

A large photograph of the Golden Gate Bridge in San Francisco, viewed from a low angle looking down the length of the bridge towards the other tower. The sky is blue with light clouds, and the water is visible in the distance.

# Brücken bauen mit Metadaten

## Das Microsoft Information Bridge Framework 1.5 bringt Unternehmensdaten über Web Services zu Office 2003

von Marcel Gnoth



Auch wenn der moderne „Information Worker“ in der Regel virtuos mit Excel, Word oder Outlook umgehen kann, die in SAP oder anderen Unternehmensanwendungen enthaltenen Geschäftsdaten müssen oft noch in einer separaten Anwendung lokalisiert und per Copy & Paste übernommen werden. Mit dem Information Bridge Framework, kurz IBF, erhalten .NET-Entwickler die Gelegenheit, Unternehmensdaten „intelligent“ in eine Office-2003-Anwendung zu übernehmen. Das Entwicklungswerkzeug ist Visual Studio .NET 2003, das durch das IBF um eine entsprechende Vorlage und einen Metadateneditor erweitert wird.

Zwei Dinge gleich vorweg: Hinter dem IBF steckt ein anspruchsvolles Konzept und es ist keine Anwendung, die man herunterlädt, installiert und sofort anfangen kann,

### kurz & bündig

#### Inhalt

Das IBF ist ein Framework für Entwickler, das die durch Metadaten gesteuerte Übernahme von Unternehmensdaten vereinfacht

#### Zusammenfassung

Das IBF ist eine Möglichkeit, um Unternehmensdaten, etwa aus einem CRM, kontextsensitiv und vor allem benutzerfreundlich in einem Office-Dokument oder einer Webseite zur Verfügung zu stellen

mit ihr zu arbeiten. Das IBF ist daher weder etwas für „Schnell-Mal-Eben“-Lösungen, noch kommt es für die Abteilungsebene oder gar den „Heimgebrauch“ in Frage. Es handelt sich, der Name deutet es bereits an, um ein Set aus Tools, Schnittstellen und Spezifikationen, mit deren Hilfe Entwickler mit Visual Studio .NET Office-Erweiterungen erstellen, die, durch Metadaten gesteuert, per Web Services Daten von einer beliebigen Datenquelle abrufen und im Aufgabenbereich einer Office-Anwendung zur Verfügung stellen. Microsoft hat das IBF im Mai 2004 angekündigt, die Version 1.0 kurz darauf fertig- und Anfang 2005 eine Version 1.5

zur Verfügung gestellt, die neben einem Debugger unter anderem auch eine auf dem BizTalk Server 2004 XSL Mapper basierende XSL-Transformation umfasst und als Client zusätzlich InfoPath und den Internet Explorer unterstützt.

Das IBF soll für jene Aufgabenstellungen eine Lösung sein, bei denen es darum geht, Unternehmensdaten in ein Office-Dokument zu übernehmen, und bei denen nicht nur die Rohdaten, sondern auch der Kontext der Daten übernommen werden soll. Die Unternehmensanwendungen werden im modernen Fachenglisch als *Line of Business (LoB)* bezeichnet. Dazu zählen nicht nur bekannte Anwendungen wie SAP,

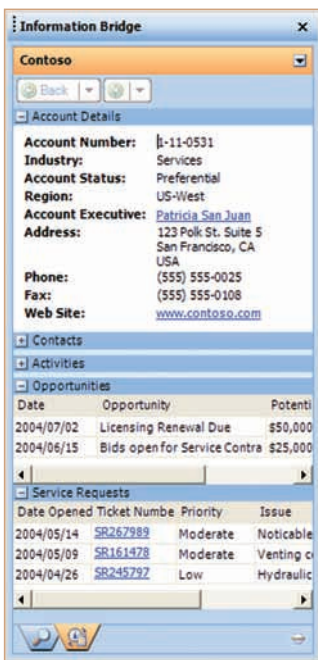


Abb 1: Der IBF-Aufgabenbereich stellt kontextbezogene Informationen zu einem Kunden dar

nummer alle gewünschten Daten zur Verfügung stellt.

### Im Mittelpunkt steht der Aufgabenbereich

Eine vielseitige Neuerung von Office 2003 ist der Aufgabenbereich, der über STRG+F1 geöffnet wird, und über den eine Anwendung bestimmte Zusatzfunktionen wie Recherche, Arbeitsbereich und Formatierungen zur Verfügung stellt. Dieser Bereich kann um beliebige andere Funktionalitäten erweitert werden. Das Besondere am Aufgabenbereich ist, dass er kontextsensitiv eingesetzt werden kann. So ist es üblich, dass ein Smart Tag in einem Dokument bestimmte Bereiche, wie eine Kunden- oder Service-Requestnummer erkennt und im Aufgabenbereich automatisch passende Informationen über ganz individuelle Steuerelemente anzeigt, mit deren Hilfe sich zum Beispiel Detailinformationen abrufen lassen. So wird der Aufgabenbereich zur Informationsschaltzentrale. Abbildung 1 zeigt einen Aufgabenbereich, der die Kundeninformationen aus der fiktiven „Microsoft-Demofirma“ Contoso anzeigt.

Sondern auch spezifische Firmenlösungen. Sind zum Beispiel Kunden-, Bestell- und Artikelinformationen in unterschiedlichen Systemen abgelegt, müssen diese Informationen für ein Angebot aus diesen Systemen zusammengetragen und in das Word- oder Excel-Dokument übernommen werden. Genau diesen, im Grunde alltäglichen Herausforderungen, stellt sich das IBF. Anstatt die Rohdaten aus verschiedenen Anwendungen per Copy & Paste übernehmen zu müssen, arbeitet der Anwender in einem Aufgabenbereich, der zum Beispiel nach Eingabe einer Kunden-

Bei IBF-Anwendungen ist der Aufgabenbereich der Ort, an dem Informationen erfasst und verändert werden. Aber IBF soll Applikationen wie die SAP-GUI nicht ablösen. Diese bieten nach wie vor den kompletten Zugriff auf die LoB-Applikation. Das IBF versteht sich vielmehr als eine Ergänzung für jene Situationen, in denen ein Anwender auf bestimmte In-

formationen des LoB-Systems zugreifen möchte. Ein Beispiel wäre das Anlegen eines neuen Kunden, das Abrufen zusätzlicher Informationen oder die Aufnahme einer Bestellung. Diese Tätigkeiten könnten dank einer IBF-Anwendung ohne SAP-GUI durchgeführt werden. Da sich die IBF-Anwendung auf eine Teilaufgabe beschränkt und die Anwender zudem in ihrer vertrauten Office-Anwendung arbeiten, werden nicht nur Fehleingaben ausgeschlossen, auch die ansonsten oft recht aufwändigen Schulungen könnten entfallen oder zumindest reduziert werden.

### Web Services – der Dreh- und Angelpunkt

Für die Kommunikation mit den LoB-Applikationen werden überwiegend Web Services eingesetzt, die über so genannte Ports, ähnlich wie beim BizTalk Server, aufgerufen werden. Damit die IBF-Engine diese Daten auch anzeigen kann, werden sie mit XML-Transformationen in IBF-Schema konformes XML umgesetzt. So können Daten ohne Programmierung im Aufgabenbereich dargestellt werden. Bietet die LoB-Applikation auch schreibende Web Services, so können über IBF auch Daten verändert werden. Abbildung 2 gibt einen Überblick über die IBF-Architektur. Der IBF-Client stellt Smart Tags zur Verfügung, die nach bestimmten Zeichenkombinationen in einem Word-, Excel-Dokument oder einer E-Mail suchen. Erkennt dann ein Smart Tag eine gesuchte Zeichenfolge, erscheint eine Liste mit Aktionen, welche an die IBF-Engine auf dem Client weitergeleitet werden. Diese greift dann über ihre Ports auf beliebige Web Services zu und liest oder schreibt Daten. Doch woher weiß die IBF-Engine, was sie zu tun hat? Dafür gibt es im Unternehmen irgendwo einen Webserver, auf dem zwei Meta-Data-Web-Services für die IBF-Clients installiert sind: Einer zum Lesen und einer zum Schreiben der Metadaten. Der Web Service zum Schreiben, wird z.B. von Visual Studio .NET verwendet, wenn die IBF-Lösung auf den Server hochgeladen wird. Die meisten Clients werden aber nur den Web Service zum Lesen verwenden. Die Clients laden sich die Metadaten-Informationen und wissen dann, auf welche Web Services sie zugreifen sollen und wie die Daten im Aufgabenbereich darzustellen sind. Die Metadaten sind das eigentli-

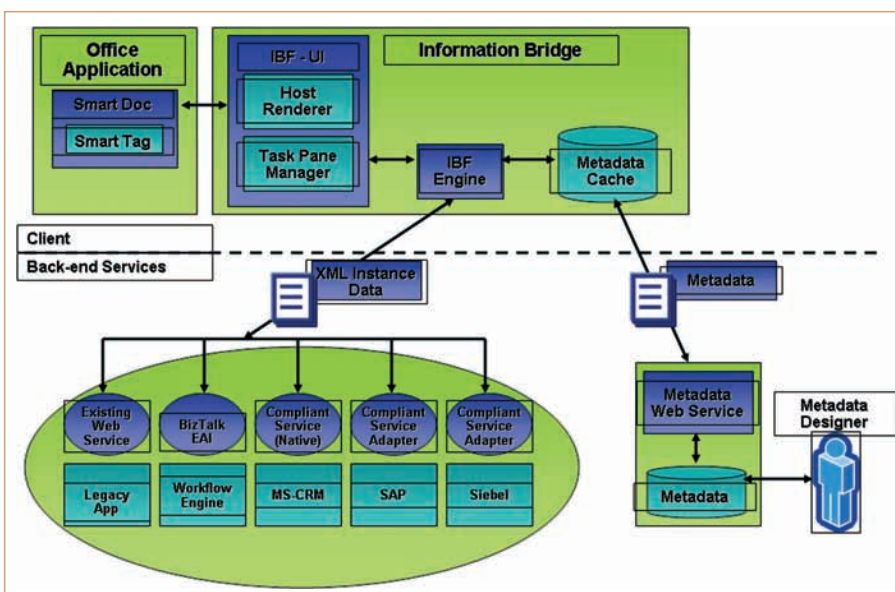


Abb 2: Die IBF-Architektur im Überblick

che *Programm* und bestehen aus einem großen XML-Dokument, in dem alles genau beschrieben ist. Jetzt könnte man fast sagen: Super, deklarative Programmierung, alles klar. Aber in den großen XML-Dokumenten sind Fehler nur schwer zu finden. Dies war einer der Gründe, warum die IBF-Entwicklung mit der Version 1.0 nur wahren „XML-Fetischisten“ Spass machen konnte. Seit Anfang des Jahres ist zum Glück eine Version 1.5 erhältlich, die unter anderem eine bessere Unterstützung beim Erstellen der Metadaten durch einen visuellen Editor für Visual Studio .NET umfasst.

### Installation und Download

Das IBF wird nicht als Ganzes, sondern in Gestalt der einzelnen Komponenten heruntergeladen [5]. Vorausgesetzt werden Windows Server 2003 mit IIS und SQL Server 2000/2005 für den IBF-Dienst. Die Client-Komponenten können auf dem gleichen oder einem anderen Rechner installiert werden. Der Client setzt

---

## Die Vorbereitung ist das A und O

---

Office 2003 (mit .NET-Unterstützung) voraus. Sie müssen daher bei der Office-Installation darauf achten, dass die *Primary Interop Assemblies* (PIA) mitinstalliert wird. Achtung, die Server-Komponenten liegen noch in der Version 1.0. Microsoft hat mit der 1.5-Version vor allem die Entwicklung erleichtern wollen. Eine ausführliche Installationsanleitung ist unter [3] zu finden. Wenn Sie Client und Server auf unterschiedlichen Rechnern installieren, denken Sie daran, dass der Client über zwei Ports auf den Server zugreifen können muss.

### IBF-Entwicklung planen

Die Entwicklung eines IBF-Projekts bedarf einer sorgfältigen Vorbereitung. Überlegen Sie sich als Erstes, welche Geschäftsobjekte (*Entities*) Sie darstellen wollen, bei komplexeren Lösungen auch wie diese zusammenhängen (ein Kunde, viele Bestellungen, usw.). Dann überlegen Sie, wie die Geschäftsobjekte dargestellt werden sollen (*Views*). Und schließlich, was soll mit den *Entities* getan werden, welche *Operations* soll es geben? Damit stehen nun die wichtigsten Elemente einer IBF-Lösung fest. Ein guter Anfang ist es, zunächst die Web Services zu erstellen. Auch wenn Sie später vielleicht auf SAP zugreifen wollen, aber noch kein SAP verfügbar haben, können Sie die Web Services durch ein kleines Web-Service-Projekt in Visual Studio .NET „simulieren“. Sie sollten nur darauf achten, dass am Ende das gleiche WSDL herauskommt. Testen Sie Ihre Web Services und benutzen Sie ein kleines Tool namens *WsdToMetadata.exe*, das aus dem WSDL Metadaten erzeugt.

### Der Metadaten-Explorer von Visual Studio .NET

Zu Ihrer Web-Service-Projektmappe können Sie nach der Installation der IBF-Entwicklungskomponenten ein Projekt vom Typ *Information Bridge Meta Data Projects* hinzufügen, in das die über WSDL erzeugten Metadaten importiert werden. Die weitere Arbeit wird mit dem *MetaDataExplorer* (Abbildung 3) vorgenommen. In der Version 1.5 kommen noch einige Assistenten

Anzeige

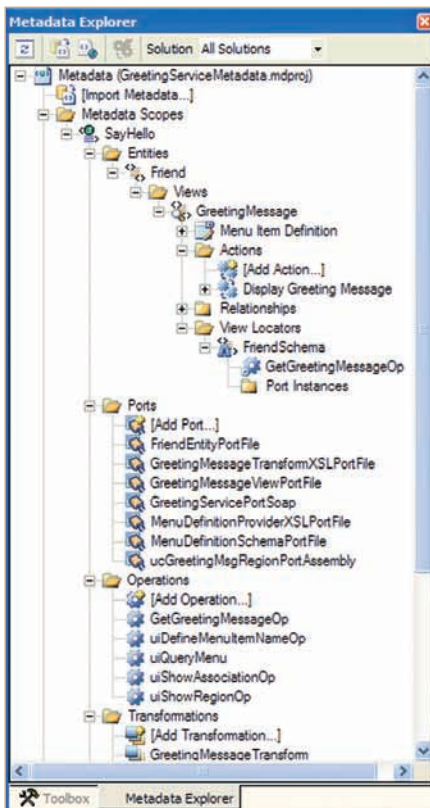


Abb. 3: Der *MetaDataExplorer* von VS.NET

hinzu. Im Metadaten-Explorer werden alle Teile der IBF-Lösung als Baum dargestellt. Dabei gibt es zwei Hauptäste. Zum einen Daten und Logik, dazu zählen Web Services, Ports, Operations und Entities. Und zum anderen die Darstellung der Informationen über Views, Menüs aber auch WinForms-Dialoge und *UserControls*. Tabelle 1 zeigt die verschiedenen Metadaten-typen.

Im Projektmappen-Explorer von Visual Studio .NET finden Sie unterhalb des IBF-Projektes nur eine einzige Datei, eine XML-Datei. Sie enthält alle Metadaten, die später zum Server hochgeladen und von dort an die Clients verteilt werden. Unter

### Die Komponenten des IBF

- Eine Client-Komponente, die für das Anlegen des Aufgabenbereichs zuständig ist
- Die IBF-Engine, die auf dem Client installiert wird
- Metadaten-Designer für Visual Studio .NET 2003
- BizTalk Mapper für XSL-Transformationen
- Ein Architecture Planning Guide
- Ein Upgrade- and Installations-Guide

[4] finden Sie ein einfaches Beispiel – eine Art „*Hello World*“ für das IBF, das es aber immerhin auf stolze 243 Zeilen bringt. Außerdem gehören zu einer IBF-Lösung noch *SmartTag-Recognizer* und eventuell *UserControls*. Der zweite Teil dieses Artikels, der nach der BASTA! 2005 erscheinen wird, wird die Umsetzung ausführlich vorstellen.

### Alternativen zur IBF

Natürlich können auch mit klassischen VBA-Mitteln dem Anwender Informationen aus Datenbanken über Web Services zur Verfügung gestellt werden. Das Einbinden eines Web Services ist in VBA dank des Web-Service-Toolkits grundsätzlich kein Problem. Aber über VBA kann der Aufgabenbereich nicht programmiert werden und es stehen auch nicht die bekannten Werkzeuge aus dem .NET Framework zur Verfügung. Eine weitere Alternative wäre die Implementierung des *ISmartDocument-Interface* mit Visual Basic 6, einer anderen COM-Programmiersprache, oder auch in Visual Studio .NET. Richtig Spass macht das aber mit keiner der aufgezählten Alternativen. Die Schnittstelle ist recht altmodisch und bei der Entwicklung passieren schnell Fehler. Eine bessere Variante und der direkte Konkurrent zu IBF sind die Visual Studio Tools für Office 2005. Damit kann auf sehr komfortable Art und Weise der Aufgabenbereich programmiert und Daten aus beliebigen Datenquellen über Smart-Tag-Aktionen zur Verfügung gestellt werden. Für den Anwender spielt es am Ende des Tages keine Rolle, auf welche Weise die Lösung umgesetzt wird – für den Entwickler dagegen schon. IBF bietet einen eher deklarativen Ansatz. Auch wenn IBF-Lösungen mit *UserControls* und WinForms-Dialogen ergänzt werden können, kann eine IBF-Lösung auch komplett ohne Code und mit guten XML-Kenntnissen umgesetzt wer-

| Metadata       | Ein Projekt der IBF-Lösung           |
|----------------|--------------------------------------|
| Entity         | Geschäftsobjekt                      |
| View           | eine Ansicht eines Geschäftsobjektes |
| Ports          | Quelle für XML, Daten, Schemata, ... |
| Schema         | XML-Schema                           |
| Transformation | wandelt XML in anderes XML ☺         |

Tabelle 1: Metadaten-typen

den. Das IBF versteht sich zudem nicht als Alternative, sondern als Ergänzung zum BizTalk Server, da letzterer zum Beispiel eine Transaktionsunterstützung mitbringt.

### Ausblick

In diesem Artikel ging es um einen ersten Überblick über das IBF, das sicher nicht in die üblichen Schubladen der Entwicklungstools passt und es daher auch eine Weile dauern wird, bis sich die Rolle des IBF und vor allem die damit verbundenen Möglichkeiten erschließen. Mit der Version 1.5 hat sich IBF zu einer ernsthaften Alternative für die Entwicklung von Office-Anwendungen in Unternehmen entwickelt. Wenn auch die Lernkurve am Anfang etwas größer sein mag, das IBF bietet umfassende Möglichkeiten für eine enge Integration zwischen Office-2003- und LoB-Anwendungen. Gerüchten zufolge plant Microsoft sogar eine Integration der IBF-Funktionalität in den Windows-Nachfolger Longhorn im Zusammenspiel mit Office 12, wenngleich sich in diesem sich schnell wandelnden und stets strategischen Überlegungen unterworfenen Bereich alles wieder ändern kann. ●

*Marcel Gnoth ist Leiter der Entwicklungsabteilung bei der Berliner NTeam GmbH. Neben seiner Arbeit als Softwareingenieur führt er Trainings durch, publiziert in Fachmagazinen und tritt regelmäßig als Sprecher auf der BASTA! auf. Seine Schwerpunkte liegen bei verteilten Informationssystemen und der Office-Integration. Kontakt: marcel@gnoth.net und www.gnoth.net.*

### ● Links & Literatur

- [1] Download der IBF-1.5-Komponenten: [www.microsoft.com/downloads/details.aspx?FamilyId=D7FA9F2F-3512-4163-A4AB-A36EEA3B6163&displaylang=en](http://www.microsoft.com/downloads/details.aspx?FamilyId=D7FA9F2F-3512-4163-A4AB-A36EEA3B6163&displaylang=en)
- [2] IBF-Newsgruppe auf msnews.microsoft.com: [microsoft.public.office.informationbridge](http://microsoft.public.office.informationbridge)
- [3] Microsoft Office Information Bridge Framework 1.5 Solution Development Guide: [go.microsoft.com/fwlink/?LinkId=40731](http://go.microsoft.com/fwlink/?LinkId=40731)
- [4] Einfacher Walkthrough: [blogs.officezealot.com/desai/archive/2004/09/19/2959.aspx](http://blogs.officezealot.com/desai/archive/2004/09/19/2959.aspx)
- [5] IBF auf MSDN: [msdn.microsoft.com/office/understanding/ibframework/default.aspx](http://msdn.microsoft.com/office/understanding/ibframework/default.aspx)
- [6] Folien zum Office-Power-Workshop auf der BASTA! Spring 2005: [www.gnoth.net/BASTA/Feb02.htm](http://www.gnoth.net/BASTA/Feb02.htm)