

# Windows Workflow Foundation 4

## BRUIKBAAR GEREEDSCHAP VOOR .NET-ONTWIKKELAAR

Maurice de Beijer

Met de release van het .NET framework 3.0 kwam Microsoft met de eerste versie van Windows Workflow Foundation. Nu krijgen we het .NET framework versie 4 dus kunnen we verwachten dat er een nieuwere versie van Windows Workflow Foundation komt met meer en mooiere mogelijkheden. Dat is grotendeels ook wel zo maar niet helemaal op de manier waarop je, gezien hoe het meestal gaat, zou verwachten.

### Windows Workflow Foundation

heeft een aantal verschillende toepassingsgebieden:

#### Transparantie van de applicatie

Door de processen waar de gebruiker mee te maken krijgt op een grafische manier weer te geven, is het veel makkelijker om de applicatie aan een (power) gebruiker uit te leggen. Waar we vroeger met hulpmiddelen als Visio losse diagrammen maakten, die niet echt werden uitgevoerd, hebben we nu workflow diagrammen die ook echt worden uitgevoerd.

Dit lijkt een klein voordeel maar in de praktijk wordt documentatie vaak niet bijgewerkt en kijken gebruikers of analisten naar een diagram dat eigenlijk niet klopt. Bij workflow kijken ze naar de echte code in diagramvorm en kunnen ze er zelfs met de Visual Studio debugger doorheen stappen.

#### Ontwikkelaars productiviteit

Veel zakelijke applicaties hebben te maken met dezelfde problemen die eigenlijk niet met de applicatie van doen hebben. Denk hierbij aan dingen als het bewaren van de status van langlopende processen als er niets gebeurt, reageren op gebeurtenissen of juist het gebrek er aan zoals een factuur die niet wordt betaald. Daarnaast moeten we vaak vastleggen wat er wanneer is gebeurd. Dit zijn allemaal zaken die door Windows Workflow Foundation worden opgelost zonder dat de ontwikkelaar daar iets specifiek voor hoeft te doen.

#### Applicatie scripts

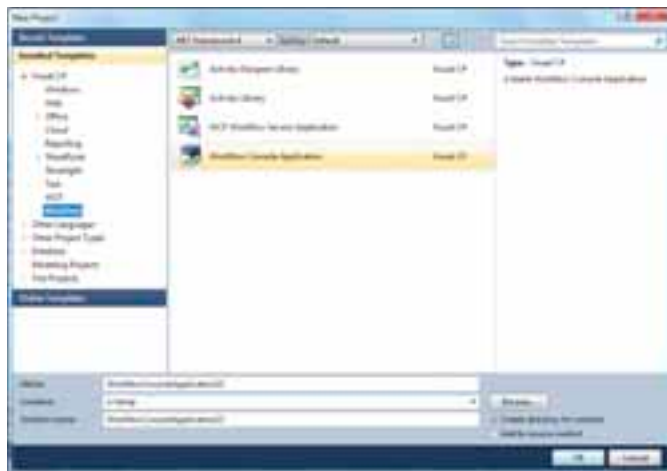
Door het gemak waarop de workflow designer – die onderdeel is van het .NET framework zelf in plaats van Visual Studio – binnen een applicatie kan worden gebruikt, is het heel goed mogelijk om een eindgebruiker eenvoudige workflows te laten maken die als een soort script kunnen worden uitgevoerd.

De ontwikkelaar heeft hierbij alle controle over welke activiteiten een gebruiker wel en niet kan gebruiken.

### Schone lei

In tegenstelling tot wat er meestal gebeurt, is Microsoft in dit geval niet met de oude versie van Windows Workflow Foundation verder gegaan om hier bugs te herstellen en nieuwe features aan toe te voegen. Bij het beoordelen van de bekende problemen en wat er nodig was om deze op te lossen kwam men tot de conclusie dat er dusdanig veel moest worden veranderd dat bestaande applicaties toch niet zonder grote aanpassingen zouden werken. En als iedereen zijn applicatie toch moet wijzigen was er eigenlijk geen goede reden om met de oude code door te gaan. Zodoende is er besloten met een schone lei te beginnen en is Windows Workflow Foundation 4 een geheel nieuw product geworden dat niet backwards compatible is met de eerdere Windows Workflow Foundation 3.

Maar wat dan met applicaties die Windows Workflow Foundation 3 gebruiken? Een goede vraag, want dat is een onderdeel van het bestaande .NET framework en kan niet zomaar verdwijnen. Uiteraard gebeurt dat ook niet. Ontwikkelaars die WF3 gebruiken hebben eigenlijk 3 keuzes:



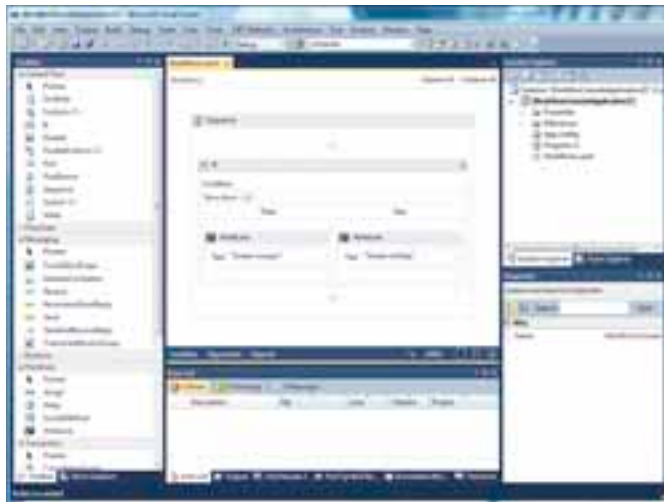
AFBEELDING 1: DE VERSCHILLENDE WORKFLOW 4 PROJECTEN IN VISUAL STUDIO 2010.

- Ze kunnen hun applicatie herschrijven om WF4 te gebruiken in plaats van WF3. Om hiermee te helpen is er zowel documentatie als een WF Migration Kit die code automatisch probeert om te zetten;
- Ze kunnen hun applicatie op het .NET 3 platform laten en niet upgraden naar .NET 4. Uiteraard heeft dit als nadeel dat er nergens nieuwe features gebruikt kunnen worden;
- Ze kunnen de applicatie naar .NET 4 upgraden maar de WF3 stack blijven gebruiken. Alle WF3 classes zijn namelijk gewoon overgezet naar .NET 4 en gewoon te gebruiken. In feite hebben we in .NET 4 dus twee verschillende workflow stacks.

Een bestaande workflow ontwikkelaar heeft dus meer dan genoeg keuzemogelijkheden hoe om te gaan met de overstap van .NET 3 naar .NET 4.

## Een eerste workflow maken

De beste manier om de mogelijkheden van WF4 te bekijken is door deze te gebruiken. Als we Visual Studio 2010 opstarten zien we onder de Workflow opties vier verschillende projecttypes staan. De eenvoudigste om mee te beginnen is de Workflow Console Application. Als we dat doen krijgen we gelijk een lege workflow en de benodigde code om de workflow uit te voeren. Laten we bij deze laatste code beginnen, want er is niet veel nodig - zeker vergeleken met de WF3 templates. De WorkflowInvoker klasse bevat een statische methode Invoke die we de workflowdefinitie meegeven die moet worden uitgevoerd. De WorkflowInvoker is maar één van de drie mogelijke manieren om een workflow uit te voeren en dan wel op een synchrone manier. Het wordt ook wel een workflow als een functie genoemd. Naast de WorkflowInvo-



AFBEELDING 2: DE WORKFLOW DESIGNER IN VISUAL STUDIO 2010.

ker is er ook een WorkflowApplication voor asynchrone executie en een WorkflowServiceHost als we IIS gebruiken om onze workflows uit te voeren.

### Workflows zijn nu XAML

Als we naar de workflow zelf gaan kijken, is het eerste dat opvalt de XAML-bestandextensie. Dit is exact hetzelfde XAML-formaat dat binnen Windows Presentation Foundation wordt gebruikt. Het wordt nu alleen niet gebruikt om UI controls op een scherm te zetten, maar om activiteiten aan een workflow toe te voegen en deze te configureren. Als we de workflow openen zien we een vrijwel leeg scherm met links een toolbar met de meeste standaard

(Advertentie)

2 juni 2010 • Hotel Lapershoek Hilversum

## Pragmatisch ontwikkelen met .NET

*Best practices in .NET projecten*

met **Sander Hoogendoorn**



.NET.NET .NET.NET .NET.NET .NET.NET .NET.NET

De onderwerpen van dit seminar dragen direct bij aan het succesvol ontwikkelen van software. Tijdens het seminar bespreekt Sander Hoogendoorn, principal technology officer bij Capgemini en lid van de Visual Studio Advisory Board bij Microsoft, het toepassen van software architectuur, het opzetten van frameworks en het hanteren van de juiste ontwerp patronen. Het seminar geeft veel praktijkvoorbeelden over het toepassen van dergelijke patronen in de dagelijkse praktijk, maar toont ook diverse zeer leerzame anti-voorbeelden. Het geeft de deelnemers een helder inzicht in de positieve bijdrage die deze technieken leveren aan projecten, het motiveert de deelnemers en biedt talrijke handvatten voor het verbeteren van de kwaliteit en onderhoudbaarheid van uw software ontwikkeling.

Bent u betrokken bij software development in .NET? **Dan mag u dit seminar niet missen!**

**Kortom: in dit praktische seminar wordt .NET in al zijn facetten en toepassingsmogelijkheden behandeld.**

### AANTREKKELIJKE KORTINGEN

Vroegboekvoordeel en korting bij meerdere deelnemers van één bedrijf! Dit seminar wordt uitgevoerd onder auspiciën van het vakblad [Software Release Magazine](#). Abonnees profiteren van korting op de deelnemersprijs. Meer informatie vindt u op [www.arrayseminars.nl](http://www.arrayseminars.nl).

DATUM	<b>2 juni 2009</b>
LOCATIE	<b>Hotel Lapershoek Hilversum</b>
TIJD	<b>Van 9.30 uur tot 17.00 uur</b>
REGISTRATIE	<b><a href="http://www.arrayseminars.nl">www.arrayseminars.nl</a></b>

- Denken in applicatie-architecturen
- Het opzetten en gebruiken van frameworks
- Het effectief opzetten van de user interface
- Realiseren van use cases in tasks
- Bouwen met factories, domeinobjecten en bedrijfsregels
- Patronen voor het ontsluiten van de back-end
- Model driven development en domain specific languages
- Deelnemers waardeerden de vorige editie met een 8!

**Array** SEMINARS



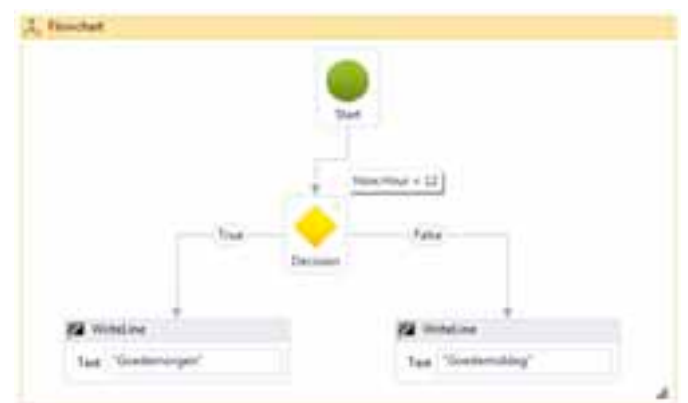
aanwezige activiteiten. Door deze activiteiten naar de designer te slepen kunnen we onze workflow wat nuttigs laten doen.

De XAML die door de designer wordt gegenereerd is uiteraard gewone XML en we kunnen dat ook als XML bekijken of bewerken. De volgende workflow XAML doet precies hetzelfde als de eerdere workflow in de designer.

```
<Activity x:Class="WorkflowConsoleApplication23.Workflow2"
  xmlns="http://schemas.microsoft.com/netfx/2009/xaml/activities"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Sequence>
    <If Condition="[System.DateTime.Now.Hour < 12]">
      <If.Then>
        <WriteLine Text="Goedemorgen" />
      </If.Then>
      <If.Else>
        <WriteLine Text="Goedemiddag" />
      </If.Else>
    </If>
  </Sequence>
</Activity>
```

LISTING 1: DE XML BRONCODE VOOR EEN WORKFLOW.

Overigens wordt de workflow nu nog door Visual Studio gecompileerd en daarna uitgevoerd. Als we dat liever iets dynamischer willen en pas tijdens het uitvoeren de XAML laden, dan is dat niet veel lastiger. De twee regels code in Listing 2 zijn hier genoeg voor.



AFBEELDING 3: EEN FLOWCHART WORKFLOW.

## Voor algemene zaken als het versturen en ontvangen van WCF-berichten, foutafhandeling of transacties zijn allemaal standaardactiviteiten aanwezig.

```
var workflow = ActivityXamlServices.Load(@"..\..\Workflow2.xaml");
WorkflowInvoker.Invoke(workflow);
```

LISTING 2: EEN WORKFLOW DYNAMISCH LADEN EN UITVOEREN.

Overigens is XAML eigenlijk niets meer dan een object instansiatie taal die in dit geval verschillende activiteiten maakt en properties zet. We kunnen dezelfde workflow dus ook gewoon in code uitschrijven zoals hieronder aangegeven.

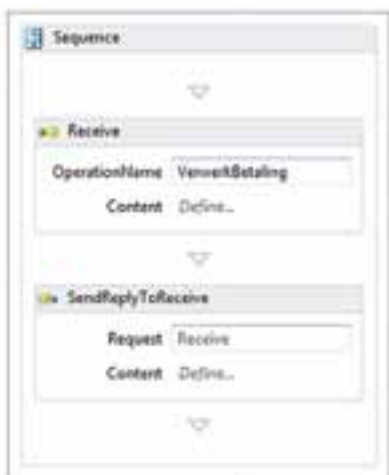
```
var workflow = new Sequence()
{
    Activities =
    {
        new If()
        {
            Condition = DateTime.Now.Hour < 12,
            Then = new WriteLine()
            {
                Text = "Goedemorgen"
            },
            Else = new WriteLine()
            {
                Text = "Goedemiddag"
            }
        }
    }
};
WorkflowInvoker.Invoke(workflow);
```

LISTING 3: DEZELFDE WORKFLOW MAAR DAN MET C# CODE GESCHREVEN.

Ongeacht welke van deze methodes we kiezen is het effect exact hetzelfde. Het is alleen een kwestie van smaak over hoe een workflow kan worden gemaakt. Overigens is de inzichtelijkheid voor analisten en power gebruikers een belangrijk voordeel van Windows Workflow Foundation en dat wordt uiteraard alleen bereikt als we de designer gebruiken.

### Flowchart workflows

De hiervoor gebruikte workflow was een zogenaamde sequentiële workflow en die zijn behoorlijk rigide, maar daardoor wel makkelijk te begrijpen in hun manier van werken. Er is ook een tweede



AFBEELDING 4: EEN WCF BERICHT ONTVANGEN MET EEN STANDAARD RECEIVE ACTIVITEIT.

stijl en dat is de flowchart. Bij een flowchart is de ontwikkelaar veel vrijer om complexe relaties tussen activiteiten te maken. Ik verwacht dan ook dat flowcharts in de praktijk veel zullen worden gebruikt. Overigens is een flowchart gewoon – net als een sequentie – een activiteit en zijn de verschillende stijlen in één workflow door elkaar te gebruiken.

### Workflow Activiteiten

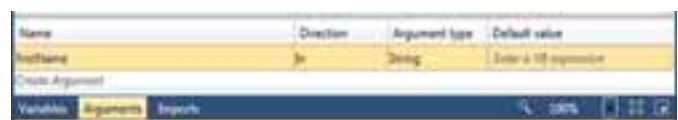
De verschillende activiteiten zijn eigenlijk de basisbouwstenen van een workflow. Standaard levert Microsoft er een flink aantal met het .NET framework mee. Een complete lijst gaat te ver maar voor algemene zaken als het versturen en ontvangen van WCF-berichten, foutafhandeling of, zowel kort- als langlopende, transacties zijn allemaal standaardactiviteiten aanwezig.

Maar de standaardactiviteiten zijn allemaal hele algemene activiteiten. Windows Workflow Foundation wordt pas echt nuttig als we een collectie van activiteiten maken die onze eigen zakelijke toepassing beschrijft. Wat voor activiteiten dit zijn is echter afhankelijk van de toepassing en dat is dan ook de reden dat Microsoft die niet meeleverd. Het is dan ook aan de ontwikkelaars die WF4 in gaan zetten om die te maken. Bij het gebruik van WF3 was het maken van activiteiten nog wel eens een lastig punt waar met veel dingen rekening moest worden gehouden. Gelukkig is dat één van de punten waar workflow sterk verbeterd is. Niet alleen zijn de meeste activiteiten veel makkelijker te maken, maar de manier waarop de runtime met onze activiteiten omgaat is zo dat we in de meeste gevallen niets expliciets hoeven te doen. De workflow runtime snapt gewoon de intentie en doet het gevraagde. Gevolg is dat sommige activiteiten van 200+ regels in WF3 terug gaan naar 20 regels code in WF4.

### Variabelen en Arguments

Een workflow, zoals hierboven, die alleen maar beslissingen kan nemen is al snel heel beperkt. We zullen dus ook met data moeten kunnen werken. De manier waarop we dat doen is door gebruik te maken van variabelen, een plaats om data op te slaan, en argumenten, de in- en uitvoer voor een activiteit. Bij het ontwikkelen van een eigen activiteit geven we op welke invoer we verwachten en wat het resultaat is. Dit doen we met behulp van argumenten. Hoe we dat doen is een beetje afhankelijk van hoe we onze activiteit ontwikkelen. Doen we dat in XAML, een workflow is eigenlijk gewoon een activiteit, dan gebeurt dit in het Argument venster van de workflow designer.

Zijn we daarentegen activiteiten in code aan het maken dan gebeurt dit door properties van het type InArgument of OutArgument te gebruiken (zie Listing 4).



AFBEELDING 5: HET WORKFLOW DESIGNER ARGUMENTS VENSTER.

```

public sealed class MyWriteLine : CodeActivity
{
    public InArgument<string> Text { get; set; }

    protected override void Execute(CodeActivityContext context)
    {
        string text = context.GetValue(this.Text);
        Console.WriteLine(text);
    }
}

```

LISTING 4: EEN ACTIVITEIT MET EEN ARGUMENT IN CODE DEFINIËREN.

## Visual Basic Expressions

Uiteraard moeten we ook nog iets met alle data in variabelen en de in- en uitvoerargumenten kunnen doen. Om dat te doen maken we gebruik van expressies binnen een workflow. Een merkwaardig ding aan expressies is dat ze altijd in Visual Basic syntax zijn, ongeacht of de rest van de code in het project in C# of Visual Basic geschreven wordt. Dit klinkt lastig, maar aangezien het gewoon om .NET expressies gaat is het verschil tussen C# en Visual Basic niet echt een groot probleem.

## Een workflow als een WCF Service

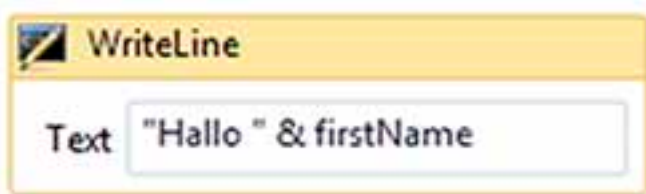
Tot nu toe hebben we een workflow gewoon met een console applicatie gestart. Dat is prima voor een kortlopende workflow, maar zodra we langlopende processen in een workflow gaan modelleren wordt dit lastiger. Een console applicatie draait namelijk niet altijd. Er zijn echter processen, zoals Windows services, die dat wel doen. Eén daarvan is Internet Information Server (IIS), een applicatie die zijn rol als webserver al lang is ontgroeid en tegenwoordig een volwaardige applicatieserver is. En door gebruik te maken van een Workflow Service Applicatie krijgen we een workflow die binnen IIS draait en waar we WCF-berichten gebruiken om met de workflow te communiceren.

## Als de AppFabric uitkomt krijgen we meer mogelijkheden om binnen IIS manager onze workflow te beheren

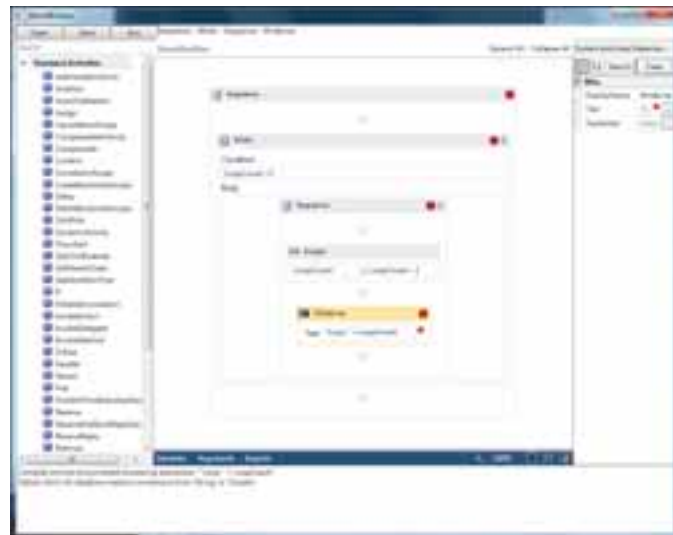
Met het huidige workflow project template is het al prima te doen maar in de toekomst, als de AppFabric uitkomt, krijgen we nog veel meer mogelijkheden om binnen de IIS manager onze workflows te beheren. Het AppFabric komt echter pas na .NET 4 zelf uit, op dit moment is er een bèta 2.

## De Windows Workflow Designer

De workflow designer die we in Visual Studio 2010 gebruiken is grotendeels een onderdeel van het .NET framework en niet van



AFBEELDING 6: EEN VISUAL BASIC EXPRESSIE OM EEN BOODSCHAP AF TE DRUKKEN.



AFBEELDING 7: DE WORKFLOW DESIGNER IN EEN EIGEN APPLICATIE.

Visual Studio 2010 zelf. Dat betekent dat we de designer ook in onze eigen applicaties kunnen gebruiken. Die applicatie moet dan wel met Windows Presentation Foundation geschreven zijn aangezien de workflow designer een, complex, WPF control is. Het gebruiken van de workflow designer maakt een hoop mogelijk, zo kunnen we de gebruikers (delen van) onze workflows aan laten passen. Het is ook mogelijk om workflow als een scripttaal in onze applicaties in te zetten. Een gebruiker kan dan met behulp van de workflow designer en door ons geleverde activiteiten bepaalde veel voorkomende acties binnen een applicatie automatiseren.

## Conclusie

Met Windows Workflow Foundation 4 is er voor de .NET ontwikkelaar weer een bruikbaar stuk gereedschap bijgekomen. Er is veel geleerd van de fouten die met de eerste versie van Windows Workflow Foundation in .NET 3 gemaakt zijn en deze versie is een stuk makkelijker te gebruiken voor de gemiddelde .NET ontwikkelaar. Zeker een product waar je als professioneel .NET ontwikkelaar kennis van moet nemen.

### Links

- WF and WCF in .NET 4  
<http://msdn.microsoft.com/en-us/netframework/aa663328.aspx>
- De WF Migration Kit  
<http://wf.codeplex.com/releases/view/41401>
- The Problem Solver blog  
<http://msmvps.com/blogs/theproblemsolver/default.aspx>

**Maurice de Beijer**, is een onafhankelijk software consultant en trainer bij DevelopMentor. Hij is te bereiken via [maurice.de.beijer@gmail.com](mailto:maurice.de.beijer@gmail.com).



A high-contrast, black and white profile photograph of a man with glasses, looking towards the left. The lighting is dramatic, highlighting the contours of his face and the texture of his hair. The background is a plain, light color.

*Ontmoet Dennis Doomen.  
Autoriteit op het gebied van  
.Net architectuur, co-presentator op  
TechED en ontwikkelaar van Fluent  
Assertions. Je leert hem kennen op  
dotnetmag.nl. De codes die Dennis  
ontwikkelt, het verhaal erachter, zijn  
informatiebronnen en meer. Want  
dotnetmag.nl biedt je alle mogelijk-  
heden om te ontwikkelen, linken, met  
dotnet'ers in Nederland te sparren en  
elkaar verder te helpen. Laat je horen  
en registreer je op dotnetmag.nl*

