

Markus Widl

## BizTalk Server 2010

Administration, Entwicklung und Einsatz



# Auf einen Blick

<b>1</b>	<b>BizTalk Server-Einführung .....</b>	<b>17</b>
<b>2</b>	<b>Einführung in die Entwicklung einer BizTalk-Anwendung .....</b>	<b>69</b>
<b>3</b>	<b>Einführung in die Administration einer BizTalk-Anwendung .....</b>	<b>181</b>
<b>4</b>	<b>BizTalk-Einsatz .....</b>	<b>225</b>
<b>5</b>	<b>Monitoring und Debugging .....</b>	<b>391</b>

# Inhalt

Vorwort .....	13
---------------	----

<b>1</b>	<b>BizTalk Server-Einführung .....</b>	<b>17</b>
1.1	Warum gerade BizTalk? .....	17
1.1.1	Kommunikation mit heterogenen Systemen .....	17
1.1.2	Was macht BizTalk besser? .....	20
1.2	BizTalk Server-Komponenten .....	21
1.2.1	Messaging Engine .....	21
1.2.2	Orchestration Engine .....	22
1.2.3	B2B-Integration .....	24
1.2.4	Business Rules .....	24
1.2.5	Business Activity Monitoring .....	25
1.2.6	RFID .....	26
1.2.7	Verwaltung und Überwachung .....	26
1.2.8	Tools .....	27
1.3	BizTalk oder was? .....	27
1.3.1	BizTalk Server oder Windows Workflow Foundation? .....	28
1.3.2	BizTalk Server oder SharePoint? .....	28
1.3.3	BizTalk Server oder AppFabric? .....	31
1.4	Editionen .....	35
1.4.1	Enterprise Edition .....	35
1.4.2	Standard Edition .....	35
1.4.3	Branch Edition .....	36
1.4.4	Developer Edition .....	37
1.4.5	Enthaltene Adapter .....	37
1.5	Installation .....	40
1.5.1	Einrichten einer Entwicklungsumgebung .....	40
1.5.2	Installation des BizTalk Servers .....	44
1.5.3	Installation des BizTalk Adapter Packs .....	63
1.6	BizTalk-Werkzeuge .....	63
1.6.1	Management-Konsole BizTalk Server Configuration .....	64
1.6.2	Management-Konsole BizTalk Server Administration .....	65
1.6.3	Entwicklungsumgebung Visual Studio .....	65
1.7	Zusammenfassung .....	68

## 2 Einführung in die Entwicklung einer BizTalk-Anwendung 69

2.1	XML, Flatfile, EDI .....	69
2.1.1	XML-Grundlagen .....	70
2.1.2	Flatfile-Grundlagen .....	93
2.1.3	EDI-Grundlagen .....	94
2.2	Anlegen eines neuen BizTalk-Projekts .....	97
2.3	Bestandteile einer BizTalk-Anwendung .....	101
2.3.1	BizTalk-Assemblys .....	102
2.3.2	Schemas .....	103
2.3.3	Mappings .....	104
2.3.4	Orchestrations .....	105
2.3.5	Pipelines .....	106
2.3.6	Nicht-BizTalk-Assemblys .....	107
2.3.7	Anwendungskonfiguration .....	107
2.4	BizTalk Editor .....	108
2.4.1	Erstellung von Schemas mit dem BizTalk Editor .....	109
2.4.2	Erstellung von Schemas mithilfe von Assistenten .....	118
2.5	BizTalk Mapper .....	129
2.5.1	Erzeugen von Links .....	131
2.5.2	Map Grid und Map Pages .....	133
2.5.3	Functoids .....	134
2.5.4	Arbeiten mit großen Schemas .....	147
2.6	Orchestration Designer .....	148
2.6.1	Orchestration View .....	149
2.6.2	Flussdiagramm .....	153
2.6.3	Properties-Fenster .....	153
2.6.4	Toolbox .....	153
2.6.5	Transaktionen .....	171
2.7	Pipeline Designer .....	175
2.7.1	Pipeline-Stages .....	176
2.7.2	Pipeline-Komponenten .....	177
2.7.3	Eigene Pipeline-Komponenten .....	177
2.7.4	Standard-Pipelines .....	178
2.8	Deployment im Entwicklungssystem .....	179
2.9	Zusammenfassung .....	180

## 3 Einführung in die Administration einer BizTalk-Anwendung 181

3.1	BizTalk Message Flow .....	181
3.2	BizTalk-Anwendungen .....	184

3.3	BizTalk Server Administration Console .....	186
3.3.1	Group Hub Page .....	187
3.3.2	Platform Settings .....	187
3.3.3	Anwendungen anlegen und entfernen .....	193
3.3.4	Anwendungsbestandteile in der BizTalk Server Administration Console .....	194
3.3.5	Die Tore zur Welt: Ports .....	196
3.4	Anwendungs-Deployment von BizTalk zu BizTalk .....	209
3.4.1	Deployment mit der BizTalk Server Administration Console .....	210
3.4.2	Bindings übertragen .....	221
3.4.3	Deployment in der Kommandozeile .....	222
3.5	Zusammenfassung .....	223

## **4 BizTalk-Einsatz ..... 225**

4.1	Arbeiten mit Dateien .....	229
4.1.1	Arbeitsweise .....	229
4.1.2	Konfiguration .....	231
4.1.3	Beispiel: Weiterleitung von Dateien ohne Orchestrierung .....	232
4.1.4	Beispiel: Weiterleitung von Dateien mit Orchestrierung .....	242
4.2	FTP-Adapter ansprechen .....	255
4.2.1	FTP im BizTalk Server 2010 .....	255
4.2.2	Voraussetzungen für den Einsatz .....	255
4.2.3	Basiskonfiguration .....	256
4.2.4	Vorbereitungen .....	257
4.2.5	Beispiel .....	262
4.2.6	Read-Only-Dateien .....	267
4.3	E-Mails empfangen und versenden .....	267
4.3.1	Arbeitsweise .....	268
4.3.2	Optionen .....	268
4.3.3	Basiskonfiguration .....	270
4.3.4	Vorbereitungen .....	271
4.3.5	Beispiel .....	273
4.4	Webservices konsumieren und publizieren .....	281
4.4.1	Arbeitsweise .....	281
4.4.2	Basiskonfiguration .....	282
4.4.3	Beispiel 1: Konsumieren eines Webservice .....	282
4.4.4	Beispiel 2: Publizieren eines Webservice .....	295

4.5	Zugriff auf SQL-Datenbanken .....	304
4.5.1	Arbeitsweise .....	304
4.5.2	Basiskonfiguration .....	305
4.5.3	Beispiel .....	305
4.6	SharePoint-Anbindung mit und ohne InfoPath .....	322
4.6.1	Arbeitsweise .....	322
4.6.2	Einschränkungen .....	324
4.6.3	Voraussetzungen .....	325
4.6.4	Basiskonfiguration .....	326
4.6.5	Beispiel .....	327
4.7	Dynamics CRM-Anbindung .....	346
4.7.1	Installation überlisten .....	346
4.7.2	Arbeitsweise .....	348
4.7.3	Basiskonfiguration .....	349
4.7.4	Beispiel .....	349
4.8	Korrelationen und mehr .....	361
4.8.1	Umsetzung .....	362
4.8.2	Fertig? .....	389
4.9	Zusammenfassung .....	390

## **5 Monitoring und Debugging ..... 391**

5.1	Monitoring .....	391
5.1.1	Message Tracking .....	392
5.1.2	Failed Message Routing .....	402
5.1.3	System Center Operations Manager .....	404
5.2	Debugging .....	410
5.2.1	Orchestration Debugging .....	410
5.2.2	Map Debugging .....	414
5.3	Zusammenfassung .....	416

## **A Anhang ..... 417**

A.1	Kommandozeilenreferenz BTSTask.exe .....	417
A.1.1	AddApp .....	417
A.1.2	AddResource .....	418
A.1.3	ExportApp .....	426
A.1.4	ExportBindings .....	427
A.1.5	ExportSettings .....	428
A.1.6	ImportApp .....	428
A.1.7	ImportBindings .....	429

A.1.8	ImportSettings .....	430
A.1.9	ListApp .....	430
A.1.10	ListApps .....	431
A.1.11	ListPackage .....	431
A.1.12	ListTypes .....	432
A.1.13	RemoveResource .....	432
A.1.14	RemoveApp .....	432
A.1.15	UninstallApp .....	433
A.2	Administration mit Windows PowerShell .....	433
A.2.1	Funktionen des BizTalkFactory PowerShell Providers .....	436
A.2.2	Installation und Start des PowerShell-Providers .....	436
A.2.3	Navigationsparadigma .....	438
A.2.4	BizTalk-Cmdlets .....	440
A.2.5	PowerShell-Scripte .....	442
A.3	Inhalt des Downloadpakets .....	443
Index	.....	445

*Gleich geht's los. Doch vorher möchte ich Ihnen noch kurz ein paar einleitende Sätze zum Inhalt dieses Buches mit auf den Weg geben.*

## Vorwort

### Warum ein Buch zum BizTalk Server?

Natürlich gibt es auch zum Microsoft BizTalk Server 2010 im Web einige Websites und Blogs. Warum dann noch ein Buch dazu? Verglichen mit anderen Produkten wie Exchange und SharePoint spielt BizTalk im Web eine eher untergeordnete Rolle, was letztendlich auch dem deutlich kleineren Verbreitungsgrad geschuldet ist. Umso schwieriger ist es, eine gute, qualitativ gleichbleibende Quelle als Unterstützung bei der Entwicklung und dem Betrieb von BizTalk-Anwendungen zu finden. Dieses Buch soll hier helfen, die Informationslücke zu füllen. Es verfolgt keinen rein theoretischen Ansatz, sondern zeigt anhand einer Reihe von Szenarien auch den konkreten Einsatz, um Ihnen zu helfen, das Gelernte in die Praxis umzusetzen.

### An wen sich das Buch richtet?

Können Sie eine oder mehrere der folgenden Fragen mit »Ja« beantworten, könnte das Buch für Sie eine gute Unterstützung darstellen:

- ▶ Sie wollen wissen, wozu BizTalk eigentlich eingesetzt wird?
- ▶ Sie stehen vor der Frage, ob BizTalk das geeignete Produkt zur Umsetzung Ihres nächsten Projekts ist?
- ▶ Sie wollen eine solide Basis im Umgang mit BizTalk erwerben?
- ▶ Sie wollen oder sollen eine BizTalk-Anwendung entwickeln, haben das aber noch nie oder nur in Teilaspekten gemacht?
- ▶ Ihre Entwickler haben eine BizTalk-Anwendung geschrieben, und Sie sollen jetzt die Administration übernehmen?

Ich setze in diesem Buch keine besonderen Vorkenntnisse voraus. Sie sollten aber ein Basiswissen in der Programmierung mit einer beliebigen Sprache mitbringen. Es ist auch sehr hilfreich, wenn Sie schon einmal mit *Visual Studio* oder einer anderen ähnlich komplexen Entwicklungsumgebung gearbeitet haben. Kenntnisse in C# sind von Vorteil, da einige Aspekte der Entwicklung einer BizTalk-



Anwendung mit dieser Programmiersprache erfolgen. Wenn Sie dann noch mit den im administrativen Umfeld gängigen Management-Konsolen umgehen können, besitzen Sie das nötige Handwerkszeug, um direkt richtig loszulegen.

### An wen sich das Buch nicht richtet?

Beantworten Sie dagegen die folgenden Fragen mit »Ja«, ist vielleicht eine andere Quelle für Sie besser geeignet:

- ▶ Sie suchen eine umfassende und vor allem vollständige Beschreibung sämtlicher BizTalk-Komponenten inklusive *BAM*, *BRE* und *RFID*?

Insbesondere diese Komponenten werden hier nur ganz am Rande angesprochen. Eine angemessene Würdigung dieser Themen würde den Rahmen dieses Buches deutlich sprengen.

- ▶ Sie verwenden den BizTalk Server bereits seit Jahren, haben mehrere Projekte damit umgesetzt und können die Entwicklung und Administration notfalls auch im Schlaf?

Dann schauen Sie sich die Inhaltsübersicht genau an, ob ich Ihnen mit diesem Buch nicht vielleicht doch noch ein paar neue Kenntnisse vermitteln kann.

### Und der Inhalt?

Ich beginne in **Kapitel 1, »BizTalk Server-Einführung«**, mit einer allgemeinen Einführung in das Thema. Dazu gehören eine Betrachtung der mitgelieferten Komponenten und eine Platzierung des BizTalk Servers neben der *Windows Workflow Foundation*, neben *SharePoint* und der *AppFabric*. Auch lernen Sie, welches Feature in welcher BizTalk-Edition fehlt. Damit Sie im weiteren Verlauf des Buches auch mit BizTalk arbeiten können, werden die Installation, Konfiguration und die verschiedenen Werkzeuge grundlegend besprochen.

Es folgt daraufhin **Kapitel 2, »Einführung in die Entwicklung einer BizTalk-Anwendung«**. Da anwendungsintern bei BizTalk grundsätzlich mit *XML-Techniken* gearbeitet wird, beginnt das Kapitel mit einer Besprechung dieses Komplexes. Auch *Flatfile*- und *EDI-Strukturen* werden betrachtet. Ein wichtiger Teil besteht im Anlegen und Konfigurieren eines BizTalk-Projekts in Visual Studio. Sie lernen daher hier die Bestandteile einer BizTalk-Anwendung kennen und die verschiedenen Designer, grafischen Editoren und Assistenten. Zuletzt wird das Deployment Ihrer Anwendungen auf einen BizTalk Server zur Entwicklung gezeigt.

In **Kapitel 3, »Einführung in die Administration einer BizTalk-Anwendung«**, lernen Sie den Nachrichtenfluss und die daran beteiligten Komponenten und deren Konfiguration kennen. Sie arbeiten mit der Management-Konsole zur Adminis-

tration der Anwendungen und ihrer Bestandteile. Am Ende des Kapitels erfahren Sie mehr über die verschiedenen Möglichkeiten, eine fertige BizTalk-Anwendung auf mehrere Server zu verteilen.

Mit **Kapitel 4, »BizTalk-Einsatz«**, verfolgen Sie eine ganze Palette verschiedener Szenarien, die Schritt für Schritt die Entwicklung, Konfiguration und den Test unterschiedlicher Einsatzgebiete beschreiben. Sie werden dabei die bisher gewonnenen Kenntnisse umsetzen und noch einige weitere Details kennenlernen. *Dateiverarbeitung, FTP, E-Mail, Webservices, SQL-Datenbanken, SharePoint* und *Dynamics CRM* sind hier die wichtigsten Themen.

Da auch bei BizTalk-Anwendungen nicht immer alles glattgeht, lernen Sie in **Kapitel 5, »Monitoring und Debugging«**, Techniken zur Nachverfolgung der verarbeiteten Nachrichten und zur Fehlersuche kennen.

Das Buch schließt mit einem **»Anhang«**, der verschiedene Administrationsmöglichkeiten in der alten *CMD.exe* und der Windows PowerShell aufzeigt. Außerdem finden Sie dort eine Beschreibung des Downloadpakets zu diesem Buch. Es sind nämlich auf der Bonus-Seite zu Ihrem Buch unter <http://www.galileocomputing.de/bonus-seite> alle Beispiele, insbesondere die Szenarien, zum Download für Sie bereitgestellt. Weitere Informationen zur Nutzung der Bonus-Seite finden Sie auf der ersten, in blau gehaltenen Seite des Buches.

Manche Absätze sind mit dem rechts gezeigten Symbol gekennzeichnet. Diese Absätze stellen Kommentare zu den entsprechenden Schritten dar. Ich möchte Ihnen damit die einzelnen Aktionen näher vorstellen. [«]

Dieses Icon sehen Sie bei Kästen, die Antworten auf wichtige Fragestellungen geben. [!]

Das dritte dargestellte Symbol finden Sie bei Kästen, die zusätzliche Informationen zum beschriebenen Themengebiet enthalten. [ + ]

### Dankeschön!

Ein Vorwort ist auch ein guter Platz, um all denen zu danken, die zum Entstehen dieses Buches beigetragen haben. Zuerst genannt seien hier *Jan Watermann*, der den ersten Kontakt zu Galileo Press hergestellt hat, und *Sebastian Kestel*, mein Lektor, der sich immer geduldig meiner Anliegen angenommen hat. Aber auch alle anderen Galileos sollen hier nicht unerwähnt bleiben, tragen sie doch im Hintergrund einen großen Teil zum Entstehen dieses Buches bei.

Dank auch an *Jens Gerecke* von Microsoft, der das Fachlektorat übernommen hat. Dank seiner Hilfe hat so manche Erläuterung noch den letzten Schliff erhalten.

Auch möchte ich meiner Schwester *Angelika Widl*, die in Kürze ihr Studium des Kommunikationsdesigns an der Hochschule für Gestaltung in Schwäbisch Gmünd abschließen wird, danken. Sie hat einige der Schaubilder dieses Buches selbst erstellt.

Zu guter Letzt gilt mein Dank meiner Familie, insbesondere meiner Frau *Bettina* und unseren beiden Kindern. Sie haben vor allem in den vergangenen Wochen öfter auf mich verzichten müssen, wenn ich mich wieder einmal zum Schreiben in mein Büro zurückgezogen habe.

Ihnen bzw. euch allen ein herzliches Dankeschön!

So, nun aber genug der Vorrede. Ich wünsche Ihnen viele neue Erkenntnisse und Erfolg mit Ihren BizTalk-Projekten!

Schömberg,  
**Markus Widl**

*markus@widl.de*  
*twitter.com/markuswidl*

*BizTalk? Oder lieber Windows Workflow Foundation, AppFabric oder sogar SharePoint? Lesen Sie, wann BizTalk die richtige Wahl für Sie ist.*

## 1 BizTalk Server-Einführung

Dieses Kapitel vermittelt Ihnen wichtige Basisinformationen zum *Microsoft BizTalk Server 2010*, zu seinen Einsatzgebieten und den mitgelieferten Komponenten. Sie lernen hier die unterschiedlichen Editionen und deren Bestandteile kennen. Darüber hinaus werden die Installation und die anschließende Konfiguration besprochen. Das Kapitel schließt mit einer Vorstellung der BizTalk-Werkzeuge für Administratoren und Entwickler.

### 1.1 Warum gerade BizTalk?

Warum sollte sich mein Unternehmen dafür entscheiden, den BizTalk Server einzusetzen? Welche Vorteile ergeben sich dadurch? Wie kann ich Kosten sparen, wenn doch eine BizTalk-Lizenz nicht gerade billig ist? Warum entwickle ich die nötige Funktionalität nicht einfach selbst? Solche und ähnliche Fragen treten immer wieder auf, wenn das Gespräch auf den möglichen Einsatz des BizTalk Servers zu sprechen kommt. In diesem Abschnitt versuche ich, diese Fragen einleitend zu klären.

Heutzutage arbeiten Unternehmen mit vielen unterschiedlichen Anwendungen auf unterschiedlichen Plattformen, etwa aus den Bereichen *LOB* (= *Line of Business*), *CRM* (= *Customer Relationship Management*) und *ERP* (= *Enterprise Resource Planning*). Diese Anwendungen laufen aber nicht isoliert auf jeweils einer eigenen Insel. Zur Abbildung von Geschäftsprozessen ist eine Kommunikation zwischen den Anwendungen erforderlich, um Daten beispielsweise abzufragen oder abzugleichen.

#### 1.1.1 Kommunikation mit heterogenen Systemen

Nehmen wir einen typischen Bestellprozess: Eine Kundenbestellung geht ein. Dies kann bereits über verschiedene Kanäle passieren, etwa über die firmenei-

gene Shop-Website, per Mail oder per Telefon. Diese Bestellung muss auf Plausibilität, Vollständigkeit und Korrektheit geprüft werden. Anschließend wird das aktuelle Kreditlimit des Kunden ermittelt und daraufhin entschieden, ob die Bestellung automatisch freigegeben wird oder zunächst von einem Vertriebsmitarbeiter manuell freigegeben werden muss. Das Kreditlimit wird dabei im CRM-System des Kunden verwaltet, mit dem dann eine entsprechende Kommunikation stattfinden muss.

Der manuelle Genehmigungsprozess soll über das Vertriebsportal im Intranet abgewickelt werden. Auch hier müssen entsprechende Daten ausgetauscht werden.

Wird die Bestellung freigegeben, kommt mit der ERP-Lagerverwaltung das nächste System zum Einsatz. Der aktuelle Lagerbestand wird ermittelt. Reicht dieser für die Bestellung nicht aus, soll automatisch beim Lieferanten eine Nachbestellung ausgeführt werden. Der voraussichtliche Liefertermin wird ermittelt.

Sobald die bestellten Produkte vorrätig sind, wird die Lieferung ausgeführt und beim Zahlungsdienstleister eine Einzugsanweisung vorgenommen.

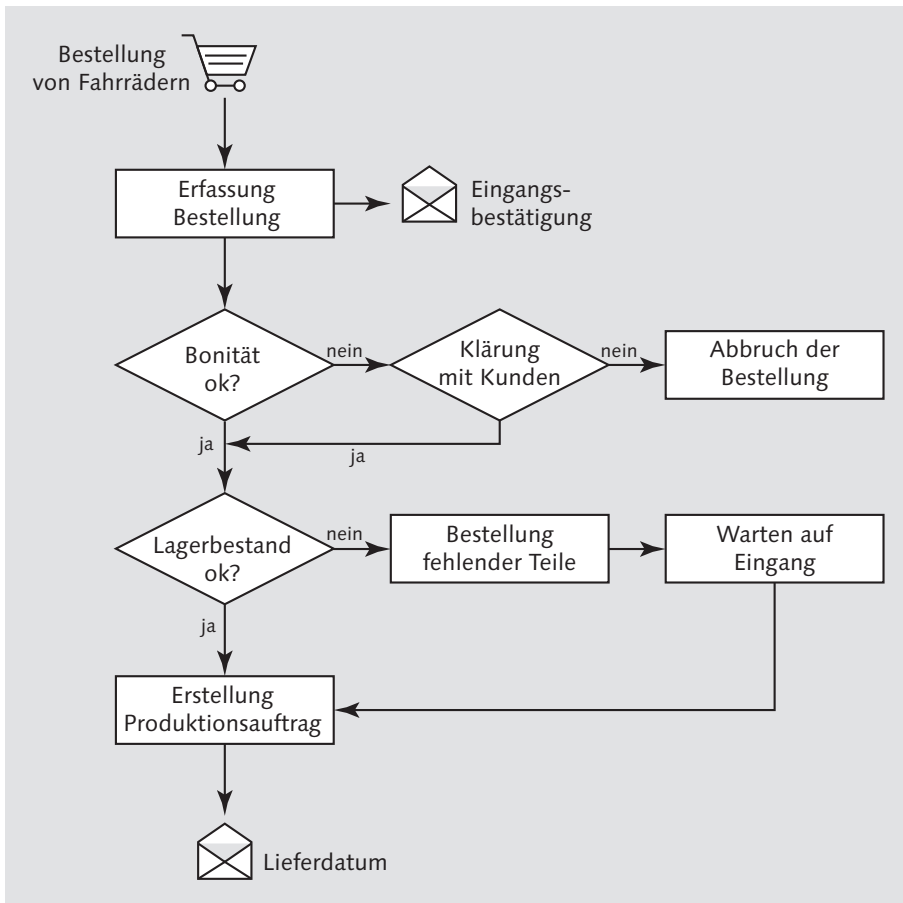
Natürlich soll der Kunde bei den einzelnen Prozessschritten über E-Mails und eine Webseite im Online-Shop auf dem Laufenden gehalten werden.

Der Vertriebsleiter und der Geschäftsführer wollen möglichst in Echtzeit prozessübergreifend Einblick in die Vorgänge erhalten, etwa hinsichtlich der Bestell- und Umsatzzahlen und beispielsweise auch der Auswertungen der Dauer einzelner Prozessschritte. So ist es unter anderem von geschäftsentscheidender Bedeutung, wie lange durchschnittlich die manuelle Freigabe einer Bestellung durch einen Vertriebsmitarbeiter dauert. Ist diese Dauer zu groß, mag das ein Hinweis auf die Notwendigkeit sein, das Vertriebspersonal aufzustocken oder den Prozess anzupassen.

Abbildung 1.1 gibt den, zugegebenermaßen vereinfachten, Prozess wieder.

An diesem fiktiven Beispiel sind bereits einige Aspekte zu erkennen, bei denen BizTalk eingesetzt werden kann. Wir haben dort unterschiedliche Systeme mit unterschiedlicher Datenhaltung. Über den Online-Shop werden die Bestellungen beispielsweise in einer SQL-Datenbank abgelegt. Das CRM-System erlaubt keinen direkten Datenbankzugriff, bietet aber stattdessen eine Palette an Webservices zur Abfrage an. Beim ERP-System könnte es ähnlich sein, nur die Webservices haben dann sicher einen anderen Aufbau.

So haben wir es hier bereits bei der unternehmensinternen Kommunikation der Anwendungen untereinander mit unterschiedlichen Schnittstellen und Nachrichtentypen zu tun, die zwischen den Anwendungen ausgetauscht werden müssen. Allerdings bleibt es oft nicht dabei. So müssen externe Handelspartner angebunden werden, mit denen z.B. über *EDI-Nachrichten* (= *Electronic Data Interchange*) kommuniziert wird.



**Abbildung 1.1** Ein vereinfachter Bestellprozess

Natürlich könnten wir jetzt eine Anwendung mit der von uns favorisierten Programmiersprache schreiben, doch ist der Aufwand beim Ansprechen der unterschiedlichen Schnittstellen und Protokolle (SMTP für die Statusmails, SOAP/WCF für Webservices unterschiedlichster Art etc.) enorm.

Die am Prozess beteiligten Daten müssen ständig in bestimmte Formate umgewandelt werden, um der jeweiligen Anwendungsschnittstelle gerecht zu werden. Und wenn sich dann in Zukunft einmal eine Anwendungsschnittstelle ändert oder eine Anwendung komplett ersetzt wird, ist der Anpassungsaufwand erheblich.

Nicht selten entstehen hier oft schwer zu wartende »Codemonster«.

### 1.1.2 Was macht BizTalk besser?

Mit dem Einsatz des BizTalk Servers kann diese Problematik erheblich verringert werden. BizTalk ist darauf ausgelegt, Nachrichten in den unterschiedlichsten Formaten aus unterschiedlichen Quellen zu empfangen, zu verarbeiten und an die unterschiedlichsten Ziele zu senden. Out of the box werden Flatfile-, also Textstrukturen allgemein, XML- und EDI-Strukturen unterstützt. In den Marketingunterlagen ist hier von der Integration heterogener Systeme die Rede.

Um Sie als Entwickler beim Ansprechen unterschiedlicher Anwendungen, Datenbanken und Protokolle zu entlasten, wird BizTalk mit einer ganzen Palette sogenannter *Adapter* für die unterschiedlichsten Einsatzgebiete ausgeliefert. So gibt es Protokolladapter (z. B. Dateizugriff, SMTP, POP3, FTP), Technologieadapter (z. B. Zugriff auf SQL-Datenbanken) und Anwendungsadapter (z. B. für Dynamics CRM und SharePoint).

Eine Übersicht über die mitgelieferten Adapter finden Sie in Abschnitt 1.4.5, »Enthaltene Adapter«.

Auch die Entwicklung eigener Adapter oder der Einkauf bei Drittherstellern ist möglich. Nutzen Sie diese Adapter, müssen Sie zwar das Nachrichtenformat für die jeweilige Anwendung berücksichtigen, Sie müssen sich dafür aber kaum mehr mit der eigentlichen Schnittstelle auseinandersetzen.

Der WSS-Adapter für den SharePoint-Zugriff vereinfacht beispielsweise die eigentlichen Webservice-Aufrufe. Statt entsprechenden Code zu entwickeln, wird der Adapter konfiguriert.

Im Regelfall decken diese Adapter nicht die komplette Schnittstelle ab, aber doch zumindest die Basisfunktionalität. Wenn mehr erforderlich ist, steht der Weg, die Schnittstelle über Code anzusprechen, immer noch offen.

Ein weiterer Vorteil liegt darin, dass unabhängig davon, wie die eingelesenen oder versendeten Nachrichten aussehen, BizTalk-intern Nachrichten immer als XML-Struktur behandelt werden. Die Nachrichten werden also »normalisiert«,

wodurch sie dann mit den immer gleichen Werkzeugen verarbeitet werden können. Anwendungsbestandteile werden so universell einsetzbar.

Wenn Sie selbst über keinerlei XML-Kenntnisse verfügen, ist das kein Beinbruch. Zu Beginn des Kapitels 2, »Einführung in die Entwicklung einer BizTalk-Anwendung«, lernen Sie die notwendigen wesentlichen Grundzüge dieser Technologie kennen. Außerdem werden Sie oftmals statt direkt mit XML-Techniken wie *XSD* (= *XML Schema*), *XPath* (= *XML Path Language*) und *XSLT* (= *XSL Transformation*) mit vereinfachenden grafischen Designern und Assistenten arbeiten.

Zu guter Letzt denkt BizTalk nicht nur an die Produktivität der Entwickler, sondern auch an die Administratoren, die letztendlich dafür Sorge tragen müssen, dass die fertigen Anwendungen auch in Produktivumgebungen unterbrechungsfrei laufen. Sollte das einmal nicht der Fall sein, müssen sie in der Lage sein, die Ursache des Problems schnell zu erkennen und zu beheben. Dafür bringt BizTalk eine Reihe verschiedener Werkzeuge mit, die den Administrator bei seinen Aufgaben wie Installation, Konfiguration, Deployment und Monitoring unterstützen.

## 1.2 BizTalk Server-Komponenten

Der BizTalk Server setzt sich aus einer Reihe unterschiedlicher Komponenten zusammen. Nicht für jeden Einsatzfall sind immer alle Komponenten erforderlich. Das Kernstück ist die *Messaging Engine* zur Nachrichtenverarbeitung.

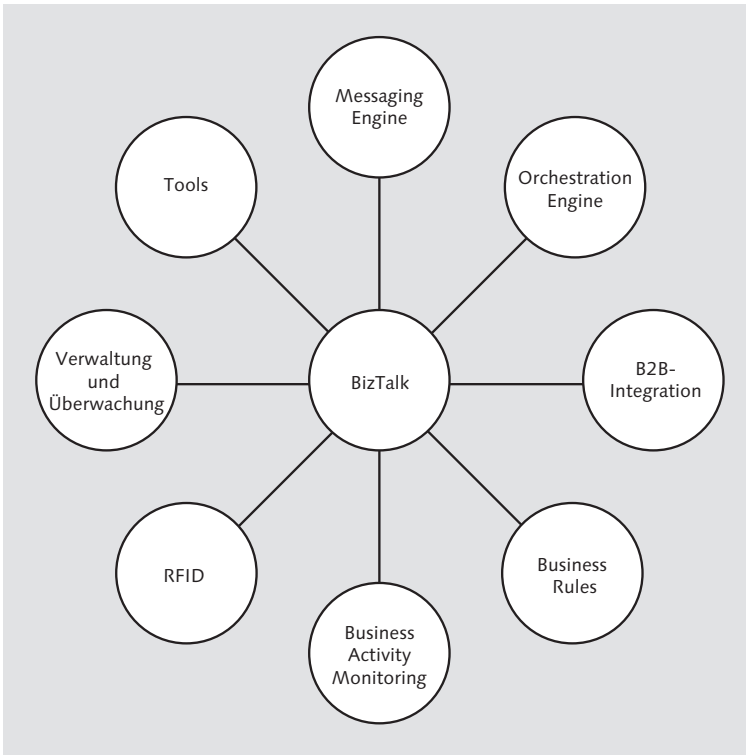
### 1.2.1 Messaging Engine

Die Nachrichtenverarbeitung ist Aufgabe der Messaging Engine des BizTalk Servers, eines der Herzstücke dieses Systems. Ihre Aufgaben sind folgende:

- ▶ Empfangen eingehender Nachrichten
- ▶ Ermittlung des Nachrichtenformats
- ▶ Auswertung der Nachrichten zur Bestimmung der Bearbeitung
- ▶ Auslieferung von Nachrichten an Empfänger
- ▶ Nachrichtenverfolgung

Der Messaging Engine steht aber noch eine ganze Reihe weiterer Komponenten zur Seite, wie Abbildung 1.2 zeigt.





**Abbildung 1.2** Die Herzstücke des BizTalk Servers

Mit den meisten dieser Komponenten werden wir im Laufe dieses Buches arbeiten. Doch zunächst einmal folgt eine kurze Erläuterung der einzelnen Komponenten.

### 1.2.2 Orchestration Engine

Die *Orchestration Engine* arbeitet eng mit der *Messaging Engine* zusammen. Bei der Entwicklung einer BizTalk-Anwendung können Sie *Orchestrations* anlegen. Diese stellen die Logik eines Programms dar, werden aber primär nicht über Programmcode (z. B. Abfragen, Schleifen oder Funktionsaufrufe) entwickelt, sondern über ein grafisches Flussdiagramm. Ein Beispiel dafür sehen Sie in Abbildung 1.3.

Die einzelnen Elemente, die sogenannten *Shapes*, besitzen Eigenschaften, über die sie konfiguriert werden. Je nach Shape-Typ kann auch Programmcode hinterlegt werden.

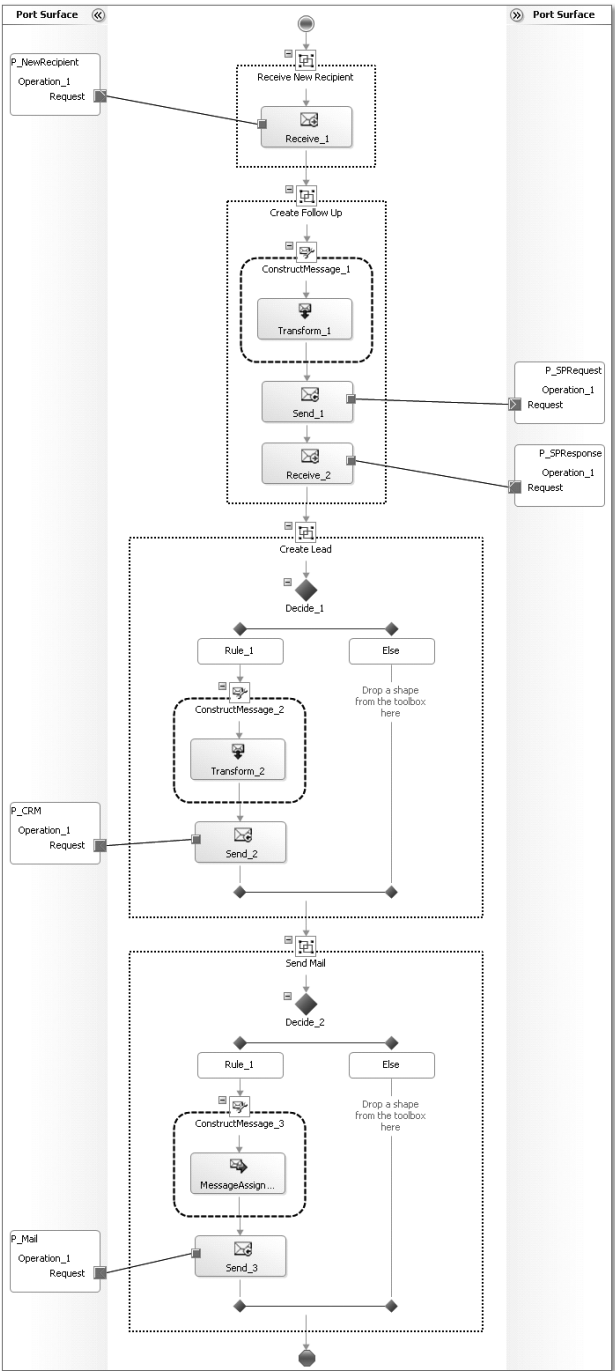


Abbildung 1.3 Mit einer Orchestration wird die Logik einer BizTalk-Anwendung entwickelt.

Solche Orchestrations werden über die Orchestration Engine ausgeführt, eine Art *Workflow Engine*. Sie nehmen die von der Messaging Engine empfangenen Nachrichten entgegen und verarbeiten sie entsprechend dem Flussdiagramm. Die Orchestrations befinden sich zusammen mit anderen Anwendungsbestandteilen in *.NET-Assemblys*, im BizTalk-Fall handelt es sich dabei um DLL-Dateien.

### 1.2.3 B2B-Integration

Hiermit ist die Anbindung von Handelspartnern (*Trading Partners*) gemeint (*B2B* = *Business to Business*), beispielsweise über den Austausch von *EDI-Nachrichten*.

Für jeden Handelspartner kann eine eigene Konfiguration angelegt werden, wobei der Umstand berücksichtigt wird, dass jeder Partner ein eigenes Nachrichtenformat haben kann (und in der Praxis oft auch hat). Außerdem können ohne Änderungen am eigentlichen Programmcode der Anwendung (den Orchestrations) Handelspartner hinzugefügt und von dort wieder entfernt werden, und das allein durch reine Konfiguration (siehe Abbildung 1.4).

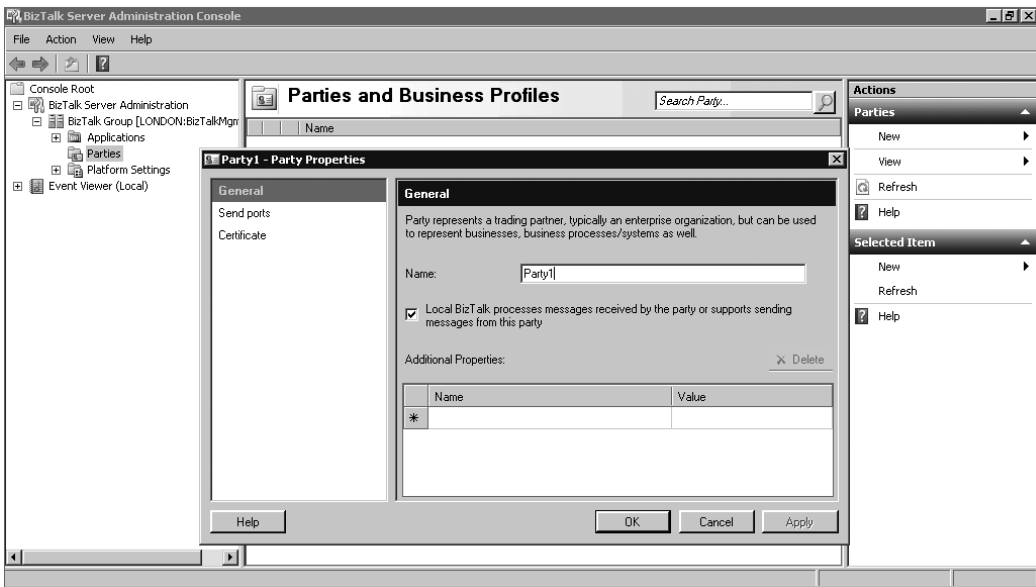


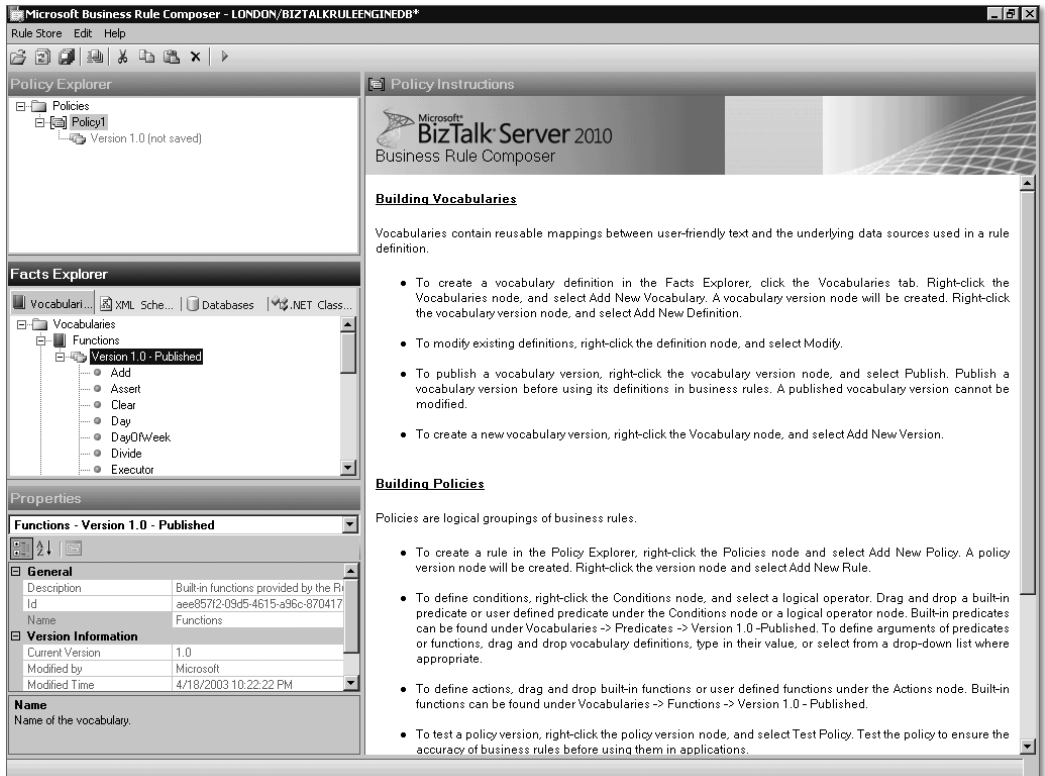
Abbildung 1.4 Die Handelspartnerkonfiguration im BizTalk Server

### 1.2.4 Business Rules

Sie können in den Orchestrations auch Entscheidungen fest hinterlegen, beispielsweise unter welchen Bedingungen welches Versandunternehmen beauftragt werden soll. Dies hätte jedoch den Nachteil, dass bei jeder Änderung die

Orchestration und damit das .NET-Assembly angepasst werden müsste. Es wäre also auch ein Entwickler für derart alltägliche Änderungen erforderlich.

Mit der *Business Rules*-Komponente können Sie derartige Regeln vom Anwender über ein separates Tool, den *Business Rule Composer*, vornehmen lassen (siehe Abbildung 1.5).



**Abbildung 1.5** Der Business Rule Composer zur Definition von Geschäftsregeln

Die Business Rules werden dann von Orchestrations aus aufgerufen, was die Änderung an Prozessen vereinfachen kann.

### 1.2.5 Business Activity Monitoring

Mithilfe der *Business Activity Monitoring*-Komponenten (BAM) können Sie BizTalk anweisen, die während der Ausführung auftretenden Daten zu protokollieren und in einer frei abfragbaren Datenbank und über das *BAM-Portal* (eine spezielle SharePoint-Anwendung) in Berichten bereitzustellen. Hiermit könnten

Auswertungen zu den bestellten Produkten, Umsatzzahlen und auch Zeitdauern für einzelne Prozessschritte ermittelt werden (siehe Abbildung 1.6).



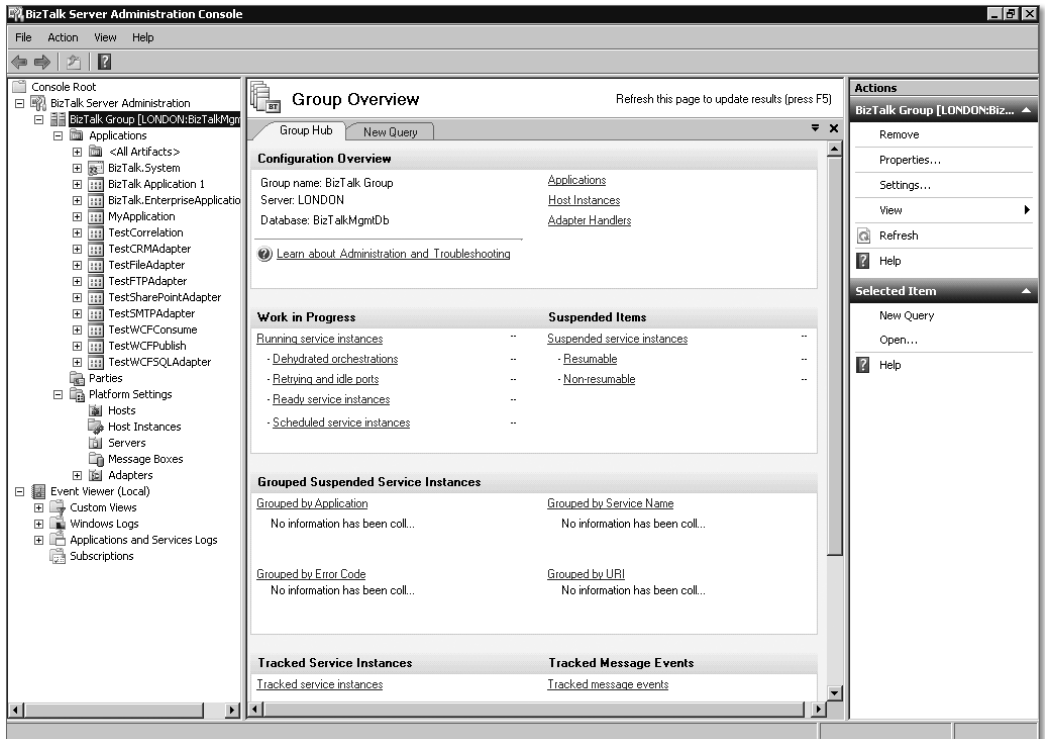
Abbildung 1.6 Das BAM-Portal zur Auswertung von Prozessdaten

## 1.2.6 RFID

Seit einigen Versionen besitzt BizTalk auch eine Out-of-the-box-Unterstützung von *RFID-Daten* (= *Radio Frequency Identification*). Dieser Bereich kommt hier im Buch nicht weiter zur Sprache, da das Thema für eine ausreichende Würdigung zu viel Platz beanspruchen würde.

## 1.2.7 Verwaltung und Überwachung

Zur Verwaltung des BizTalk Servers, dort laufender Anwendungen und der Überwachung des Nachrichtenaustauschs gibt es eine Palette an Werkzeugen und Management-Konsolen. Ich werde Ihnen hier einige davon vorstellen (siehe Abbildung 1.7).



**Abbildung 1.7** Mit der BizTalk Server Administration Console werden BizTalk-Anwendungen verwaltet.

### 1.2.8 Tools

Nicht zuletzt gibt es auch übergreifende Tools, wie eine Erweiterung von *Visual Studio 2010*, mit dem die Entwicklung von komplexen BizTalk-Anwendungen ermöglicht wird.

## 1.3 BizTalk oder was?

Bei jedem neuen Projekt müssen Sie die Entscheidung treffen, auf Basis welcher Produkte und Technologien dieses umgesetzt werden soll. Manchmal gibt es dabei nicht die eine Lösung, sondern mehrere Alternativen oder eine Kombination daraus. Dieser Abschnitt soll Ihnen eine Entscheidungshilfe an die Hand geben, wann BizTalk, die Windows Workflow Foundation, SharePoint (hier insbesondere der Workflow-Bereich) und AppFabric jeweils ihre Vorteile ausspielen können.

### 1.3.1 BizTalk Server oder Windows Workflow Foundation?

Nachdem Sie nun gelernt haben, dass der BizTalk Server mit der *Orchestration Engine* auch eine Workflow Engine enthält, stellt sich die Frage, in welchen Szenarien eher ein BizTalk Server und wann die *Windows Workflow Foundation* (WF) aus dem .NET Framework ab Version 3 zum Einsatz kommen sollte, insbesondere da Letztere von Microsoft kostenlos abgegeben wird. Ein wichtiges Entscheidungskriterium liegt in der erforderlichen Kommunikation der Lösung mit verschiedenen Anwendungen, Protokollen und Datenbanken. Als Faustregel gilt dabei: Benötigen Sie für Ihre Lösung eine Workflow-Funktionalität, die sich primär innerhalb einer Anwendung abspielt, ist das ein Hinweis auf die WF. Je mehr Kommunikation Sie mit Anwendungen (z. B. LOB-Anwendungen wie SAP, Dynamics NAV, Dynamics CRM), Protokollen (z. B. FTP, Webservices) und Datenbanken (z. B. SQL, Oracle) gemeinsam benötigen, umso eher kann hier der BizTalk Server für Sie von Vorteil sein. Einige Gründe, die dabei für BizTalk sprechen, sind folgende:

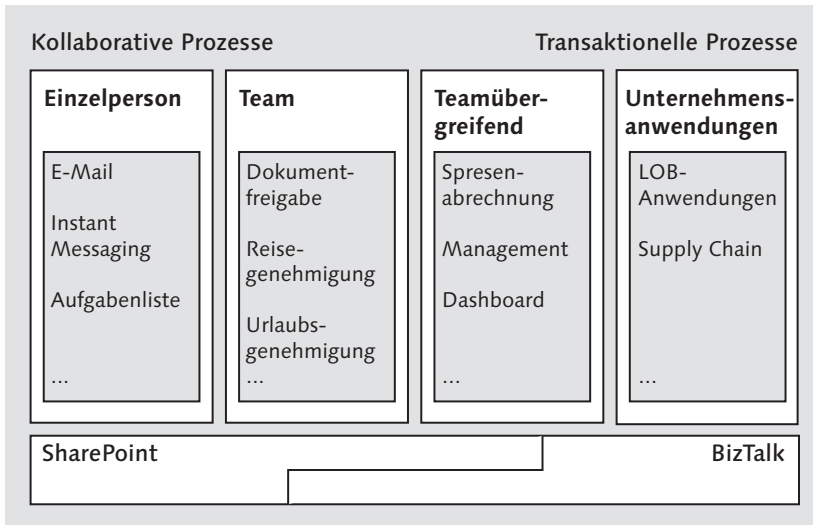
- ▶ Ein zentraler Hub, über den die Kommunikation abgewickelt wird. Entsprechend müssen Sie bei Änderungen am System, beispielsweise beim Austausch einer beteiligten Anwendung, nur an einer Stelle Anpassungen vornehmen.
- ▶ Eine zentrale Stelle, die für die Überwachung und Fehlersuche bei Kommunikationsproblemen verantwortlich ist.
- ▶ Ein potenziell geringerer Entwicklungsaufwand, da der BizTalk Server bereits out of the box mit seinen Adaptern die Verbindung zu diversen Anwendungen, Protokollen und Datenbanken über Konfigurationen aufbauen kann.
- ▶ Die Möglichkeit, Anwendungen über die Kommunikation und beteiligte Daten zu erstellen, etwa mithilfe der optionalen BizTalk-Komponente *Business Activity Monitoring* (BAM).

Die Entscheidung für BizTalk oder die WF hängt allerdings noch von vielen anderen Faktoren ab, z. B. von dem verfügbaren Budget, Kenntnissen der Entwickler und der vorhandenen Serverinfrastruktur.

### 1.3.2 BizTalk Server oder SharePoint?

Neben der Frage, ob Workflows mit BizTalk oder der WF umgesetzt werden sollen, steht vielleicht auch der mögliche Einsatz von SharePoint (also entweder der *SharePoint Server 2010* oder die kostenlose Variante *SharePoint Foundation 2010*) im Raum. Der Einfachheit halber ist im Folgenden verkürzt stets von *SharePoint* die Rede.

Beide Produkte haben Stärken bei unterschiedlichen Workflow-Szenarien (siehe Abbildung 1.8).



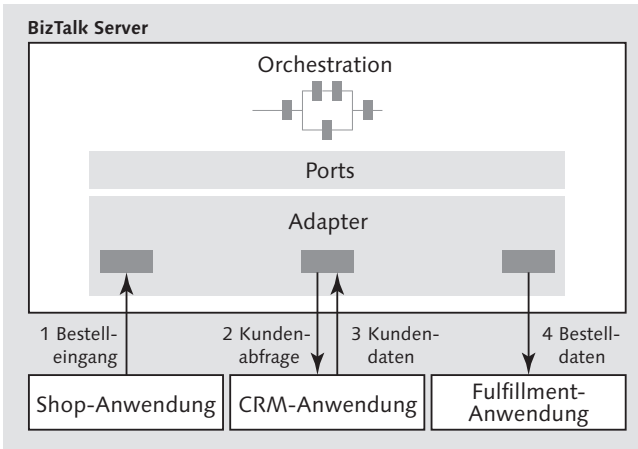
**Abbildung 1.8** Prozesse lassen sich in unterschiedliche Arten einteilen und werden von unterschiedlichen Produkten abgedeckt.

Sie sehen im Diagramm vier verschiedene Prozessarten: *Einzelperson*, *Team*, *Teamübergreifend* und *Unternehmensanwendungen*. Die drei linken Arten erfordern eine hohe Benutzerinteraktion, etwa zur Genehmigung eines Dokuments. Hier kann SharePoint seine Stärke mit ausgefeilten Benutzeroberflächen im Browser und mit der Integration in den Office-Applikationen ausspielen. BizTalk wäre hier nur schwer einzusetzen, da die Möglichkeit zur einfachen Benutzerinteraktion fehlt. Diese müsste über externe Lösungen realisiert werden.

Dagegen kann BizTalk bei der Prozessart am rechten Rand punkten. Durch das Adapterkonzept ist die Kommunikation mit externen Systemen in BizTalk potenziell einfacher als in SharePoint, da die Schnittstellen dort oft aufwendig mit eigenem Programmcode angesprochen werden müssen.

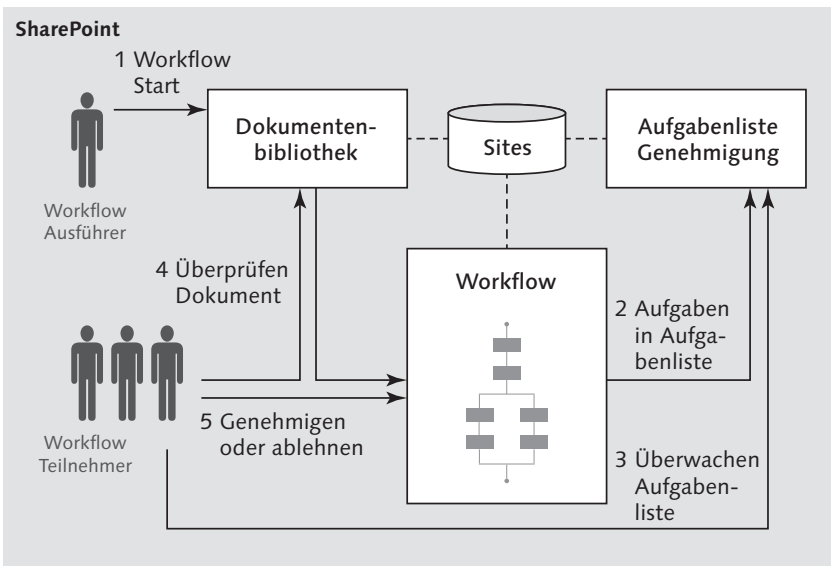
In manchen Situationen stellt sich aber nicht die Frage, ob BizTalk oder SharePoint zum Einsatz kommen soll, sondern es gibt auch Bereiche, in denen sich die beiden Produkte ergänzen. Ein Beispiel: Der Geschäftsprozess »Bestellung« soll umgesetzt werden. Dazu müssen Daten aus verschiedenen Anwendungen ausgelesen und hinterlegt werden. Außerdem muss eine dafür verantwortliche Person über die Plausibilität und über die Freigabe entscheiden. Abbildung 1.9 veranschaulicht dies anhand eines typischen BizTalk-Workflows mit einem hohen Anwendungskommunikationsanteil.





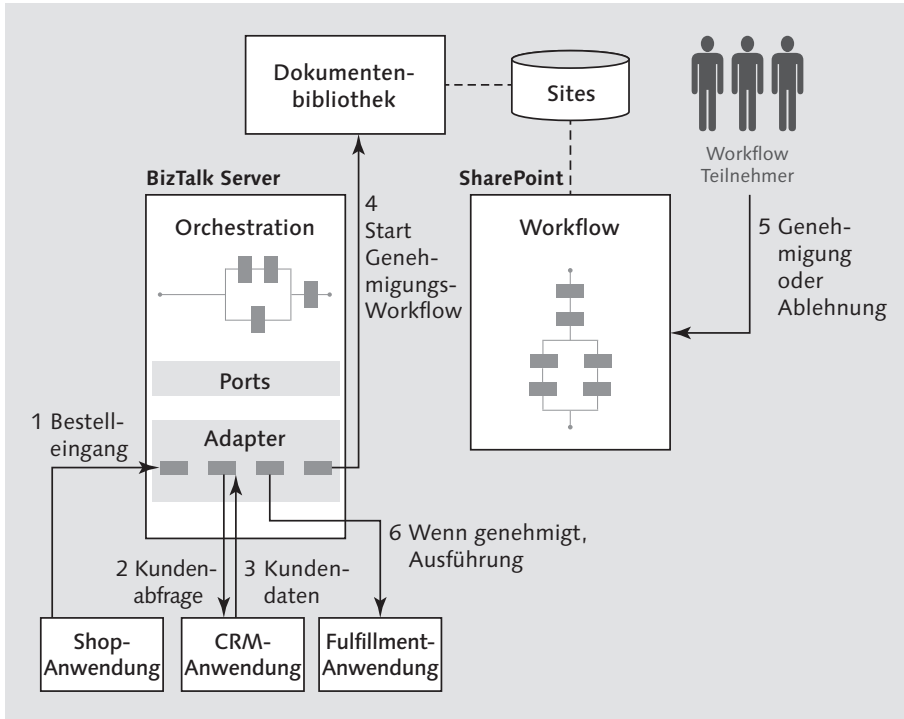
**Abbildung 1.9** Ein typischer Workflow, wie er von BizTalk abgedeckt wird

Es fehlt hier jedoch der Genehmigungsschritt. Dieser könnte in SharePoint wie in Abbildung 1.10 aussehen.



**Abbildung 1.10** Ein typischer Workflow, wie er von SharePoint ausgeführt wird

Zur Umsetzung der Lösung könnten nun beide Produkte gemeinsam dort eingesetzt werden, wo sie ihre jeweilige Stärke haben. BizTalk bei der Anwendungs- und SharePoint bei der Anwenderkommunikation. Kombiniert sieht der Workflow wie in Abbildung 1.11 aus.



**Abbildung 1.11** BizTalk und SharePoint arbeiten zusammen.

Das »führende« System wäre hier der BizTalk Server. Er erzeugt aus den ermittelten Daten ein Dokument, das in SharePoint abgelegt wird. SharePoint deckt dann den Genehmigungsprozess ab (Benachrichtigung der genehmigenden Person, Formulare etc.). BizTalk wartet auf das Ergebnis der Genehmigung und verfährt dann entsprechend.

### 1.3.3 BizTalk Server oder AppFabric?

Mit *AppFabric* stellt Microsoft eine weitere *Middle-Tier-Technologie* bereit. Hier stellt sich die Frage, wie diese gegenüber BizTalk platziert wird und ob BizTalk in Zukunft vielleicht durch AppFabric ersetzt werden soll. Der Begriff *AppFabric* ist eigentlich zu ungenau, denn es gibt zwei separate Produkte mit unterschiedlicher Ausprägung, die beide auf AppFabric »hören«:

- *Windows Azure AppFabric*
- *Windows Server AppFabric*

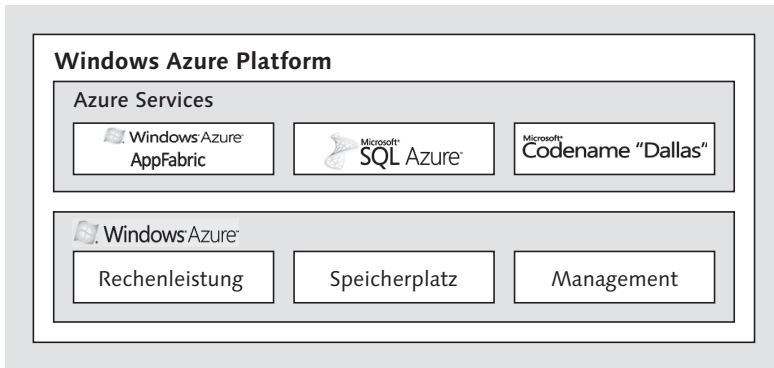
Das erstgenannte Produkt gehört zum Cloud-Universum *Microsoft Azure Platform*, das zweite ist für die lokale Installation gedacht.

## Windows Azure AppFabric

AppFabric richtet sich speziell an Anwendungsentwickler, die aus dem *.NET Framework 4* insbesondere die Komponenten *WCF* (= *Windows Communication Foundation*) und *WF* (= *Windows Workflow Foundation*) einsetzen. Dem *.NET Framework* fehlt bis heute ein geeignetes *Middle-Tier-Hosting*. AppFabric soll diese Lücke füllen, liefert es doch eine Serverinfrastruktur, mit der die auf Basis der *WF* entwickelte Geschäftslogik ausgeführt und über *WCF* angesprochen wird.

Windows Azure AppFabric ist eine von vier Kernbestandteilen aus der Microsoft Azure Plattform (siehe Abbildung 1.12). Diese sind:

- ▶ *Windows Azure*, die Betriebsumgebung oder, wie Microsoft es nennt, das Betriebssystem für Azure-Anwendungen auf Basis des Windows Server 2008.
- ▶ *SQL Azure*, die Cloud-Variante des SQL Servers für relationale Datenbanken.
- ▶ *Dallas*, der Codename für einen Marketplace für kommerzielle und freie Informationen aller Art, die in eigenen Anwendungen genutzt und bereitgestellt werden können. Beispiele hierfür sind Kartenmaterial, Börsenkurse, Wetterdaten, Nachrichten und Bilder. Auch das Publizieren als Datenanbieter ist möglich.
- ▶ *Windows Azure AppFabric*, das hier näher vorgestellt wird und im Wesentlichen aus den Komponenten *Access Control Service (ACS)* zur Authentifizierung und Autorisierung sowie einem *Service Bus* besteht, der als Kommunikationsinfrastruktur zwischen Anwendungen fungiert.



**Abbildung 1.12** Die Komponenten des Microsoft Azure-Universums

Windows Azure AppFabric wurde während des Entwicklungszeitraums eine Zeit lang als *BizTalk Services* titulierte und danach auch noch als *.NET Services*.

Die über AppFabric kommunizierenden Anwendungen müssen dabei nicht zwangsweise in der Cloud ausgeführt werden, auch die Anbindung von *On Pre-*

mise-Systemen, also lokal installierten Anwendungen, ist mit AppFabric möglich. Die Anwendungen müssen dabei noch nicht einmal zum selben Unternehmen gehören. Dabei löst AppFabric ein wichtiges Problem: On Premise-installierte Anwendungen sind im Regelfall über eine Firewall von außen abgeschottet. Sollen die Anwendungen aber über eine Schnittstelle verfügbar gemacht werden, ist entweder eine Konfigurationsänderung an der Firewall, das Platzieren der Anwendung(sschnittstelle) innerhalb der DMZ (= *Demilitarized Zone*) oder eines *Extranets* erforderlich. Nicht so, wenn AppFabric zum Einsatz kommt. Die Kommunikationspartner verbinden sich nicht direkt miteinander, sondern indirekt über AppFabric. Der Kommunikationsaufbau geschieht dabei vom inneren geschützten Netzwerk nach außen. AppFabric vermittelt dann die Kommunikation, läuft selbst aber in der Cloud, physisch also in Microsofts Rechenzentren.

## Windows Server AppFabric

Windows Server AppFabric wird lokal installiert und erweitert die Management-Konsole der IIS (= *Internet Information Services*) um WCF-Konfigurationsoptionen. Dies war dringend nötig, denn die bisherige WCF-Konfiguration über den *WCF Service Configuration Editor* ist sowohl für Entwickler also auch Administratoren eine echte Herausforderung. Das AppFabric-eigene Dashboard sehen Sie in Abbildung 1.13.

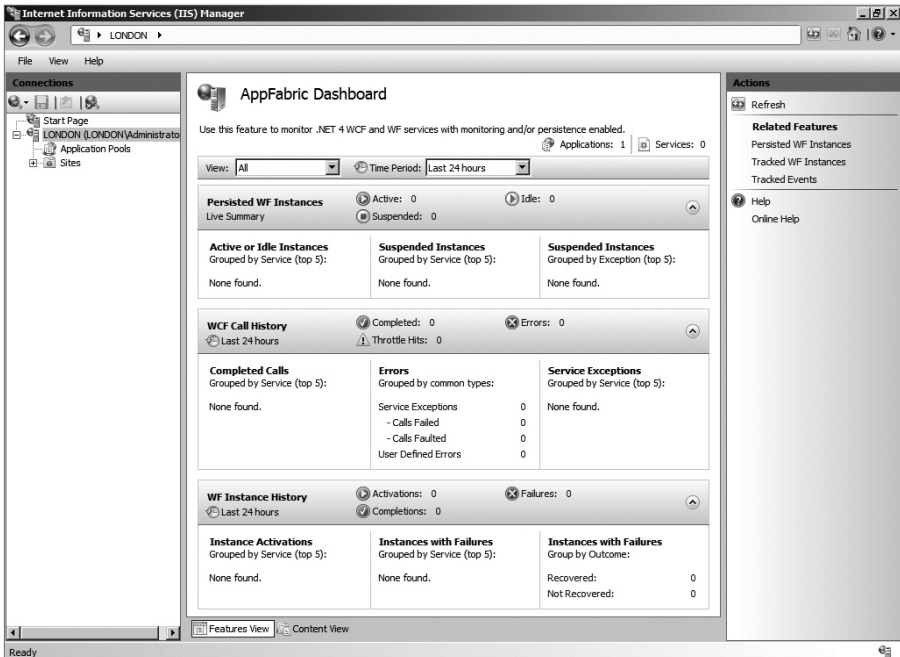


Abbildung 1.13 Die AppFabric-Konfiguration innerhalb der IIS-Konsole

## BizTalk und AppFabric im Vergleich

Beide Produkte verfolgen ein ähnliches Ziel, die Kommunikation von Anwendungen untereinander zu regeln, doch sind ihre Ansätze unterschiedlich. Während BizTalk nachrichtenorientiert ist (»XML Schema first«) ist AppFabric dienst- und workflowzentriert (»Code first«).

Im Vergleich zeigt sich, dass AppFabric zusammen mit WCF und WF aus dem .NET Framework 4 folgende Features abdeckt:

- ▶ Skalierbarer, zuverlässiger, überwachbarer Host, der auch *Load Balancing* (Lastverteilung) unterstützt
- ▶ Nachrichtenverfolgung und Archivierung
- ▶ Unterstützung von Suspend-/Resume- und Persistenz-Mechanismen
- ▶ Transaktionsverarbeitung
- ▶ Werkzeuge für Administratoren zur Verwaltung und Überwachung

BizTalk deckt diese Features ebenfalls ab, doch liefert er out of the box noch einiges mehr:

- ▶ Adapterkonzept
- ▶ Unterstützung von EDI, RFID, EDIFACT, X12, HL7, HIPAA, RosettaNet und SWIFT
- ▶ Auswerten von Flatfile-Strukturen
- ▶ Aufteilen einer großen Nachricht in mehrere kleine
- ▶ konfigurierbare Wiederholungsmechanismen für den Fehlerfall
- ▶ Business Activity Monitoring (BAM)
- ▶ Business Rules Engine (BRE)
- ▶ weiterführende grafische Werkzeuge und Assistenten für die Entwicklung

Diese Liste erhebt nicht den Anspruch, vollständig zu sein, hilft aber bei der Beantwortung der Frage, welches Produkt bei einer konkreten Aufgabenstellung zum Einsatz kommen soll. Manchmal geht es aber auch gar nicht um ein Entweder-oder, sondern um eine gemeinsame Lösung mit mehreren Komponenten. Auf dem Installationsmedium des BizTalk Servers finden Sie die optionale Komponente *AppFabric Connect*, mit der BizTalk-Anwendungen direkt mit AppFabric kommunizieren können.

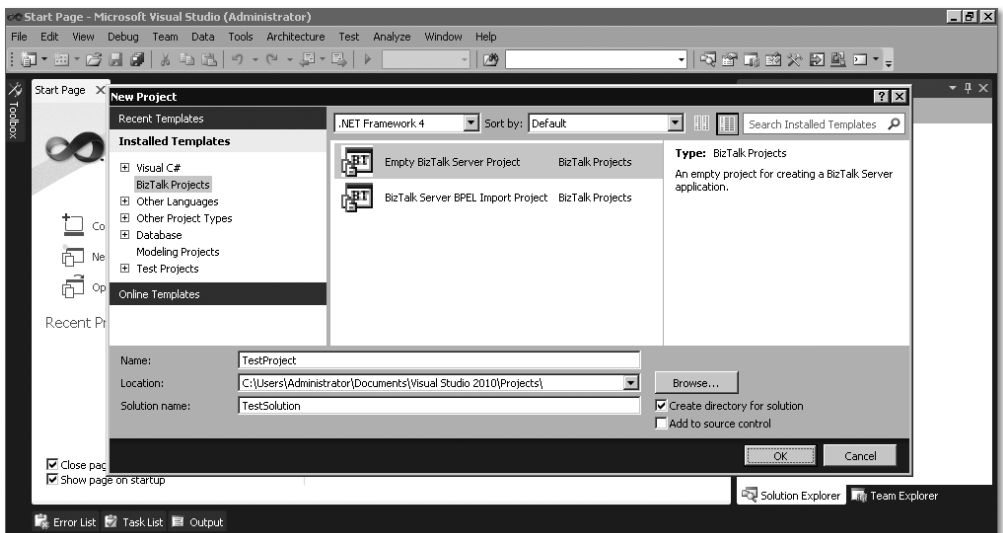
In diesem Buch spielt AppFabric aus Platzgründen keine weitere Rolle. Wir stürzen uns hier ganz auf BizTalk.

## 2.2 Anlegen eines neuen BizTalk-Projekts

Legen Sie ein neues *BizTalk-Projekt* an, gehen Sie grundsätzlich immer nach den gleichen Schritten vor. Nachdem Sie das Projekt in Visual Studio erstellt haben, sollten Sie auch gleich einige grundlegende Konfigurationen für das Deployment, das heißt das Übertragen der BizTalk-Anwendung aus Visual Studio heraus auf einen Entwicklungs-BizTalk Server, vornehmen. Diese Angaben könnten Sie auch zu einem späteren Zeitpunkt angeben, aber wenn Sie es gleich nach dem Anlegen des Projekts machen, vergessen Sie es später nicht.

Zunächst legen Sie also ein neues Visual Studio-Projekt basierend auf der Vorlage *Empty BizTalk Server Project* an:

1. Starten Sie Visual Studio.
2. Wählen Sie den Menübefehl **FILE • NEW • PROJECT**.
3. Im Vorlagenfenster wählen Sie die Kategorie **BIZTALK PROJECTS** und markieren dort die Vorlage **EMPTY BIZTALK SERVER PROJECT** (siehe Abbildung 2.11).



**Abbildung 2.11** Ein neues BizTalk-Projekt wird angelegt.

4. Geben Sie einen Namen für das neue Projekt an (hier *TestProject*) und einen Pfad, unter dem alle zum Projekt gehörenden Dateien abgelegt werden sollen.
5. Visual Studio möchte außerdem einen *Solution Name* wissen. In einer Solution können ein oder auch mehrere zusammengehörende Projekte zusammengefasst und gemeinsam geöffnet werden. Oftmals werden Sie nur eine

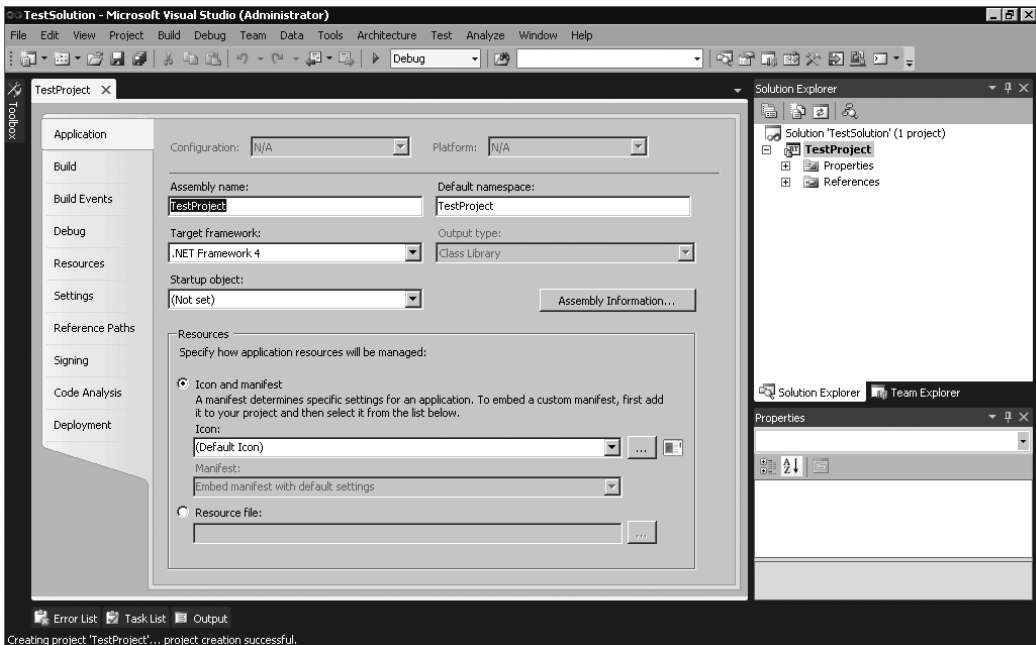
Solution mit einem einzelnen Projekt anlegen. Behalten Sie aber stets im Hinterkopf, dass in einer Solution auch mehrere Projekte enthalten sein können. Das macht die Verwaltung in größeren Szenarien einfacher. Die Solution kann einen eigenen Namen haben, der nicht identisch mit dem Projekt sein muss (hier `TestSolution`).

Nachdem Visual Studio die Solution und das Projekt gemäß den Angaben erzeugt hat, folgt nun die Basiskonfiguration. Hier sind zwei Punkte besonders wichtig:

- *Deployment-Einstellungen* für den Entwicklungs-BizTalk Server
- Angeben einer *SNK-Datei* (*Strong Name Keyfile*) zum Signieren des Assemblys

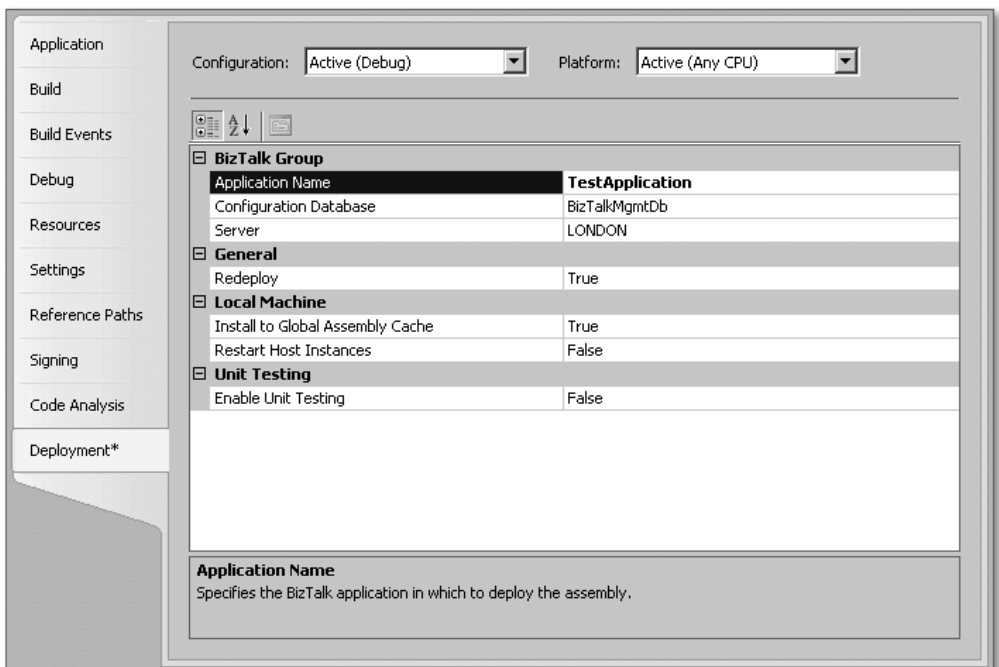
Diese Angaben nehmen Sie über folgende Schritte vor:

1. Im Menü **PROJECT** wählen Sie den Eintrag **TESTPROJECT PROPERTIES**. Alternativ dazu können Sie im **Solution Explorer** auch mit der rechten Maustaste auf den Projektnamen klicken und im sich daraufhin öffnenden Menü **PROPERTIES** auswählen. Es erscheint ein Fenster mit einigen Registerkarten (siehe Abbildung 2.12).



**Abbildung 2.12** Die Projekteigenschaften werden über verschiedene Registerkarten verwaltet.

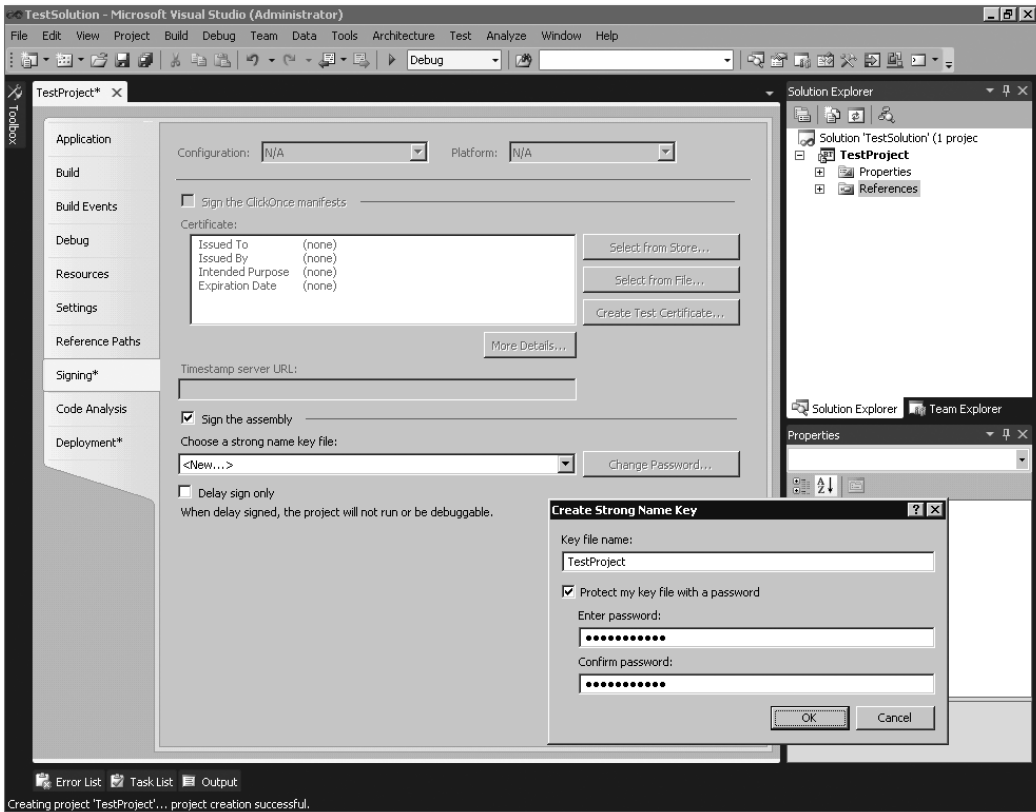
2. Wechseln Sie zur Registerkarte DEPLOYMENT, und geben Sie unter APPLICATION NAME einen Namen für Ihre BizTalk-Anwendung ein (hier *TestApplication*). Diesen Namen werden Sie in der BIZTALK SERVER ADMINISTRATION CONSOLE wiederfinden. Theoretisch könnten Sie den Namen auch weglassen, dann werden die Projektbestandteile auf dem BizTalk Server der sogenannten *Default Application* hinzugefügt. Allerdings ist es dann nicht einfach, alle zur Anwendung gehörenden Bestandteile zu erkennen und getrennt voneinander zu verwalten.
3. Bei SERVER geben Sie den Namen Ihres BizTalk Servers an.
4. Unter CONFIGURATION DATABASE wählen Sie die Management-Datenbank, die bei der Installation des BizTalk Servers angelegt wurde (siehe Abbildung 2.13).



**Abbildung 2.13** Die Deployment-Einstellungen

5. Wechseln Sie nun zur Registerkarte SIGNING, und markieren Sie das Auswahlfeld SIGN THE ASSEMBLY. In der Combobox wählen Sie den Eintrag NEW und geben in dem sich daraufhin öffnenden Fenster einen beliebigen Dateinamen für ein *Keyfile* und ein Kennwort an (siehe Abbildung 2.14).
6. Speichern Sie Ihre Angaben, oder schließen Sie das Eigenschaften-Fenster.





**Abbildung 2.14** BizTalk-Assemblies müssen für die Aufnahme in den GAC digital signiert werden.

## [!]

### Warum muss signiert werden?

Als Ergebnis des Visual Studio-Projekts wird ein .NET-Assembly in Form einer DLL-Datei erzeugt. Zur Ausführung der Anwendung auf dem BizTalk Server muss diese DLL-Datei in den sogenannten GAC (= *Global Assembly Cache*) aufgenommen werden. Dabei handelt es sich um den Ordner %windir%\Microsoft.NET\assembly (siehe Abbildung 2.15). Vielleicht kennen Sie als GAC auch den Ordner %windir%\assembly. Dieser kommt bei .NET Framework 2- bis 3.5-Assemblies zum Einsatz. Da eine Anwendung für den BizTalk Server 2010 als *Target Framework* gezwungenermaßen Version 4 voreingestellt hat, kommen die BizTalk-Assemblies auch in den neuen GAC vom .NET Framework 4. In ihm werden globale Assemblies abgelegt, die systemweit zum Einsatz kommen sollen. Im GAC können auch mehrere Versionen desselben Assemblys abgelegt werden.

Damit ein Assembly in den GAC aufgenommen werden kann, muss es digital signiert werden. Diese Aufgabe übernimmt Visual Studio mithilfe des Keyfiles, das dort automatisch erzeugt und zum Projekt hinzugefügt wird (Dateiendung PFX).

Die digitale Signatur ist aus Sicherheitsgründen erforderlich. Ein digital signiertes Assembly lässt sich nicht unbemerkt mit Schadcode ausstatten. Außerdem wird die Versionsverwaltung des Assemblys mithilfe der Signatur durchgeführt. Visual Studio kann das Assembly automatisch in den GAC des BizTalk Servers aufnehmen. Manuell wäre dies mithilfe des Kommandozeilentools GACUTIL.EXE möglich, das Bestandteil des .NET Frameworks ist.

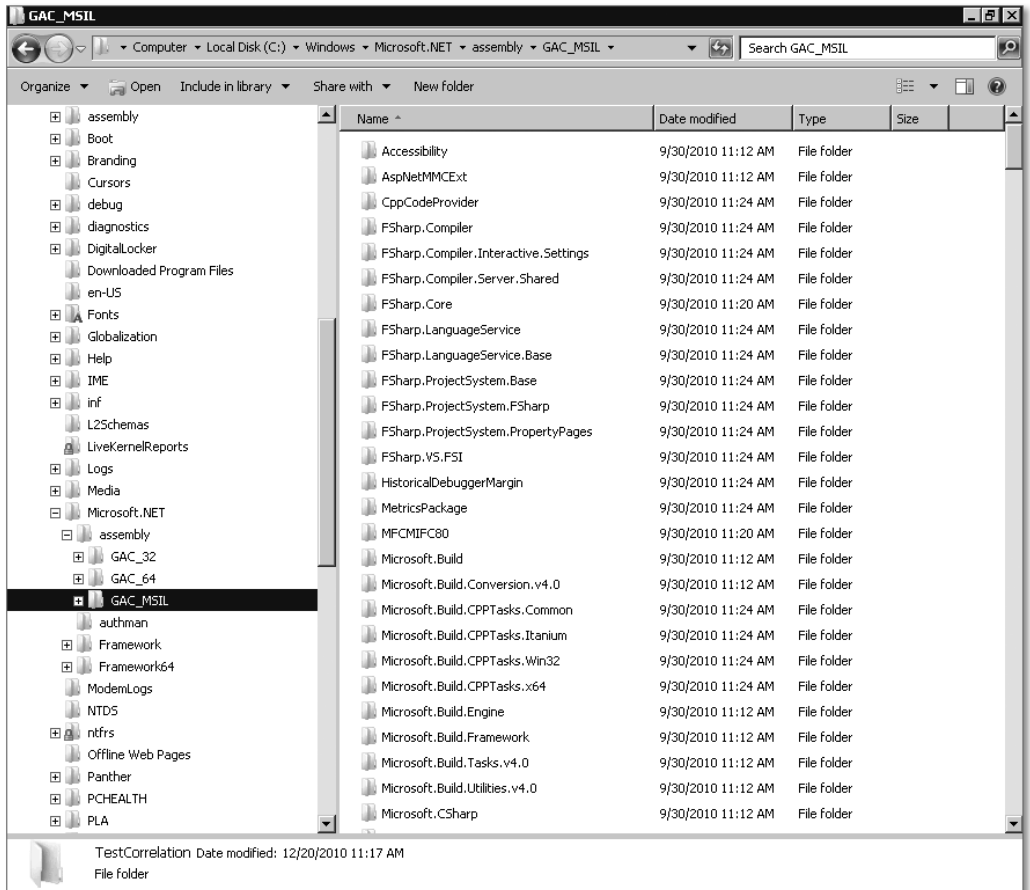
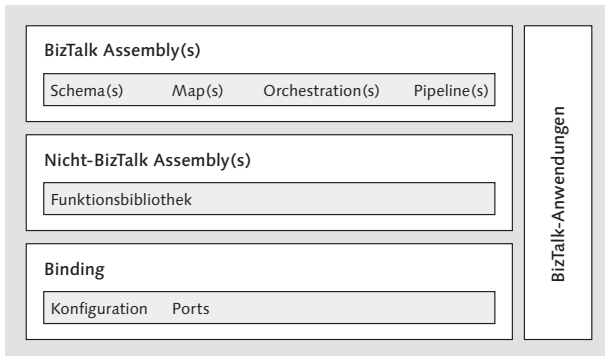


Abbildung 2.15 Im GAC des BizTalk Servers müssen die Assemblys abgelegt werden.

## 2.3 Bestandteile einer BizTalk-Anwendung

Je nach Szenario besteht eine *BizTalk-Anwendung* aus unterschiedlichen Komponenten. Abbildung 2.16 skizziert diese Komponenten.



**Abbildung 2.16** Eine BizTalk-Anwendung besteht aus Assemblys und aus dem Binding.

Das Schaubild zeigt BizTalk-Assemblys, Nicht-BizTalk-Assemblys sowie eine Konfiguration, das *Binding*. Während die Assemblys im GAC abgelegt werden, wird die Konfiguration in der Management-Datenbank des BizTalk Servers abgelegt, die während der Installation erzeugt wurde.

All diese Bestandteile, also Assemblys plus Konfiguration, machen eine BizTalk-Anwendung aus, und sie werden auch zusammen verwaltet.

### 2.3.1 BizTalk-Assemblys

Ein wesentlicher Bestandteil einer BizTalk-Anwendung ist das bzw. sind die BizTalk-Assemblys. Dabei handelt es sich um .NET-Assemblys, die in Form von DLL-Dateien mithilfe von Visual Studio und den dort integrierten BizTalk-Entwicklerkomponenten erstellt werden. Ein solches Assembly kann folgende Inhalte haben:

- ▶ Schemas
- ▶ Mappings
- ▶ Orchestrations
- ▶ Pipelines

Nicht jede Anwendung erfordert alle vier Komponenten, doch werden Sie diese oft gemeinsam antreffen. Man spricht hierbei auch von den sogenannten *BizTalk-Artefakten*. Diese Bestandteile können in einem einzelnen Assembly oder aufgeteilt in mehreren Assemblys vorhanden sein und sogar mehrfach in verschiedenen Versionen in unterschiedlichen Assemblys.

Es stellt sich hier die Frage, welche Aufgabe diese Elemente jeweils haben.

### 2.3.2 Schemas

Damit Ihre BizTalk-Anwendung sinnvoll auf die Werte von eingehenden Nachrichten zugreifen kann, muss der Aufbau der Nachricht bekannt sein. Dazu gehört beispielsweise, welche Elemente und Attribute vorhanden sind, ihre Reihenfolge, ob sie sich wiederholen können, welche Datentypen vorliegen etc. Man spricht hier vom sogenannten *Nachrichtenschema*.

Dieses Schema ist nicht nur für eingehende Nachrichten wichtig, sondern auch für ausgehende sowie für Nachrichten, die nur anwendungsintern zum Einsatz kommen. Nachrichtenschemas erzeugen Sie in Visual Studio mit dem *BizTalk Editor* aus den Entwicklerkomponenten des BizTalk Servers (siehe Abbildung 2.17).

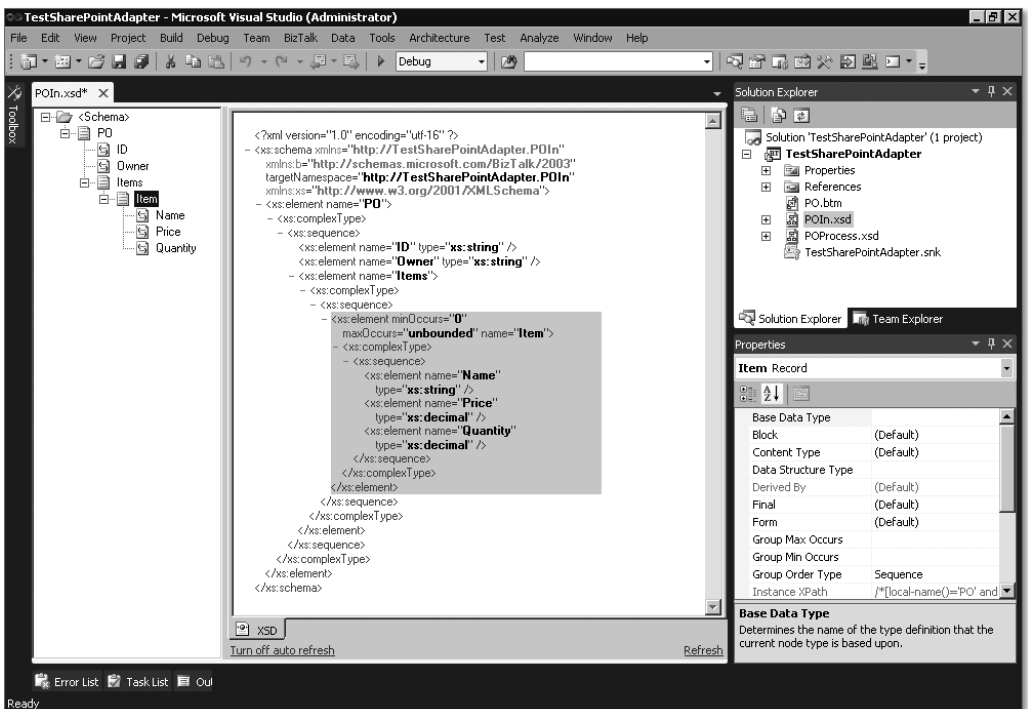


Abbildung 2.17 Mit dem BizTalk Editor definieren Sie Nachrichtenschemas.

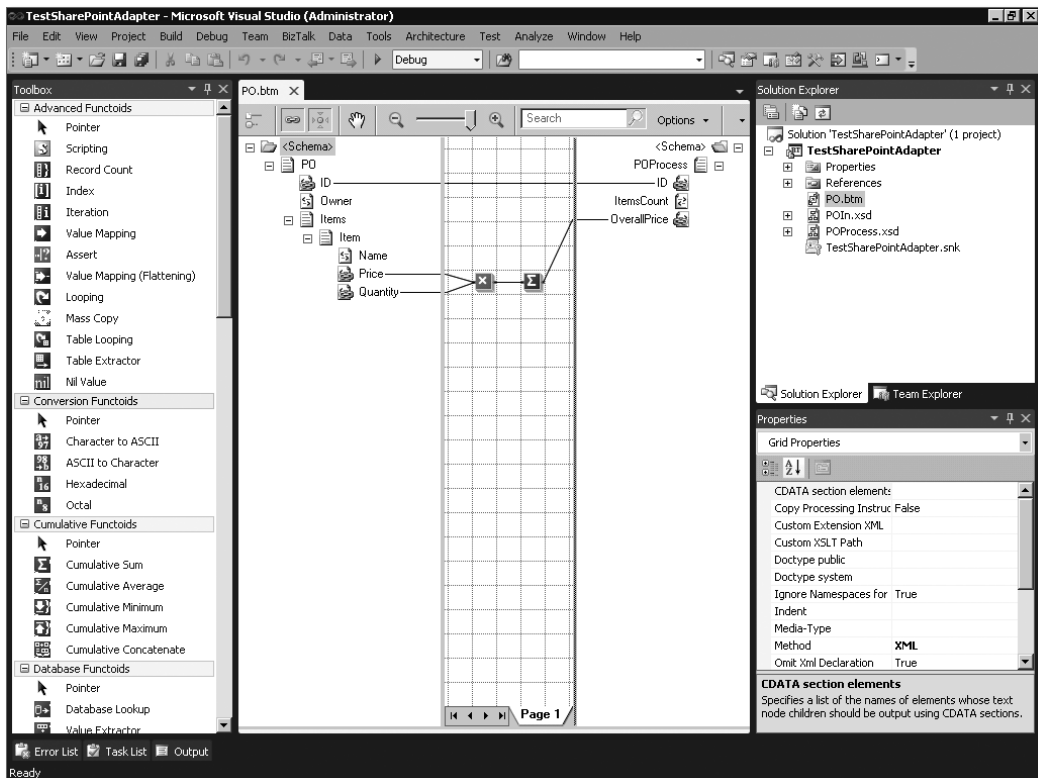
Im BizTalk Editor manipulieren Sie nicht den *XSD-Code* im rechten Bereich des Fensters, sondern links die Baumstruktur. Über das Properties-Fenster (normalerweise rechts unten in Visual Studio befindlich) geben Sie Zusatzinformationen wie Datentypen an.

Können Sie das Nachrichtenschema nicht frei definieren, weil es beispielsweise von einer Anwendung oder einem Kommunikationspartner bereits festgelegt wurde, hilft oft der Einsatz eines Assistenten, mit dessen Hilfe Sie das Schema von bestehenden Nachrichtendateien halb automatisch auslesen können.

Den BizTalk Editor lernen Sie in Abschnitt 2.4 noch näher kennen.

### 2.3.3 Mappings

Oft müssen Sie die Werte aus Nachrichten, die einem bestimmten Schema folgen, in andere Nachrichten mit unterschiedlichem Schema überführen. Diese Beziehungen definieren Sie über den *BizTalk Mapper* aus den Entwicklerkomponenten des BizTalk Servers in Visual Studio (siehe Abbildung 2.18).



**Abbildung 2.18** Mit dem BizTalk Mapper legen Sie Beziehungen zwischen Quell- und Zielschemas fest.

Im BizTalk Mapper geben Sie links das Quell- und rechts das Zielschema an. Natürlich müssen Sie vorher diese Schemas bereits definiert haben. Über Verbin-

dungslinien setzen Sie die Schemas in Beziehung. Dabei ist oft keine 1:1-Zuordnung möglich, sondern der unterschiedliche Nachrichtenaufbau, andere Datentypen etc. müssen berücksichtigt werden. Auch kann es durchaus sein, dass für die neue Nachricht Berechnungen durchgeführt (z. B. Anzahl  $\times$  Preis) oder Werte aus einer Datenbank nachgeschlagen werden müssen (z. B. Produktinformationen, basierend auf einer ID aus der Quellnachricht). In diesen Fällen setzen Sie die sogenannten *Functoids* aus der Toolbox im BizTalk Mapper ein. Es handelt sich dabei um Funktionsbausteine aus den folgenden Bereichen:

- ▶ String Functoids
- ▶ Mathematical Functoids
- ▶ Logical Functoids
- ▶ Date/Time Functoids
- ▶ Conversion Functoids
- ▶ Scientific Functoids
- ▶ Cumulative Functoids
- ▶ Database Functoids
- ▶ Advanced Functoids

Die Functoids platzieren Sie auf dem sogenannten *Map Grid* zwischen Quell- und Zielschema und stellen die Verbindung zwischen den Schema-Einträgen und den Functoids her.

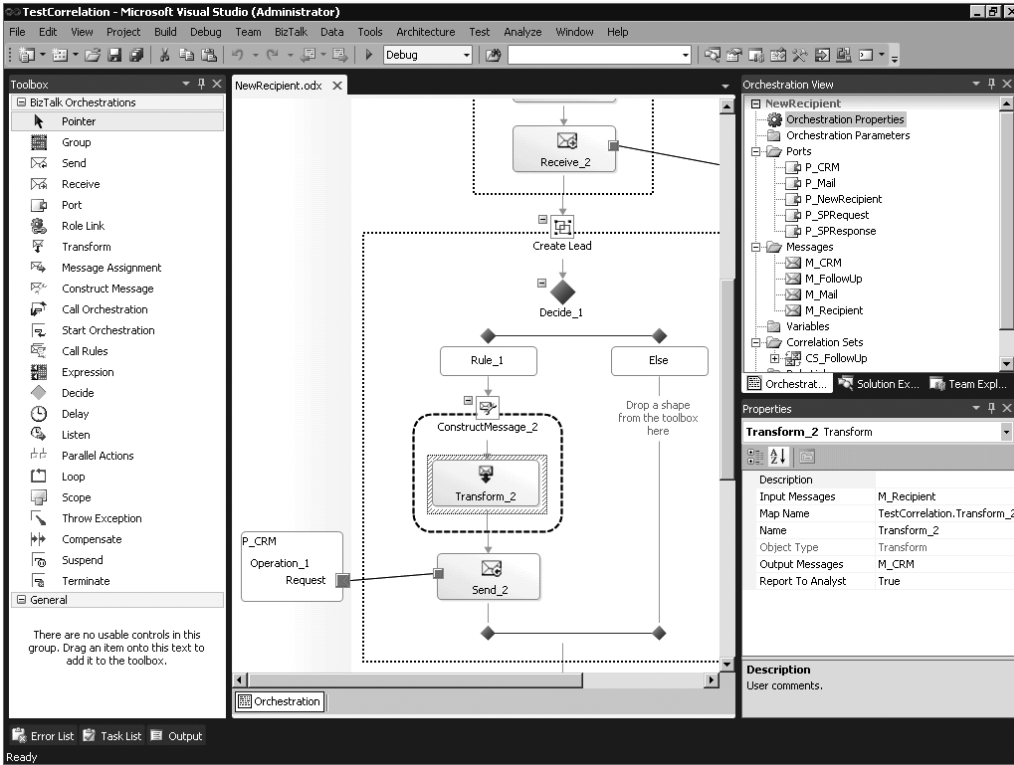
In der Toolbox finden Sie auch das *Scripting-Functoid*, bei dem Sie Programmcode, z. B. in Form einer in C# geschriebenen Funktion, hinterlegen können, um Nachrichtenwerte zu verarbeiten. Mit diesem Functoid stellen Sie auch spezielle Funktionen bereit, die in der Functoids-Liste nicht berücksichtigt sind. Mappings tragen in Visual Studio die Dateiendung *BTM*.

Abschnitt 2.5 beschreibt den BizTalk Mapper im Detail.

### 2.3.4 Orchestrations

Mit Orchestrations definieren Sie die Logik Ihrer BizTalk-Anwendung. An Stellen, an denen Sie bei herkömmlichen Anwendungen Programmcode schreiben würden, legen Sie bei einer BizTalk-Anwendung Orchestrations an. Diese werden über den *Orchestration Designer* aus den Entwicklertools des BizTalk Servers in Visual Studio als Flussdiagramm erzeugt (siehe Abbildung 2.19). Mit einer Orchestration definieren Sie, was wie und in welcher Reihenfolge mit eingehenden Nachrichten passieren soll. Orchestrations tragen in Visual Studio die Dateiendung *ODX*.

Der Orchestration Designer wird in Abschnitt 2.6 näher vorgestellt.



**Abbildung 2.19** Mit dem Orchestration Designer definieren Sie die Logik Ihrer BizTalk-Anwendung.

### 2.3.5 Pipelines

Eingehende und ausgehende Nachrichten werden in BizTalk über Pipelines verarbeitet, etwa zum Entkomprimieren/Komprimieren, Entschlüsseln/Verschlüsseln etc. BizTalk liefert eine Reihe von Pipelines für Standardszenarien mit. Reichen diese für Ihre BizTalk-Anwendung nicht aus, entwerfen Sie über den *Pipeline Designer* aus den BizTalk Server-Entwicklertools in Visual Studio eigene Pipelines (siehe Abbildung 2.20).

In Visual Studio wird dabei zwischen *Receive Pipelines* und *Send Pipelines* unterschieden, die jeweils aus unterschiedlichen Stufen bestehen, den sogenannten *Stages*. Die Dateiergung von Pipelines lautet *BTP*.

Mehr zum Pipeline Designer lesen Sie in Abschnitt 2.7.

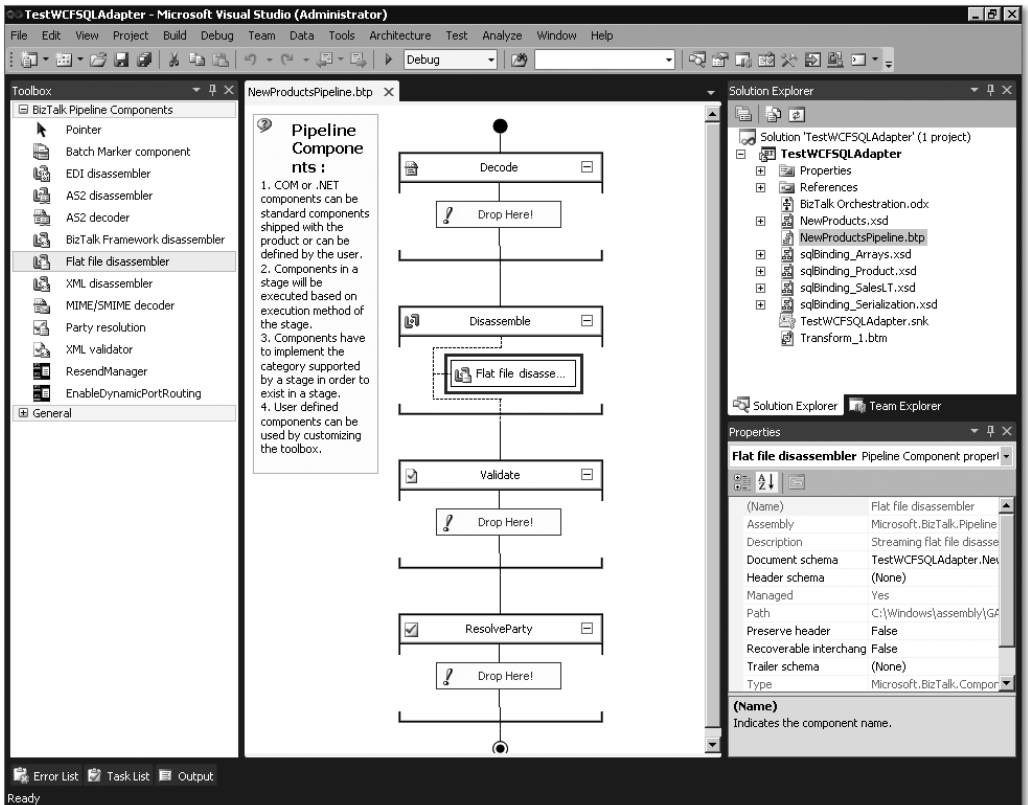


Abbildung 2.20 Mit dem Pipeline Designer entwerfen Sie eigene Pipelines.

### 2.3.6 Nicht-BizTalk-Assemblies

Neben dem/den BizTalk-Assemblys/-Assembly können zu den Bestandteilen einer Anwendung auch noch weitere .NET-Assemblys gehören, die an sich von BizTalk unabhängig sind und beispielsweise Funktionsbibliotheken enthalten. Diese Funktionen können von den BizTalk-Assemblys aus aufgerufen werden.

### 2.3.7 Anwendungskonfiguration

Doch nur die Assemblys allein machen noch keine lauffähige BizTalk-Anwendung aus, auch wenn sie wichtige Bestandteile davon enthalten. Es fehlt noch die Konfiguration, das sogenannte *Binding*. Dabei handelt es sich um wesentliche Einstellungen, etwa *Receive Ports* und *Send Ports*, die Ein- und Austrittspunkte für Nachrichten in und aus BizTalk-Anwendungen und gegebenenfalls auch die Zuordnung dieser Ports zu den Orchestrierungen aus den Assemblys.



*Genug der Theorie. Verfolgen Sie nun Schritt für Schritt verschiedene Einsatz-Szenarien.*

## 4 BizTalk-Einsatz

Beim Lesen der vorangegangenen Kapitel haben Sie viele Konzepte, Funktionen und Verfahrensweisen des BizTalk Servers kennengelernt. Es ist nun Zeit, das erworbene Wissen anhand verschiedener Szenarien in die Praxis umzusetzen.

Ich empfehle Ihnen, zu Beginn insbesondere das erste Szenario in Abschnitt 4.1, »Arbeiten mit Dateien«, durchzugehen, da dort einige sich immer wiederholende Schrittfolgen besprochen werden. Ob Sie dann alle weiteren Szenarien sequenziell bearbeiten oder je nach Lust und Laune zu bestimmten Szenarien springen, überlasse ich Ihnen.

Bei der jeweiligen Szenariobeschreibung finden Sie auch Hinweise, in welchem Beispiel welcher Adapter und welche BizTalk-Artefakte zum Einsatz kommen. Das gibt Ihnen die Möglichkeit, gezielt bestimmte Beispiele anzusteuern.

Ein umfassendes Szenario finden Sie in Abschnitt 4.8, »Korrelationen und mehr«.

### Aufgeführte Beispiele zum Download

Auf der Bonus-Seite zu diesem Buch, die Sie unter <http://www.galileocomputing.de/bonus-seite> finden, können Sie die Visual Studio-Projekte aller hier aufgeführten Beispiele herunterladen.

[+]

### Überblick über die behandelten Szenarien

#### **Szenario 1: Arbeiten mit Dateien**

Mittelpunkt des Szenarios bildet der *File-Adapter*. Nach einer Einführung in diesen Adapter werden zwei Beispiele besprochen:

- Beispiel 1: Weiterleitung von Dateien ohne Orchestration

Dateien werden mit dem File-Adapter aus Ordner A gelesen und in Ordner B abgelegt. Dabei wird die Konfiguration der Anwendung mit Receive Port und Send Port besprochen. Der Send Port wird über einen Filter angewiesen, die empfangenen Nachrichten weiterzuleiten.

Eingesetzte BizTalk-Adapter: File

Eingesetzte BizTalk-Artefakte: –

► Beispiel 2: Weiterleitung von Dateien mit Orchestration

Als Erweiterung von Beispiel 1 wird das Verarbeiten der eingelesenen Nachricht nicht über einen Filter des Send Ports realisiert, sondern über eine Orchestration. Das Anlegen einer einfachen Orchestration und das Deployment des Visual Studio-Projekts werden hier ausführlich erläutert.

Eingesetzte BizTalk-Adapter: File

Eingesetzte BizTalk-Artefakte: Orchestration

**Szenario 2: FTP-Server ansprechen**

Nach der Besprechung des *FTP-Adapters* wird das Entwicklungssystem um die optionale FTP-Komponente der IIS erweitert, damit ein FTP-Server für das Szenario vorhanden ist. Wie im ersten Szenario geht es hier um die grundsätzliche Arbeitsweise des Adapters. Auch hier wird ein Ordner eines FTP-Servers überwacht, Dateien werden ausgelesen und in einem weiteren FTP-Ordner abgelegt. Eine Orchestration kommt nicht zum Einsatz.

Eingesetzte BizTalk-Adapter: FTP

Eingesetzte BizTalk-Artefakte: –

**Szenario 3: E-Mails empfangen und versenden**

In diesem Szenario werden die Adapter *POP3* und *SMTP* besprochen. Für das Beispiel wird gezeigt, wie das optionale Feature *SMTP-Server* des Windows Server 2008 R2 installiert und konfiguriert wird, um das Beispiel selbst nachvollziehen zu können. Dort werden mithilfe des SMTP-Adapters E-Mails versendet. Der Inhalt der E-Mail wird dabei über eine dynamische Konfiguration mithilfe einiger C#-Codezeilen innerhalb einer Orchestration definiert.

Der Ablauf gestaltet sich wie folgt: Ein Schema für eingehende XML-Dateien wird angelegt. Diese Dateien sollen über den File-Adapter eingelesen werden, und daraufhin soll eine Orchestration instanz starten. Die Datei enthält die Angaben für die E-Mail. In der Orchestration wird ein dynamischer Send Port mit dem SMTP-Adapter konfiguriert und der Versand in die Wege geleitet.

Eingesetzte BizTalk-Adapter: SMTP, File

Eingesetzte BizTalk-Artefakte: Schema, Orchestration

#### **Szenario 4: Webservices konsumieren und publizieren**

Der Zugriff und die Bereitstellung auf Webservices werden in diesem Szenario über die Adapter *WCF-Basic-Http* und *WCF-WSHttp* durchgeführt. Besprochen werden zwei Beispiele:

► Beispiel 1: Konsumieren eines Webservice

Über ein separates Visual Studio-Projekt wird ein einfacher Webservice auf Basis von WCF entwickelt und gestartet (Addition zweier Zahlen und Rückgabe der Summe). Mithilfe des *BizTalk WCF Service Consuming Wizard* werden dann für diesen Webservice die notwendigen Schemas und eine Orchestration erzeugt. Über eine Eingangsnachricht, die die zwei zu addierenden Zahlen enthält, wird die Orchestration-Instanz gestartet. Die Daten aus der Datei werden dann über ein Mapping in das erforderliche Nachrichtenformat überführt, und der Aufruf des Webservice wird durchgeführt. Das Ergebnis wird in eine Datei geschrieben.

Eingesetzte BizTalk-Adapter: WCF-Basic-Http

Eingesetzte BizTalk-Artefakte: Schema, Map, Orchestration

► Beispiel 2: Publizieren eines Webservice

Eine Orchestration wird als Webservice über eine *IIS Web Site* publiziert und von einer separaten C#-Konsolenanwendung aus angesprochen. Das Publizieren des Webservice wird mithilfe des *BizTalk WCF Service Publishing Wizard* vorgenommen.

Eingesetzte BizTalk-Adapter: WCF-WSHttp

Eingesetzte BizTalk-Artefakte: Schema, Orchestration

#### **Szenario 5: Zugriff auf SQL-Datenbanken**

Für den Zugriff auf SQL-Datenbanken wird der *WCF-SQL-Adapter* aus dem *BizTalk Adapter Pack* herangezogen. Im Beispiel werden für eine Flatfile-Datei ein passendes Schema sowie eine Receive Pipeline angelegt, um entsprechend aufgebaute Dateien in der Anwendung auswerten zu können. Diese Dateien enthalten Informationen zu neuen Produkten. Mit dem *Add Adapter Wizard* werden die nötigen Schemas einer Produkttabelle aus einer SQL-Datenbank erzeugt. Die Produktdaten aus der Datei sollen dieser Tabelle hinzugefügt werden.

Eingesetzte BizTalk-Adapter: WCF-Custom mit sqlBinding (Send)

Eingesetzte BizTalk-Artefakte: Schema (Flatfile und XML), Orchestration, Pipeline

### **Szenario 6: SharePoint-Anbindung mit und ohne InfoPath**

Im Szenario werden die Komponenten des *WSS-Adapters* sowie seine Installation, Konfiguration und potenzielle Einsatzgebiete mit und ohne *InfoPath* für den Zugriff auf *SharePoint*-Inhalte besprochen.

Im Beispiel wird dann zunächst ein Schema erzeugt. Dieses Schema fungiert als Grundlage für ein *InfoPath*-Formular. Das Formular wird in einer *SharePoint*-Formularbibliothek als Vorlage publiziert. Ausgefüllte Formulare werden in der Bibliothek abgelegt.

Diese Bibliothek wird dann in einer *BizTalk*-Anwendung vom *WSS-Adapter* als Nachrichtenquelle überwacht. Eingegangene Nachrichten werden ausgewertet, in einem Mapping diverse Berechnungen durchgeführt, und das Ergebnis wird in einer weiteren *SharePoint*-Bibliothek abgelegt.

Eingesetzte *BizTalk*-Adapter: *WSS*

Eingesetzte *BizTalk*-Artefakte: Schema, Map, Orchestration

### **Szenario 7: Dynamics CRM-Anbindung**

Grundlage für die *Dynamics CRM*-Anbindung ist hier der separat erhältliche *CRM-Adapter* von Microsoft. Die Installation und Konfiguration des Adapters wird hier besprochen.

Im darauf folgenden Beispiel wird ein Schema für die Eingangsnachrichten erzeugt. Diese werden über den *File-Adapter* eingelesen und enthalten Daten für neue Kontakte zum Anlegen im *CRM*-System. Auch hier erzeugt ein Assistent die notwendigen Schemas. Es wird ein Mapping angelegt, das die Werte der Eingangsnachricht in ein Format bringt, wie es der *CRM-Adapter* voraussetzt. Die Antwort des Adapters wird über den *File-Adapter* ins Dateisystem geschrieben.

Eingesetzte *BizTalk*-Adapter: *CRM*, *File*

Eingesetzte *BizTalk*-Artefakte: Schema, Map, Orchestration

### **Szenario 8: Korrelationen und mehr**

Das letzte Szenario fasst einige Punkte der bisherigen Szenarien zusammen und verwendet mit der Korrelation eine bisher nur theoretisch angesprochene Technik. Das Ziel dabei ist, regelmäßig neue Datensätze aus einer *SQL*-Datenbank auszulesen. Mit diesen Daten wird ein *InfoPath*-Formular gefüllt und in einer *SharePoint*-Bibliothek abgelegt. Die Anwendung wartet dann auf die Bearbeitung durch eine Person und reagiert mit dem Anlegen eines Leads in *Dynamics CRM* und/oder durch das Absenden einer dynamisch generierten Mail.

Eingesetzte BizTalk-Adapter: WCF-Custom mit sqlBinding (Receive), WSS, SMTP, CRM

Eingesetzte BizTalk-Artefakte: Schema, Map, Orchestration

## 4.1 Arbeiten mit Dateien

Wir beginnen mit der Anbindung einer Anwendung an das Dateisystem. Dabei sollen Dateien aus lokalen Ordnern und Netzwerkfreigaben gelesen und in diese geschrieben werden. Zum Einsatz kommt dabei mit dem *File-Adapter* einer der einfacheren Vertreter seiner Gattung.

### 4.1.1 Arbeitsweise

BizTalk bringt für den Empfang, also das Einlesen von Dateien, und den Versand, das Schreiben von Dateien, den File-Adapter out of the box mit. Mit ihm sind sowohl der Zugriff auf das lokale Dateisystem als auch auf Netzwerkfreigaben möglich.

#### File-Adapter in Receive Locations

Setzen Sie den File-Adapter in einer Receive Location ein, konfigurieren Sie einen Pfad, der überwacht werden soll. Der Adapter überprüft dann in einem regelmäßigen, konfigurierbaren Intervall (*Polling Intervall*) den Inhalt des Pfades. Das Polling Intervall hat aber keine Auswirkung, wenn der Pfad auf einen lokalen Ordner eines NTFS-formatierten (= New Technology File System) Laufwerks zeigt. Hier wird der Dateiempfang über Events eingeleitet, die vom NTFS-Filesystem erzeugt werden. Wird eine Datei gefunden, sperrt sie der Adapter, um eine Änderung des Dateiinhalts während der Verarbeitung zu verhindern. Nachdem sie eingelesen wurde, wird die Datei gelöscht, um eine Mehrfachverarbeitung auszuschließen.

Um eine möglichst schnelle Verarbeitung zu gewährleisten, verarbeitet der File-Adapter nicht jede Datei einzeln, sondern in Gruppen (*Batches*), wodurch der Datendurchsatz erhöht werden kann, weil unter anderem die Anzahl der Zugriffe auf die Message Box minimiert wird. Im Standardfall ist die Größe eines Batches auf 20 Dateien konfiguriert. Die Receive Location wird also bis zu 20 Dateien (wenn so viele nicht vorhanden sind, auch weniger, aber nicht mehr) auf einmal einlesen und im Bearbeitungsprozess weiterleiten. Ist Ihre Anwendung aber dazu gedacht viele MB-große Nachrichten zu verarbeiten, kann es sich lohnen, die maximale Nachrichtenanzahl eines Batches zu verkleinern. Alternativ dazu können Sie auch die Maximalgröße in Bytes eines Batches verändern. Diese steht standardmäßig auf 100KB.

### Voraussetzungen

Damit der File-Adapter auf den Pfad zugreifen kann, müssen Sie dafür sorgen, dass der Receive Handler der Receive Location (also das Benutzerkonto der Host Instance) mit folgendem Rechten ausgestattet ist:

- ▶ Zeigt der Pfad in das lokale Dateisystem: *List Folder, Read Data, Delete Subfolder, Delete Files*
- ▶ Zeigt der Pfad auf eine Netzwerkfreigabe: *Full Control*

Eine weitere Voraussetzung ist das *NTFS-Dateisystem* (= *New Technology File System*). Beachten Sie außerdem, dass der File-Adapter nur Dateien entgegennimmt, bei denen weder das *Read Only*- noch das *System-Attribut* gesetzt sind. Derartige Dateien bleiben einfach im Pfad unverarbeitet liegen.

### File-Adapter in Send Ports

Bei der Konfiguration des File-Adapters für die Senderichtung geben Sie einen Pfad zu einer Datei an. Dieser Pfad kann Platzhalter, etwa das Sternchen als Wildcard-Zeichen, enthalten. Anstelle des Sternchens wird beim Anlegen einer Datei die sogenannte *Message ID* gesetzt. Dabei handelt es sich um ein eindeutiges Identifizierungsmerkmal, mit dem BizTalk jede Nachricht kennzeichnet. Neben dem Sternchen können Sie auch sogenannte *Makros* als Platzhalter verwenden. Mit diesen greifen Sie auf verschiedene Metadaten zu, die BizTalk mit jeder Nachricht verwaltet. Die eben angesprochene Message ID gehört ebenfalls dazu. Tabelle 4.1 zeigt eine Liste der möglichen Makros und ihre Bedeutung.

Makro	Bedeutung
%datetime%	<i>UTC-Datum</i> im Format YYYY-MM-DDThhmmss (Beispiel: 2005-11-12T160000)
%datetime_bts2000%	<i>UTC-Datum und -Zeit</i> im Format YYYYMMDDhhmmsss, wobei sss für Sekunden und Millisekunden steht (Beispiel: 2005111201020345 bedeutet 12.11.2005 um 10:20:34 Uhr und 500 Millisekunden.)
%datetime.tz%	<i>Lokalzeit plus GMT-Zeitzone</i> im Format YYYY-MM-DDThhmmssTZD (Beispiel: 2005-11-12T102034+100)
%DestinationParty%	Name des Handelspartners (Metadatum BTS.DestinationParty)
%DestinationPartyQualifier%	Qualifier des Handelspartners (Metadatum BTS.DestinationPartyQualifier)
%MessageID%	<i>Globally Unique Identifier (GUID)</i> der Nachricht (Metadatum BTS.MessageID)

**Tabelle 4.1** Makros des File-Send-Adapters

Makro	Bedeutung
%SourceFileName%	Name der ursprünglichen Datei, die vom File-Adapter gelesen wurde. Enthalten ist der Dateiname samt Suffix ohne den Pfad. Wurde die Nachricht nicht über den File-Adapter empfangen, wird der Platzhalter nicht ersetzt (Metadatum <code>FILE.ReceivedFileName</code> ). Dieses Makro nutzen Sie nur, wenn die zu versendende Nachricht identisch mit der empfangenen Nachricht ist.
%SourceParty%	Name des Handelspartners, von dem die ursprüngliche Nachricht empfangen wurde. Dieses Makro nutzen Sie nur, wenn die zu versendende Nachricht identisch mit der empfangenen Nachricht ist.
%SourcePartyQualifier%	Qualifier des Handelspartners, von dem die ursprüngliche Nachricht empfangen wurde. Dieses Makro nutzen Sie nur, wenn die zu versendende Nachricht identisch mit der empfangenen Nachricht ist.
%time%	UTC-Zeit im Format <code>hhmmss</code> (Beispiel: 102034)
%time.tz%	Lokalzeit plus GMT-Zeitzone im Format <code>hhmmssTZD</code> (Beispiel: 102034+100)

**Tabelle 4.1** Makros des File-Send-Adapters (Forts.)

Wollen Sie im Dateinamen Werte aus den Nachrichten selbst einsetzen, müssen Sie den Port in einer Orchestration über Programmcode konfigurieren. Der Port muss dann dynamisch konfiguriert werden. Wie das funktioniert, werden wir in Abschnitt 4.3, »E-Mails empfangen und versenden« noch sehen.

**Voraussetzungen**

Der File-Send-Adapter benötigt unter dem Zielpfad *Read-/Write-Rechte*, um die Datei schreiben zu können.

**4.1.2 Konfiguration**

Um die Arbeit mit dem File-Adapter kennenzulernen, legen wir im Folgenden eine einfache Anwendung an. Ein Ordner soll überwacht werden. Dateien, die in diesem Ordner abgelegt werden, sollen entgegengenommen und in einem anderen Ordner abgespeichert werden. Also eine einfache Verschiebe-Operation.

Wir werden dabei zunächst die komplette Anwendung über die BizTalk Server Administration Console anlegen, ohne etwa eine Orchestration mit Visual Studio zu definieren. Das werden wir in einem späteren Beispiel nachholen.

### Basiskonfiguration in der BizTalk Server Administration Console

Mit Ausnahme der Send Handler und der Receive Handler bietet der File-Adapter keine weitere Konfigurationsmöglichkeit bei den Platform Settings der BizTalk Server Administration Console. Wir können also direkt mit der Anwendung beginnen.

#### 4.1.3 Beispiel: Weiterleitung von Dateien ohne Orchestrierung

Das Beispiel besteht aus folgenden Schritten:

1. Anlegen der Anwendung
2. Anlegen des Receive Ports
3. Anlegen des Send Ports
4. Start und Test der Anwendung

#### Schritt 1: Anlegen der Anwendung

1. Öffnen Sie die BizTalk Server Administration Console aus dem Windows-Startmenü.
2. Klicken Sie mit der rechten Maustaste auf den Pfad BIZTALK SERVER ADMINISTRATION • BIZTALK GROUP • APPLICATIONS, und wählen Sie im Kontextmenü den Befehl NEW • APPLICATION.
3. Geben Sie als Anwendungsnamen `TestFileAdapter` ein, und schließen Sie das Fenster mit OK (siehe Abbildung 4.1).

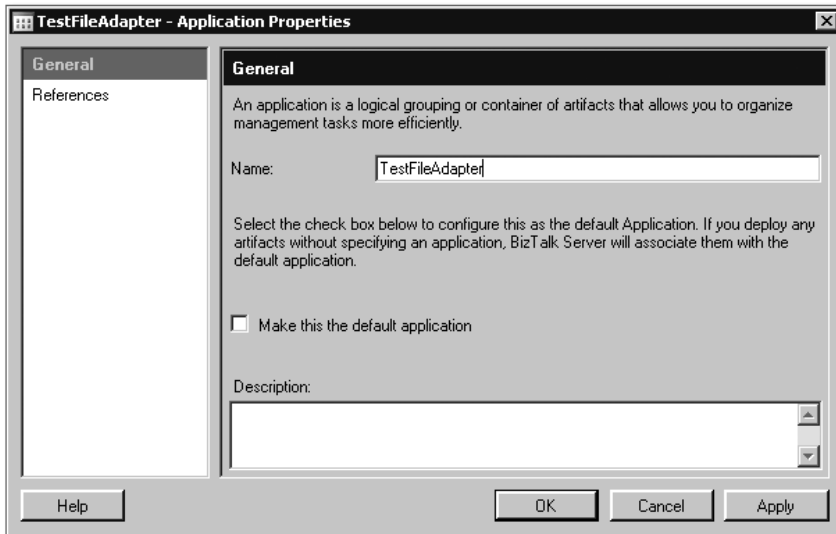


Abbildung 4.1 Eine neue Anwendung wird angelegt.



# Index

%datetime% 230  
%datetime.tz% 230  
%datetime\_bts2000% 230  
%DestinationParty% 230  
%DestinationPartyQualifier% 230  
%MessageID% 230  
%SourceFileName% 231  
%SourceParty% 231  
%SourcePartyQualifier% 231  
%time% 231  
%time.tz% 231  
&amp; 76  
&apos; 76  
&gt; 76  
&lt; 76  
&quot; 76  
.NET Framework 28, 32, 34, 51, 100, 185  
.NET Services 32  
.NET-Assembly 24, 25, 98, 100, 102, 107,  
136, 164, 165, 178, 179, 185  
<!-- 75  
--> 75

## A

---

Abonnement → Subscription  
Accelerator 35, 36  
Access Control Service → ACS  
Access Restrictions 297  
AccessViolationException 156  
ACID 171  
ACS 32  
Active Directory 46, 436  
Adapter 20, 29, 34, 37, 192, 200, 202  
    *Adapter Pack* 35, 36, 37, 39, 63, 305  
    *Anwendungsadapter* 20  
    CRM 37, 228, 229, 346, 362  
    DB2 39  
    *eBusiness* 39  
    EMS 38  
    *EnterpriseOne* 38  
    File 20, 37, 196, 200, 225, 226, 228, 229  
    FTP 20, 37, 226, 255  
    FTPS 37  
    *Host Applications* 38  
    *Host Files* 39

HTTP 37, 47  
LOB 36, 192  
MSMQ 37  
OneWorld 38  
Oracle 40  
PeopleSoft 38  
POP3 20, 37, 226, 268  
Protokolladapter 20, 192  
Receive 181, 196  
Rendezvous 38  
SAP 39  
Send 184  
SMTP 20, 38, 192, 226, 229, 268, 269,  
362  
SOAP 38, 47, 281  
SQL 20, 40, 304  
Technologieadapter 20  
WCF 38, 281  
WCF LOB Adapter SDK 40  
WCF-Basic-Http 281  
WCF-Basic-http 227  
WCF-Custom 227, 229, 305  
WCF-SQL 227, 304, 361  
WCF-WSHttp 227, 281, 282  
WebSphere 39  
WSS 20, 38, 47, 51, 191, 196, 200, 228,  
229, 322, 339, 361  
Add 282  
AddApp 222  
AddResource 222  
ADO.NET 142, 304  
ADOMD.NET 52  
AdventureWorksLT 306  
Agent 405  
Alias 436  
All Group 112  
All In-Progress Service Instances 394  
AMD 41  
AMD-V 41  
Anwenderkommunikation 30  
Anwendung 184  
    *anlegen* 193  
    *beenden* 206  
    *Bestandteile* 101, 194  
    *deinstallieren* 194, 220  
    *entfernen* 193

*exportieren* 193, 215  
*importieren* 193, 218  
*installieren* 218  
*Konfiguration* 107, 204, 206  
*starten* 206  
 Anwendungskommunikation 30  
 Any Attribute 114, 140  
 Any Element 113, 140  
 AppFabric 27, 31  
     *AppFabric Connect* 34  
     *Windows Azure AppFabric* 31, 32  
     *Windows Server AppFabric* 31, 33  
 Application View 408  
     *Application Artifact View* 409  
     *Application State View* 409  
     *Group State View* 409  
     *Host State View* 409  
 Archivierung 191  
 Argument 434  
 Artefakt 102, 211  
 ASCII 140  
 ASP.NET 190  
     *Development Server* 284  
 Assistent 65, 67  
     *Adapter Wizard* 350  
     *Add Adapter Wizard* 227, 311, 363  
     *Application Installation Wizard* 219  
     *BizTalk Flat File Schema Wizard* 307  
     *BizTalk WCF Service Consuming Wizard*  
         227, 281, 285  
     *BizTalk WCF Service Publishing Wizard*  
         227, 282, 295, 299  
     *Data Connection Wizard* 333  
     *Flatfile Schema* 94, 121  
     *Import MSI Wizard* 218  
     *Port Configuration Wizard* 159, 246  
     *Publishing Wizard* 375  
     *Schema* 118  
     *XML Schema* 81, 118  
 asynchron 163  
 Atomic 155, 171  
 Attach 412  
 Attachment 268  
 Attribut 75, 86, 103, 109, 110, 115  
 Attribute Group 113  
 Auschecken → Check Out  
 Ausfallsicherheit 188  
 Ausnahmeregel 261  
 Ausrufezeichen 245

Auto-Scrolling 148  
 Azure 32  
     *Microsoft Azure Platform* 31  
     *SQL Azure* 32  
     *Windows Azure* 32

## B

---

B2B 24, 35, 36  
 BAM 25, 28, 36, 44  
     *PlugIn* 48  
     *Portal* 25, 47, 51  
 Batch 229, 326  
 Baumstruktur 73, 109  
 Benutzerkonto 46, 51, 55, 57, 61, 64, 190,  
     201  
 Berechtigung 230, 231, 335, 359, 362  
 Best Practice Analyzer → BPA  
 Bestellanforderung → Purchase Order  
 Binding 54, 102, 107, 185, 210, 211, 215,  
     219  
     *importieren* 293  
     *übertragen* 221  
 BizTalk Adapter Pack → Adapter  
 BizTalk Administrators Group 59  
 BizTalk B2B Operators Group 59  
 BizTalk Editor 67, 80, 103, 108  
 BizTalk Expression Editor 161, 165, 167,  
     168  
 BizTalk Group 56, 178, 183, 190, 191, 218  
 BizTalk Management Database → Manage-  
     ment-Datenbank  
 BizTalk Mapper 67, 90, 104, 129, 337, 377,  
     383, 414  
 BizTalk MessageBox Database → Message  
     Box  
 BizTalk Operators Group 59  
 BizTalk Runtime → Runtime  
 BizTalk Server Administration 56, 64, 65,  
     99, 179, 180, 186, 193, 194, 197, 204  
 BizTalk Server Configuration 54, 64  
 BizTalk Services 32  
 BizTalk Tracking Database → Tracking-  
     Datenbank  
 BizTalk.System 178, 217  
 BizTalk-Assembly 102, 184, 194, 210, 215,  
     218  
 BizTalkDTADb → Tracking-Datenbank  
 BizTalkFactory PowerShell Provider →  
     PowerShell-Provider

BizTalkMgmtDb → Management-Datenbank  
 BizTalkMsgBoxDb → Message Box  
 BizTalkServerApplication 190  
 BizTalkServerIsolatedHost 190  
 Body 268  
 Bound 203  
 BPA 45, 62  
 BPM 35  
 BRE 164  
     *Business Rules Editor* 164  
 BTM 105, 130, 414  
 BTP 106, 175  
 BTSTask.EXE 179, 222, 417, 433, 436  
     AddApp 417  
     AddResource 418  
     ExportApp 426  
     ExportBindings 427  
     ExportSettings 428  
     ImportApp 428  
     ImportBindings 429  
     ImportSettings 430  
     ListApp 430  
     ListApps 431  
     ListPackage 431  
     ListTypes 432  
     RemoveApp 432  
     RemoveResource 432  
     UninstallApp 433  
 BTS-WIP 236, 238  
 Business Activity Monitoring → BAM  
 Business Profile 187  
 Business Rule 24  
     *Business Rule Composer* 25  
 Business-to-Business → B2B

## C

C# 48, 49, 105, 136, 165, 227, 295, 377  
 CAB 52  
 CDATA 76  
     76  
 Check Out 324  
 Cloud 31, 32  
 Cmdlet 434, 440  
     Add-ApplicationReference 440  
     Add-ApplicationResource 440, 441  
     Deploy-Policy 440  
     Disable-ReceiveLocation 441  
     Enable-ReceiveLocation 441, 442  
     Enlist-Orchestration 440  
     Enlist-SendPort 441  
     Enlist-SendPortGroup 441  
     Export-Application 440, 441  
     Export-Bindings 440, 442  
     Export-Policy 440  
     Get-ApplicationResourceSpec 440  
     Get-Command 440  
     Get-EventLog 434  
     Get-ExecutionPolicy 443  
     Get-ItemProperty 438  
     Get-Service 434  
     Import-Application 440  
     Import-Bindings 440  
     Import-Policy 440  
     New-PSDrive 437  
     Remove-ApplicationReference 440  
     Restart-HostInstance 440, 442  
     Restart-Service 434  
     Set-DefaultApplication 440  
     Set-ExecutionPolicy 443  
     Set-Location 436  
     Start-Application 440, 442  
     Start-HostInstance 440  
     Start-Orchestration 440  
     Start-SendPort 441  
     Start-SendPortGroup 441  
     Start-Service 434  
     Stop-Application 440, 442  
     Stop-HostInstance 440  
     Stop-Orchestration 440  
     Stop-SendPort 441, 442  
     Stop-SendPortGroup 441  
     Stop-Service 434  
     Undeploy-Policy 441  
     Unenlist-Orchestration 440  
     Unenlist-SendPort 441  
     Unenlist-SendPortGroup 441  
 Codeplex 223, 435  
 Collection 92  
 Commandlet → Cmdlet  
 Comma-Separated Value → CSV  
 Communication Pattern 248  
 Compensation 172  
     Block 172  
 Complex Element → Element  
 complexType 84  
 Connection String 143  
 Consistent 171

Consume Adapter Service 311  
 Contract 284  
 Contribute 326  
 Correlation Set 151, 379  
     *Following* 152, 380  
     *Initializing* 152, 379  
 Correlation Type 151, 153, 379  
 CRM 17, 18, 20, 28, 313, 346, 382  
 crm\_action 354  
 CSV 93  
 Customer Experience Improvement Program  
     52  
 Customer Relationship Management → CRM

## D

---

Data Lock 171  
 Dateisystem 229  
 Datentyp 83  
 Debugging 391, 410  
 default 86  
 Default Application 99  
 dekodieren 176  
 Delegate 165  
 Delimited Structure 93, 121  
 Dependencies 217  
 Deployment 97, 164, 179, 202, 209, 226  
     *Einstellungen* 98  
     *Kommandozeile* 222  
 Deployment View 409  
     *BAM Components View und BAM Alerts*  
         409  
     *Deployment State View* 409  
     *Hosts State View* 409  
     *Runtime Alerts* 409  
     *Runtime Component Views* 409  
     *Runtime Role State View* 409  
 Designer 65, 67  
 Dienst 190  
 Direct 249  
 Distinguished Field 116, 275, 277  
 DLL 100, 102, 118, 184  
 DMZ 33  
 Document Type Definition → DTD  
 DOCX 70  
 Dokumentenbibliothek 191, 196, 200, 323  
 Domänencontroller 55, 57  
 Download-Paket 443  
 Drag-and-Drop 158, 249  
 DTD 118  
 Duplicate Detection 390  
 Durable 171  
 Durchschnittswert 142  
 Dynamic 250

## E

---

EAI 35  
 EDI 19, 20, 24, 34, 69, 94, 133, 181  
     *Message Types* 95  
 EDIFACT 34, 94, 95  
     *DELFOR* 95  
     *DESADV* 95  
     *IFTMIN* 95  
     *IFTSTA* 95  
     *INVOIC* 95  
     *INVRPT* 95  
     *ORDERS* 95  
     *ORDRSP* 95  
     *PAYMUL* 95  
     *PAYORD* 95  
     *PRICAT* 95  
     *PRODAT* 95  
 Edition 35  
     *Branch* 36, 37  
     *Developer* 37, 63  
     *Enterprise* 35, 36, 37, 44, 63  
     *Standard* 35, 37, 63  
 Electronic Data Interchange → EDI  
 Element 73, 103, 109, 110, 115  
     *einfach* 83  
     *komplex* 84  
 ElseIf 167  
 encoding 73  
 Enterprise Resource Planning → ERP  
 Enterprise SSO 56  
 Entität 348  
 Entity-Referenz 76  
 entkomprimieren 106, 182  
 entschlüsseln 106, 176, 182  
 Entwicklungsumgebung 40  
 ERP 17, 18  
 ErrorReport 403  
 Eskalation 169  
 Event 165  
 Event Type 396  
 Eventlog 277, 413  
 Excel 48

Exception 139  
 Exception Handler 155, 157, 169, 172  
 Execution Policy 443  
   *AllSigned* 443  
   *RemoteSigned* 443  
   *Restricted* 443  
   *Unrestricted* 443  
 ExportApp 222  
 ExportBindings 222  
 Extended Markup Language → XML  
 Extensible Stylesheet Language Transforma-  
   tion → XSLT  
 External Data Representation → XDR  
 Extranet 33

## F

---

F11 414  
 F5 252  
 F9 416  
 Failed Message Routing 391, 402  
 Feature 56  
 Fehlerbehandlung → Exception Handler  
 File-Receive-Adapter 236  
 File-Send-Adapter 238  
 Filter 207, 209, 225, 226, 402  
 Filter Expression 251  
 Firewall 33, 261  
 fixed 87  
 Flatfile 20, 34, 69, 93, 177, 179, 181, 197,  
   203  
 Flussdiagramm 67, 105, 148, 153, 160, 167  
 Footer 93, 122  
 For-Each 92  
 FTP 28, 196  
 FTP Server  
   *Kommandos* 255  
 FTP Site 258  
 FTP-Receive-Adapter 263  
 FTP-Send-Adapter 265  
 FTP-Server 226  
 FTPS-Server 255  
 Full Stop 242  
 Functoid 105, 130, 133, 134, 414  
    $10^n$  146  
   *Absolute Value* 145  
   *Add Days* 143  
   *Addition* 134, 145  
   *Advanced Functoids* 136  
   *Arc Tangent* 146  
   *ASCII to Character* 141  
   *Assert* 139  
   *Base-Specified Logarithm* 146  
   *Character to ASCII* 140  
   *Common Logarithm* 146  
   *Concatenate* 141  
   *Conversion Functoids* 140  
   *Cosine* 146  
   *Cumulative Average* 142  
   *Cumulative Concatenate* 142  
   *Cumulative Functoids* 141  
   *Cumulative Maximum* 142  
   *Cumulative Sum* 141  
   *Database Functoids* 142  
   *Database Lookup* 142  
   *Date* 143  
   *Date / Time Functoids* 143  
   *Date and Time* 144  
   *Division* 134, 145  
   *Equal* 138, 144  
   *Error Return* 143  
   *Format Message* 143  
   *Get Application ID* 143  
   *Get Application Value* 143  
   *Get Common ID* 143  
   *Get Common Value* 143  
   *Greater Than* 144  
   *Greater Than or Equal To* 144  
   *Hexadecimal* 141  
   *Index* 138  
   *Integer* 145  
   *IsNil* 144  
   *Iteration* 138  
   *Less Than* 144  
   *Less Than or Equal To* 144  
   *Logical AND* 144  
   *Logical Date* 144  
   *Logical Existence* 144  
   *Logical Functoids* 139  
   *Logical Functoids* 138, 144  
   *Logical NOT* 144  
   *Logical Numeric* 144  
   *Logical OR* 144  
   *Logical String* 144  
   *Looping* 140  
   *Lowercase* 147  
   *Mass Copy* 132, 140  
   *Mathematical Functoids* 145

*Maximum Value* 145  
*Minimum Value* 145  
*Modulo* 145  
*Multiplication* 145, 337  
*Natural Exponential Function* 146  
*Natural Logarithm* 146  
*Nil Value* 140  
*Not Equal* 144  
*Octal* 141  
*Record Count* 137, 338  
*Remove Application ID* 143  
*Round* 145  
*Scientific Functoids* 146  
*Scripting* 105, 136, 377, 416  
*Set Common ID* 143  
*Sine* 146  
*Size* 147  
*Square Root* 145  
*String Concatenate* 147  
*String Extract* 147  
*String Find* 146  
*String Functoids* 146  
*String Left* 147  
*String Left Trim* 147  
*String Right* 147  
*String Right Trim* 147  
*Subtraction* 145  
*Table Extractor* 140  
*Table Looping* 140  
*Tangent* 146  
*Time* 144  
*Uppercase* 134, 147  
*Value Extractor* 143  
*Value Mapping* 138  
*Value Mapping (Flattening)* 140  
*X^Y* 146

Funktionsbibliothek 107, 185

## G

---

GAC 100, 102, 136, 179, 185, 194, 218, 219  
 GACUTIL.EXE 101  
 Genehmigung 29  
 Generate Instance 115  
 Generic 165  
 Geschäftsprozess 17, 29, 361  
 Global Assembly Cache → GAC  
 Globally Unique Identifier → GUID

GMT 230  
 Group Hub Page 187, 392, 394  
 GUID 230, 377

## H

---

Handelspartner → Trading Partner  
 Handler  
     *Receive* 192, 201  
     *Send* 192  
 Handlungsstrang 166, 167, 169  
 HAT 392  
 Header 93, 122, 268  
 Health and Activity Tracking → HAT  
 Heterogene Systeme 20  
 HIPAA 34, 35, 37  
 HKCU 436  
 HKLM 436  
 HL7 34, 35, 37  
 Höherstufung → Promotion  
 Host 35, 36, 187, 188, 192, 201, 204  
     *Host Instance* 188  
     *In-Process* 61, 190  
     *Isolated* 61, 190, 282  
     *Standard* 190  
 Host Integration Server → Host  
 HTML 70, 75, 90  
 HTTP 190, 281  
 HTTPS 281  
 Hub 28, 36  
 Hydration 393  
 Hyperthreading 45  
 Hyper-V 43

## I

---

If-Then-Else 166  
 IIS 33, 47, 226, 257  
 IIS Host 217  
 IIS Web Site 227, 282, 298  
 ImportApp 222  
 ImportBindings 222  
 Inbound Logical Ports 253  
 Inbound Maps 202  
 InboundId 365  
 InfoPath 228, 322, 324, 327, 329, 361, 371  
     *InfoPath Filler* 324  
     *InfoPath Forms Services* 324  
 In-Parameter 162, 163

Input Instance 115  
 Input Parameter 134  
 Installation 40, 44  
 InstallDTD.vbs 118  
 InstallWFX.vbs 118  
 Intel 41  
 IntelliSense 161  
 Interface 284  
 Internet Explorer 71, 284  
 Internet Information Services (IIS)  
   Manager 258  
 Internet Information Services → IIS  
 IOException 156  
 ISAPI 190  
 ISO 8601 143  
 Isolated 171  
 Itanium 45

## J

---

Java 90  
 Job 399  
 JScript.NET 136

## K

---

Kennwort 214  
 Klasse 79  
 Kommentar 75  
 Komponente  
   .NET 175  
   AS2 decoder 177  
   AS2 disassembler 177  
   AS2 encoder 177  
   Batch Marker component 177  
   BizTalk Framework assembler 177  
   BizTalk Framework disassembler 177  
   COM 175  
   EDI assembler 177  
   EDI disassembler 177  
   eigene 177  
   EnableDynamicPortRouting 177  
   Flat file assembler 177, 179  
   Flat file disassembler 122, 177, 179, 310  
   MIME/SMIME decoder 177  
   MIME/SMIME encoder 177  
   Party resolution 177  
   ResendManager 177

*XML assembler* 177, 179, 203  
*XML disassembler* 177, 178, 197  
*XML validator* 177  
 komprimieren 106, 182, 183  
 Konfiguration → Binding  
 Konsolenanwendung 295  
 Korrelation 225, 228, 305, 361

## L

---

Lastverteilung → Load Balancing  
 Lead 228, 348, 361  
 Line of Business → LOB  
 Link 130, 131  
   *Direct Link* 132  
   *Link by Name* 132  
   *Link by Structure* 132  
   *Mass Copy* 132  
 ListApp 222  
 ListPackage 222  
 ListTypes 222  
 Lizenz 35  
 Load Balancing 34, 188, 191  
 LOB 17, 28  
 Locals-Fenster 416  
 Long Running 155, 171, 172  
 Lookup value 143

## M

---

Mail 226, 386  
 Makro 230, 270, 339  
 Management Pack 391, 405  
   *Installation* 406  
 Management-Datenbank 56, 59, 102, 179,  
   187, 191, 211, 218, 220, 437  
 Map Grid 105, 133, 134  
 Map Page 133  
 Map → Mapping  
 Mapping 102, 104, 130, 160, 181, 182,  
   183, 196, 201, 202, 227, 228, 229, 290,  
   337, 353  
   *Debugging* 414  
 max Occurs 85  
 Message 117, 151, 160, 161, 245  
 Message Box 35, 36, 59, 60, 172, 183, 191,  
   207, 229, 391, 399  
   *Master* 191

Message Database → Message Box  
Message Details 397  
Message Event 395  
Message Flow 181, 397  
Message ID 230, 238  
Message Tracking 391, 392, 398, 399  
Message Type 246  
Messaging Engine 21, 22  
Metadatum 403  
Microsoft Software Installation → MSI  
Microsoft Update 62  
Microsoft XML Core Services → MSXML  
Microsoft.BizTalk.DefaultPipelines 178  
Middle-Tier 31  
Middle-Tier-Hosting 32  
min Occurs 85  
Monitoring 391  
MP 406  
MSI 179, 193, 215, 216, 219  
MSXML 90  
Multi-part Message Type 152

## N

---

Namensraum 79, 82, 83  
    *Präfix* 82  
Namespace → Namensraum  
Name-Wert-Paar 75  
Native 115  
NAV 28  
Navigationsparadigma 436  
Netzwerkfreigabe 229  
New Technology File System → NTFS  
Newsletter 361  
Nicht-BizTalk-Assembly 107, 185, 218  
Node 87, 131  
    *Attribute* 88  
    *Eigenschaften* 115  
    *Element* 88  
    *Root* 88  
Non-BizTalk-Assembly → Nicht-BizTalk-  
    Assembly  
NTFS 46, 229, 230

## O

---

Objekt 79  
ODETTE 95  
ODX 105, 149

Office Open XML → OOXML  
On Premise 33  
OOXML 70  
Operation 249  
Operations Manager 2007 R2 Agent 406  
Operations Manager Console 406, 408  
optional 86  
Oracle 28  
Orca 216  
Orchestration 22, 24, 102, 105, 162, 163,  
    165, 183, 188, 202, 204, 209, 226, 227,  
    228, 229, 243, 275, 288, 296, 314, 354,  
    368, 401  
    *beenden* 206  
    *Debugger* 154, 400, 410  
    *Designer* 67, 105, 148, 243  
    *Engine* 22, 28  
    *Instanz* 183  
    *Parameters* 150  
    *Properties* 150  
    *starten* 206  
    *View* 148, 149  
    *Zustände* 206  
Outbound Logical Ports 253  
Outbound Maps 202  
Out-Parameter 162, 163  
Output 116  
Output Instance 115

## P

---

Pan Mode 148  
Parameter 434  
Partial Stop 242  
Partially Started 240  
Party 187  
PDF 177, 178, 183, 197  
PFX 98, 100  
Pipeline 67, 69, 102, 106, 122, 175, 196,  
    197, 201, 227  
    *Designer* 67, 106, 175, 310  
    *Komponenten* 177  
    *PassThruReceive* 178, 197  
    *PassThruTransmit* 178, 203  
    *Receive* 106, 175, 176, 182, 196, 309  
    *Send* 106, 175, 176, 183  
    *Stage* 176  
    *Standard* 197, 203  
    *Standardpipelines* 178



*XMLReceive* 178, 197  
*XMLTransmit* 179, 203  
 Platform Settings 187  
 Policy 195  
 Polling Intervall 229  
 POP3 196  
 Port 150, 196  
     *Dynamic One-way Send Port* 202, 273  
     *Dynamic Solicit-Response Send Port* 203  
     *logisch* 204, 253  
     *One-way-Receive* 197, 233  
     *physisch* 204, 253  
     *Receive* 107, 122, 151, 158, 181, 190,  
         195, 196, 198, 202, 203, 225, 233  
     *Request-Response-Receive* 197, 202, 233  
     *Send* 107, 151, 158, 183, 188, 190, 195,  
         196, 198, 200, 202, 203, 225, 226, 237,  
         402  
     *Send dynamisch* 226  
     *Send Port Group* 195, 203  
     *Send Port Zustände* 203  
     *Static One-way Send Port* 202, 237, 273  
     *Static Solicit-Response Send Port* 202  
     *Type* 152  
     *Zwei-Wege* 202  
 Port Surface 150, 158  
 Port Type 248  
 Positional Structure 93, 121  
 PowerShell 222, 433  
     *Engine* 442  
     *Windows PowerShell ISE* 442  
 PowerShell-Provider 223, 435, 436  
     *Installation* 436  
 Präfix 198, 204  
 prohibited 86  
 Projekt 97  
     *Debug-Configuration* 139  
     *Release-Configuration* 139  
 Projektvorlage 65, 66  
 Promotion 116, 153, 207, 209, 398  
     *Quick Promotion* 116  
 Properties-Fenster 149, 153  
 Property Field 116  
 Property Schema 153  
 PS1 442  
 Purchase Order 121, 327, 336

## Q

---

Qualifier 230  
 Quellnachricht 129

## R

---

Radio Frequency Identification → RFID  
 Read Only-Attribut 230, 240  
 Receive Location 181, 188, 195, 196, 198,  
     200, 203  
 Receive Pipeline 181  
 Record 109, 110, 115  
 Recordset 142  
 Reference Counter 399  
 Ref-Parameter 162, 163  
 Refresh 252  
 Relevance Tree View 148  
 Relevant Links 148  
 RemoveApp 222  
 RemoveResource 222  
 Repeating Table 330  
 required 86  
 Resource 196  
 RFID 26, 34, 35, 36, 37  
 Role Link 152, 194  
     *Type* 153  
 Rollback 172  
 Root 73  
 RosettaNet 34, 35, 37  
 Running Service Instances 392, 394, 412  
 Runtime 56, 60

## S

---

S/MIME 176  
 SAP 28, 305  
 sapBinding 305  
 Saxon 90  
 Schema 102, 103, 108, 160, 196, 226, 227,  
     228, 229, 274, 285, 295, 307, 327, 336  
     *Complex Type* 110  
     *CRM* 350  
     *Destination* 129  
     *EDI* 96  
     *Flatfile* 94, 118, 179  
     *Footer* 179  
     *groß* 147  
     *Header* 179

- Instanz* 108
- Simple Type* 110
- Source* 129
- SQL* 311, 363
- Trailer* 179
- validieren* 115
- XML* 79, 110
- schema valid → schemagültig
- schemagültig 80
- Schleife 92, 167
- Schreibrecht 191
- SCOM 391, 404
- Scope 141
- Script 442
  - Post-Processing* 211
  - Pre-Processing* 211
- SDK 66
- Secure Sockets Layer → SSL
- SELECT 305
- sequence 84
- Sequence Group 112
- Serialisierung 177
- Server Manager 47
- Servers 191
- Service Bus 32
- Service Instance 395, 397
- Shape 22, 411
  - Abbruchverhalten* 154
  - Business Rules Engine* 154
  - Call Orchestration* 150, 162, 163, 164, 170
  - Call Rules* 164
  - Compensate* 157, 173
  - Construct Message* 162
  - Decide* 116, 164, 166, 167, 382
  - Delay* 168, 169
  - Entscheidungen* 154
  - Expression* 116, 165, 273, 413
  - Group* 154, 368
  - Gruppierungen* 153
  - Listen* 169
  - Loop* 116, 167
  - Message Assignment* 161, 162, 273, 276, 386
  - Nachrichtenerstellung* 154
  - Nachrichtenübermittlung* 154
  - Orchestrationsaufruf* 154
  - Parallel Actions* 167, 170
  - Port* 150, 158, 160
  - Programmcode* 154
  - Receive* 158, 169
  - Role Link* 160
  - Scope* 155, 171
  - Send* 158
  - Start Orchestration* 150, 163, 164
  - Suspend* 169, 170
  - Terminate* 170
  - Throw Exception* 157
  - Transform* 160, 162, 290, 316, 356, 376, 383
  - Warteverhalten* 154
- SharePoint 20, 27, 28, 228, 322, 327, 376
  - SharePoint Products Configuration Wizard* 51
  - SPF* 28, 51
  - SPS* 28
  - Web Application* 326
- SharePoint Adapter Web Service 322, 325
- SharePoint Foundation → SharePoint
- SharePoint Server Object Model 322
- SharePoint Server → SharePoint
- Sibling 112
- Sicherheitsgruppe 57, 59, 61, 190
- Signatur 182, 197
- signieren 98, 100, 183
- Simple Element → Element
- Single Sign-on 56
- SMTP 19
- SMTP Server 192, 271
- SMTP.EmailBodyTextCharset 277
- SnapIn 434
- Snapshot 40, 42, 43
- SNK → PFX
- SOA Factory-Community 435
- SOAP 19, 190, 304
  - Header* 293
- Solution 97
  - Solution Explorer* 149
- Specify Later 249
- Spedition 195
- SQL 18, 28
- SQL Server 49, 55, 57, 183
  - SQL Server 2008 Express* 49
- sqlBinding 227, 229, 305
- SQL-Datenbank 227, 228, 304, 361
- SQL-Server
  - Agent* 399, 410
  - Configuration Manager* 399

SQLXML 52  
 SSL 47, 255, 268  
 SSO Administrators Group 347  
 Stage 175  
   *Assemble* 177, 179  
   *Decode* 176  
   *Disassemble* 176, 178, 179, 310  
   *Encode* 177  
   *Pre-Assemble* 177  
   *Resolve Party* 176, 178  
   *Validate* 176  
 Standardanwendung → Default Application  
 Standardnamensraum → Namensraum  
 Started 240  
 Status  
   *In Breakpoint (Active)* 412  
   *Resumable* 396  
   *Suspend* 169  
   *Suspended* 176, 391, 396  
   *Suspended (Resumable)* 172  
 Stopped 240  
 Stored Procedure 304, 305  
 stornieren 173  
 Strong Name Keyfile → SNK  
 Subjekt 434  
 Subscription 183, 207, 239, 250, 399, 402  
 Suffix 199  
 Summe 141  
 Suspended Service Instances 394  
 SWIFT 34, 35, 37, 95  
 synchron 162, 170  
 System Center Operations Manager → SCOM  
 System.BizTalk  
   *BizTalkBinding* 214  
 System.Xml.XmlDocument 246  
 System-Attribut 230, 240  
 Systemsteuerung 47, 194, 220

## T

---

Target Environment 214, 219  
 Target Framework 100  
 Target Staging Environment 219  
 targetNamespace 83  
 TestMap Input 416  
 Testumgebung 209  
 TLS 255  
 Toolbox 105, 134, 149, 153, 175

Tooltip 136  
 Tracked Message Events 394, 395, 397  
 Tracked Service Instances 394, 395, 400  
 Tracking 187  
 Tracking Database → Tracking-Datenbank  
 Tracking-Datenbank 59, 391, 399  
 TRADACOMS 94  
 Trading Partner 24, 160, 187, 194, 230  
 Transaktion 34, 157, 171  
   *Transaction Type* 155, 171  
 Transformation 90, 129, 207  
 Translation 130  
 Transport Level Security → TLS  
 Transport Type 200, 203  
 Trennzeichen 93, 121  
 Try/Catch 155  
 T-SQL 304  
 type 83  
 TypedPolling 365

## U

---

Überwachung → Monitoring  
 unbounded 85  
 Unexpected Error 277  
 Unicode 73, 90, 121  
 Uniform Resource Identifier → URI  
 UninstallApp 222  
 Unit-Test 209  
 Upgrade 44  
 URI 79, 293  
 use 86  
 UTC 230

## V

---

Validate Instance 116  
 Validate Schema 116  
 value-of 93  
 Variable 151  
 VB.NET 48, 136  
 VBS 118  
 Verb 434  
 Verkettung 142, 333  
 Verknüpfung → Link  
 verschlüsseln 106, 177, 182, 183  
 version 73  
 View 323  
 Virtual PC 41, 43

Visual Basic.NET → VB.NET  
Visual Studio 27, 46, 48, 65, 71, 97, 102,  
185, 193, 203, 213, 401, 414  
VMware Workstation 41, 42  
Voraussetzungen  
    *Hardware* 45  
    *Software* 46  
Vorlage  
    *BizTalk Orchestration* 149  
    *Empty BizTalk Server Project* 97  
    *Flat File Schema* 109  
    *Map* 130  
    *Schema* 109  
VT 41

## W

---

W3C 70, 109  
WCF 19, 32, 34, 227  
    *WCF Service Configuration Editor* 33  
WCF LOB Adapter SDK 304  
WCF Service 282, 283, 304  
Web location 283  
Web Service Description Language → WSDL  
Web-Ressource 216  
Webservice 18, 28  
    *konsumieren* 227, 281  
    *publizieren* 227, 281, 282, 295  
Well-Formed XML → WFX  
WF 27, 32, 34  
WFX 118  
While 167  
Wildcard 230  
Windows  
    *Windows 7* 44, 46  
    *Windows Server 2008* 44, 46  
    *Windows Vista* 44, 46  
Windows Communication Foundation →  
    WCF  
Windows Firewall → Firewall  
Windows Management Framework → WMF  
Windows Management Instrumentation →  
    WMI  
Windows Platform SDK 216  
Windows PowerShell → PowerShell

Windows Update 62  
Windows Workflow Foundation → WF  
Windows-Sicherheitsgruppe →  
    Sicherheitsgruppe  
WMF 436  
WMI 211  
wohlgeformt 77  
Workflow 27, 28  
    *Workflow Engine* 24, 28  
Wrapper 177  
Write Cache 238  
WS-\*-Protokoll 281  
WSDL 282, 285  
WS-I Basic Profile 281  
WSS-Receive-Adapter 340  
WSS-Send-Adapter 342

## X

---

X12 34, 94  
XDR 118  
XML 20, 21, 69, 70, 181  
    *Deklaration* 73, 82  
    *Prolog* 73, 82  
    *Regelwerk* 77  
XML Path Language → XPath  
XML Schema Definition → XSD  
XML Schema → XML  
XML-Editor 213  
xmlns 79, 83  
XPath 21, 71, 87, 92  
xs 82  
XSD 21, 80, 103, 108, 118  
XSL 414  
XSL Transformation → XSLT  
XSLT 21, 71, 90, 136, 160  
    *Processor* 90

## Z

---

Zeichensatz 73  
Zertifikat 176  
Zielnachricht 129  
ZIP 70, 177  
Zoom 148