

# Die wichtigsten Vorbereitungen treffen

# 4

## In diesem Kapitel

- ▶ Wie Sie sich mit dem IIS eine praktische Entwicklungsumgebung einrichten
- ▶ Wie sich physische und virtuelle Verzeichnisse unterscheiden
- ▶ Wie Sie Ihren MySQL-Datenbankserver betriebsbereit machen

---

Nachdem Sie Datenbank und Webserver mit Skriptingfunktion installiert und damit die Grundlagen geschaffen haben, möchten Sie natürlich auch Datenbanken anlegen und mit ihnen arbeiten. In diesem Kapitel erfahren Sie, was Sie tun müssen, um sich auf Ihrem PC eine clevere Entwicklungsumgebung einzurichten, mit der Sie bequem arbeiten können und die Ihren späteren Live-Auftritt gut vorbereitet.

## Wer dieses Kapitel lesen sollte

Wenn Sie

- ✓ zum ersten Mal einen eigenen Webserver installiert haben
- ✓ zum ersten Mal mit serverseitigen Skripten arbeiten
- ✓ zum ersten Mal mit MySQL und PHP arbeiten

finden Sie diesem Kapitel wichtige Informationen, die Ihnen die ersten Schritte auf dem Weg zum Webdatenbank-Projekt erleichtern.

## Zu Hause ist es doch am schönsten: localhost

Denken Sie noch einmal kurz daran, wie wir im Kapitel 1 *Was ist ein Webserver eigentlich?* die Kommunikation zwischen Browser und Webserver beschrieben haben. Über eine Internetadresse wie `meinlieblingswebseite.com` konnten Sie einen anderen Webserver im Internet ausfindig machen lassen und irgendwelche Internetseiten von diesem Server anfordern und sich in Ihrem Browser anzeigen lassen.

Wenn Sie nur auf Ihrem eigenen PC arbeiten, befindet sich der Browser ja auf demselben Computer wie Ihr Webserverprogramm. Wenn Sie jetzt HTML- oder Skriptdateien von Ihrem Server mit dem Browser anfordern möchten, funktioniert die Kommunikation nun im Grunde zwar genauso, nur entfällt hier natürlich der Umweg über das Internet.

Ihr eigener Rechner hat eine eigene, fest eingestellte lokale Adresse, die nur auf Ihrem PC gültig ist, und über die Sie ihn mit Client-Programmen wie Ihrem Webbrowser ansprechen können. Standardmäßig heißt diese Adresse `localhost`. Wenn Sie einen IIS- oder PWS-Webserver verwenden, können Sie ihn alternativ auch über die Adresse `oemcomputer` erreichen.

## ***Ein Blick in die Zukunft***

Während Sie experimentieren und entwickeln, brauchen Sie sich noch keine großen Gedanken um Geschwindigkeits- und Sicherheitsaspekte im echten Online-Betrieb zu machen. Schließlich sind Sie vermutlich vorläufig die einzige Person, die auf Ihre Webdatenbank-Applikation zugreifen wird. Wir beschränken uns deshalb erst einmal auf die allernötigsten Einstellungen, um eine Webdatenbank-Applikation mit serverseitigen Skripten auf einem freistehenden PC zum Laufen zu bringen. Gleichzeitig bereiten wir aber schon mal alles so vor, dass Sie ein Projekt später ohne viel Mühe in den Produktivbetrieb übernehmen können. (Mehr dazu finden Sie in Teil IV *An den Start gehen*.)

## ***Für IIS- und Access-Nutzer***

### ***Vom Baum der Erkenntnis naschen und sinnvoll Verzeichnisbäume pflanzen***

Mit Ratschlägen ist es so eine Sache. Sie kennen ja alle die Geschichte von der Schlange und dem Apfel. Schließlich verbringen wir nur deshalb fünf Tage die Woche mit Arbeit vor dem Computer. Hätte Eva damals einfach nicht auf die Ratschläge der Schlange gehört, gäbe es heute wahrscheinlich auch keine Webdatenbanken und wir könnten stattdessen alle Ferien im Paradies machen. Aber nun gut, wir müssen eben in den sauren Apfel beißen und das Beste aus allem machen. Und damit das auch möglichst einfach und erfreulich wird, tun wir das natürlich mit System.

Sie haben es vermutlich schon geahnt: Wir haben natürlich einen Vorschlag in der Tasche, zu dem wir Sie gerne überreden möchten. Unser Vorschlag ist, dass Sie als Allererstes eine kleine Verzeichnisstruktur anlegen, in der Sie dann für den Rest des Buches arbeiten werden.

Wenn Sie den IIS neu installiert haben, wird für Sie schon automatisch eine Standardwebsite eingerichtet, die auch bereits aktiv ist und ASP-Dateien enthält. Wenn Sie nach dem Start des Webservers in Ihren Browser

`http://localhost/`

eintippen, können Sie diese Seiten ansehen.

Dabei handelt es sich um die Dokumentation des IIS 5.1 Webservers, die auch die ASP-Dokumentation beinhaltet. Dort finden Sie auch viele Tipps und Hinweise zu weiterführenden Fragen beim Entwickeln.

An dieser Basiskonfiguration wollen wir zum Entwickeln erst einmal nichts verändern. Stattdessen schlagen wir Ihnen vor, dass Sie die Projekte in gesonderten Verzeichnissen ablegen, so dass Sie die Hilfeseiten wie gehabt erreichen.



Sie müssen diesem Vorschlag nicht unbedingt folgen, um ein Webdatenbank-Projekt mit ASP an den Start zu kriegen. Wenn Sie noch nicht so recht wissen, was die Zusammenhänge zwischen physischen Ordnern und virtuellen Verzeichnissen eines Webservers sind, sind Sie mit diesem Vorschlag aber schon mal auf der sicheren Seite.

### Ein Beispiel für den Einstieg



Wenn Sie die Dateien `miniskripte.zip`, `bsp1verzeichnis.zip` und `bsp2katalog.zip` von unserer Webseite heruntergeladen haben, werden Sie darin eine Verzeichnisstruktur vorfinden, die Projekte in verschiedenen Ordnern untergliedert wie in Abbildung 4.1.

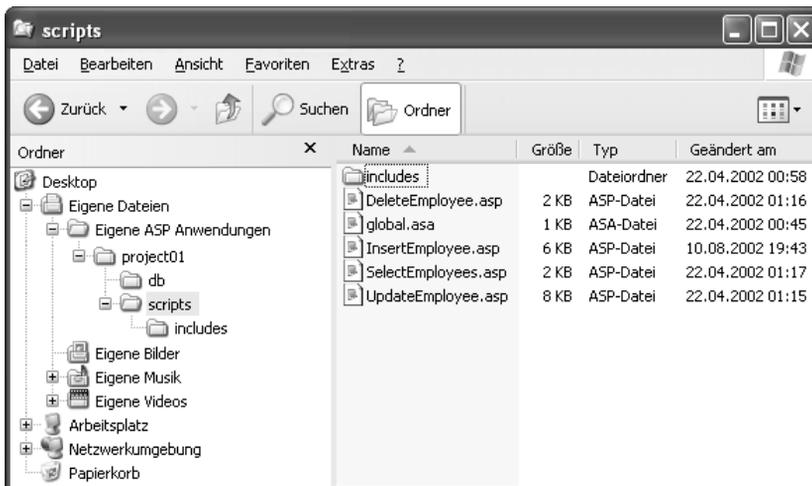


Abbildung 4.1: Verzeichnisstruktur für Webdatenbank-Projekte

Um diese Struktur zu übernehmen, entpacken Sie einfach die Dateien in ein beliebiges Verzeichnis, beispielsweise nach Eigene Dateien.

### Dateitypen in verschiedene Ordner packen

Es ist immer empfehlenswert, zunächst Projektordner anzulegen, in denen Sie dann alle Dateien abspeichern, die irgendwie im jeweiligen Webdatenbank-Projekt verwendet werden. Bei der Arbeit mit vielen verschiedenen Skriptdateien kann man nämlich ziemlich schnell die Übersicht verlieren, welche Datei zu welchem Projekt gehört.

Außerdem sollten Sie für die verschiedenen Arten von Dateien in einem Webdatenbank-Projekt zusätzlich separate Unterordner anlegen:

- ✓ einen Ordner für die ASP-Dateien
- ✓ einen weiteren Ordner für Dateien, die speziell in diesem Webdatenbank-Projekt in Ihre ASP-Dateien eingebunden werden
- ✓ einen Ordner, in dem Sie später mit Access die Webdatenbank anlegen
- ✓ einen Ordner für Grafiken, die Sie auf Ihrer Website verwenden



Später, wenn Sie ein Projekt live auf einen Webserver aufspielen möchten, können Sie dann ohne viel Mühe für die verschiedenen Ordner je nach enthaltenem Dateityp speziell zugeschnittene Zugriffsrechte vergeben. Zum Entwickeln brauchen Sie sich um diese Feineinstellungen des Webserver zwar noch nicht zu kümmern. Wenn Sie durch das geschickte Anlegen von Ordnern diesen späteren Arbeitsschritt zu Beginn schon berücksichtigen, sparen Sie sich später dafür ein Menge lästiger Aufräumarbeiten in Ihren Verzeichnissen.

### Eigene Skriptverzeichnisse für den IIS scharfschalten

Nun müssen Sie dem Webserver noch Bescheid sagen, in welchen Verzeichnissen sich ausführbare ASP-Dateien befinden. Der Webserver legt anhand Ihrer Angaben eine *virtuelle Verzeichnisstruktur* an, über die man später mit dem Browser in der Website navigieren kann.

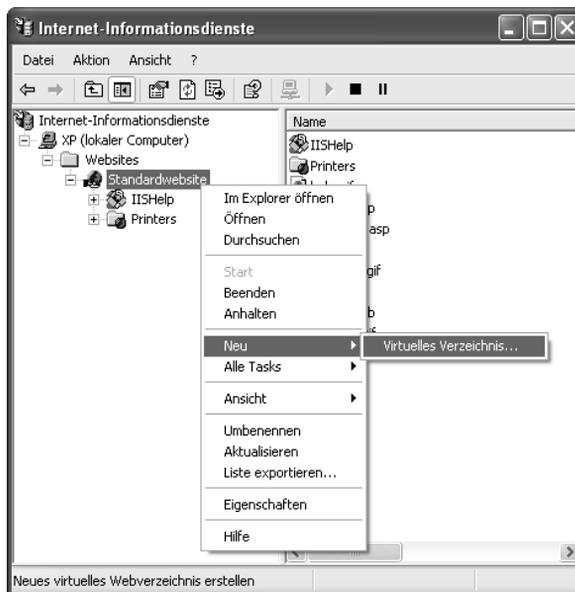


Abbildung 4.2: Ein neues virtuelles Verzeichnis erstellen

1. Dazu öffnen Sie unter **START|SYSTEMSTEUERUNG|VERWALTUNG** das Programm **INTERNET-INFORMATIONSDIENSTE**.
2. Zum Entwickeln legen Sie die virtuellen Verzeichnisse am besten unterhalb der **STANDARDWEBSITE** an, um die Webserver-Dokumentation an Ort und Stelle zu erhalten. Klicken Sie dazu mit der linken Maustaste einmal auf **STANDARDWEBSITE**. Klicken Sie dann einmal mit der rechten Maustaste, so dass das Kontextmenü erscheint. Wählen Sie dazu **NEU|VIRTUELLES VERZEICHNIS ...** aus.
3. Der Assistent zum Erstellen virtueller Verzeichnisse führt Sie von nun an Schritt für Schritt zum Ziel. Er wird Sie bitten, zu jedem neuen virtuellen Verzeichnis eine paar Voreinstellungen vorzunehmen: ein Alias, die Angabe des physischen Verzeichnisses und die Zugriffsberechtigungen.

### *James Bond alias 007*

Ein *Alias* ist ein Name für ein virtuelles Verzeichnis eines Webservers. Eine gute Entsprechung für ein Alias ist ein Pseudonym. Der Besucher Ihrer Website wird für gewöhnlich nur die Pseudonyme für die Verzeichnisse erfahren und mit ihrer Hilfe durch die Website-Verzeichnisse navigieren.

Der Webserver weiß, welche physischen Verzeichnisse wirklich hinter diesen Pseudonymen stecken und sucht dann automatisch an der richtigen Stelle nach den angeforderten Dateien.

Sie können jeden x-beliebigen Namen als Alias angeben, allerdings darf er keine Sonderzeichen wie Leerstellen oder Umlaute enthalten. Die kann Ihr Webserver nämlich nicht verarbeiten.

### *Skriptdateien müssen ausführbar sein*

Das virtuelle Verzeichnis, in dem sich die ASP-Skripte befinden, ist normalerweise das Basisverzeichnis Ihrer Website. So bezeichnet man das Verzeichnis, in dem sich die Startseite Ihrer Website befindet. Als *Alias* empfiehlt sich deshalb ein sprechender Name für Ihr Projekt, zum Beispiel `project01` für Ihr erstes Webdatenbank-Projekt.

Als *physisches Verzeichnis* wählen Sie den Ordner aus, den sie zuvor physisch für Ihre ASP-Dateien angelegt haben, also beispielsweise

```
d:/Eigene Dateien/Eigene ASP-Projekte/project01/scripts
```

Bei den *Zugriffsberechtigungen* können Sie es zum Entwickeln bei den Standardeinstellungen belassen: **SKRIPTS AUSFÜHREN** und **LESEN**.

Nachdem Sie das virtuelle Verzeichnis für Ihre ASP-Seiten so eingerichtet haben, können Sie sie ganz einfach mit Ihrem Browser aufrufen. In unserem Beispiel wäre das

`http://localhost/project01/`

oder

`http://localhost/project01/default.asp`



Wenn Sie unterhalb eines physischen Ordners Unterordner angelegt haben, werden diese automatisch mit denselben Einstellungen auch in die virtuelle Verzeichnisstruktur als Unterordner übernommen.

### Virtuelle Welten bauen

Auf diese Weise können Sie für Ihren Webserver eine eigene virtuelle Verzeichnisstruktur anlegen, die ganz anders aussehen kann als die echte Verzeichnisstruktur, die Sie in Ihrem Explorer oder Datei-Manager sehen können.

Wenn Sie unser Beispiel nachgebaut haben, würde die virtuelle Verzeichnisstruktur so aussehen wie in Abbildung 4.3.

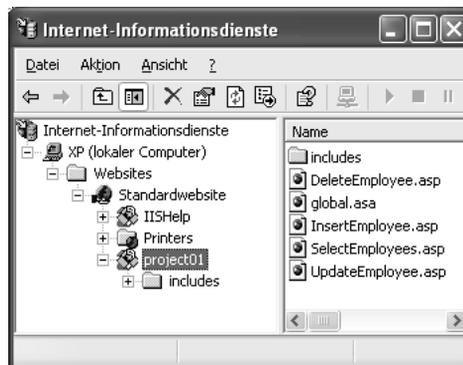


Abbildung 4.3: Vorschlag für eine virtuelle Verzeichnisstruktur

### Die Datenbank bleibt im geheimen Kämmerchen

Der Ordner, der die Datenbankdateien enthält, wird normalerweise nicht als virtuelles Verzeichnis freigegeben. Wenn Sie das auf einem echten Webserver tun würden, könnten nämlich wildfremde Menschen die komplette mdb-Datei mit Ihren Daten einfach mit einem Browser von Ihrem Webserver anfordern und downloaden. Alles, was sie dazu tun müssten, ist, die richtige Internetadresse aufzurufen. Das ist meistens nicht gewünscht und deshalb teilt man dem Webserver am besten überhaupt nicht erst mit, dass es diesen Ordner gibt.

## Wenn Sie Apache und MySQL benutzen

### Alle Wege führen nach htdocs

Das Standardwebverzeichnis des Apache-Webrowsers ist das Verzeichnis `htdocs`. Immer, wenn jemand von ihm eine bestimmte Internetseite anfordert, sucht der Webserver in diesem Verzeichnis nach der entsprechenden Datei. Wo sich dieses Verzeichnis genau befinden soll, können Sie bei der Installation des Webrowsers festlegen.

Damit auch alles funktioniert, müssen Sie anschließend noch ein paar Einstellungen vornehmen:

- ✓ Damit der Webserver das `htdocs`-Verzeichnis auch benutzen kann, muss er der Besitzer dieses Verzeichnisses und `all` seiner Dateien sein.
- ✓ Wenn Sie Apache frisch installiert haben, ist es zunächst so eingestellt, dass es nur Dateien mit der Endung `.html` akzeptiert. Damit auch PHP-Dateien vom Webserver auf den richtigen Weg gebracht werden, müssen Sie ihm mitteilen, dass alle Dateien mit der Endung `.php` an den PHP-Skriptinterpreter weitergereicht werden sollen

Alle dazu notwendigen Einstellungen an Ihrem Webserver können Sie nach der Installation in der Datei `httpd.conf` vornehmen. Sie ist die zentrale Konfigurationsdatei des Apache-Webrowsers und bietet Ihnen viele Einstellungsmöglichkeiten, um den Webserver an Ihre Bedürfnisse anzupassen. Eine schrittweise Anleitung zur Konfiguration des Apache für PHP-Dateien und zur Vergabe von Verzeichnisrechten finden Sie im Anhang A *LAMP installieren*.

Unterhalb Ihres `htdocs`-Verzeichnisses können Sie auch weitere Unterordner anlegen, um die Dateien Ihrer Webdatenbank-Projekte strukturiert unterzubringen. Sie sind dann automatisch auch für den Webserver zugänglich.

### Ordnung schaffen im htdocs-Verzeichnis

Damit Sie beim Ausprobieren und Entwickeln am Ende nicht ein unübersichtliches Durcheinander von PHP-Dateien in seinem `htdocs`-Ordner anrichten, ist es immer eine gute Idee, unterhalb des `htdocs`-Ordners weitere Ordner für die verschiedenen Projekte anzulegen.

Innerhalb jedes Projektes sollten Sie separate Unterordner einrichten:

- ✓ einen Ordner für Bilder
- ✓ einen Ordner für PHP-Klassen, die innerhalb eines Projektes auf verschiedenen Seiten verwendet werden. (Worum es sich bei PHP-Klassen genau handelt und warum man sie braucht, werden wir im Teil II natürlich noch eingehend erläutern.)



Wenn Sie die Dateien `miniskripte.tar.gz`, `bsp1verzeichnis.tar.gz` und `bsp2katalog.tar.gz` von unserer Webseite heruntergeladen haben, werden Sie darin diese Verzeichnisstruktur vorfinden. Wenn Sie sie übernehmen möchten, entpacken Sie einfach die Pakete nach `htdocs`.

Ihr `htdocs`-Verzeichnis könnte also beispielsweise so aussehen wie in Abbildung 4.4.

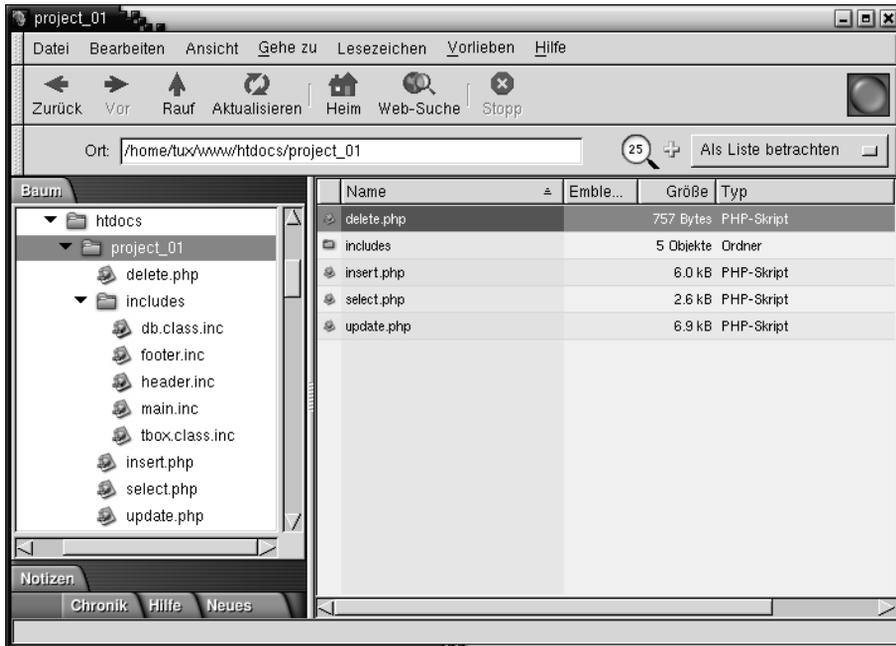


Abbildung 4.4: `htdocs`-Verzeichnisstruktur

Um später die Startseite des Webdatenbank-Projektes `projekt01` aufzurufen, würden Sie nun diese Adresse in Ihren Browser eintippen:

`http://localhost/project01/`

oder

`http://localhost/project01/index.php`