

# Information Bridge Framework 1.5

## EENVOUDIG EEN KOPPELING NAAR LOB-TOEPASSINGEN ONTWIKKELEN MET BEHULP VAN IBF 1.5

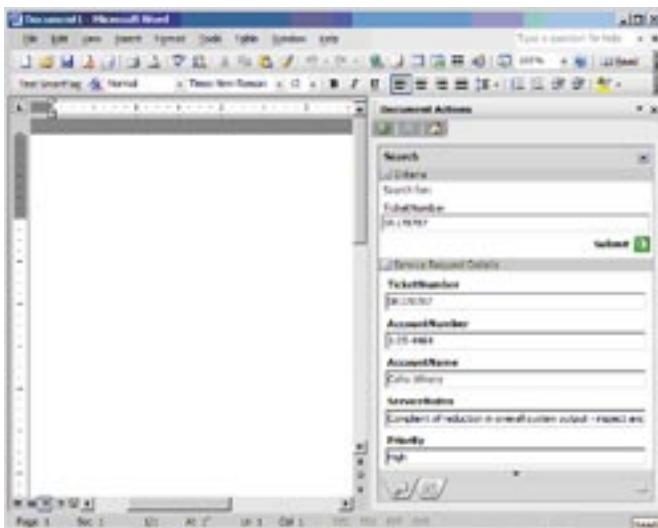
Information Bridge Framework (IBF) is een framework waarmee vanuit Microsoft Office LOB-applicaties kunnen worden benaderd. Over versie 1.0 van dit product is in een eerder .NET Magazine al een artikel geschreven waarin de basisprincipes van IBF werden uiteengezet. In het huidige artikel laten we zien hoe je met behulp van een aantal wizards en de resourcekit uit versie 1.5 snel een eenvoudige IBF-oplossing kunt ontwikkelen.

De ontwikkelomgeving is in versie 1.5 van het Information Bridge Framework sterk verbeterd. Er is een groot aantal wizards beschikbaar waarmee de verschillende onderdelen van een IBF-oplossing snel gebouwd kunnen worden, waardoor het een echte RAD-tool is geworden. Bovendien is een debugger beschikbaar die zowel de assembly-code als de IBF-events debugt. Dit maakt het ontwikkelen van IBF-oplossingen eenvoudiger, maar maakt het proces voor de ontwikkelaar ook beter inzichtelijk. Ook voor het maken van XSL-files (nodig om transformaties te maken voor bijvoorbeeld views) is een grafische interface – BizTalk Mapper – beschikbaar.

### Onderdelen IBF-oplossing

Een IBF-oplossing bestaat uit de volgende onderdelen:

1. een **webservice** die data beschikbaar stelt en voldoet aan IBF-richtlijnen
2. de **IBF-metadata service**: deze staat geïnstalleerd op de IBF-server en stelt de IBF-functionaliteit beschikbaar aan de client
3. de **IBF-user interface** (het taakvenster in Microsoft Office): de user control of het html-bestand waarin de gegevens getoond worden
4. (eventueel) de **smart tag**-component of het 'attached schema'-document

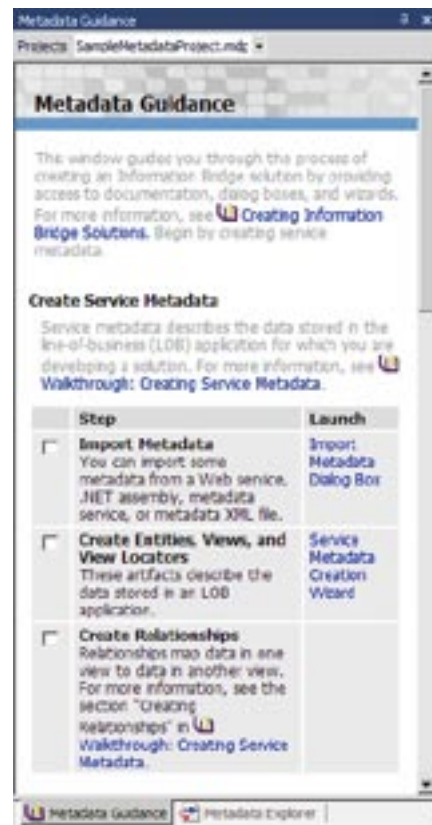


Afbeelding 1. IBF-oplossing

Gebruikers van IBF-oplossingen blijven gewoon werken binnen hun Office-applicatie en merken dus niets van de verschillende onderdelen waaruit de IBF-oplossing bestaat. Het voordeel voor gebruikers is dat ze hun Office-applicatie niet hoeven te verlaten om gegevens uit andere systemen op te zoeken of in hun document op te nemen. Het voordeel voor ontwikkelaars is dat het met behulp van het IBF-framework eenvoudiger is om smart documents te ontwikkelen, en desgewenst uit te breiden. We zullen de onderdelen van een IBF-oplossing kort behandelen voordat we laten zien hoe je een IBF-applicatie bouwt.

### Webservice

De koppeling naar line-of-business-toepassingen wordt gemaakt met behulp van webservices. Voor de eisen waaraan deze web-



Afbeelding 2. Metadata Guidance

```

<?xml version='1.0' encoding='utf-8' ?>
<!-- definitions -->
<!-- schema -->
<!-- service -->
<!-- port -->
<!-- operation -->
<!-- transformation -->
<!-- schema -->
<!-- translation -->
<!-- group -->
<!-- custom operation type -->

```

Codevoorbeeld 1. Fragmenten uit de WSDL van de webservice die we gebruiken in de IBF-oplossing

service moet voldoen, verwijzen we naar de Resource Kit en het eerdere artikel in het .NET Magazine over IBF; zie Nuttige Internetadressen. In dit artikel gebruiken we één van de webservices uit de IBF Resource Kit.

## Metadata-service

De metadata-service is de kern van de IBF-oplossing; zie afbeelding 1. Deze metadata-service wordt ontwikkeld in Visual Studio, en vervolgens naar de server gepubliceerd.

## User interface

De IBF-user interface is een speciale pane in Office. In alle Office-applicaties wordt deze zelfde pane getoond. IBF versie 1.5 ondersteunt naast Microsoft Word, Excel en Outlook ook InfoPath, en zelfs Internet Explorer. Je kunt links tonen die het IBF-pane activeren. In Outlook wordt de IBF-pane getoond wanneer Word als e-mail editor wordt gebruikt. De IBF-task pane bestaat uit twee verschillende onderdelen: de **Search**-pane en een **Custom**-pane. De Search-pane kan door gebruikers worden opgeroepen via het menu in de actieve Office-applicatie (Tools – Information Bridge). Hiervoor hoeft het geopende document dus niet gebaseerd te zijn op een speciale template of gekoppeld te zijn aan een schema. De Search-pane wordt gebruikt om gegevens uit de LOB-applicatie op te zoeken. Let op: naast de IBF-search pane blijft het Office Research-pane beschikbaar. De Custom-pane kan door de gebruiker worden geactiveerd via smart tags, of het wordt automatisch getoond via 'attached schema'-documenten. Een attached schema-document gebruik je bijvoorbeeld om een klantoverzicht te tonen wanneer de gebruiker ergens een klantnaam in moet vullen. De Custom-pane wordt dus op dezelfde manier gebruikt als in 'normale' smart documents.

## Een IBF-oplossing ontwikkelen

De Microsoft Office Information Bridge Framework Metadata Designer is een add-in in Visual Studio waarmee de IBF-oplossingen worden ontwikkeld. Deze wordt alleen op de pc van de ontwikkelaar geïnstalleerd. Naast een overzicht van de metadata in de Metadata Explorer is een groot aantal wizards beschikbaar via het Metadata Guidance-venster. (Te tonen via View – Other Windows – Metadata Guidance). Zie afbeelding 2.

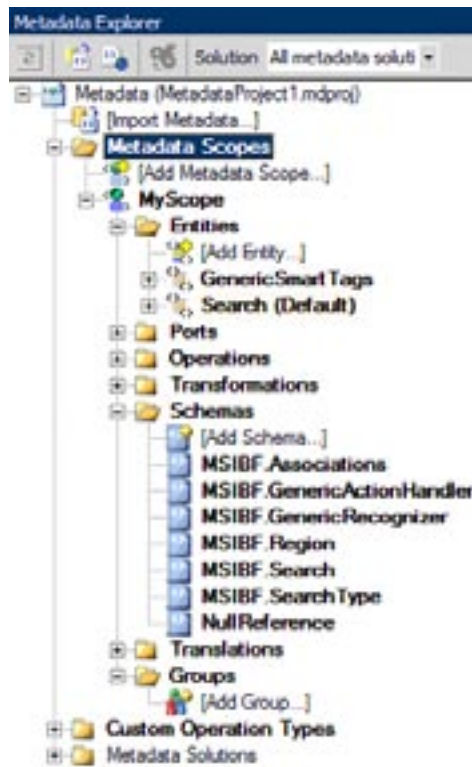
## Scenario demo-applicatie

We zullen een eenvoudige toepassing van de Search-pane en de Custom-pane laten zien. De Search-pane gebruiken we om gegevens over een ServiceRequest op te halen. De Custom-pane gebruiken we om via een smart tag detailgegevens over een ServiceRequest te tonen wanneer een gebruiker het unieke nummer van dit ServiceRequest in Word of een andere Office-applicatie typt.

## Een demo-applicatie ontwikkelen

Bij het ontwikkelen van de applicatie houden we de volgorde aan:

1. de **webservice**
2. de **IBF metadata-service**
3. de **IBF user interface**



Afbeelding 3. De Metadata Explorer

4. de **smart tag**-component of het 'attached schema'-document  
Omdat het ontwikkelen van een Research-pane als user interface het eenvoudigst is, beginnen we hiermee. Vervolgens leggen we uit hoe je de functionaliteit aan smart documents beschikbaar kunt stellen. De ruimte in dit artikel laat helaas niet toe om dit in detail toe te lichten.

## 1. Webservice

We gebruiken de webservice uit de Sample Solution uit de IBF-resource kit (<http://<Root>/informationbridge/service.asmx>). De GetServiceRequest-operation van deze webservice haalt met behulp van een TicketNumber de ServiceRequest-gegevens uit de bijgeleverde IBFSampleData-database. We zullen het ontwikkelen van de webservice hier niet uitgebreid behandelen. Het belangrijkste is dat de XML-schema's van de objecten die we opvragen, overeenkomen met de schema's in onze metadata-service. Hoe de objecten worden geserialiseerd, en dus hoe deze XML eruit ziet, kun je in je .NET-classes regelen met behulp van XML-attributen. Het is handig om de namespaces te controleren met behulp van de webservice WSDL; zie Codevoorbeeld 1. Zo zien we dat de GetServiceRequest-method inderdaad een TicketNumber heeft als input, en een ServiceRequest teruggeeft.

## 2. De IBF-metadata-service

Het echte IBF-ontwikkelwerk begint met het aanmaken van een nieuw Information Bridge Metadata Project in Visual Studio. De Resource Kit bevat een aantal templates (XML-files) voor IBF-oplossingen die de meest gebruikte elementen al bevatten. Zo kunnen we onder andere Smart Tag-functionaliteit snel later toevoegen, en hoeven niet alles zelf vanaf scratch op te bouwen. We importeren Microsoft.InformationBridge.Framework.UI.InformationBridgeScope.Metadata.xml met behulp van de **Import Metadata Dialog Box** uit de Metadata Guidance, en we geven de Metadata Scope een geschikte naam; zie afbeelding 3.

Nu hebben we het raamwerk van de metadata klaargezet. Je ziet dat de metadata georganiseerd is in Scopes en Solutions. Voor elke oplossing die je bouwt – bijvoorbeeld voor elke applicatie die je via IBF wilt ontsluiten – kun je een aparte Scope gebruiken. Maar het

is niet verplicht. De scopenaam InformationBridge wordt gebruikt voor generieke smart tag-oplossingen. Hier komen we later in het artikel op terug. Tabel 1 geeft een overzicht van de belangrijkste onderdelen van een IBF-oplossing.

### Tip

Volg de volgende stappen om een scope te verwijderen van de metadata-service

- Maak een **nieuw** IBF-project aan
- Importeer de metadata-scopes van de server (kies 'Metadata service' als import location, en selecteer de scope die je wilt verwijderen)
- Verwijder de scopes in het metadata-venster
- Kies voor het verwijderen van 'Metadata project and metadata service'
- Publiceer de metadata

De volgende stap is het aanmaken van de Entities, en bijbehorende Views. Wanneer de webservice die je gebruikt voldoet aan de IBF-richtlijnen, kun je deze gebruiken om de Entities automatisch te laten genereren. Hiervoor moeten we eerst de webservice 'importeren'; zie afbeelding 4. Dat doen we door opnieuw de **Import Metadata Dialog Box** uit de Metadata Guidance te selecteren. We kiezen voor Import Metadata from Webservice, en selecteren vervolgens GetServiceRequest. In de Metadata is nu de Operation Soap.GetServiceRequest aangeemaakt, de schema's voor ServiceRequest en TicketNumber, en de benodigde Ports.

Vervolgens gebruiken we de Service Metadata Creation Wizard om Entities aan te maken. We maken een ServiceRequest-entity aan op basis van het schema Xsd.ServiceRequest:

1. Selecteer de Service Metadata Creation Wizard (in het Metadata Guidance-venster)
2. Selecteer MyScope
3. Maak de entity ServiceRequest aan en selecteer Xsd. ServiceRequest
4. Vul de naam van de view in (bijvoorbeeld **ServiceRequestDetails**)
5. Selecteer Soap.GetServiceRequest en voltooi de wizard

Entity	Het object, de entiteit waarvan je de gegevens wilt tonen. Bijvoorbeeld Customer, of ServiceRequest
View	Definitie van de weergave van de gegevens. Bijvoorbeeld CustomerDetails of AllCustomers. Een View is altijd gerelateerd aan een Entity en wordt aangeroepen met behulp van een View Locator. In ons voorbeeld zal het TicketNumber de View Locator zijn voor een ServiceRequest-view.
Action	De actie die wordt uitgevoerd. Bijvoorbeeld het zoeken van klantgegevens of het tonen van klantgegevens.
Port	Poorten naar webservices, xsd-schema's, (smart tag) assemblies, etc.
Operation	Operations zijn de methoden die door IBF aangeroepen worden. Naast Operations zijn er ook Actions. Een Action wordt door de eindgebruiker uitgevoerd en kan uit meer Operations bestaan.
Transformation	Transformations – Transformaties worden gebruikt om data te transformeren. Transformaties kunnen zijn gedefinieerd door XSL, XPath of regular expressions, maar je kunt ook componenten aanroepen. Een Transformation is een voorbeeld van een View Locator
Schemas	Definities van alle entiteiten

Tabel 1. Elementen van een IBF-oplossing

### 3. De IBF-user interface

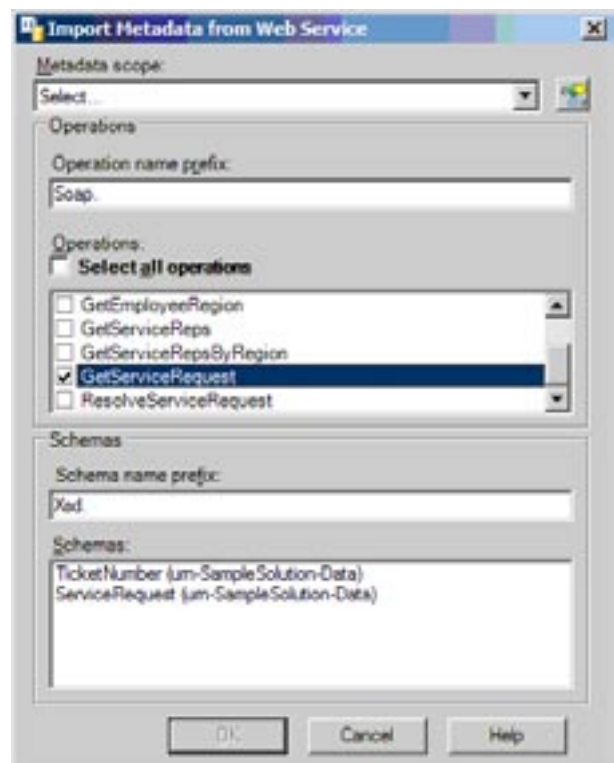
Nu kunnen we aan de user interface beginnen. Met behulp van de **Region Creation Wizard** maken we een Region aan waarin de ServiceRequest-data getoond moeten worden.

1. Selecteer de Region Creation Wizard (in het Metadata Guidance-venster)
2. Maak een nieuwe Metadata-solution aan (MySampleMetadata-Solution)
3. Selecteer MyScope en vervolgens de entity ServiceRequest en de view ServiceRequestView
4. Maak de actie **ShowServiceRequestDetails** aan. (Vul de naam, header en description in). Deze actie hebben we straks nodig om de region in Microsoft Office te activeren
5. Maak een Windows Forms-based region aan (**ServiceRequestDetails**), in een nieuw project. We kiezen voor een Forms-based region omdat dit meer mogelijkheden biedt om de functionaliteit uit te breiden dan HTML
6. Voltooi de wizard

De Region Creation Wizard maakt een compleet nieuw C#-project aan, waarin een class beschikbaar is die de ServiceRequest-gegevens toont in de gewenste weergave. Het enige dat we handmatig moeten doen, is het aanmaken van een strong named key (typ hiervoor **sn -k keyfilename** in the Visual Studio 2003 command prompt) en deze key aan de assembly toevoegen. Dit maakt het mogelijk de assembly door de client te laten trusten. En dat is belangrijk wanneer we de oplossing opleveren.

### Opmerking

Wanneer je tijdens het ontwikkelen in je IBF-oplossing verwijzingen maakt naar assemblies – bijvoorbeeld de assembly waarin de Region-controls staan! – worden de locaties van deze assemblies opgenomen in de metadata. Deze locatie moet dus op de client ook beschikbaar zijn. Zorg dus dat de assemblies met een setup op de client worden geïnstalleerd, of dat ze via een http-adres beschikbaar zijn.



Afbeelding 4. Webservice-metadata importeren

We hebben nu de user interface gebouwd, en een Action aangeemaakt om de pane te tonen. Voordat we gaan bepalen wanneer deze Action wordt uitgevoerd, kunnen we de Action al testen. Hiervoor navigeren we in de Metadata Explorer naar de Action (My Scope – Entities – ServiceRequest – Views – ServiceRequest-Details – Actions – ShowServiceRequest). In het snelmenu onder de rechtermuisknop kiezen we voor Build and Execute. Vervolgens moeten we een View-locator aangeven: in ons geval dus het Ticket-Number van de ServiceRequest die we willen tonen. Die XML ziet er als volgt uit:

```
<ns0:TicketNumber Id="SR178787" xmlns:ns0="urn-SampleSolution-Data" />
```

Codevoorbeeld 2. ServiceRequest View Locator

Wanneer we op OK klikken wordt de IBF-pane met ServiceRequest-informatie getoond. We hebben dus nog geen Office-context, maar kunnen al wel controleren of de oplossing werkt. Wanneer je de Action opent om deze te testen, zie je dat de Action gekoppeld is aan de Operation die de Region toont. Op deze plaats kun je ook extra functionaliteit aan de Action toevoegen door extra Operations toe te voegen. Wanneer je bijvoorbeeld meer Regions wilt tonen – bijvoorbeeld een Customer met al zijn ServiceRequests – voeg je de bijbehorende Operations aan de Action toe. Met de Action-property SequentialExecution kun je aangeven of de acties sequentieel – en dus altijd in dezelfde volgorde – moeten worden uitgevoerd.

## IBF Search-functionaliteit toevoegen

We gaan nu de Region die we hebben aangemaakt in de IBF Search-pane tonen. Een gebruiker kan dan een TicketNumber invullen in de IBF Search-pane, waarna vervolgens de bijbehorende ServiceRequest in ons Region-control wordt getoond. Om de benodigde metadata aan te maken gebruiken we de **Search Definition Wizard**. Uiteraard selecteren we weer onze scope, entity, view en view locator. Vervolgens houden we voor de meeste stappen de default-waarden aan die automatisch worden ingevuld. Handmatig vullen we alleen de Search Category in: de naam van de zoekactie. Vervolgens geven we aan dat het TicketNumber ID het zoekcriterium vormt. De XPath-query hiervoor kunnen we met de user interface laten genereren; zie afbeelding 5.

```
Dim strName As String
Dim serviceType As Type
Dim hostAppType As Type
Dim objDocumentService As IDocumentService
serviceType = GetType(IDocumentService)
objDocumentService = regionFrameProxy.Host.Mediator.GetService(serviceType)
hostAppType = objDocumentService.Application.GetType()
Select Case CType(objDocumentService.Document.GetType(), _
System.Type).FullName
Case "Microsoft.Office.Interop.Word.DocumentClass"
Dim wordApp As Microsoft.Office.Interop.Word.Application
Dim wordDocument As Microsoft.Office.Interop.Word.Document
Dim wordSelection As Microsoft.Office.Interop.Word.Selection
wordApp = objDocumentService.Application
wordDocument = objDocumentService.Document
wordSelection = wordApp.Selection
wordSelection.TypeText (StrText)
Case "Microsoft.Office.Interop.Excel.WorkbookClass"
Dim excApp As Microsoft.Office.Interop.Excel.Application
Dim excWorkbook As Microsoft.Office.Interop.Excel.Workbook
Dim excRange As Microsoft.Office.Interop.Excel.Range
excApp = objDocumentService.Application
excWorkbook = objDocumentService.Document
excRange = excApp.Selection
excRange.Value2 = StrText
End Select
```

Codevoorbeeld 3. Vanuit de IBF-pane tekst invoegen in het Word of Excel-document

Overigens wordt de Search-pane automatisch uitgebreid wanneer je verscheidene zoekacties (Search Categories) hebt geconfigureerd. Er wordt dan een keuzelijst met zoekopties aan de gebruiker getoond. We testen de oplossing door de solution te starten, en in Word te kiezen voor Tools – Information Bridge. Het resultaat wordt getoond in afbeelding 1.

## IBF-pane aanpassen

Nu de IBF Search-pane werkt, willen we de functionaliteit van de Region-control natuurlijk uitbreiden. We kunnen bijvoorbeeld een button toevoegen om de ServiceRequest-informatie in het document te plaatsen. Codevoorbeeld 3 kun je gebruiken om vanuit een IBF-pane tekst in een Word- of Excel-document in te voegen. De code in het voorbeeld is geschreven in Visual Basic .NET; de C#- versie is beschikbaar in de Resource Kit.

Belangrijke aandachtspunten:

- Voor elke Office-applicatie is een apart objectmodel beschikbaar.
- Zet een referentie naar de assemblies van de applicaties die je wilt ondersteunen. In dit voorbeeld dus Microsoft.Office.Interop.Word en Microsoft.Office.Interop.Excel.

## Debuggen

Stel de **Metadata DebuggerMode**-property van het project in op **DebuggerMode\_Managed** om de .NET-code te debuggen. Wanneer je de reference-pane tijdens het ontwikkelen ook in Excel wilt testen, kun je in de Project Properties aangeven dat Excel wordt opgestart bij het opstarten van de solution; zie afbeelding 6.

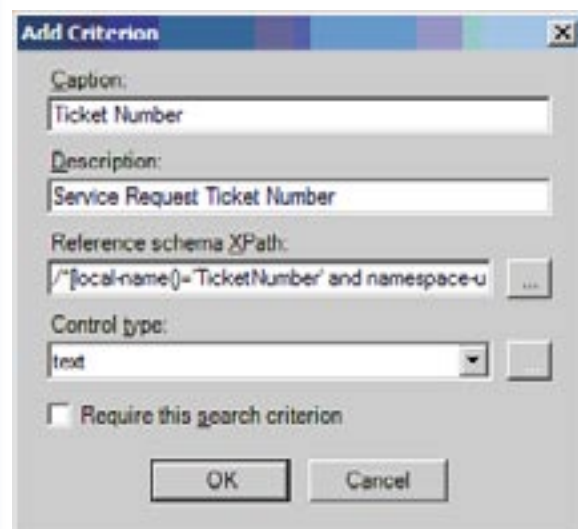
## 4. De smart tag-component of het ‘attached schema’-document

Het ontwikkelen van een smart tag of een smart document is iets complexer dan het bouwen van de Search-functionaliteit.

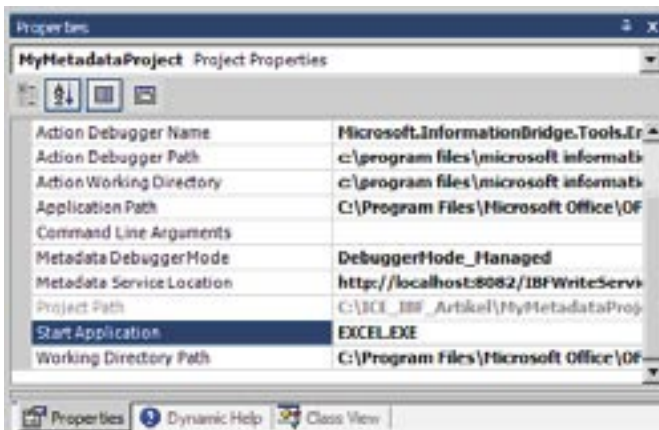
### Smart tag

Voor een smart tag is een recognizer-class nodig, die niet door de wizards kan worden gegenereerd. Deze recognizer-class werkt op dezelfde manier als de recognizer die voor ‘reguliere’ smart tags gebruikt wordt, en erft van Microsoft.InformationBridge.Framework.UI.Interop.IRecognizer. Met behulp van de Recognize-method regel je welke tekst door de smart tag wordt herkend. CommitSmartTag is de method die de smart tag activeert. Hierin geef je ook aan welke Action moet worden uitgevoerd – bijvoorbeeld een View tonen met een bepaalde View Locator; zie codevoorbeeld 4.

De sample solution bevat een duidelijk codevoorbeeld. In dit artikel laten we alleen zien hoe je de metadata zo configureert dat deze recognizer wordt geactiveerd. We zullen het ontwikkelen van de



Afbeelding 5. Zoekcriterium aangeven in de Search Definition Wizard



Afbeelding 6. IBF-oplossing testen in Excel

class zelf niet behandelen. De metadata scope met de naam **InformationBridge** is gereserveerd voor smart tags die in alle documenten beschikbaar zijn (dus niet alleen als een document aan een bepaald schema is gekoppeld). We importeren dus opnieuw het XML-bestand (Microsoft.InformationBridge.Framework.UI.InformationBridgeScope.Metadata.xml) in onze metadata en hernoemen de Scope dit keer niet.

Volg de volgende stappen om een IBF SmartTag Recognizer te activeren:

1. Configureer de **GenericSmartTags** Port:  
De Location-property moet gelijk zijn aan de locatie van de smart tag assembly
2. Configureer de **GenericRecognizer** Port in de Information Bridge-scope (dus niet in de Scope van je eigen solution)  
Stel de Data-property van de Recognizer in, zodat deze jouw recognizer-class gebruikt. Zie codevoorbeeld 5.
3. Maak een Recognizer-actie aan.  
Navigeer hiervoor in de InformationBridge Scope naar GenericSmartTags – Views – GenericRecognizers. Het rechtermuis-knop-snelmenu bevat de optie Add Action.  
Vul dezelfde naam in als in de XML van de GenericRecognizer, kies als Type EnterContext, en als Parameterschema MSIBF: GenericRecognizer.
4. Configureer de actie zodat deze de juiste Operations uitvoert: LoadRecognizer en ActivateRecognizer
5. Configureer beide acties: kies Data Source = Action Parameters en Transformation Instance = GenericRecognizer

## Smart Document

De ruimte laat het helaas niet toe om het ontwikkelen van een IBF-smart document hier in detail te beschrijven. Het belangrijkste dat je moet weten is het volgende: wanneer een Word-document is gekoppeld aan een schema met dezelfde naam als een IBF Scope, dan zal deze IBF-functionaliteit automatisch geactiveerd worden. Wanneer de cursor in een XML-element staat met dezelfde naam als een Entity uit IBF wordt de default-View van deze Entity getoond. Op deze manier kan zowel documentspecifieke als contextspecifieke informatie in de task-pane worden getoond. De IBF-client is het enige dat op de client geïnstalleerd hoeft te worden. Wanneer een nieuwe smart tag is ontwikkeld, moet je deze alleen aan de metadata op de IBF-server toevoegen. Dit is dus een snelle en relatief eenvoudige manier om smart documents te ontwikkelen en te deployen.

## Sterk verbeterd

Ten opzichte van het Information Bridge Framework 1.0 is de ontwikkelomgeving voor IBF 1.5 sterk verbeterd. Met behulp van een groot aantal wizards kan snel een IBF-oplossing worden ontwikkeld. Wanneer LOB-oplossingen via IBF worden ontsloten in Office is het belangrijk dat eerst wordt nagedacht over het ontwerp

```
<ContextInformation MetadataScopeName="MyScope"
  EntityName="ServiceRequest"
  ViewName="ServiceRequestDetails"
  ReferenceSchemaName="Xsd. TicketNumber (urn-SampleSolution-Data)"
  xmlns="http://schemas.microsoft.com/InformationBridge/2004/
    ContextInformation">
  <Reference>
    <ns0:TicketNumber Id="SR178787" xmlns:ns0="urn-SampleSolution-Data" />
  </Reference>
</ContextInformation>
```

Codevoorbeeld 4. Smart tag-context voor tonen van een ServiceRequestDetails-view

```
<msibf:GenericRecognizer Id="ServiceRequestRecognizer"
  TypeName="MyNamespace.ServiceRequestRecognizer"
  xmlns:msibf="http://schemas.microsoft.com/InformationBridge/2004">
  <YourInitializationData
    xmlns="http://schemas.microsoft.com/InformationBridge/2004/Metadata" />
</msibf:GenericRecognizer>
```

Codevoorbeeld 5. Data-property van de Recognizer

van de webservices. Wanneer deze basis goed is, is het betrekkelijk eenvoudig om de gegevens in Microsoft Office te tonen. Dankzij het generieke mechanisme voor smart tags en regions kan veel sneller functionaliteit aan Office worden toegevoegd dan in de 'klassieke' smart documents. Het belangrijkste verschil tussen de IBF-smart tag en de smart tag uit Office is dat de assembly nu aangeroepen wordt door IBF in plaats van door Office. Hierdoor hoeft op de client alleen de IBF-client te worden geïnstalleerd.

**Ingeborg Struijk** is Development Consultant bij Microsoft Services. Haar e-mailadres is [ingebors@microsoft.com](mailto:ingebors@microsoft.com)

### Nuttige internetadressen

- e-book over IBF: [www.bluevisionsoftware.com/WebSite/BookInfo.aspx?ID=15](http://www.bluevisionsoftware.com/WebSite/BookInfo.aspx?ID=15)
- Walkthrough op internet, deel 1: [www.dotnet-news.com/lien.aspx?ID=11999](http://www.dotnet-news.com/lien.aspx?ID=11999)
- deel 2: [www.codeproject.com/useritems/IBFWalkthroughPart2.asp?msg=1084712](http://www.codeproject.com/useritems/IBFWalkthroughPart2.asp?msg=1084712)
- Artikel in .NET Magazine #6. Het Information Bridge Framework 1.0. De brug van Microsoft Office 2003 naar line-of-business-toepassingen, door: Wim Uyttersprot, Patrick Tisseghem, Gert Servranckx, U2U: [www.microsoft.nl/netmagazine6](http://www.microsoft.nl/netmagazine6)
- IBF downloads en Resource Kit: <http://msdn.microsoft.com/office/understanding/ibframework/downloads/>
- MSDN Webcast: Introduction to Microsoft Office Information Bridge Framework <http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?EventID=1032266518>

( advertentie Microsoft Press )



**Customizing the Microsoft® .NET Framework Common Language Runtime**  
ISBN: 0-7356-1988-3  
Auteur: Steven Pratschner  
Pagina's: 400