwel alle opzichten een uitstekende opvolger is van het krachtige, maar verouderde Visual Age for Java. De ontwikkelomgeving is veel intuïtiever dan die van Visual Age for Java en kent dus een kortere leercurve. De interface is snel en gebruikersvriendelijk en doet wat dat betreft denken aan een product van Microsoft. Verder is het product stabiel en is de file-based Java-omgeving meer in overeenstemming met de producten van de concurrentie.

## Het grote verschil tussen WSAD en Microsoft's .NET is de keuze van de programmeertaal

Door de betere aansluiting op moderne internettechnologieën verwachten we dat WSAD binnen twee jaar de voorganger Visual Age for Java uit de markt zal hebben geprijsd. Waarschijnlijk is het ook het streven van IBM om Visual Age for Java snel te vervangen. Zo biedt WSAD al ondersteuning voor WebSphere 4.0, waar de nieuwste versie van Visual Age for Java alleen WebSphere 3.5 ondersteunt. Daarnaast ondersteunt WSAD de deployment van EAR-, WAR- en JAR-files en het gebruik van EJB 1.1 in een J2EE-omgeving, allemaal features die in Visual Age for Java ontbreken.

Het streven om alle onderdelen van een webapplicatie in één omgeving te integreren kunnen wij alleen maar

toejuichen. Een betere integratie met front end, applicatieserver en web services zal ongetwijfeld leiden tot snellere ontwikkeltrajecten, al is het alleen maar omdat niet langer kennis van een groot aantal tools benodigd is. Een volledige integratie van technieken is helaas nog niet bereikt. Voor een aantal activiteiten is nog steeds Visual Age for Java vereist, zoals het werken met de Enterprise Access Builder, het gebruik van de Persistence Builder voor het genereren van Java Beans en het visueel ontwerpen van Java-applicaties met behulp van de Visual Composition Editor. Om deze redenen kan WSAD dan ook samen met Visual Age for Java worden aangeschaft in de zogenaamde Application Developer Box. Deze bevat dan ook nog IBM's applicatieserver WebSphere versie 4.01, IBM's database DB2 versie 7.2 en het versiebeheerprogramma Rational Clearcase LT. Integratie van deze producten met WSAD is uiteraard mogelijk.

De combinatie van Visual Age for Java en WSAD in één pakket is naar onze mening onnodig en het gelijktijdig uitvoeren van deze tools is een te zware belasting voor een gemiddeld systeem. Van IBM mag worden verwacht dat een volgende versie van WSAD wordt uitgerust met de volledige functionaliteit van Visual Age for Java.

*Carlo Smits*

werkt als Java programmeur bij de sector Financial Services van Cap Gemini Ernst & Young en is te bereiken via [carlo.smits@cgey.nl](mailto:carlo.smits@cgey.nl)