

# Microsoft Office InfoPath 2003 toolkit for Visual Studio .NET

BUSINESSLOGICA PROGRAMMEREN MET VISUAL STUDIO .NET EN INFOPATH

**Bedrijven en instellingen hebben te maken met veel papieren administratie. Voor elk proces is wel een formulier te bedenken. Microsoft biedt met InfoPath de mogelijkheid deze papierstroom te reduceren en beter te beheren. Hoe je dit met de InfoPath toolkit aanpakt zien we aan de hand van een project dat is gecreëerd in Visual Studio .NET.**

Aan de orde komen de volgende onderwerpen: het starten en compileren van een InfoPath-project, de diverse functionaliteiten van Visual Studio .NET voor InfoPath en andersom, InfoPath en eventhandlers, uit welke objecten bestaat het InfoPath-formulier, en debugging en deployment van een InfoPath-formulier. Kennis van Microsoft Office InfoPath 2003, XML en Visual Studio .NET is vereist.

Microsoft Office 2003 InfoPath is een hulpmiddel voor het verwerken van formulieren. Een onderdeel van InfoPath is de editor waarmee gemakkelijk formulieren gebouwd kunnen worden. Deze formulieren kunnen daarna worden geïntegreerd in andere systemen, zoals Microsoft BizTalk Server 2004, SharePoint Portal Server 2003 of met webservices. In 2004 is door Microsoft de InfoPath 2003 toolkit voor Visual Studio .NET en InfoPath service pack 1 uitgebracht. Microsoft wil met deze toolkit het gebruik van InfoPath vergroten en vergemakkelijken. De toolkit zorgt voor een integratie met Visual Studio .NET. Bekende voordelen van Visual Studio .NET, zoals debugging en code completion, kan de ontwikkelaar nu ook toepassen in InfoPath. Vanzelf sprekend staat het gehele .NET Framework nu ter beschikking voor InfoPath-toepassingen.

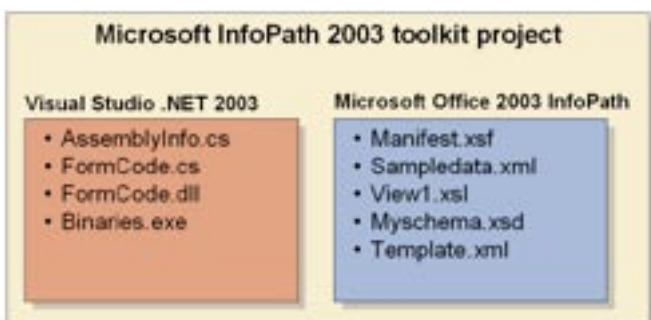
## InfoPath toolkit

De InfoPath toolkit biedt ontwikkelmogelijkheden door een combinatie tussen InfoPath en Visual Studio.NET. InfoPath gebruikt xsf-form-definities, xml-data-bestanden, xsd-schema-bestanden en xsl logica-bestanden. De InfoPath toolkit combineert deze bestanden met standaard codebestanden in Visual Studio .NET en gooit ze samen in één Visual Studio .NET-project. In afbeelding 1 zie je de schematische weergave van het InfoPath toolkit-project en de verdeling van bestanden tussen InfoPath en Visual Studio .NET.

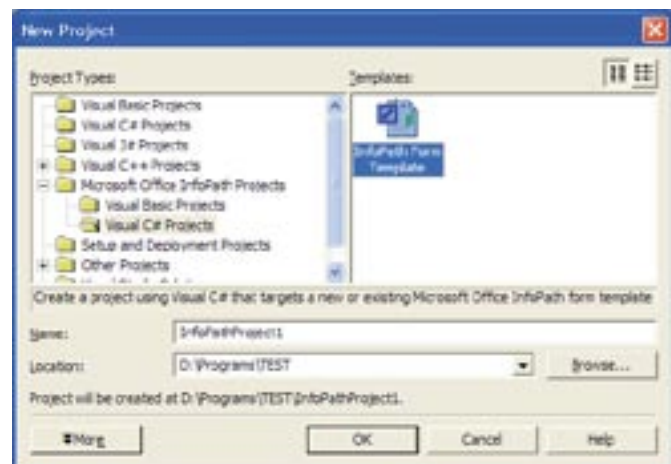
## Het starten van een InfoPath-project in Visual Studio .NET

Wanneer je de InfoPath toolkit hebt gedownload en geïnstalleerd, kun je aan de slag met het InfoPath-formulier. Afbeelding 2 toont een nieuwe InfoPath-template in Visual Studio .NET. Bij het aanmaken van het nieuwe project vraagt Visual Studio .NET of je een bestaand InfoPath-formulier wilt gebruiken voor het project of dat je een nieuw InfoPath-formulier wilt aanmaken. Automatisch wordt InfoPath opgestart en kun je het InfoPath-formulier direct gaan vormgeven. In afbeelding 3 ziet je het Visual Studio .NET-project.

In Visual Studio .NET begint het project met het geopende bestand FormCode.cs. Dit bestand dient als een zogenaamd 'code-behind'-bestand van het InfoPath-formulier. In dit codebestand wordt de class met de projectnaam gedefinieerd. Deze class is de startup-class voor het InfoPath-formulier. Hierin wordt alle code gedefinieerd die betrekking heeft op de logica van formulier. Nadat het formulier is voorzien van velden en grafische onderdelen kunnen we logica gaan coderen in Visual Studio .NET. Zoals eerder beschreven bestaat er een class met jouw projectnaam. Deze class bevat in het begin van het project maar twee methoden, de \_Startup- en de \_ShutDown-methode. De \_Startup-methode wordt getriggerd wanneer je een nieuw of bestaand formulier opent en InfoPath wordt gestart. De \_ShutDown wordt aangeroepen wanneer je het formulier of InfoPath afsluit. In codevoorbeeld 1 zie je de twee methoden.



Afbeelding 1. Het InfoPath-project



Afbeelding 2. De InfoPath-template in Visual Studio .NET

```
private XDocument thisXDocument;
private Application thisApplication;

public void _Startup(Application app, XDocument doc)
{
    thisXDocument = doc;
    thisApplication = app;

    // You can add additional initialization code here.
}

public void _Shutdown()
{
}
}

```

Codevoorbeeld 1. De \_Startup- en \_ShutDown-methoden

Wanneer het formulier wordt opgestart, krijgt de \_Startup-methode twee variabelen mee: een applicatie-variabele en een variabele van het type Xdocument. De applicatie-variabele heeft een relatie met de Microsoft Office InfoPath-applicatie, en de Xdocument-variabele heeft een relatie met het huidige formulier.

### Het definiëren van events vanuit InfoPath

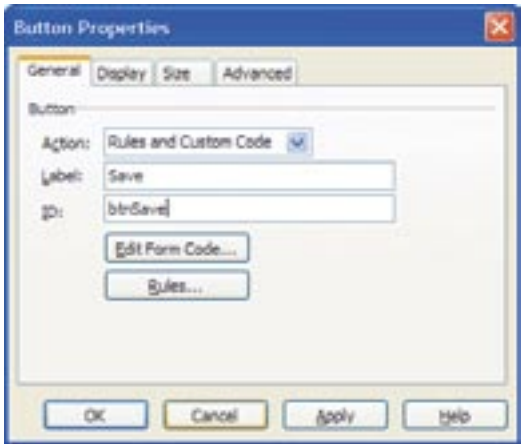
Bij andere Visual Studio .NET-projecten zoals een Windows-applicatie hebben we te maken met events. Deze events kunnen geplaatst worden op diverse controls. In InfoPath kunnen we nu ook achter diverse controls - zoals het formulier, velden en knoppen - events hangen. Deze events zijn gemakkelijk te koppelen vanuit InfoPath. In tabel 1 zie je een overzicht van een aantal events dat toegepast kan worden op een InfoPath-formulier.

Zoals je ziet zijn er verschillende events beschikbaar. De events zijn makkelijk te koppelen aan de controls via de control-properties. Wanneer we een event aan een control koppelen, wordt in Visual Studio .NET een event-methode aangemaakt, waarin je eigen logica kunt toevoegen. In afbeelding 4 zie je hoe je vanuit InfoPath een eventhandler-methode kunt zetten achter een knop.

Door op Edit Form Code te drukken wordt de methode OnClick achter de control btnSave gemaakt. In codevoorbeeld 2 zie je een voorbeeld van een eventhandler bij de knop btnSave.

### Objectmodel in InfoPath en Service Pack 1

InfoPath werkt met diverse objecten, maar de voornaamste zijn het UI-object, het Xdocument-object en het View-object. Het UI-object biedt de gebruiker een aantal methoden, zodat hij met de userinterface (in dit geval InfoPath) kan communiceren. Een voorbeeld hiervan is de Alert-methode. Deze methode zendt een bericht naar de userinterface met een gedefinieerde tekst. Met InfoPath 2003 service pack 1 is er een nieuw UI-object bijgekomen, het UI2-object. Dit object erft van het originele UI-object, maar biedt een



Afbeelding 4. Het aanmaken van een event achter een control



Afbeelding 3. Het InfoPath Visual Studio .NET-project

aantal nieuwe methoden. Er is bijvoorbeeld een methode Confirm bijgekomen, die gebruikers een keuzeberichtenbox aanbiedt in de userinterface. Om het UI-object aan te spreken, werkt het InfoPath-objectmodel met een UIObject-interface die gedefinieerd staat binnen het Xdocument-object.

Het Xdocument-object representeert het formulier wat betreft data en gedrag. Zoals beschreven kunnen we via het Xdocument de userinterface aanspreken via de UIObject-interface die gedefinieerd staat binnen de property UI. We kunnen naast de userinterface ook de onderliggende of ingevulde data aanspreken van het formulier. Dit wordt gedaan aan de hand van de DOM-property in het Xdocument-object. De DOM-property representeert een XML DOM-document met alle data van het huidige formulier. Codevoorbeeld 3 bevat een voorbeeld van het aanspreken van het UI- en het Xdocument-object.

Wat we nu zien in codevoorbeeld 3 is dat de methode VisitDate\_OnBeforeChange de data in het formulier checkt aan de hand van het XML DOM-document. Een nadeel is dat, als we velden op data willen controleren, we gebruik moeten maken van het XML DOM-document.

Het is mogelijk om een wrapper te maken, deze wijze levert je het voordeel op dat je met XPATH-expressies kunt gaan werken. Je kunt dus zeer complexe XPATH-queries op je data loslaten die complexe data structuren retourneren.

In InfoPath kun je een formulier ontwerpen met diverse views. Deze views bevatten verscheidene weergaven van de data achter het formulier. Vanuit de code kunnen we deze views aanspreken met het View-object. Het Xdocument-object heeft twee properties, de View-property die de op dat moment geselecteerde view representeert en de ViewInfos-property die de collectie bevat van alle views in het huidige formulier. In de view-collectie van het InfoPath-formulier is één enkele view de default-view. We kunnen

Event-naam	Omschrijving
OnLoad	Dit event wordt getriggerd wanneer een formulier wordt geladen.
OnSwitchView	Dit event wordt getriggerd wanneer er in het formulier wordt gewisseld tussen diverse views.
OnContextChange	Dit event wordt getriggerd nadat een OnLoad of een OnSwitchView is getriggerd en de huidige context van het formulier wijzigt.
OnSign	Dit event wordt getriggerd wanneer een formulier digitaal wordt gesigneerd.
OnClick	Dit event wordt getriggerd wanneer er bijvoorbeeld op een knop wordt gedrukt.
OnBeforeChange	Dit event wordt getriggerd wanneer er in een veld data wordt toegevoegd voordat de wijziging van de veld waarde wordt doorgevoerd.
OnValidate	Dit event wordt getriggerd wanneer er een veld wordt gevalideerd met de eventueel ingestelde validatieregels in InfoPath.
OnAfterChange	Dit event wordt getriggerd wanneer er in het veld data is toegevoegd en de waarde is doorgevoerd.

Tabel 1. Overzicht InfoPath-formulier events

```
[InfoPathEventHandler(MatchPath="btnSend",
EventType=InfoPathEventType.OnClick)]
public void btnSave_OnClick(DocActionEvent e)
{
    e.XDocument.Save();
}
}
```

Codevoorbeeld 2. Eventhandler-voorbeeld

```
[InfoPathEventHandler(MatchPath="/Customers/VisitDate",
EventType=InfoPathEventType.OnBeforeChange)]
public void VisitDate_OnBeforeChange(DataDOMEvent e)
{
    IXMLDOMNode phone =
thisXDocument.DOM.selectSingleNode(@" /Customers/PhoneDate")
    if (phone.text == "")
    {
        e.ReturnMessage = " Phone date should be filled in
before Visit date.";

        e.ReturnStatus = false;

        return;
    }

    e.ReturnStatus = true;
    return;
}
}
```

Codevoorbeeld 3. Het UI-object en het Xdocument-object

```
private void SetView(string viewName)
{
    thisXDocument.View.SwitchView(viewName);
}

private void ShowDefaultView()
{
    ViewInfos viewInfos = thisXDocument.ViewInfos;
    foreach(ViewInfo viewInfo in viewInfos)
    {
        If(viewInfo.IsDefault)
        {
            thisXDocument.UI.Alert("The default view is: " + viewInfo.Name);
        }
    }
}
}
```

Codevoorbeeld 4. Het selecteren van een andere view

bijvoorbeeld de geselecteerde view direct wijzigen in InfoPath. In codevoorbeeld 4 zie je een voorbeeld van het wijzigen van de geselecteerde view. Ook wordt een voorbeeld getoond om de default-view te laten zien.

## Het debuggen

Debuggen van een InfoPath-project gaat bijna op dezelfde manier als bij elk ander Visual Studio .NET-project. Wanneer het project gestart wordt, zal in InfoPath een preview van het formulier weer-geven. Binnen Visual Studio kunnen we net als bij andere projec-ten de diverse functionaliteiten van het formulier debuggen.

## Deployment

Het deployen van een InfoPath-project doen we nadat we een release-versie hebben gebouwd. Vervolgens kunnen we kijken naar

de bin-directory van de release-map van het project. Hierin staat de assembly van het project en een xsn InfoPath-template. De xsn-template kunnen we vanaf nu gebruiken in InfoPath. De dll bevat de .NET assembly en moet mee gedeploteerd worden met de xsn. Door middel van een verwijzing naar deze assembly kan elk InfoPath-formulier de template aanroepen.

## Tot slot

Naast de genoemde functionaliteiten zijn er nog vele andere moge-lijkheden met de InfoPath 2003 toolkit for Visual Studio .NET. Digitale handtekeningen en externe databronnen aanspreken van-uit de code zijn functionaliteiten die deze toolkit nog meer biedt. InfoPath was eerst een product voor de Office-gebruiker en niet zozeer voor de .NET- ontwikkelaar. Nu echter kunnen we het product vanuit een volstrekt ander oogpunt bekijken. Met een complete integratie in Visual Studio .NET is de InfoPath 2003 tool-kit for Visual Studio .NET een nieuwe uitdaging. InfoPath 2003 service pack 1, dat het objectmodel van InfoPath flink uitbreidt, biedt nieuwe functionaliteiten voor de gebruiker en voor de ont-wikkelaar.

**Willem Boeré** is werkzaam bij Avana de Nederland

### Nuttige internetadressen

<http://msdn.microsoft.com/office/infopath>  
InfoPath 2003 for Visual Studio .NET SDK

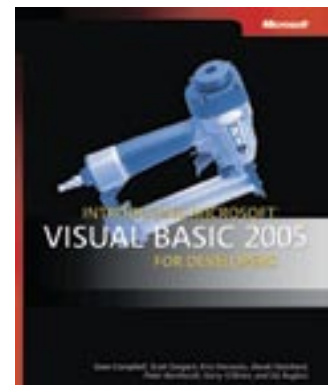
( advertentie Microsoft Press )



### Developing Solutions with Microsoft InfoPath

ISBN: 0-7356-2116-0

Auteurs: Matthew Blain, Patrick Halstead en Vani Mandava-Teredesai



### Introducing Microsoft Visual Basic 2005 for Developers

ISBN: 0-7356-2058-X

Auteurs: Campbell, Swigart, Horrocks, Hatchard, Bernhardt, O'Brien, Rugless