3 Der Internet Information Server

3.1 »IIS« oder »Internet-Informationsdienste«?

Bis Windows 2000 heißt der Webserver von Microsoft »Internet Information Server« oder kurz »IIS«. Ab Windows 2000 verwendet Microsoft den Begriff »Internet-Informationsdienste«, wohl um klarzustellen, dass der IIS nun ein integraler Bestandteil von Windows ist. Viel Unterschied gibt es allerdings nicht zwischen dem IIS 4 und den Internet-Informationsdiensten. Beide laufen als Windows-Dienst und bieten prinzipiell denselben Umfang an Möglichkeiten. Ich verwende in diesem Buch der Einfachheit halber die Bezeichnung »IIS«, die auch ein Kürzel ist für »Internet Information Services«, und meine damit den IIS 5, der unter Windows 2000 läuft.

3.2 Die Komponenten des IIS

Der IIS ist einmal ein Webserver, der neben den normalen Webserver-Eigenschaften (Abruf von HTML- und anderen Dateien, Aufruf von CGI-Programmen) auch ASP- und ASP.NET-Programme ausführen kann. Er integriert aber auch andere Server, wie beispielsweise den FTP-Server. ASP(.NET)-Programme arbeiten zudem häufig mit weiteren Servern, Diensten oder Komponenten wie beispielsweise COM+ oder dem Indexdienst. Damit Sie einen Überblick erhalten, welche Server, Dienste und Komponenten für die Internetprogrammierung wichtig sind, beschreibe ich diese im Folgenden kurz. Auf einige der beschriebenen Server, Dienste und Komponenten gehe ich im ASP.NET-Teil näher ein.

Der FTP-Server

Der in den IIS integrierte FTP-Server gibt, wie jeder FTP-Server, einem FTP-Client Zugriff auf freigegebene Ordner und Dateien auf dem Dateisystem des Rechners. Dazu legen Sie Ordner und Dateien im Basisverzeichnis des FTP-Servers ab (normalerweise ist das der Ordner *C:\Inetpub\ftproot*) oder verknüpfen beliebige Ordner als virtuelle FTP-Ordner mit dem FTP-Server. Unter Windows 2000 vergeben Sie für diese Ordner und Dateien Zugriffsrechte. In der Administration des FTP-Servers legen Sie fest, ob ein anonymer Zugriff möglich ist. Lassen Sie diesen zu, kann jeder Benutzer auf die Dateien zugreifen.

Lassen Sie den anonymen Zugriff nicht zu, haben nur Benutzer Zugriff, die als normale Windows-Benutzer registriert sind und unter Windows einen passenden Zugriff auf die entsprechende Datei bzw. auf den Ordner besitzen. Die Dateien auf Ihrem System können Sie z.B. im Internet Explorer über die URL *ftp://localhost* abrufen.

Der NNTP-Server

Der NNTP-Server ist ein einfacher Newsserver, über den Sie eigene Newsgroups einrichten können. Dieser Server steht nur unter Windows NT Server und Windows 2000 Server zur Verfügung.

Der virtuelle SMTP-Server

Der virtuelle SMTP-Server ermöglicht das Senden von E-Mails in beliebigen Programmen. Die E-Mail wird dazu entweder in Textform im SMTP-Protokoll in ein spezielles Verzeichnis abgelegt (womit sie automatisch versendet wird) oder mit Hilfe von COM- oder .NET-Objekten versendet. Der SMTP-Server übernimmt die Weiterleitung an den Empfänger. Dazu ist nichts weiter notwendig als ein Internetzugang. Das Buch behandelt das Senden von E-Mails ab Seite■■ 1035.

Die FrontPage-Servererweiterungen

Die FrontPage-Servererweiterungen sind spezielle Komponenten, die auf dem Webserver installiert werden. Diese Komponenten ermöglichen externen Programmen wie Microsoft FrontPage und Visual Studio den direkten Zugriff auf die Dateien einer Website. Ein Entwickler kann eine Website damit auch von entfernten Rechnern aus in einer passenden Entwicklungsumgebung bearbeiten.

Die ISAPI-Schnittstelle

Über das Internet Server API (ISAPI) können Sie mit beliebigen Programmiersprachen Internetanwendungen entwickeln, wenn Sie auf den Komfort der ASP(.NET)-Programmierung verzichten wollen. Der IIS stellt Ihnen dazu eine ISAPI-Schnittstelle zur Verfügung. Eine Internetanwendung wird dazu in Form einer *.dll*-Datei kompiliert. Diese Datei exportiert einige durch den ISAPI-Standard festgelegte Funktionen, die vom Webserver aufgerufen werden. Zur Programmierung können Sie spezielle Funktionen aus den API-DLLs des ISAPI nutzen. Sie können ISAPI-Erweiterungen und ISAPI-Filter programmieren. ISAPI-Erweiterungen werden über einen URL aufgerufen und enthalten Internetprogramme, die – wie ASP – auch HTML-Seiten erzeugen können. ISAPI-

Filter sind intern mit bestimmten einfachen Ereignissen verknüpft, die beim Abrufen eines Webdokuments auftreten. Dazu gehören z.B. Ereignisse wie das Verarbeiten der HTTP-Header einer Anforderung und die Authentifizierung des Clients. Ein ISAPI-Filter wird automatisch vom Webserver aufgerufen, wenn eine Webanforderung das Ereignis auslöst, für das der Filter registriert ist.

ISAPI-Erweiterungen sind mittlerweile veraltet, da ASP und vor allen Dingen ASP.NET wesentlich einfacher zu programmieren sind und mehr Möglichkeiten bieten. ISAPI-Filter sind u.U. noch sinnvoll, da diese Filter besondere Techniken erlauben. So können Sie beispielweise eine von Windows unabhängige Benutzer-Authentifizierung programmieren (was aber auch recht einfach unter ASP.NET möglich ist, wie ich ab Seite ■■1041 zeige): Ein ISAPI-Filter reagiert auf das Ereignis »Benutzer-Authentifizierung«, liest die im HTTP-Header mitgelieferten Login-Informationen aus und vergleicht diese mit den in einer Datenbank gespeicherten Benutzerkonten.

Weitere Informationen zu ISAPI finden Sie in der Online-Dokumentation des IIS (*localhost/iisHelp/iis/misc/default.asp*), indem Sie »ISAPI« als Suchbegriff eingeben, oder im MSDN (*msdn.microsoft.com/library*), indem Sie nach »Developing ISAPI Extensions» und »Developing ISAPI Filters« suchen.

Der Zertifizierungsserver

Der Zertifizierungsserver (Microsoft Certificate Server) erstellt und verwaltet digitale Zertifikate. Digitale Zertifikate bestätigen die Echtheit von Unternehmen und Einzelpersonen im Internet. Diese Zertifikate werden für viele abgesicherte Anwendungen im Internet benötigt. Der sichere Zugriff auf Webseiten über SSL (Secure Sockets Layer) ist z.B. nur mit gültigen Zertifikaten möglich. Der Person oder dem Unternehmen wird dazu ein öffentlicher Schlüssel zugeordnet, der im Zertifikat zusammen mit dem Namen der Person bzw. des Unternehmens enthalten ist. Abgesicherte Dokumente werden mit dem Zertifikat signiert, womit der Empfänger überprüfen kann, ob das Dokument wirklich vom Sender kommt. Der Empfänger muss dazu lediglich den öffentlichen Schlüssel des Senders kennen.

Der Zertifizierungsserver steht nur in den Server-Varianten der Microsoft-Betriebssysteme zur Verfügung.

Der Indexdienst

Der Indexdienst, der ab Windows 2000 integraler Bestandteil des Betriebssystems ist, ermöglicht die Volltextsuche in Dokumenten. Mit Hilfe dieses Dienstes können Sie mit wenig Programmieraufwand Suchseiten für das Dateisystem oder das Web eines Rechners entwickeln. Der Indexdienst verwaltet einzelne Kataloge mit Informationen zu den indizierbaren Dateien eines Systems, die in regelmäßigen Abständen im Hintergrund aktualisiert werden. Jeder Katalog kann das gesamte Datei- oder Websystem oder nur einen Teil davon indizieren. Im Indexdienst sind Filter registriert, mit deren Hilfe dieser auch Dokumente indizieren kann, die keine reinen Textdateien sind. Die Einrichtung und Programmierung des Indexdienstes, der früher übrigens als Index Server bezeichnet wurde, wird ab Seite ■■1007 behandelt.

Der Microsoft Transaction Server (MTS) bzw. COM+

Der Microsoft Transaction Server (MTS) ist unter Windows NT 4 ein separater Server, den Sie mit dem Windows NT Option Pack installieren können. In Windows 2000 sind die Dienste, die der MTS unter Windows NT anbietet, in Form der so genannten Komponentendienste (Component Services) bereits in COM+ integriert. Das mittlerweile veraltete Component Object Model (COM) stellt eine Infrastruktur zur Verfügung, die ermöglicht, dass Anwendungen Objekte verwenden können, deren Klassen in separaten Dateien (so genannten COM-Komponenten) in binärer Form gespeichert sind. Der MTS bzw. die Komponentendienste von COM+ erweitern die COM-Infrastruktur um wichtige Features wie ein flexibles und ressourcenschonendes Komponentenmanagement und das Handling von Transaktionen[1]. Um einem weit verbreiteten Irrtum direkt vorzubeugen: Der MTS ist nicht ausschließlich dazu da, Transaktionen zu handeln. Eine weitere, von Transaktionen zunächst unabhängige Aufgabe des MTS ist, die auf einem System installierten Komponenten so zu verwalten, dass möglichst wenig Ressourcen verbraucht werden und dass das Instanzieren von Objekten aus diesen Komponenten möglichst performant ist.

1 Als Transaktion wird in der Computerwelt eine datenverändernde Aktion bezeichnet. Werden innerhalb der Aktion mehrere unterschiedliche Datensätze (in einer oder mehreren Datenbanken) verändert, stellt eine Transaktion sicher, dass entweder alle oder – im Fehlerfall – gar keine Veränderungen stattfinden.

Komponenten können von normalen Anwendungen und von Webanwendungen verwendet werden. Besonders für Webanwendungen in klassischen ASP-Seiten spielen solche Komponenten eine große Rolle. Eine ASP-Seite, die für das eigentliche Programm eine Komponente verwendet, ist wesentlich schneller als eine ASP-Seite, die das Programm selbst enthält. Dummerweise erzeugt jeder Aufruf einer ASP-Seite eine neue Instanz der Komponente oder zumindest einer Klasse dieser Komponente. Greifen sehr viele Benutzer gleichzeitig auf die ASP-Seite zu, sind die Ressourcen des Rechners schnell ausgeschöpft. Ist die Komponente allerdings im MTS bzw. in COM+ registriert, werden die Ressourcen massiv geschont. Der MTS bzw. COM+ erzeugt dazu nur eine bestimmte Maximalanzahl an Komponenten und verteilt die verfügbaren dynamisch an die einzelnen Clients (in unserem Fall an die ASP-Seiten). Komplexe Techniken ermöglichen dabei eine optimale Performance.

Der MTS bzw. COM+ ist für Internetprogrammierer wichtig, die große Websites entwickeln, auf die gleichzeitig sehr viele Benutzer zugreifen.

3.3 IIS-Administration für Programmierer

Den IIS können Sie über ein Snap-In in der Microsoft Management Console (MMC) administrieren. Alternativ steht Ihnen in den Server-Versionen von Windows NT/2000/XP noch eine webbasierte Administration über ASP-Programme zur Verfügung, die Sie über die Adresse *localhost/IISAdmin* erreichen. In älteren Windows-Versionen können Sie daneben auch noch den Personal Web Manager verwenden, der aber nur sehr rudimentäre Administrations-Features bietet und häufig zu Verwirrungen führt weil er das gesuchte Feature nicht anbietet, obwohl es in der IIS-Administration vorhanden ist.

Die MMC bietet die besten Möglichkeiten und zeigt vor allen Dingen alle Optionen, was bei den anderen Administrations-Anwendungen nicht unbedingt der Fall ist. Ich beschreibe deshalb hauptsächlich die Administration unter der MMC.

Eine andere Möglichkeit der IIS-Administration ist die Verwendung von speziellen COM-Komponenten des IIS. So können Sie Programme entwickeln, die Administrationsaufgaben, wie beispielsweise das Anlegen von neuen virtuellen Webordnern, übernehmen. Diese Variante beschreibe ich aber nicht.

Die webbasierte Administration

Wenn der IIS unter Windows NT Server, Windows 2000 Server oder Windows XP Server läuft, können Sie zur Administration ASP-Dokumente verwenden. Der IIS installiert dazu eine spezielle separate Website, die Sie über den Port 8807 erreichen. Der virtuelle Ordner *IISAdmin* enthält die Dateien zur Administration. Die Administration erreichen Sie also folgendermaßen:

```
http://Rechnername:8807/IISAdmin
```

So können Sie einen Webserver auch recht einfach über das Intranet oder Internet administrieren. Die Administrationswebsite ist allerdings per Voreinstellung besonders geschützt. Sie lässt keine anonyme Anmeldung und auch nicht die Anmeldung über eine Standardauthentifizierung (über einen Login-Dialog im Browser) zu.

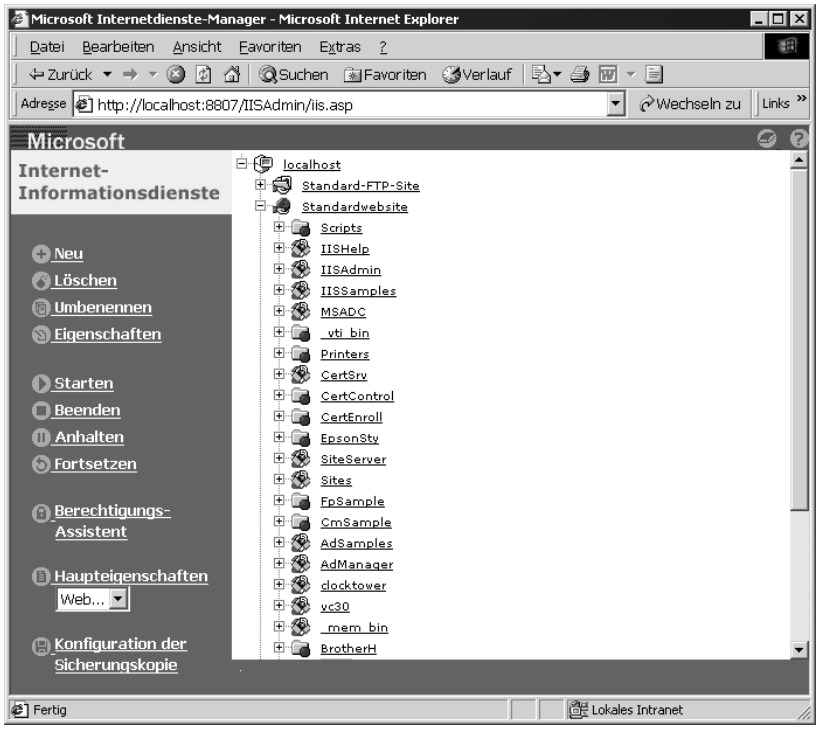

Abbildung 3.1: Die webbasierte Administration unter Windows 2000 Server (mit installiertem Site Server)

Lediglich Anwender, die in einem Intranet arbeiten, den Windows Explorer verwenden und ein Benutzerkonto auf dem Server besitzen, können sich zunächst über die so genannte integrierte Windows-Authentifizierung am Server anmelden (was allerdings automatisch nach den Logindaten des Benutzers am lokalen Rechner erfolgt). Diese Einstellungen können Sie natürlich auch ändern, so dass auch eine Anmeldung über das Internet möglich ist (wie ich es noch ab Seite■■ 101 beschreibe).

Ich beschreibe die webbasierte Administration nicht weiter, weil diese prinzipiell identisch mit der MMC-Administration ist.

Die MMC-Administration

Das MMC-Snap-In zur IIS-MMC-Administration ist in der Datei *iis.msc* definiert, die Sie unter Windows NT und Windows 2000 im Ordner *\Winnt\System32\Inetsrv* finden. Ein Doppelklick auf diese Datei öffnet die MMC. Sie können diese Datei aber auch anders öffnen: Wählen Sie dazu unter Windows NT START / PROGRAMME / WINDOWS NT OPTION PACK / MICROSOFT INTERNET INFORMATION SERVER / INTERNETDIENSTE-MANAGER. Unter Windows 2000 öffnen Sie den Ordner VERWALTUNG in der Systemsteuerung und starten dort den INTERNETDIENST-MANAGER. Die Administration ist unter den verschiedenen Windows-Versionen nahezu identisch. Ich beschreibe deshalb nur die neuere Version (IIS 5 unter Windows 2000).

Abbildung 3.2 zeigt die IIS-Administration in der MMC mit bereits aufgeklapptem Ordner für die Standardwebsite. Ordner mit einem einfachen Ordnersymbol bezeichnen Dateiordner, die im Basisordner des IIS gespeichert sind. Der Basisordner des IIS ist der Stammordner für HTTP-Anforderungen. Eine Datei, die im Basisordner gespeichert ist, kann über die direkte Angabe des Dateinamens im URL angesprochen werden:

```
http://Zaphod/default.htm
```

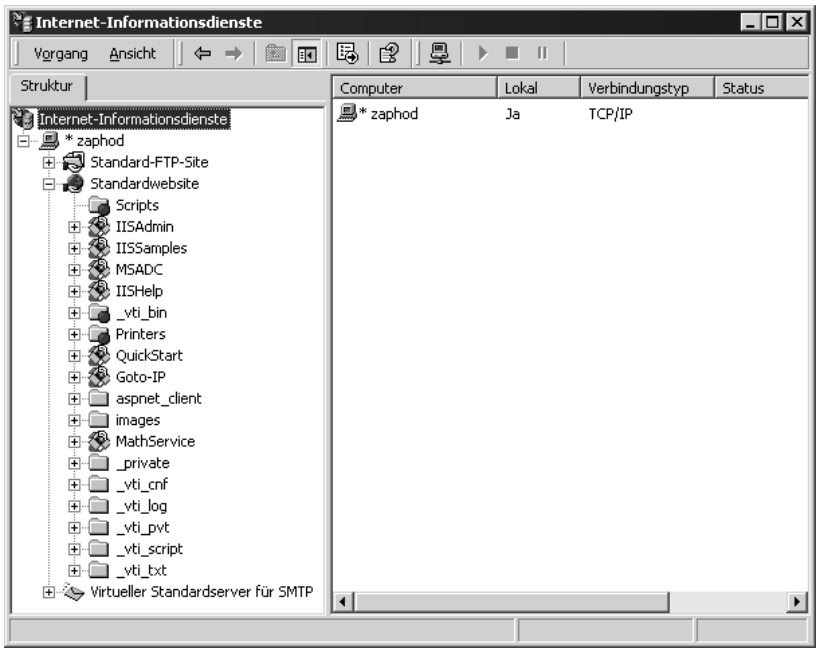

Abbildung 3.2: Das Snap-In zur IIS-Administration in der MMC unter Windows 2000 Professional

Alle Ordner, die Sie im IIS-Basisordner anlegen, werden automatisch in der IIS-Administration als Webordner angezeigt (wenn Sie bei laufender MMC neue Ordner anlegen, müssen Sie dazu die Ansicht über das Kontextmenü des Standardwebsite-Ordners aktualisieren).

Ordner, die nicht als Webanwendung laufen und die direkt im Root-Ordner des IIS gespeichert sind, werden über ein normales Ordnersymbol gekennzeichnet.

Ordner mit einer Weltkugel symbolisieren virtuelle Ordner. Virtuelle Ordner können von einem Browser aus wie normale Webordner angesprochen werden. Die Dateien dieser Ordner werden aber in physikalischen Ordnern gespeichert, die irgendwo auf den Festplatten des Systems angelegt sind (und eben nicht im Basisordner des IIS).

Ordner, die mit dem Symbol gekennzeichnet sind, das wie ein geöffneter Kasten aussieht, laufen als Webanwendung. Solche Ordner ermöglichen, dass die darin laufenden ASP-Programme Daten für einzelne Sitzungen oder für die gesamte Anwendung zwischenspeichern (was mit

normalen Ordnern nicht geht). Daneben wird das wichtige Debuggen unter ASP erst dann möglich, wenn ein Ordner als Webanwendung konfiguriert ist (unter ASP.NET ist das allerdings keine Voraussetzung). Ab Seite ■■97 gehe ich noch näher auf Webanwendungen ein. Ob ein Webanwendungs-Ordner im Basisverzeichnis des IIS gespeichert ist oder ein virtueller Ordner ist, können Sie am Symbol nicht erkennen.

Die Standardwebsite und die maximale Anzahl von Verbindungen

Der IIS erzeugt bei seiner Installation eine Standardwebsite. Diese ist mit der Standard-IP-Adresse des Rechners verknüpft. Unter den Server-Versionen von Windows NT, 2000 oder XP können Sie auch mehrere Websites parallel verwalten. So können Sie eine Website beispielsweise mit einem anderen Port als dem Standardport (80) verknüpfen oder mit einer anderen IP-Adresse (die mit einer separaten Netzwerkkarte verknüpft ist). Komplexe Systeme können so recht einfach in eine (besonders geschützte) Website zur Administration des Systems und die normale Benutzer-Website aufgeteilt werden. Eine andere Möglichkeit ist, mehrere Websites für verschiedene Anwendungsbereiche zu erzeugen und diese separat abzusichern. Die Administration der Benutzerrechte für multiple Websites ist damit wesentlich übersichtlicher, als wenn dazu virtuelle Ordner in einer einzigen Website verwendet werden. Unter den Professional-Versionen ist die Erzeugung mehrerer Websites allerdings nicht möglich. Hier müssen Sie mit virtuellen Ordnern arbeiten.

Die Professional-Versionen sind zudem in der maximalen Anzahl gleichzeitiger Verbindungen auf 10 beschränkt. Ein Einsatz im Internet ist für diese Versionen damit ausgeschlossen. Im Intranet sollte diese Limitierung aber normalerweise keine Probleme bereiten.

3.3.1 Das Basisverzeichnis

Der IIS verwaltet die Webdateien einer Website primär in einem Basisverzeichnis, das in den Eigenschaften der Website eingestellt werden kann. Per Voreinstellung ist dies der Ordner *\Inetpub\wwwroot* direkt unter dem Stammordner des Systemlaufwerks. Dieser Ordner ist die physikalische Repräsentanz des virtuellen Root-Ordners. Alle Dateien in diesem und allen Unterordnern können über relative Pfandangaben mit einem Webbrowser abgerufen werden, sofern dies nicht durch

Sicherheitseinstellungen unterbunden wird. In der Administration des IIS können Sie dem Root-Ordner aber virtuelle Ordner zuordnen, deren Inhalt in einem beliebigen anderen Ordner gespeichert ist. Für einen Webordner geben Sie auch an, welchen grundsätzlichen Zugriff Clients auf die enthaltenen Ordner und Dateien besitzen. So können Sie beispielsweise das LESEN, das SCHREIBEN und das AUSFÜHREN von Anwendungen für einen Ordner erlauben oder verbieten.

3.3.2 Eigenschaften einer Website

In den Eigenschaften einer Website können Sie eine Menge Einstellungen für die gesamte Website vornehmen. So können Sie beispielsweise die IP-Adresse festlegen, die mit der Website verbunden ist, und den Port definieren, auf dem der Webserver eingehende Anforderungen empfängt. Die Eigenschaften erreichen Sie am schnellsten, wenn Sie den Befehl EIGENSCHAFTEN im Kontextmenü des Website-Eintrags betätigen. Abbildung 3.3 zeigt die Eigenschaften der Standardwebsite.

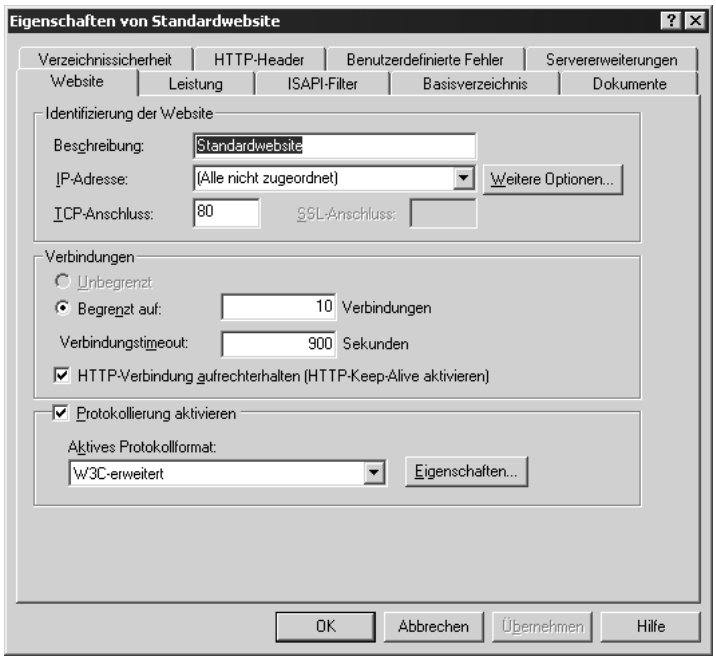

Abbildung 3.3: Die Eigenschaften der Standardwebsite

Übersicht über die Register der Website-Eigenschaften

Zur Übersicht beschreibt die folgende Auflistung zunächst die einzelnen Register der Website-Eigenschaften. Einige Register werden auf den folgenden Seiten noch näher erläutert.

- WEBSITE: Im Website-Register (Abbildung 3.3) stellen Sie die Grundeinstellungen der Website, wie z.B. deren Beschreibung und das Timeout für Verbindungen, ein. Diese Einstellungen beschreibe ich im folgenden Abschnitt.
- LEISTUNG: Im Leistung-Register können Sie die Leistung der Website optimieren und einschränken. So können Sie z.B. die Netzwerkbandbreite auf einen bestimmten Wert reduzieren, damit Ihr Netz durch den IIS nicht überlastet wird. Eine Einschränkung der CPU-Verwendung ist dann u.U. sinnvoll, wenn auf dem Rechner noch weitere Server laufen.
- ISAPI-FILTER: In diesem Register binden Sie ISAPI-Filter ein (siehe Seite ■■84).
- BASISVERZEICHNIS: Hier stellen Sie die Grunddaten des Ordners ein, in dem die Dateien und Unterordner des Webs verwaltet werden. Dazu gehören das physikalische Verzeichnis und die Rechte, die Internetanwender auf diesem Ordner besitzen (Lesen, Schreiben, Ausführen etc.). Für Unterordner (die dieses Register auch besitzen) konfigurieren Sie hier ebenfalls die Einstellungen einer eventuellen Webanwendung. Webanwendungen beschreibe ich ab Seite■■ 97.
- DOKUMENTE: In diesem Register stellen Sie ein, welches Dokument als Standard-Dokument verwendet wird, wenn der Anwender keine Dateiangabe in die URL integriert (also nur eine Pfadangabe macht). Außerdem können Sie hier ein HTML-Dokument angeben, das als Fußzeile für alle Dokumente des Web verwendet wird. Solche Fußzeilen sind aber nicht zu empfehlen. Zum einen wird die Zeile immer unter dem eigentlichen Dokument angezeigt und nicht am Ende des Webbrowser-Fensters, zum anderen wird die Fußzeile nicht angezeigt, wenn der Anwender keinen Dateinamen in die URL integriert und folglich das Standarddokument verwendet wird.

- VERZEICHNISSICHERHEIT: Hier können Sie u.a. einstellen, ob Benutzer sich in das System einloggen müssen, wenn sie versuchen ein Dokument abzurufen. Außerdem können Sie den Zugriff auf bestimmte IP-Adressen oder Domänennamen beschränken (nur Windows NT/2000/XP Server) und die sichere Kommunikation über SSL mit Server-Zertifikaten einrichten.
- HTTP-HEADER: In diesem Register können Sie festlegen, welche Werte im Header einer HTML-Seite an den Client zurückgegeben werden. Der HTTP-Header wird genutzt, um Informationen zwischen Client und Server auszutauschen. Der Client überträgt z.B. Informationen über die Art und Version des Webbrowsers im Header.
- BENUTZERDEFINIERTE FEHLER: Wie Sie ja bereits wissen, antwortet ein Webserver auf eine Anforderung mit einem HTTP-Status. Einige dieser Stati symbolisieren Fehler. Der Fehler 404 wird z.B. erzeugt, wenn ein Benutzer einen unbekannten Dateinamen in der URL verwendet. Diese Fehler sind mit HTML- oder ASP-Dateien verknüpft, die der IIS zum Client sendet, wenn der Fehler auftritt. Im Register BENUTZERDEFINIERTE FEHLER können Sie diese Verknüpfungen bearbeiten und natürlich auch mit eigenen Fehlermeldungsseiten verknüpfen. Bei der Erzeugung der neuen Fehlerseiten können Sie die voreingestellten HTML-Dateien als Basis verwenden. In einem Web ist eine Anpassung dieser Seiten schon deshalb wichtig, um Angaben zum technischen Support in die Fehlermeldung integrieren zu können. Sie finden die Default-Fehlerdokumente in *\WINNT\help\iisHelp\common*. ASP.NET nutzt diese Fehlermeldungs-Seiten allerdings nicht mehr. Die Fehlerbehandlung unter ASP.NET wird ab Seite ■■521 behandelt.
- SERVERERWEITERUNGEN: In diesem Register können Sie die FrontPage-Servererweiterungen grundlegend administrieren. Hier können Sie z.B. festlegen, ob die Erstellung von Dokumenten im Web (über ein kompatibles Programm) möglich ist. Die eigentliche Administration der FrontPage-Servererweiterungen ist übrigens eine MMC-Snapin, das finden Sie bei Windows 2000 unter SYSTEMSTEUERUNG / VERWALTUNG / ADMINISTRATOR FÜR SERVERERWEITERUNGEN. Die Administration der FrontPage-Servererweiterungen wird in diesem Buch nicht weiter behandelt.

Das Register »Website«

In Website-Register (Abbildung 3.3) stellen Sie die Grundeinstellungen der Website, wie z.B. deren Beschreibung und das Timeout für Verbindungen, ein.

- BESCHREIBUNG: Eine einfache Beschreibung der Website für interne Zwecke.
- IP-ADRESSE und TCP-Anschluss: In dieser Einstellung können Sie die IP-Adresse und den Port der Website einrichten. Jede auf einem System eingerichtete Website muss eine eindeutige IP-Adresse/Port-Kombination besitzen. Die Voreinstellung der IP-Adresse (ALLE NICHT ZUGEORDNET) bedeutet, dass der Webserver mit allen IP-Adressen des Rechners verknüpft ist. In einem einfachen System kann der Webserver dann im Intranet über die feste, private IP-Adresse und im Internet über die vom Provider dynamisch zugeordnete Adresse angesprochen werden. Wenn Sie den Internet-Zugriff unterbinden wollen (was eine interessante Sicherheitseinstellung ist), geben Sie hier nur die feste Adresse des Rechners an. Beachten Sie dann aber, dass Sie nun Probleme bekommen, wenn Sie den Rechner über seinen Namen oder über *localhost* ansprechen. Auf lokalen Systemen werden diese Namen in die Adresse 127.0.0.1 umgewandelt, die dann ja nicht mit der Website verbunden ist. Sie müssen in dem Fall also auf dem lokalen System immer mit der IP-Adresse arbeiten. Der Zugriff von entfernten Systemen funktioniert aber auch über den Rechnernamen. Falls auf dem Server mehrere Websites installiert sind (was ja nur unter den Server-Versionen von Windows NT, 2000 und XP möglich ist), muss jeder dieser Sites eine separate IP-Adresse oder – bei gleicher Adresse – ein separater Port zugeordnet werden. In der Einstellung TCP-ANSCHLUSS stellen Sie den Port ein, an dem die Website auf Anforderungen reagiert. Normale Websites laufen unter dem Port 80. Spezielle Websites, z.B. eine Website für Administrationszwecke, laufen meist unter anderen Portnummern (was ein zusätzlicher Sicherheitsgewinn ist, denn diese Ports sind ja nicht allgemein bekannt).
- VERBINDUNGEN: Hier stellen Sie die Anzahl der gleichzeitig möglichen Verbindungen und das Verbindungs-Timeout ein. Der IIS kann unter Windows NT/2000/XP Server theoretisch unendlich viele Verbindungen bearbeiten (unter den anderen Windows-Ver-

sionen sind nur maximal 10 Verbindungen möglich). Zu viele gleichzeitige Verbindungen begrenzen aber die Performance. Wenn Sie die Performance Ihres Web erhöhen wollen, begrenzen Sie die Anzahl der gleichzeitig möglichen Verbindungen. Ein Webbrowser, der versucht eine Verbindung aufzubauen, erhält die HTTP-Meldung 403.9 »Zugriff verboten: Zu viele Benutzer verbunden«, wenn die Anzahl der möglichen Verbindungen damit überschritten würde. Das Verbindungs-Timeout regelt die Zeit, die eine Verbindung noch geöffnet bleibt, wenn ein Client sich zwischenzeitlich nicht meldet. Ist das Timeout abgelaufen, wird die Verbindung getrennt und steht damit anderen Clients zur Verfügung. Das Timeout hat allerdings nur dann Auswirkungen, wenn die Anzahl der Verbindungen begrenzt ist. Stehen noch genügend freie Verbindungen zur Verfügung, vergibt der IIS dem Client einfach eine neue Verbindung, wenn dieser sich zurückmeldet.

- PROTOKOLLIERUNG: Der IIS ist in der Lage, jeden Zugriff auf die Website zu protokollieren. Er verwendet dazu einfach Textdateien. Webmaster können an Hand dieser Dateien Rückschlüsse auf die Webzugriffe ableiten. In den Eigenschaften können Sie einstellen, ob die Protokollierung aktiviert ist und wie die Dateien verwaltet werden. Per Voreinstellung erzeugt der IIS pro Tag eine separate Datei, die Sie im Ordner *\WINNT\system32\LogFiles\W3SVC1* finden.

3.3.3 Eigenschaften der Webordner

Jeder Webordner besitzt ähnliche Eigenschaften wie der Stammordner. Lediglich die Register WEBSITE, LEISTUNG, ISAPI-FILTER und SERVERERWEITERUNGEN fehlen.

Für jeden Webordner können Sie Einstellungen vornehmen, die von der Konfiguration der Website abweichen. So können Sie z.B. die Zugriffsmöglichkeiten auf einen Webordner unabhängig von der Website-Einstellung einschränken oder auch erweitern. Abbildung 3.4 zeigt die Eigenschaften eines virtuellen Ordners, den ich für die Beispiele dieses Buchs eingerichtet habe.

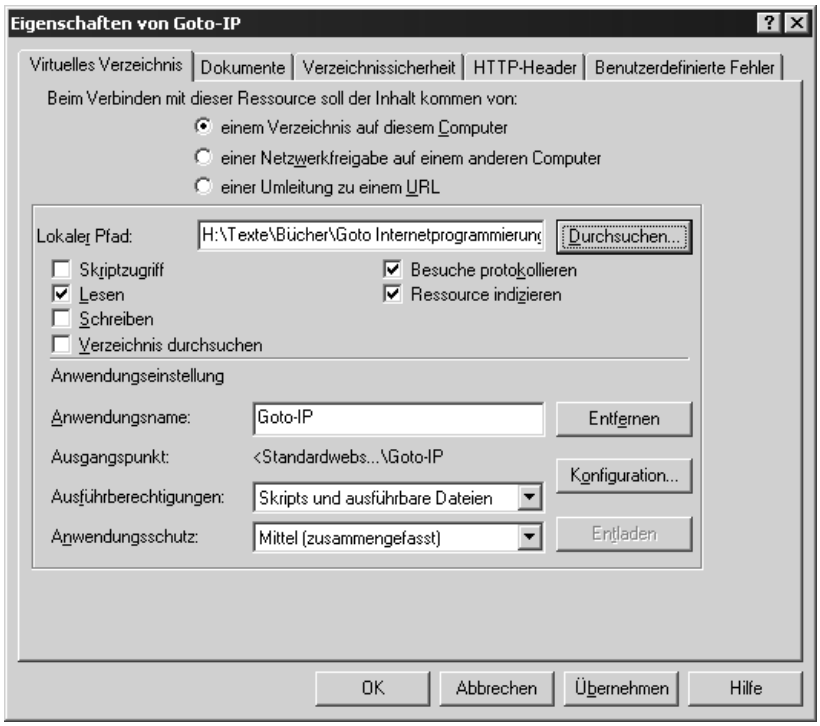

Abbildung 3.4: Eigenschaften eines Webordners

3.3.4 Webanwendungen

Den Stammordner und jeden Unterordner einer Website können Sie als »Anwendung« definieren. Eine Anwendung ermöglicht:

- Sitzungen, über die Sie unter ASP für jeden verbundenen Client im IIS Daten speichern können (das globale Speichern unter ASP.NET ist davon aber nicht betroffen),
- das Speichern von globalen Daten, die sich alle Clients teilen (wieder nur für ASP, nicht für ASP.NET),
- die Verwendung von externen Klassenbibliotheken in ASP.NET-Webanwendungen (die in einem Unterordner *Bin* gespeichert werden)
- und das Debuggen von ASP-Programmen (das ASP.NET-Debuggen ist davon aber nicht betroffen).

Die Bezeichnung »Anwendung« ist für diesen Zweck etwas irreführend. Es handelt sich ja schließlich nicht um ein Programm, das von Anwendern benutzt wird, sondern um eine spezielle Konfiguration eines Webordners.

Ist ein Webordner nicht als Anwendung konfiguriert, können Sie in ASP-Seiten keine globalen Daten speichern (in ASP.NET-Seiten ist das globale Speichern aber auch dann möglich). Erst eine Anwendung ermöglicht unter ASP die Verwaltung globaler Daten für die gesamte Anwendung oder nur für die einzelnen Sitzungen. Deshalb ist die Bezeichnung »Anwendung« wohl angebracht.

Sie erreichen die Konfiguration für Webanwendungen über das Register VERZEICHNIS in den Eigenschaften eines Webordners. Im Bereich Anwendungseinstellung können Sie die Anwendung konfigurieren. Falls noch keine Anwendung besteht, können Sie eine über den ERSTELLEN-Schalter erzeugen. Über den KONFIGURIEREN-Schalter erreichen Sie die Konfiguration (was auch sonst ...).

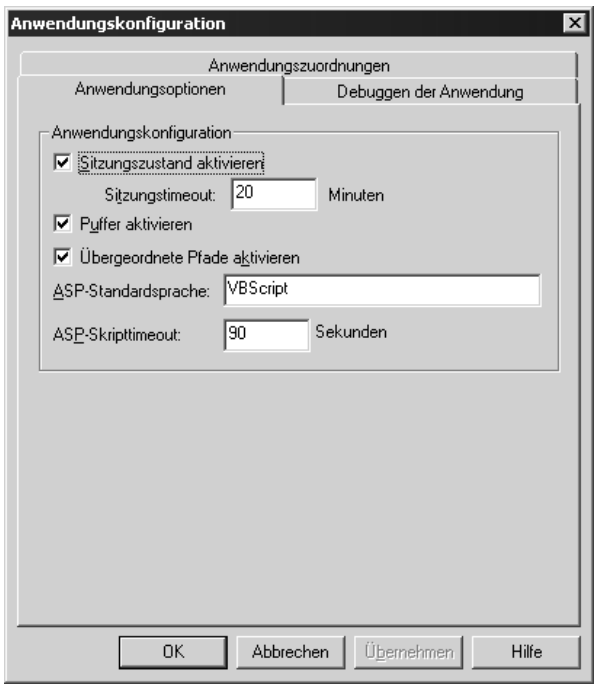

Abbildung 3.5: Die Konfiguration einer Webanwendung

Sitzungen

Wenn Sie für eine ASP-Webanwendung Sitzungen ermöglichen, wird jede Verbindung eines Clients als eine Sitzung betrachtet. Der Sinn solcher Sitzungen ist das Speichern globaler Daten. Damit können Sie auf einfache Weise Daten zwischen einzelnen ASP-Dokumenten austauschen. Für einen Online-Shop müssen Sie sich z. B. die ID des Anwenders merken, um den Warenkorb identifizieren zu können. Diese ID wird üblicherweise in einer Sitzungsvariable verwaltet. In der Konfiguration können Sie zudem das Timeout einer Sitzung einstellen. Die Voreinstellung ist 20 Minuten. Ruft der Anwender innerhalb dieser Zeit kein Dokument ab, wird die Sitzung beendet. Die globalen Sitzungsdaten gehen dann verloren. In ASP.NET gilt diese Einstellung allerdings nicht, weil ASP.NET Sitzungen unabhängig vom IIS verwaltet. In ASP.NET werden die Grundeinstellungen für Sitzungen in der Datei *machine.config* vorgenommen, können aber auch in einer speziellen Anwendungskonfigurationsdatei definiert sein. Die ASP.NET-Anwendungskonfiguration wird ab Seite■■ 662 behandelt. Das Timeout der Sitzung können Sie in ASP(.NET) auch über das `Session`-Objekt anpassen.

Globale Daten

In ASP und ASP.NET können Sie globale Daten für eine gesamte Anwendung oder für die einzelnen Sitzungen speichern. Diese Daten können dann grundsätzlich von allen ASP(.NET)-Dokumenten innerhalb der Anwendung gesetzt und abgefragt werden. Anwendungsglobale Daten existieren nur einmal. Wenn eine Sitzung anwendungsglobale Daten ändert, ist die Änderung sofort in allen anderen Sitzungen verfügbar. Sitzungsglobale Daten werden für jede Sitzung separat gespeichert. Eine Änderung in einer Sitzung betrifft andere Sitzungen nicht. Das Kapitel 10 zeigt ab Seite ■■621, wie Sie in ASP.NET mit den Objekten `Application` und `Session` globale Daten speichern.

Debuggen

Im Register DEBUGGEN DER ANWENDUNG der Anwendungskonfiguration können Sie das Debuggen von ASP-Programmen ein- und ausschalten. Wenn Sie das Debuggen einschalten, können Sie Fehler, die in ASP-Dokumenten auftreten, lokalisieren. Entwicklungsumgebungen wie Visual Studio oder Visual Interdev schalten diese Option allerdings normalerweise automatisch ein, wenn Sie die Anwendung in der Entwicklungsumgebung starten.

Abbildung 3.6: Einstellung des Debuggens in der Anwendungskonfiguration eines Webordners

Serverseitiges Debuggen bedeutet, dass der Debugger auf demselben Rechner ausgeführt wird wie der IIS. Dieses Debuggen funktioniert (fast) immer. Beim clientseitigen Debuggen wird der Debugger auf einem entfernten Rechner ausgeführt. So weit ich weiß, hat Microsoft diese Art des Debuggens nie für ASP ermöglicht. Die Einstellungen auf diesem Register der Anwendungskonfiguration haben übrigens keinen Einfluss auf das Debuggen von ASP.NET-Anwendungen. Diese Anwendungen können Sie auch dann debuggen, wenn im IIS das Debugging abgeschaltet ist. Ich beschreibe das Debuggen von ASP.NET-Programmen in Kapitel 6 ab Seite ■■356.

Anwendungszuordnungen

Im Register ANWENDUNGSZUORDNUNGEN stellen Sie die Zuordnung von Dateitypen zu ISAPI-Anwendungen ein. Voreingestellt sind die Dateitypen, mit denen der IIS bereits umgehen kann. Dazu gehören auch ASP-Programme (*.asp*) und ASP.NET-Anwendungen (*.aspx*). Wenn Sie selbst eine ISAPI-Anwendung entwickelt oder eine gekauft haben, kön-

nen Sie diese Anwendung hier mit einem Dateityp verbinden. Ruft der Benutzer eine Datei mit der entsprechenden Endung ab, wird die Datei über die ISAPI-Anwendung gesendet und dort verarbeitet.

3.4 IIS-Sicherheit

Ein Web, das Sie mit dem IIS eingerichtet haben, ist zunächst nicht abgesichert. Jeder Benutzer im Intranet oder – wenn der Computer, auf dem der IIS läuft, Zugang zum Internet hat – im Internet kann die dort gespeicherten Dokumente normalerweise zumindest lesen (sofern Sie nicht das Recht dazu in den Eigenschaften des Webordners ausgeschaltet haben). Ist für eine Webanwendung das Recht eingeschaltet, dass Skripte oder normale Programme ausgeführt werden dürfen, kann ein Client Programme starten, die in dem Webordner gespeichert sind. Ist zusätzlich das Schreibrecht gesetzt, kann ein Benutzer die zu startenden Programme sogar vom Client in den Webordner übertragen und danach ausführen. Wenn Sie ein »Gefahrensucher« sind (wie der im Film »Kentucky Fried Movie«), sollten Sie für das gesamte Web alle Rechte vergeben und abwarten. Es dauert bestimmt nicht lange, bis ein Hacker Ihren IIS entdeckt und die »Administration« Ihres Rechners übernimmt ...

Sie können Ihren Webserver aber auch absichern. Dazu gehören zwei Dinge: Die Verzeichnissicherheit und der allgemeine Schutz vor Angreifern. Mit der Verzeichnissicherheit können Sie einrichten, dass nur bestimmte Benutzer Zugriff auf das Web oder bestimmte Seiten im Web besitzen. Die Rechte dieser Benutzer können Sie über die Windows-Rechte auf Dateien und Ordner einstellen. Die Verzeichnissicherheit schützt Ihr System aber nicht vor Hackern, die über einige Sicherheitslöcher in das System eindringen und dort Schaden anrichten oder Daten stehlen können.

Der folgende Abschnitt zeigt zunächst, wie Sie Verzeichnissicherheit erreichen. Danach gehe ich noch auf den Schutz vor Angreifern ein.

3.4.1 Verzeichnissicherheit

Der IIS ist eng in das Windows-Sicherheitskonzept integriert. Mit der Verzeichnissicherheit können Sie ein ganzes Web oder einzelne Webordner so absichern, dass nur Benutzer Zugriff erhalten, die ein Konto

auf dem System besitzen. Die Konfiguration ist recht einfach. Voraussetzung dafür ist aber, dass Sie das NTFS-Dateisystem auf den Computern verwenden, die die Webdateien speichern. Ohne NTFS können Sie die Zugriffsrechte auf einzelne Dateien und Ordner nicht für einzelne Benutzer oder Gruppen definieren.

Dieser Abschnitt behandelt lediglich die in den IIS integrierten Features zur Absicherung des Zugriffs auf Webressourcen. Für diese, von ASP und ASP.NET unterstützte Art der Absicherung müssen alle Webbenutzer als Windows-Benutzer registriert werden. Wenn Sie Ihre Webbenutzer dagegen selbst (z.B. in einer Datenbank) verwalten wollen, müssen Sie in ASP 2/3 die Authentifizierung komplett selbst programmieren (oder alternativ den komplizierten Membership Server des Site Servers verwenden). In ASP.NET ist die Unterstützung einer eigenen Benutzerverwaltung dagegen bereits sehr gut integriert, erfordert aber auch etwas Programmierung. Ab Seite ■■1041 geht das Buch darauf ein.

Der anonyme Benutzer

Wenn Sie für einen Webordner den anonymen Zugriff erlauben (was die Voreinstellung ist), kann normalerweise jeder Webbenutzer auf die Dateien in diesem Ordner zugreifen, ohne sich authentifizieren zu müssen. Der IIS mappt den anonymen Webbenutzer auf das Windows-Benutzerkonto *IUSR_Rechnername*. Auf dem Rechner *Zaphod* heißt dieses Konto z.B. *IUSR_Zaphod*. Beim Zugriff auf die Dateien im Web werden die Rechte von *IUSR_Rechnername* verwendet. Normalerweise besitzt dieses Konto aber gar keine Rechte auf Ordnern und Dateien, da Windows standardmäßig dem Konto *Jeder* für alle Ordner und Dateien im System Lese- und Ausführungsberechtigungen vergeben hat. Wenn Sie die Rechte des Benutzerkontos *Jeder* für eine Datei entfernen, hat zunächst kein Internetanwender Zugriff auf die Datei. In abgesicherten Windows-Systemen, bei denen das Benutzerkonto *Jeder* meist gar keine Rechte besitzt, müssen Sie dem Konto *IUSR_Rechnername* Leserechte auf alle Webdateien vergeben, wenn Sie einen anonymen Zugang ermöglichen wollen.

Wenn Sie das einmal auf Ihrem lokalen System testen wollen, sollten Sie beachten, dass der IIS per Voreinstellung in lokalen Webs auch immer versucht, die Rechte des auf dem Client in Windows eingeloggten Benutzers zu verwenden. Um diesen Mechanismus auszuschalten, deak-

tivieren Sie die Option INTEGRIERTE WINDOWS-AUTHENTIFIZIERUNG in den Authentifizierungs-Eigenschaften des Webordners.

Ein Benutzer, der keine Leserechte für die angeforderte Datei besitzt, erhält die HTTP-Fehlermeldung 401.3 – »Zugriff verweigert wegen ACL[2] auf Ressource«.

Absichern eines Webordners

Wenn Sie einen Webordner absichern wollen, öffnen Sie das Register VERZEICHNISSICHERHEIT in den Eigenschaften des Ordners. Hier wählen Sie den BEARBEITEN-Schalter für die Steuerung des anonymen Zugriffs und der Authentifizierung.

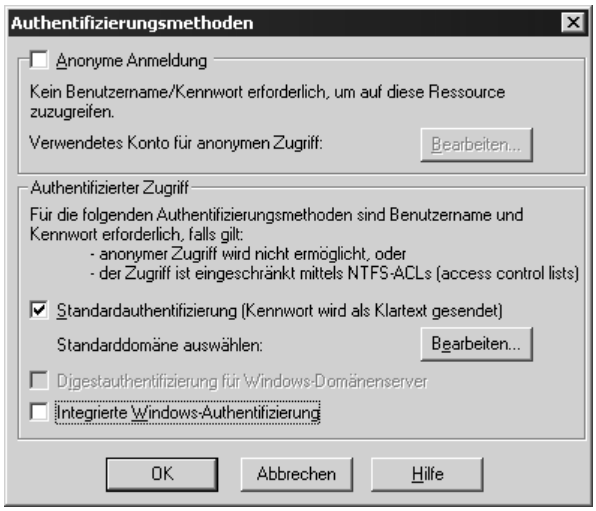

Abbildung 3.7: Der Dialog zur Einstellung der Benutzer-Authentifizierung

Hier wählen Sie die anonyme Anmeldung ab. Nun können Sie noch entscheiden, welche der Authentifizierungsmethoden Sie zulassen. Die Standardauthentifizierung verwendet den in den gängigen Browsern eingebauten Login-Dialog und sendet die Login-Informationen über

2 Jedes Objekt in einem Windows-System (z.B. eine Datei oder ein Ordner) besitzt eine Access Control List (ACL), die aus einzelnen ACEs (Access Control Entries = Zugriffs-Kontroll-Einträgen) besteht. Ein ACE enthält eine Security ID (SID), die ein Windows-Konto identifiziert (z.B. das Konto eines Benutzers oder einer Gruppe), und eine Zugriffsmaske, die die Zugriffsrechte für die SID definiert. Über die ACL verwaltet Windows also die Zugriffsrechte auf ein Objekt.

HTTP in Textform zum Webserver. Der Mechanismus dazu ist im HTTP1.1-Standard definiert. Ruft ein Client über HTTP eine geschützte Datei ab, gibt der Webserver den Status 401 »Access Denied« mit der Information »WWW-Authenticate: Basic realm="*Rechnername*"« im Header zurück. Der Client erkennt an dem WWW-Authenticate-Header, dass eine Authentifizierung erwünscht ist, und reagiert normalerweise mit der Anzeige eines Login-Dialogs. Abbildung 3.8 zeigt einen solchen Dialog im Internet Explorer.

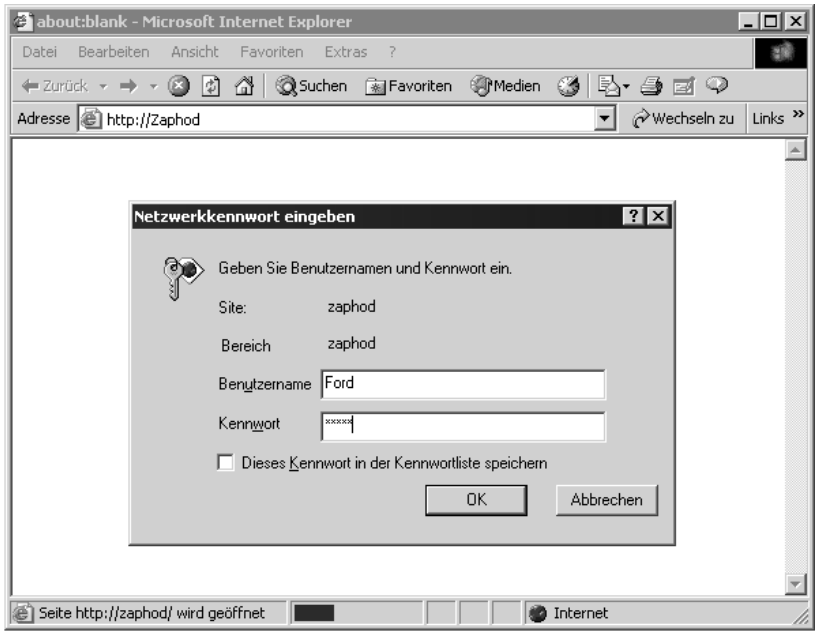

Abbildung 3.8: Login in ein geschütztes Web mit der Standardauthentifizierung

Betätigt der Benutzer dort den OK-Schalter, sendet der Client die Anforderung nochmals, allerdings nun mit Base64-dekodierten-Login-Informationen:

```
GET / HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
*/*
Accept-Language: de
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01;
 Windows NT 5.0)
Host: Zaphod
Connection: Keep-Alive
Authorization: Basic Rm9yZDpnYWxheHk=
```

Base64 ist ein Codierungsverfahren, das verwendet wird, um binäre Daten und Texte, die nicht dem 7-Bit-ASCII-Code entsprechen, so zu kodieren, dass diese über den 7-Bit-ASCII- oder den auf IBM-Maschinen verwendeten EBCDIC-Code dargestellt werden können. Damit ist sichergestellt, dass diese Daten über das Internet ohne Probleme verwendet werden können. Für die Codierung und Decodierung stehen im Internet viele Tools und Bibliotheken zur Verfügung. Auf der Seite *www.fourmilab.ch/webtools/base64/* finden Sie z.B. ein kleines Windows-Programm zum Kodieren und Dekodieren von Dateien. Der dekodierte Authorisierungscode des Beispiels oben ist »Ford:galaxy«. Angreifer (Hacker) können diese Informationen natürlich auslesen, wenn sie Zugriff auf das Netz haben.

Die integrierte Windows-Authentifizierung arbeitet mit dem Windows-Login des Benutzers. Diese Authentifizierung ist nur sinnvoll für Intranets. Im Internet kann man nicht davon ausgehen, dass der Benutzer Windows verwendet und sich auch im System eingeloggt hat. Die integrierte Windows-Authentifizierung besitzt im Intranet den Vorteil, dass die Benutzer sich nicht separat in das Web einloggen müssen. Ein Nachteil ist, dass der Benutzer sich neu in Windows einloggen muss, wenn er unter einem anderen Namen Zugriff auf das Web erhalten will. Ein weiterer Nachteil ist, dass diese Authentifizierungsmethode nicht über Proxies oder Firewalls hinweg und nur mit dem Internet Explorer funktioniert.

Wenn Sie beide Authentifizierungsmethoden einschalten, versucht das System zuerst die integrierte Windows-Authentifizierung. Ist diese nicht möglich bzw. hat der Windows-Benutzer nicht die erforderlichen Rechte, wird die Standardauthentifizierung verwendet.

Benutzer und Rechte einer geschützten Website

In einem geschützten Web müssen Sie Rechte für den Webordner oder einzelne Dateien definieren. Wenn Sie die voreingestellten Rechte über-

nehmen, kann jeder in Windows registrierte Benutzer das Web verwenden. Ein Zugriff nicht registrierter Benutzer ist allerdings nicht möglich. Das liegt daran, dass zum einen das Benutzerkonto *Jeder* standardmäßig alle Rechte auf dem Webordner und den enthaltenen Dateien besitzt (mit »Jeder« sind allerdings nur alle in Windows registrierten Benutzer gemeint, nicht ein beliebiger Anwender). Zum anderen mappt der IIS in einem geschützten Web den zugreifenden Benutzer nicht mehr auf das Konto *IUSR_Rechnername* (das ja auch zum Konto *Jeder* gehört).

Unter ASP.NET können Sie zusätzlich zum hier beschriebenen Verfahren eine eigene Benutzerauthentifizierung einrichten, bei der die Benutzerdaten entweder in eigenen Programmen oder über eine Konfigurationsdatei verwaltet werden. ASP.NET ermöglicht damit auch das Verwalten der Benutzerdaten in einer Datenbank, was besonders für Websites interessant ist, die sehr viele Benutzer verwalten. Die Einrichtung und Programmierung unter ASP.NET beschreibe ich aber erst in Kapitel 14, weil dazu einiges an ASP.NET-Grundwissen notwendig ist.

Für den normalen Zugriff auf die Website und die Ausführung von ASP-Programmen müssen Sie für Benutzer, die Zugriff auf die Website besitzen sollen, entweder in einer Windows-Domäne oder auf dem Server Benutzerkonten einrichten. Der Browser sendet schließlich nur den Namen und das Passwort zum IIS. Der IIS prüft, ob ein lokaler oder ein Domänenbenutzer mit dem gesendeten Namen existiert und ob das Passwort korrekt ist. Wenn Sie die integrierte NT-Sicherheit verwenden wollen, müssen Sie die lokalen Benutzer allerdings auch auf dem Client mit demselben Passwort verwalten wie auf dem Server, wenn Sie keine Domäne einsetzen.

Legen Sie die Webbenutzer zunächst in Windows an, falls das noch nicht geschehen ist. In Windows 2000 verwenden Sie dazu den Benutzermanager, den Sie unter der Systemsteuerung in VERWALTUNG / COMPUTERVERWALTUNG / LOKALE BENUTZER UND GRUPPEN finden (Abbildung 3.9).

Verwalten Sie die Webbenutzer idealerweise in speziellen (Web-)Gruppen (die Sie u.U. neu anlegen müssen). Die Rechtevergabe wird damit erheblich vereinfacht. Für einen geschützten Online-Shop würde ich z.B. die Gruppen »Shop-Kunden«, »Shop-Administratoren« und »Shop-Sachbearbeiter« anlegen. Shop-Kunden haben lediglich lesen-

den Zugriff auf Seiten, die für eine Bestellung relevant sind, Shop-Administratoren können auf alle Seiten zugreifen und Shop-Sachbearbeiter besitzen Leserechte für die Seiten, die für die Verarbeitung der Bestellungen verwendet werden.

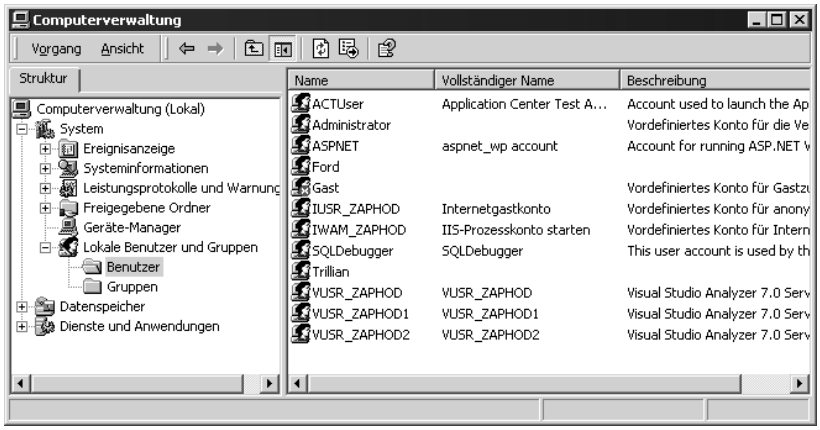

Abbildung 3.9: Der Benutzermanager von Windows 2000

Gehen Sie dann im Explorer zum Webordner und stellen Sie dort die Rechte für die Benutzer ein. Für normale Internet- oder Intranetbenutzer reichen Leserechte vollkommen aus. Die im Ordner enthaltenen Dateien werden ja schließlich normalerweise nur gelesen, was auch für ASP-Dateien gilt (in einigen Fällen erlauben Sie auch einen Datei-Upload, dann müssen die Benutzer auch Schreibrechte besitzen).

Entfernen Sie für eine geschützte Website auf jeden Fall den Benutzer *Jeder*. Wenn Sie diesen Benutzer bzw. dessen Rechte nicht entfernen, besitzt jeder in Windows registrierte Benutzer Zugriff auf diesen Ordner.

Oft wird eine Webanwendung in mehrere Bereiche aufgeteilt. In einem Online-Shop könnte es z.B. einen Bereich für die Kunden und einen für die Sachbearbeiter geben, die die Bestellungen der Kunden bearbeiten. Wenn Sie die zugehörigen Dokumente in separaten physikalischen Ordnern verwalten, erleichtern Sie sich die Rechteverwaltung. Windows vererbt standardmäßig alle Rechte eines Ordners an die enthaltenen Dateien und Unterordner. So definieren Sie die Rechte lediglich für den Ordner und müssen sich nicht um die enthaltenen Dateien kümmern.

Für einen Shop könnte es z.B. einen physikalischen Ordner *Shop* geben, der über einen gleichnamigen virtuellen Ordner im IIS dargestellt wird. Dieser Ordner enthält alle Dokumente, die von den Kunden verwendet werden dürfen (bzw. sollen, wir wollen ja unsere Produkte verkaufen ...). Lassen Sie nur einen geschlossenen Kundenkreis zu, verwalten Sie die Kunden als Benutzer in Windows und vergeben Leserechte für die Gruppe dieser Benutzer auf dem Shop-Ordner. Ein weiterer Ordner *Shop-Bearbeitung* enthält in diesem Beispiel Dokumente zur Bearbeitung der Bestellungen. Leserechte vergeben Sie nur für die Gruppe der Sachbearbeiter, nicht für die Kunden. Abbildung 3.10 zeigt die Rechtevergabe für die Gruppe der Kunden auf dem Ordner *Shop*. Den Dialog zur Einstellung der Rechte erreichen Sie im Windows-Explorer über die Eigenschaften des Ordners.

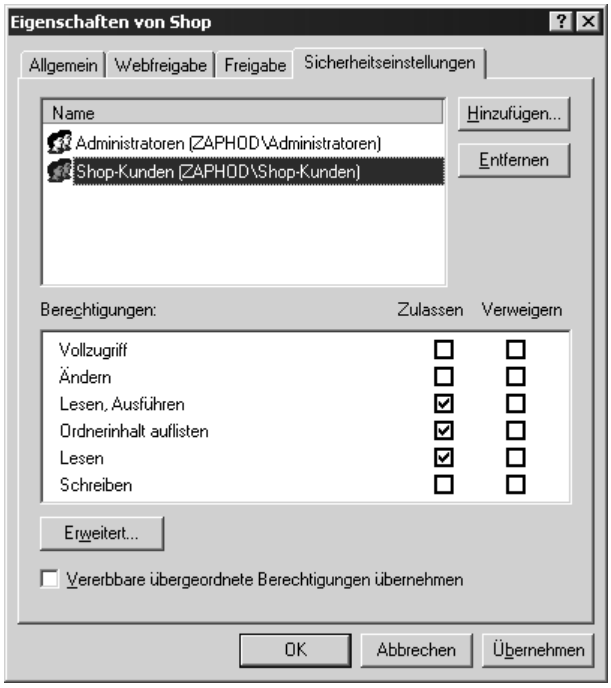

Abbildung 3.10: Vergabe der Rechte für die Kunden eines Online-Shops in Windows 2000

Achten Sie darauf, dass Sie die Option VERERBBARE ÜBERGEORDNETE BERECHTIGUNGEN ÜBERNEHMEN ausschalten, damit nicht die Rechte des übergeordneten Ordners übernommen werden.

Wenn Sie einen Webordner nur einem geschlossenen Personenkreis verfügbar machen wollen, aber ermöglichen wollen, dass dieser Kreis dynamisch erweitert wird, bringt diese Vorgehensweise einige Probleme mit sich. Neue Personen müssen ja schließlich immer auch als Windows-Benutzer angelegt werden. Wenn Sie den Benutzer automatisch z.B. nach einer Registration anlegen wollen, können Sie dazu Betriebssystembefehle (recht einfach) oder LDAP[3] (kompliziert!) verwenden. Der Membership Server, der ein Teil des Microsoft Site Server Commerce ist, löst dieses Problem dadurch, dass die Benutzer nicht in Windows, sondern in einer Datenbank verwaltet werden. Leider erfolgt die Programmierung über LDAP und ist damit recht kompliziert. Wenn Sie eine eigene Benutzerverwaltung programmieren wollen, müssen Sie diese unter ASP noch komplett selbst entwickeln (oder vordefinierte Tools verwenden). In ASP.NET ist eine flexible Benutzerverwaltung bereits integriert.

Separate Rechtevergabe für Dateien

Sie können natürlich auf einzelne Dateien separate Rechte vergeben. Gehen Sie dazu in die Eigenschaften der Datei, schalten Sie die Vererbung der übergeordneten Rechte ab und definieren Sie die Rechte für die Datei. So können Sie einzelne Dateien mehr, weniger oder anderen Benutzern zugänglich machen.

Anonymen mit abgesichertem Zugriff mischen

Oft ist es notwendig, auf einzelne Dokumente einen anonymen Zugriff zu erlauben. In einem Shop könnten z.B. alle Dokumente, die zum Bestellen verwendet werden, für alle Benutzer frei sein. Lediglich die Dokumente zum Absenden der Bestellung sind nur für registrierte Benutzer verfügbar. Wenn Sie den anonymen Zugriff auf einzelne Dokumente in einem gesicherten System erlauben wollen, stellen Sie diesen einfach in den Eigenschaften des Dokuments in der IIS-Administration ein. Für Webordner habe ich dies ja schon ab Seite■■ 103 beschrieben.

3 Das »Lightweight Directory Access Protocol« ermöglicht den Zugriff auf Verzeichnisdienste über ein Objektmodell. In Windows NT und 2000 sind bereits Verzeichnisdienste für verschiedene administrative Aufgaben im »Active Directory« enthalten.

3.4.2 Schutz vor externen Angreifern

Ein System, auf dem das aktuelle Service Pack und die aktuellen Patches für Sicherheitslöcher nicht installiert sind, enthält eine Vielzahl an Sicherheitslöchern. Nicht nur der IIS ist davon betroffen, auch verschiedene andere Microsoft-Komponenten und -Anwendungen wie z.B. der Windows Media Player. Über diese Sicherheitslöcher kann ein Angreifer das System auslesen, den Datenverkehr abhören oder sogar Programme oder Betriebssystembefehle ausführen. Ideal für den Angreifer ist, wenn er das auszuführende Programm auch noch zuvor auf den Rechner übertragen kann. Ein nicht abgesichertes System ist quasi ein Scheunentor für Angreifer.

Nicht nur die Microsoft-Sicherheitslöcher bieten Angreifern Zugang zu Ihrem System. Ein Hacker kann auch andere Techniken verwenden. Er kann z.B. das Programm *ShareSniffer* benutzen, um Rechner in einem einzugebenden IP-Bereich auf freigegebene Ordner zu scannen, und dann ganz einfach Zugriff auf diese Ordner erhalten.

Das Security Bulletin

Microsoft veröffentlicht in unregelmäßigen Abständen Artikel im so genannten Security Bulletin. Sie können diese Artikel im Internet abrufen (*www.microsoft.com/technet/itsolutions/security/current.asp*) oder per E-Mail abonnieren (ebenfalls auf der genannten Seite). Die Anzahl der darin gemeldeten Sicherheitslöcher gibt schon zu denken. Dabei sollten Sie auch noch beachten, dass die gemeldeten Sicherheitslöcher nur die sind, die Microsoft gefunden hat. Hacker finden wahrscheinlich noch andere, die Microsoft (noch) gar nicht kennt.

Schutz des Systems

Die vorhergehenden Abschnitte haben Ihnen wahrscheinlich gezeigt, dass Sie ein System, das an das Internet angeschlossen ist, absichern sollten. Dies gilt besonders dann, wenn im System der IIS läuft. Dazu gehören nicht nur die Installation der aktuellen Service Packs und Patches, sondern auch grundlegende Sicherheitseinstellungen im System.

Wie Sie ein System absichern, das den IIS ausführt, beschreibt Microsoft im Artikel »Secure Internet Information Server 4 Checklist« bzw. »Secure Internet Information Server 5 Checklist«. Sie erreichen diese

Artikel über die Seite *support.microsoft.com/support/kb/articles/q282/0/60.asp*. Daneben sollten Sie aber auf jeden Fall auch noch eine Firewall installieren. Eine hervorragende Firewall für private Zwecke ist die im Norton Internet Security enthaltene. Diese Firewall administriert sich mit nur sehr wenigen und leicht verständlichen Benutzereingriffen fast selbstständig und schützt das System sehr effektiv. Sie werden sich wundern, wie häufig Sie angegriffen werden ...