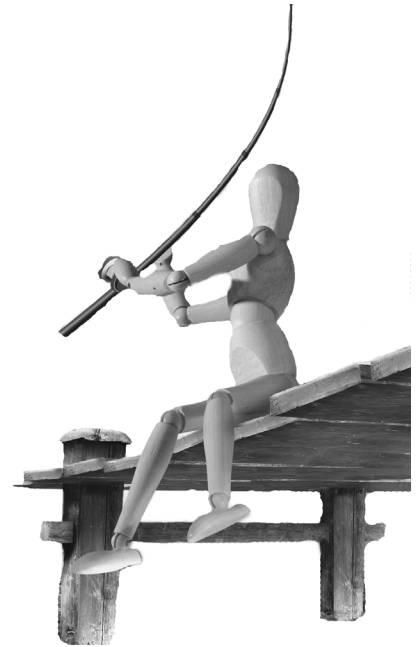


1. Installation und Konfiguration

Dieses Kapitel beschreibt die Installation eines lokalen Entwicklungssystems, das aus Apache, PHP und MySQL besteht. Dabei werden sowohl Windows als auch Linux als Entwicklungsplattform berücksichtigt.

Neben reinen Installationstipps vermittelt dieses Kapitel aber auch eine Menge Hintergrundinformationen zur Konfiguration und Administration:

- » Die Passwort-Absicherung von Apache-Verzeichnissen
- » Die Vorstellung wichtiger MySQL-Administrationswerkzeuge (mysql, mysqladmin, mysqldump, MySQL Administrator, MySQL Query Browser, MySQL Workbench)
- » Zeichensatzfragen (Latin1 oder Unicode)



1.1 Versionen und Lizenzen

Bevor Sie mit dem Installieren beginnen, sollten Sie kurz überlegen, welche Version Sie nutzen möchten und wie die rechtlichen Grundlagen aussehen. Dieser Abschnitt gibt einige Hintergrundinformationen dazu.

Nicht immer ist die neueste Software-Version auch die beste. Da es in der Open-Source-Welt nicht immer ganz einfach ist, einen Überblick über die Versionen zu bewahren, und da beinahe jedes Open-Source-Projekt ein eigenes System zur Nummerierung der Versionen verwendet, hier einige Empfehlungen (Stand: August 2009).

Apache-Versionen

Die stabile Version von Apache hat die Versionsnummer 2.2.*n*. Die Versionen 1.3.*n* und 2.0.*n* werden nicht mehr weiterentwickelt, es gibt aber noch regelmäßige Sicherheits-Updates. Für die Beispiele aus diesem Buch spielt es keine Rolle, welche Apache-Version Sie einsetzen.

Alle drei genannten Versionen funktionieren effizient und zuverlässig. Auch bei der Konfiguration der Standardeinstellungen gibt es kaum Unterschiede. Während der Arbeit an diesem Buch haben wir die aktuelle Apache 2.2.*n*-Version eingesetzt.

Seit Version 2 stehen für Apache mehrere Threading-Module zur Auswahl. Am effizientesten ist das worker-Modul, das eine besonders gute Parallelverarbeitung mehrerer Teilprozesse erlaubt. Allerdings sind nicht alle unter Unix/Linux üblichen Bibliotheken, auf die PHP selbst oder diverse PHP-Erweiterungen zugreifen, thread-sicher. Das kann in sehr seltenen Fällen zu Abstürzen führen. Auf einem Entwicklungssystem wird dieser Fall wahrscheinlich nie eintreten, auf einem Webserver, der Hunderttausende Seiten pro Tag liefert, ist die Fehlerwahrscheinlichkeit aber deutlich höher.

Aus diesem Grund empfiehlt die PHP-Website, PHP und Apache in Kombination mit dem prefork-Modul zu betreiben. Das ist weniger effizient, aber dafür stabiler.

<http://php.net/manual/en/faq.installation.php#faq.installation.apache2>

PHP-Versionen

Seit die PHP-Entwickler im Juli 2004 PHP 5.0 fertiggestellt haben, wurde PHP in vielen kleinen Schritten weiterentwickelt und optimiert. Die folgende Aufzählung nennt die wichtigsten Neuerungen in PHP 5.1, 5.2 und 5.3 aus der Sicht dieses Buchs. (Machen Sie sich keine Sorgen, wenn Sie mit den vielen Fachausdrücken und Abkürzungen vorerst nichts anfangen können. Viele Begriffe werden im Verlauf dieses Buchs noch genauer beschrieben.)

- » **PHP 5.1:** neue Datenbankschnittstelle PDO mit MySQL-Unterstützung
- » **PHP 5.2:** effizientere Speicherverwaltung, diverse neue Funktionen und Bibliotheken (filter, json, zip)
- » **PHP 5.3:** Namespaces, Lambda-Funktionen mysqlnd-Backend für die Erweiterungen mysql, mysqli und PDO. (Das bedeutet, dass die externe Bibliothek libmysql nicht mehr erforderlich ist; aus Entwicklersicht bleibt alles gleich, die Erweiterungen mysql, mysqli und PDO sind kompatibel zu den bisherigen Versionen.)

In Kurzform kann man die Entwicklung von PHP 5.0 zu 5.3 mit dem Wort »evolutionär« zusammenfassen, d.h. wirklich fundamentale Neuerungen fehlen. Grundlegend neu wird erst **PHP 6** sein: Für diese Version versprechen die Entwickler eine vollständige Unicode-Unterstützung. Es steht allerdings in den Sternen, wann diese Version endlich fertig wird (angekündigt ist sie schon lange). Wir tippen auf 2011 und schließen nicht aus, dass es vorher noch eine Version 5.4 geben wird ...

Alle Beispiele dieses Buchs setzen voraus, dass Sie zumindest PHP 5.0 einsetzen. Soweit einzelne PHP-Bibliotheken oder -Funktionen erst in späteren PHP-Versionen implementiert wurden, weisen wir in diesem Buch explizit darauf hin. Auch bei allen größeren Beispielprogrammen geben wir an, welche PHP-Version erforderlich ist. Unsere Empfehlung: Verwenden Sie aus Sicherheits- und Effizienzgründen zumindest Version 5.2 und nach Möglichkeit Version 5.3.

MySQL-Versionen

MySQL 5.1 wurde Ende 2008 fertiggestellt. Gegenüber der Vorgängerversion bietet MySQL 5.1 die folgenden Neuerungen:

- » Partitionierung (Verteilung großer Tabellen auf mehrere Datenbanken bzw. Rechner)
- » Neue Replikationsarchitektur (*row based replication*)
- » Plugin-Architektur für Tabellen-Engines
- » Ereignisse (regelmäßige Ausführung von SQL-Kommandos bzw. zeitgesteuerte Trigger)
- » Verbesserungen in der virtuellen Information_Schema-Datenbank
- » Elementare XML-Funktionen

Seit Mitte 2009 steht zudem eine Betaversion von **MySQL 5.4** zur Verfügung. Die Endversion ist für Ende 2009 versprochen, es kann aber gut sein, dass die Fertigstellung wie bei früheren MySQL-Versionen länger als erwartet dauert. (MySQL 5.2 und 5.3 gibt es nicht, d.h., es gab einen Versionssprung von 5.1 auf 5.4.) So schnell und überraschend MySQL 5.4 auftauchte, so klein ist auch die Liste der Änderungen:

- » Der InnoDB-Tabellentreiber wurde für High-End-Hardware (d.h. für Rechner mit mehr als 16 CPUs/Cores) stark optimiert und führt nun manche Benchmarks beinahe doppelt so schnell aus wie früher. Bemerken werden Sie diese Verbesserungen aber nur bei riesigen Datenbanken auf expliziter Server-Hardware.
- » Mit SIGNAL und RESIGNAL ist endlich eine ordentliche Fehlerbehandlung in Stored Procedures möglich.
- » *Prepared Statements* können nun auch mit OUT-Parametern von Stored Procedures umgehen.
- » Manche Sub-SELECTs und JOINS werden besser optimiert und wesentlich schneller ausgeführt.

Bis auf die Geschwindigkeitssteigerungen bietet MySQL 5.1 also keine nennenswerten Neuerungen. Für Entwicklungssysteme und kleinere Datenbankanwendungen ist es insofern vollkommen gleichgültig, ob Sie MySQL 5.1 oder MySQL 5.4 einsetzen! Nur Enterprise-Anwender werden von MySQL 5.4 spürbar profitieren.

Um die Kompatibilität unserer Beispiele mit MySQL 5.4 sicherzustellen, haben wir zur Arbeit an diesem Buch eine Beta-Version von MySQL 5.4 eingesetzt. Dennoch funktionieren nahezu alle Beispiele dieses Buchs auch mit MySQL 5.1 und viele sogar mit 5.0.

Unsere Empfehlung: Bis MySQL 5.4 endgültig fertiggestellt ist, sollten Sie nach Möglichkeit Version 5.1 einsetzen. Diese Version verspricht über die nächsten Jahre die höchste Stabilität und die beste Wartung. Ein späterer Wechsel auf Version 5.4 sollte problemlos sein, Änderungen am internen Format der Datenbankdateien gibt es nicht.

Selbst wenn Ihnen nur **MySQL 5.0** zur Verfügung steht, ist dies kein großes Problem. Achten Sie aber darauf, dass es sich um eine aktuelle Version innerhalb der 5.0.n-Serie handelt. In den ersten MySQL 5.0.n-Versionen gab es noch große Stabilitätsprobleme in den Bereichen Stored Procedures, Trigger und GIS-Funktionen. Das vielleicht gewichtigste Argument, das gegen den Einsatz von MySQL 5.0 spricht, ist das der Wartung. Im sogenannten »MySQL Lifecycle« verspricht MySQL die Wartung seiner Datenbank-Server für fünf Jahre ab der Fertigstellung (*general availability*). Dieser lange Wartungszeitraum gilt allerdings nur für die *Enterprise Edition*. Bei der kostenlosen *Community Edition* verkürzt sich dieser Zeitraum auf zwei Jahre. Deswegen endet deren aktive Wartung bereits Ende 2009.

<http://www.mysql.com/about/legal/lifecycle/>

Die nächste MySQL-Version wird voraussichtlich **MySQL 6** sein. Da Oracle gerade im Begriff ist, Sun und damit auch MySQL zu übernehmen, sind Prognosen über die in MySQL 6 enthaltenen Funktionen und über den Veröffentlichungszeitraum nahezu unmöglich. Ursprünglich war geplant, in MySQL 6 Online-Backup-Funktionen, eine allgemeingültige Foreign-Key-Architektur sowie diverse neue Tabellentreiber (Falcon, Maria) zu integrieren.

Das Versionsdilemma

Webhoster stellen ihren Kunden Webspace und eine Umgebung zur Verfügung, die in der Regel aus Linux + Apache + PHP + MySQL = LAMP besteht. Manchmal kommt statt Linux auch BSD oder Solaris zum Einsatz. Die Webhosting-Firmen stehen dabei vor dem im Vorwort bereits angesprochenen Dilemma: Auf der einen Seite wünschen sich viele Kunden die jeweils neueste PHP- und MySQL-Version; auf der anderen Seite muss ein Rechner, auf dem womöglich mehrere Kunden ihre Webpräsenz haben, absolut stabil laufen.

Die meisten Webhoster agieren in dieser Situation konservativ und stellen Stabilität vor Funktionsreichtum: Eine etwas ältere Software-Version, die erwiesenermaßen stabil läuft, wird der gerade aktuellsten Version vorgezogen.

Die Zeitspanne von der Fertigstellung einer neuen PHP- oder MySQL-Version bis zu deren breiter Verfügbarkeit wird noch durch die Versionspolitik der gängigen Linux-Distributionen verlängert: Die meisten Linux-Distributoren führen normalerweise keine grundlegenden Versions-Updates durch, sondern integrieren neue PHP- und MySQL-Versionen erst in der jeweils nächsten Version der Distribution. Und auch eine neue Linux-Distribution wird nicht schon am ersten Tag ihres Lebenszyklus von Webhosting-Firmen eingesetzt ...

Damit sollte klar sein, warum es durchaus ein ganzes Jahr dauern kann, bis große Webhoster die gerade neueste PHP- oder MySQL-Version anbieten. Wenn Sie also Webseiten programmieren, die später auf einem Webhosting-Server laufen sollen, sollten Sie sich bereits bei der Entwicklung auf die PHP- und MySQL-Versionen konzentrieren, die Ihr Provider anbietet. Konfigurieren Sie auch Ihr Testsystem entsprechend!

Wenn Sie nicht so lange warten möchten, müssen Sie einen sogenannten root-Server anmieten, konfigurieren und verwalten. Das kostet kein Vermögen, setzt aber gute Linux/Unix-

Kenntnisse voraus. Sie müssen sich selbst um Sicherheits-Updates kümmern, sämtliche Server-Dienste möglichst sicher konfigurieren, selbst Backups anlegen etc. – lauter Aufgaben, die sonst der Webhoster übernimmt.

Lizenzen

Apache, PHP und MySQL sind Open-Source-Software. Das bedeutet nicht nur, dass Sie diese Produkte kostenlos aus dem Internet herunterladen und anwenden dürfen, sondern auch, dass der gesamte Quellcode dieser Produkte zur Verfügung steht. Für alle drei Produkte gelten allerdings verschiedene Lizenzbedingungen, die Sie einhalten müssen.

Im Folgenden setzen wir voraus, dass Sie typische Webprojekte entwickeln, also z. B. private oder kommerzielle Webseiten. Die folgende Diskussion berücksichtigt hingegen nicht den Fall, dass Sie einen Webserver, eine Programmiersprache oder ein Datenbanksystem entwickeln, also Produkte, die dieselben oder vergleichbare Aufgaben erfüllen wie Apache, PHP oder MySQL selbst. Auch das ist selbstverständlich möglich und zulässig, allerdings ist dann ein genaues Studium der originalen Lizenztexte unumgänglich (siehe die Links etwas weiter unten).

Apache und PHP: Für diese beiden Produkte gilt sinngemäß: Die Software kann ohne Einschränkungen kostenlos eingesetzt werden, auch in kommerziellen Projekten. Sie dürfen Apache und PHP auch als Teil eines Produkts weitergeben.

MySQL: Bei MySQL ist die Lage deutlich komplizierter. Für MySQL gibt es zwei Lizenzen, die GPL (*GNU Public License*, siehe Link unten) und eine kommerzielle Lizenz. Daraus ergeben sich folgende Konsequenzen:

- » Wenn für Ihr Produkt selbst die GPL gilt (oder eine andere Lizenz, die mit der GPL kompatibel ist), ist die Weitergabe und Nutzung kostenlos erlaubt.
- » Auch wenn Sie eine Website auf der Basis von Apache, PHP und MySQL entwickeln – egal, ob diese Seite dann privat oder kommerziell genutzt wird –, ist eine kostenlose Nutzung des MySQL-Servers zulässig (*free use for those who never copy, modify or distribute*).
- » Problematisch ist hingegen die **Weitergabe** bzw. der **Verkauf** von kompletten Produkten, die MySQL als Datenbanksystem einsetzen. Angenommen, Sie entwickeln eine Buchhaltungssoftware, wobei Sie PHP als Programmiersprache und MySQL als Datenbankserver einsetzen. Jetzt möchten Sie Ihr Buchhaltungsprogramm verkaufen (wobei nicht die GPL oder eine vergleichbare Lizenz gilt): In diesem Fall muss der Käufer neben Ihrem Produkt auch eine kommerzielle Version des MySQL-Servers erwerben!

Der Grund: Ihr Buchhaltungsprogramm gilt als *derived work*, das von MySQL abhängig ist. Entweder Sie beachten nun die Spielregeln der GPL und veröffentlichen Ihr Programm ebenfalls unter der GPL oder Sie brauchen eine kommerzielle MySQL-Lizenz.

Die Firma Oracle, die 2009 Sun und damit auch MySQL gekauft hat, bietet kommerzielle Lizenzen unter dem Namen *MySQL Enterprise* in einem Abonnementmodell an. In der günstigsten Variante kostet der kommerzielle Einsatz von MySQL ca. 480 Euro pro Rechner und Jahr (Stand August 2009).

Dementsprechend stellt die Website mysql.com zwei unterschiedliche Versionen des MySQL Servers zu Auswahl: Die kostenlose *Community Edition* und die kostenpflichtige *Enterprise Edition*. Beide Versionen basieren auf demselben Code und bieten dieselben Funktionen. Abschließend noch die Links zu den vollständigen Apache-, PHP- und MySQL- und GPL-Lizenztexten und zu ergänzenden Informationen:

<http://www.apache.org/licenses/>
<http://www.php.net/license/>
<http://www.mysql.com/products/enterprise/>
<http://www.mysql.com/about/legal/licensing/>
<http://www.mysql.com/about/legal/licensing/foss-exception.html>
<http://www.gnu.org/licenses/licenses.html>

1.2 Installation unter Windows

Die folgenden Abschnitte beschreiben die Einzelinstallation von Apache 2.2, PHP 5.3 und MySQL 5.1/5.4. Die Einzelinstallation hat den Vorteil, dass Sie die Version jeder Komponente selbst auswählen können und eher verstehen, welche Programme auf Ihrem Rechner installiert sind und wie sie konfiguriert werden. Das erleichtert auch die spätere Wartung bzw. das Update einer bestimmten Komponente.

Noch bequemer ist die Installation, wenn Sie ein Komplettpaket verwenden, das alle drei Komponenten sowie diverse weitere Programme in sich vereint. Das bekannteste derartige Paket ist XAMPP für Windows:

<http://www.apachefriends.org/de/xampp-windows.html>

Der Vorteil von XAMPP gegenüber der Einzelinstallation besteht darin, dass Sie eine Menge Zeit und Nerven sparen. Dem steht als wesentlicher Nachteil der Umstand gegenüber, dass Sie keinen Einfluss auf die einzelnen Komponenten nehmen können, sondern nur das Komplettpaket als Ganzes installieren können. Insbesondere ist XAMPP ungeeignet, um Beta- oder Testversionen auszuprobieren – XAMPP enthält nämlich nur offiziell unterstützte Versionen.

Windows 7

Wir haben für dieses Kapitel Windows 7 in der 64-Bit-Variante als Grundlage genommen. Selbstverständlich funktioniert die Installation auch unter Windows Vista oder XP. Vista- und Windows-7-Anwender finden im Internet in diversen Installationsanleitungen immer wieder zwei Empfehlungen:

- » **Deaktivierung der UAC:** Nach unseren Erfahrungen ist eine Deaktivierung der UAC (*User Account Control*) nicht unbedingt für eine erfolgreiche Installation erforderlich. Sie ersparen sich aber natürlich eine Menge sinnloser Unterbrechungen durch Dialoge in der Art »Wollen Sie das wirklich?«. (Unter Windows 7 sind diese Warnungen zum Glück deutlich weniger geworden.)

Wenn Sie die UAC deaktivieren möchten, geben Sie in das Eingabefeld des Startmenüs `msconfig` ein und starten das gleichnamige Programm. Im Dialogblatt `TOOLS` wählen Sie den Eintrag `BENUTZERKONTENSTEUERUNG DEAKTIVIEREN` aus und klicken auf den Button `STARTEN`. Damit die geänderten Einstellungen aktiv werden, ist ein Neustart von Windows erforderlich. Auf demselben Weg können Sie die UAC bei Bedarf später wieder aktivieren.

- » **Installation außerhalb des Programme-Verzeichnisses:** Diese Empfehlung bezieht sich darauf, dass gewöhnliche Benutzer im Verzeichnis `C:\Programme` bzw. `C:\Program Files` keine Schreibrechte haben. Das führt zu gravierenden Problemen bei der Konfiguration und Wartung der Programme. Aus diesem Grund sollten Sie Apache, PHP und MySQL jeweils in eigene Verzeichnisse installieren. In diesem Buch verwenden wir die Verzeichnisse `C:\apache22`, `C:\php53` und `C:\mysql151` bzw. `C:\mysql154`.

IIS deaktivieren

Falls Sie auf einer Windows-Server-Version arbeiten, sollten Sie als Erstes sicherstellen, dass der Microsoft Internet Information Server (kurz IIS) nicht läuft. Die Installation von Apache in der Standardeinstellung (Port 80) scheitert, wenn IIS parallel läuft.

Zur Kontrolle bzw. zur Deinstallation des IIS starten Sie unter Windows 7 in der `SYSTEMSTEUERUNG` das Modul `PROGRAMM DEINSTALLIEREN` und klicken auf den Link `WINDOWS-FEATURES AKTIVIEREN ODER DEAKTIVIEREN`. Im nun erscheinenden Dialog stellen Sie sicher, dass die `INTERNETINFORMATIONSDIENSTE` deaktiviert sind. Unter Windows XP führen Sie `SYSTEMSTEUERUNG\SOFTWARE\WINDOWS-KOMPONENTEN` aus und deaktivieren im `ASSISTENT FÜR WINDOWS-KOMPONENTEN` die Option IIS.

Apache installieren

Apache steht unter www.apache.org (bzw. auf diversen Mirrors) als Installationsprogramm für Windows zur Verfügung. Laden Sie die `apache_2.2.n_xxx.msi`-Datei auf Ihren Rechner, und starten Sie das Installationsprogramm im Windows-Explorer durch einen Doppelklick. Es erscheint ein Dialog zur Grundkonfiguration (siehe Abbildung 1.1). Darin müssen Sie den Domain-Namen (z.B. `uranus.sol`) und den Server-Namen (ebenfalls `uranus.sol` oder `www.uranus.sol`) angeben. Diese Informationen werden auch dann benötigt, wenn Ihr Rechner nur als einzelner Testrechner läuft. Im Regelfall werden die richtigen Werte automatisch in die Dialogfelder eingetragen. Des Weiteren müssen Sie angeben, unter welcher E-Mail-Adresse der Administrator des Webservers (also Sie) erreichbar ist und ob der Webserver den Port 80 verwenden soll.



Abbildung 1.1: Apache-Installation

Im nächsten Dialogblatt wählen Sie die Variante CUSTOM, damit Sie in der Folge als Installationsort ein beliebiges Verzeichnis angeben können (z.B. C:\apache22). Den Installationsumfang belassen Sie unverändert. Nach der Installation wird Apache automatisch als Service (Dienst) eingerichtet und sofort gestartet. In der Taskleiste erscheint ein kleines Icon, das den Zustand von Apache anzeigt (ein grüner Pfeil, wenn das Programm läuft).

Apache starten und stoppen

Sie können Apache mit den Menükommandos PROGRAMME|APACHE HTTP SERVER|CONTROL APACHE SERVER|START bzw. |STOP starten bzw. beenden. Auch das Apache-Icon in der Taskleiste kann zum Starten bzw. Beenden verwendet werden. PROGRAMME|APACHE HTTP SERVER|CONFIGURE APACHE SERVER|EDIT HTTPD.CONF startet den Editor Notepad und zeigt darin die Apache-Konfigurationsdatei an.

Als ersten Test, ob Apache tatsächlich läuft, öffnen Sie in Ihrem Webbrowser die Seite `http://localhost`. Sie sollten die Standardstartseite des Webbrowsers zu sehen bekommen (»It works!«). Abschließend noch zwei Verzeichnisangaben, die für den empfohlenen Installationsort C:\apache22 gelten:

Konfigurationsdatei: C:\apache22\conf\httpd.conf
 Webdateien: C:\apache2\htdocs

MySQL 5.1/5.4 installieren

Kostenlose MySQL-Installationsdateien der *Community Edition* von MySQL 5.1 und 5.4 finden Sie in einer 32- und einer 64-Bit-Version für Windows auf den folgenden Seiten:

<http://dev.mysql.com/downloads/mysql/5.1.html>

<http://dev.mysql.com/downloads/mysql/5.4.html>

Dort entscheiden Sie sich für die Download-Datei WINDOWS MSI INSTALLER, je nach Windows-Version in der 32- oder in der 64-Bit-Version. Die *.msi-Datei speichern Sie auf Ihrer lokalen Festplatte und starten sie per Doppelklick. Im Installationsassistenten entscheiden Sie sich für die Installationsvariante CUSTOM, damit Sie im nächsten Schritt den Installationsort festlegen können (unser Vorschlag: C:\mysql51 bzw. C:\mysql54). Den Installationsumfang belassen Sie unverändert. Für den Ablauf der Installation spielt es keine Rolle, ob Sie sich für MySQL 5.1 oder 5.4 entscheiden – beide Versionen nutzen dasselbe Installationsprogramm.

MySQL-Update durchführen

Während eine Neuinstallation des MySQL-Servers zumeist problemlos gelingt, führt ein Update oft zu Problemen. In diesem Zusammenhang einige Tipps:

- » Bevor Sie mit der Installation der neuen Version beginnen, sollten Sie (noch mit der alten Version!) ein Backup aller Datenbanken durchführen. Dazu können Sie das Programm `mysqldump` oder ein Administrationsprogramm (MySQL Administrator, phpMyAdmin etc.) verwenden.
- » Beenden Sie den MySQL-Serverdienst. Dazu suchen Sie in der Systemsteuerung nach VERWALTUNG|LOKALE DIENSTE ANZEIGEN und stoppen dort den Dienst *MySQL*. (Das Installationsprogramm ist unbegreiflicherweise nicht in der Lage, den Dienst selbst zu beenden.)
- » Es kann sein, dass sich der im Folgenden beschriebene Assistent kurz vor Abschluss der Konfigurationsarbeiten über eine fehlende Tabelle beklagt (z.B. *Table 'mysql.servers' doesn't exist*). Abhilfe: Deinstallieren Sie den MySQL-Server, und löschen Sie dann die Datenbank `mysql`, indem Sie das folgende Verzeichnis manuell löschen:

```
C:\Benutzer\All Users\MySQL\MySQL Server 5.1\data\mysql
```

Anschließend installieren Sie den MySQL-Server neu. Bei der Installation wird die Datenbank `mysql` neu eingerichtet, diesmal inklusive aller für die aktuelle Version erforderlichen Tabellen.

Durch diese zugegebenermaßen recht radikale Vorgehensweise verlieren Sie leider alle MySQL-Benutzerdaten samt Passwörtern. Sie müssen diese Daten später aus Ihrem Backup wieder einspielen. (Abermals ist nicht ganz klar, warum sich das Installationsprogramm nicht einfach um ein Update der mysql-Datenbank kümmert. Ein entsprechendes Update-Script gibt es zwar, seine Ausführung setzt aber voraus, dass der MySQL-Server läuft – und diese Voraussetzung ist während der Konfiguration nicht gegeben.)

Der MySQL-Konfigurationsassistent

Der Konfigurationsassistent wird nach der Installation automatisch ausgeführt (siehe Abbildung 1.2). Bei Bedarf können Sie diesen Assistenten später mit `PROGRAMME\MySQL\MySQL SERVER\MySQL SERVER INSTANCE CONFIG WIZARD` neuerlich starten. Der Assistent erfüllt zwei Aufgaben: Er richtet die MySQL-Konfigurationsdatei ein und sichert den MySQL-Server durch ein Passwort ab. Die folgende Aufzählung beschreibt die wichtigsten Schritte des Assistenten. Sie machen nichts falsch, wenn Sie im Zweifelsfall den Vorgaben des Assistenten folgen.

- » **CONFIGURATION TYPE:** Im ersten Schritt sollten Sie sich für eine `DETAILED CONFIGURATION` entscheiden.
- » **SERVER TYPE:** Hier können Sie zwischen `DEVELOPER`, `SERVER` oder `DEDICATED SERVER` wählen. Die Option beeinflusst, welchen Teil des Hauptspeichers der MySQL-Server für sich zu reservieren versucht. Je größer der Anteil ist, desto schneller läuft der Server, aber desto stärker sind auch alle anderen Programme beeinträchtigt. Für Webentwickler ist `DEVELOPER` die optimale Einstellung. Falls Sie Benchmark-Tests durchführen, erzielen Sie bei großen Datenbanken mit `SERVER` oder `DEDICATED SERVER` wesentlich bessere Ergebnisse.
- » **DATABASE USAGE:** Die drei Optionen `MULTIFUNCTIONAL`, `TRANSACTIONAL ONLY` und `NON-TRANSACTIONAL ONLY` entscheiden darüber, für welche Tabellentypen der MySQL-Server konfiguriert wird (für Experten: `MyISAM`, `InnoDB` oder beide). Die Defaulteinstellung lautet `MULTIFUNCTIONAL`, und dabei sollten Sie es auch belassen.
- » **INNO DB TABLESPACE SETTING:** In diesem Punkt geben Sie an, wo `InnoDB`-Datenbankdateien gespeichert werden sollen (standardmäßig einfach im Installationsverzeichnis).
- » **CONCURRENT CONNECTIONS:** Dieser Punkt bestimmt die Anzahl der Datenbankverbindungen, die üblicherweise gleichzeitig offen sein werden. Solange Sie MySQL auf einem Entwicklungsrechner laufen lassen, wird die Anzahl ziemlich niedrig sein. Wählen Sie daher die Option `DECISION SUPPORT` (20 Verbindungen). Wenn der MySQL-Server dagegen im Vollbetrieb in Kombination mit einer gut frequentierten Webseite läuft, wird die Anzahl der Verbindungen wesentlich höher sein (Option `ONLINE TRANSACTION PROCESSING`).

- » **ENABLE TCP/IP NETWORKING:** Unter Windows bestehen zwei Möglichkeiten, wie der MySQL-Server mit den Anwendungen kommuniziert: über sogenannte *Named Pipes* oder über das Netzwerkprotokoll TCP/IP. Der Assistent schlägt vor, TCP/IP zu aktivieren und die Port-Nummer 3306 zu verwenden. Auch hier sollten Sie dem Vorschlag des Assistenten folgen, wenn es nicht wirklich triftige Gründe gibt, die dagegensprechen.

Die Option **ADD FIREWALL EXCEPTION FOR THIS PORT** bewirkt, dass in die Windows-Firewall eine Regel eingebaut wird, die im gesamten Netzwerk eine Kommunikation über den Port 3306 zulässt. Diese Option sollten Sie nur aktivieren, wenn Clients von anderen Rechnern auf den MySQL-Server zugreifen sollen.

Die Option **ENABLE STRICT MODE** bewirkt, dass MySQL sich weitestgehend an den SQL-Standard hält. Diese Option ist standardmäßig aktiviert – belassen Sie es dabei!

- » **DEFAULT CHARACTER SET:** MySQL unterstützt alle möglichen Zeichensätze zur Speicherung von Texten in der Datenbank. In diesem Punkt legen Sie den Defaultzeichensatz fest. Dieser gilt nur, wenn beim Datenbankdesign nicht explizit ein anderer Zeichensatz gewählt wird (d.h., die Einstellung schränkt Sie später in keiner Weise ein). Da alle Beispiele dieses Buchs den Latin1-Zeichensatz nutzen, empfiehlt es sich, diesen Zeichensatz als Defaulteinstellung zu verwenden.
- » **INSTALL AS WINDOWS SERVICE:** Der MySQL-Server kann wahlweise als *.exe-Datei oder als Windows-Dienst gestartet werden. Die zweite Variante ist komfortabler und sicherer.

Die Option **INCLUDE BIN DIRECTORY IN WINDOWS PATH** bewirkt, dass das bin-Verzeichnis der MySQL-Installation in die Variable `PATH` eingebaut wird. Das erleichtert den Aufruf von Administrationswerkzeugen von `mysql`, `mysqladmin` oder `mysqldump` erheblich. Aktivieren Sie diese Option!

- » **SECURITY OPTIONS:** In diesem wahrscheinlich wichtigsten Punkt des Assistenten legen Sie fest, wer eine Verbindung zum MySQL-Server herstellen kann und ob dabei ein Passwort erforderlich ist. Die sicherste und von den Autoren empfohlene Einstellung ist in Abbildung 1.2 zu sehen: Als einziger Benutzer wird `root` eingerichtet und durch ein Passwort abgesichert. `root` gilt als Datenbankadministrator und darf sich nur vom lokalen Rechner aus anmelden (nicht von einem anderen Rechner im Netzwerk). Darüber hinaus gibt es keine anonymen Accounts (Verbindungsaufbau ohne Passwort).

Selbstverständlich können Sie später weitere Benutzer einrichten, die auf MySQL bzw. auf bestimmte MySQL-Datenbanken zugreifen dürfen (mehr dazu finden Sie in den Abschnitten 7.2 und 15.1).



Abbildung 1.2: Der MySQL-Konfigurationsassistent

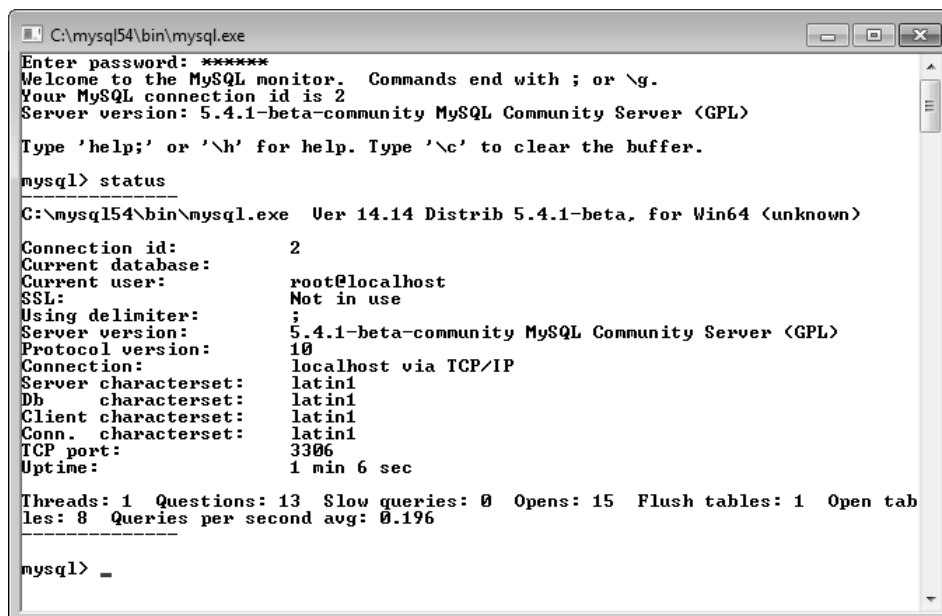


Abbildung 1.3: Der MySQL-Kommandozeileninterpreter

Testen, ob MySQL läuft

Um zu testen, ob die MySQL-Installation erfolgreich war, starten Sie den MySQL-Kommandozeileninterpreter `mysql.exe` durch `PROGRAMME|MYSQL|MYSQL SERVER|MYSQL COMMAND LINE CLIENT`. Dabei geben Sie dasselbe Passwort an, das Sie während der MySQL-Konfiguration verwendet haben. Wenn alles klappt, erscheint im Eingabefenster nun die MySQL-Eingabeaufforderung. Geben Sie als erstes Kommando `status` ein. Damit werden die wichtigsten Verbindungsparameter angezeigt. Das Ergebnis sollte in etwa wie in Abbildung 1.3 aussehen. Mit `exit` beenden Sie den Kommandozeileninterpreter.

MySQL Administrator und Query Browser installieren

Grundsätzlich können Sie mit dem gerade vorgestellten Programm `mysql.exe` und einigen ähnlichen Werkzeugen (`mysqldump.exe` und `mysqladmin.exe`, um die beiden wichtigsten zu nennen) die gesamte Administration des MySQL-Servers durchführen. Besonders praktisch ist das aber nicht. Wesentlich mehr Komfort bieten die Programme MySQL Administrator und Query Browser, die Teil der MySQL-GUI-Tools sind.

Sie finden diese Programme unter <http://dev.mysql.com> kostenlos zum Download (Rubrik `MORE DOWNLOADS|MYSQL GUI TOOLS`). Zur Installation reicht ein Doppelklick auf die `*.msi`-Datei aus. Die Programme können anschließend mit `START|PROGRAMME|MYSQL|MYSQL ADMINISTRATOR` bzw. `|MYSQL QUERY BROWSER` gestartet werden.

Tipps zur Verwendung des MySQL-Kommandozeileninterpreters sowie diverser offizieller MySQL-Administrationswerkzeuge folgen in Abschnitt 1.8. Ein weiteres Administrationsprogramm, nämlich phpMyAdmin, wird in Kapitel 7 vorgestellt. Sein größter Vorteil gegenüber den hier erwähnten Programmen besteht darin, dass es über einen Webbrowser bedient wird und sich daher auch zur Administration des MySQL-Servers eignet, der auf einem anderen Rechner läuft (z.B. beim Webhoster).

PHP 5.3 installieren

Dieser Abschnitt beschreibt die automatische Installation von PHP 5.3 mit einer MSI-Datei. Die Installationsdateien zu PHP finden Sie auf den folgenden Websites:

http://windows.php.net/download/	(offizielle Versionen)
http://snaps.php.net/	(tagesaktuelle Updates, Testversionen)

Auf den Download-Seiten stehen unzählige PHP-Varianten zur Auswahl. Damit PHP im Zusammenspiel mit Apache zuverlässig funktioniert, müssen Sie sich für eine Variante `VC6 x86 THREAD SAFE` entscheiden (nicht `VC9!`) und dem Link `INSTALLER` folgen, damit Sie die `*.msi`-Datei bekommen. Speichern Sie die Datei an einem Ort, wo Sie sie wiederfinden – Sie brauchen die MSI-Datei möglicherweise später nochmals, um die Konfiguration zu verändern!

Das Installationsprogramm zeigt zuerst die PHP-Lizenzbedingungen an. Nach deren Bestätigung müssen Sie angeben, wohin PHP installiert werden soll (unser Vorschlag: C:\php53). Im nächsten Dialog wählen Sie aus, wie PHP installiert werden soll. Wenn Sie, wie in diesem Buch beschrieben, Apache 2.2 als Webserver verwenden, ist APACHE 2.2 MODULE die richtige Option. Nun möchte das Installationsprogramm noch wissen, in welchem Verzeichnis sich die Apache-Konfigurationsdatei httpd.conf befindet (C:\apache22\conf, wenn Sie unseren Empfehlungen gefolgt sind).

Der letzte und vielleicht schwierigste Punkt betrifft die PHP-Erweiterungen, die installiert werden sollen (siehe Abbildung 1.4). Für die meisten Beispiele dieses Buchs sind die folgenden Erweiterungen ausreichend: bzip2, EXIF, GD2, Mcrypt, Multi-Byte String, MySQL, MySQLi, PDO und PDO/MySQL. Mit der Ausnahme von Multi-Byte String sind diese Erweiterungen alle bereits standardmäßig ausgewählt.

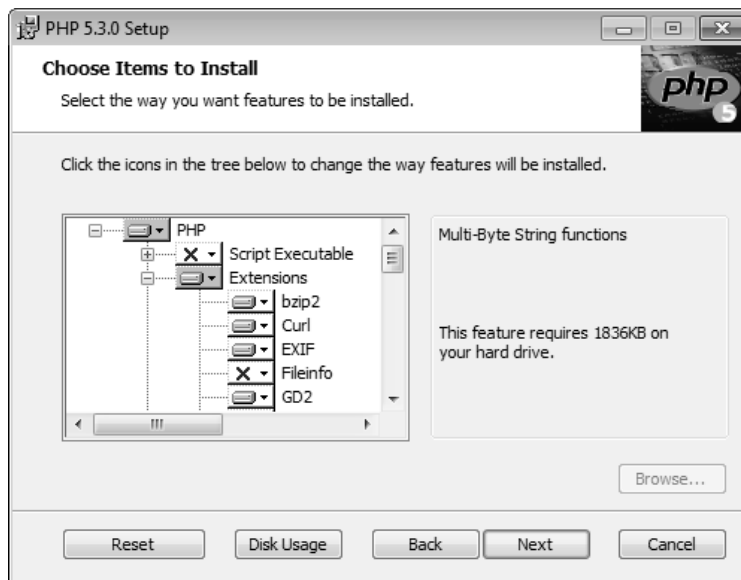


Abbildung 1.4: PHP-Installation

Widerstehen Sie der Versuchung, einfach alle Erweiterungsmodule zu installieren! Viele Module verweisen auf externe Programme bzw. Bibliotheken und funktionieren nur, wenn auch diese externen Komponenten installiert sind. Hier gilt: Weniger ist mehr!

Selbstverständlich können Sie später zusätzliche Erweiterungen installieren. Der naheliegende Weg, das PHP-Setup-Programm einfach über das Systemsteuerungsmodul PROGRAMM DEINSTALLIEREN ODER ÄNDERN zu starten, schlägt allerdings fehl: Das Installationsprogramm wird so ohne ausreichende Zugriffsrechte gestartet.

Die richtige Vorgehensweise sieht so aus: Sie suchen im Windows-Startmenü das Programm ZUBEHÖR|EINGABEAUFFORDERUNG, klicken es mit der rechten Maustaste an und wählen

ALS ADMINISTRATOR AUSFÜHREN. Im Eingabefenster starten Sie nun nochmals `php-5.n-win32-xxx.msi`, wobei Sie den vollständigen Pfad zur MSI-Datei angeben müssen. Im Setup-Dialog des Installationsassistenten wählen Sie die Variante `CHANGE`, bestätigen in den folgenden zwei Dialogen die Apache-Version und das Verzeichnis von `httpd.conf` und wählen schließlich die gewünschten Module aus, die zusätzlich installiert werden sollen.

PHP testen

Damit Apache das PHP-Modul erkennt, müssen Sie Apache neu starten. Dazu öffnen Sie am einfachsten über das Apache-Icon in der Taskbar den Apache Service Monitor und klicken auf den Button `RESTART`. Alternativ können Sie auch in der Windows-Systemsteuerung nach dem Modul `VERWALTUNGSDIENSTE` suchen und Apache dort neu starten.



Abbildung 1.5: Der Beginn der umfassenden PHP-Testseite

PHP ist nun als Erweiterung zu Apache installiert und wird zusammen mit Apache gestartet. Um zu testen, ob die Installation funktioniert hat, erstellen Sie im Verzeichnis `C:\apache22\htdocs` mit einem Editor die Datei `test.php` mit dem folgenden Inhalt:

```
<?php phpinfo(); ?>
```

Anschließend öffnen Sie mit einem Webbrowser die folgende Seite:

```
http://localhost/test.php
```

Das Ergebnis sollte wie in Abbildung 1.5 aussehen. Entscheidend ist insbesondere, dass die Testseite Abschnitte zu den Erweiterungen `mysql`, `mysqli` und `pdo_mysql` enthält (siehe auch die Abbildungen 11.1, 11.3 und 11.4 ab Seite 409).

Wenn es Probleme gibt ...

Die wahrscheinlichste Fehlerursache sind Tippfehler in den Konfigurationsdateien. Die Protokolldateien des Apache-Webservers verraten die Fehlerursache leider nur selten, aber natürlich empfiehlt es sich dennoch, als Erstes einen Blick in die Datei `C:\apache22\logs\error.log` zu werfen.

Wenn der Webbrowser die Datei `http://localhost/test.php` nicht anzeigt, sondern stattdessen einen Download-Dialog präsentiert, fehlt zumeist die Zeile `AddType application/x-httpd-php .php` in `httpd.conf` (oder sie enthält einen Tippfehler).

Wenn Sie den Verdacht haben, dass PHP die Einstellungen in `php.ini` einfach ignoriert, sollten Sie sicherstellen, dass PHP diese Datei überhaupt lädt. Von welchem Ort PHP die Konfigurationsdatei lädt, verrät die Zeile *Loaded Configuration File* der PHP-Testseite (neunte Zeile in Abbildung 1.5). Stellen Sie gegebenenfalls sicher, dass `httpd.conf` die Anweisung `PHPIniDir "C:/php53"` enthält.

Denken Sie daran, dass Änderungen in `httpd.conf` und `php.ini` erst wirksam werden, wenn Sie Apache neu starten! Achten Sie auch auf die korrekte Angabe von Verzeichnissen. In `php.ini` dient wie unter Windows \ als Trennzeichen zwischen Verzeichnissen, in `httpd.conf` hingegen wie unter Unix/Linux das Zeichen `!`.

Sobald einmal Probleme auftreten, kann Ihnen auch Ihr Webbrowser einen Streich spielen. Jeder Webbrowser verwaltet einen Pufferspeicher (Cache), in dem er die zuletzt geladenen Seiten speichert. Nach einer Fehlerkorrektur in `httpd.conf` bzw. `php.ini` kann es sein, dass der Webbrowser weiterhin die alte Seite anzeigt. Abhilfe: Löschen Sie den Cache des Webbrowsers (in Firefox im Dialogblatt EXTRAS|EINSTELLUNGEN|DATENSCHUTZ, Link KÜRZLICH ANGELEGTE CHRONIK, Button JETZT LÖSCHEN).

1.3 Paketinstallation unter Linux

Grundsätzlich ist die Installation von Apache, MySQL und PHP unter Linux ein Kinderspiel. Alle Distributionen liefern zu diesem Zweck fertige Pakete mit, die Sie mit dem Paketmanager der Distribution installieren. Fertig! Die folgenden Abschnitte fassen exemplarisch die Installation unter Fedora 11, openSUSE 11.1 und Ubuntu 9.04 zusammen.

Die einfache Installation hat freilich einen Nachteil: Sie bekommen auf diese Weise nur die Versionen, die mit der jeweiligen Distribution mitgeliefert werden – und die sind oft recht alt. Die momentan verfügbaren Distributionen (Stand August 2009) setzen auf Apache 2.2, MySQL 5.0 oder bestenfalls MySQL 5.1 und PHP 5.2. Wenn Sie nicht auf die nächste Version Ihrer Lieblingsdistribution warten möchten, gibt es zwei Wege: Entweder greifen Sie auf

XAMPP zurück oder Sie installieren bzw. kompilieren einzelne Komponenten manuell (siehe die beiden folgenden Abschnitte).

Installation unter Fedora 11

Wenn Sie mit Fedora arbeiten, installieren Sie mit `yum install` die folgenden Pakete:

- » `httpd`
- » `php, php-exif, php-gd, php-mbstring, php-mcrypt, php-mysql, php-pdo`
- » `mysql-server`

Tatsächlich werden wesentlich mehr Pakete installiert, weil sich `yum` darum kümmert, alle anderen erforderlichen Komponenten und Bibliotheken zu installieren. Bei Bedarf bieten sich auch noch die folgenden Pakete an:

- » `mysql-administrator, mysql-query-browser, phpMyAdmin`

Weder Apache noch MySQL werden gestartet. Dazu müssen Sie die folgenden vier Kommandos ausführen. Die ersten zwei Kommandos sind für den unmittelbaren Start zuständig, die anderen beiden Kommandos dafür, dass der Start in Zukunft automatisch erfolgt:

```
root# /etc/init.d/httpd start
root# /etc/init.d/mysqld start
root# chkconfig --level 35 httpd on
root# chkconfig --level 35 mysql on
```

Bei Fedora läuft Apache automatisch in dem für PHP geeigneten `prefork`-Modus (Konfiguration durch die Datei `/etc/sysconfig/httpd`). Der Administrator-Login für MySQL ist vorerst noch nicht abgesichert. Die erforderliche Vorgehensweise ist in den Abschnitten 1.8 und 7.2 beschrieben. Tabelle 1.1 fasst die wichtigen Eckdaten des LAMP-Systems unter Fedora zusammen.

ECKDATEN	ORT BZW. ACCOUNT
PHP-Konfiguration	<code>/etc/php.ini</code>
Apache-Konfiguration	<code>/etc/httpd/*</code>
Apache-DocumentRoot-Verzeichnis	<code>/srv/www/html/</code>
Apache-Account	<code>apache</code>
Apache-Init-Script (Start/Stop)	<code>/etc/init.d/httpd</code>

Tabelle 1.1: Eckdaten der Fedora-Installation

Installation unter openSUSE 11.1

Unter openSUSE installieren Sie am einfachsten mit zypper die komplette LAMP-Umgebung:

```
root# zypper install -t pattern lamp_server
```

Alternativ können Sie natürlich auch mit zypper install die folgenden Einzelpakete installieren, wobei aufgrund der automatischen Auflösung der Paketabhängigkeiten diverse weitere Pakete hinzukommen:

- » apache2-prefork
- » apache2-mod_php5, php5-exif, php5-gd, php5-mysql, php5-mbstring
- » mysql, mysql-client

Bei Bedarf bieten sich auch noch die folgenden Pakete an:

- » mysql-administrator, mysql-query-browser, mysql-workbench, phpmyadmin

Weder Apache noch MySQL werden gestartet. Dazu müssen Sie die folgenden vier Kommandos ausführen. Die ersten zwei Kommandos sind für den unmittelbaren Start zuständig, die anderen beiden Kommandos dafür, dass der Start in Zukunft automatisch erfolgt:

```
root# rcapache2 start
root# rcmysql start
root# insserv apache2
root# insserv mysql
```

Der Administrator-Login für MySQL ist vorerst noch nicht abgesichert. Die erforderliche Vorgehensweise ist in den Abschnitten 1.8 und 7.2 beschrieben. Tabelle 1.2 fasst die wichtigsten Eckdaten des LAMP-Systems unter openSUSE zusammen.

ECKDATEN	ORT BZW. ACCOUNT
PHP-Konfiguration	/etc/php5/apache2/php.ini
Apache-Konfiguration	/etc/apache2/httpd.conf
Apache-DocumentRoot-Verzeichnis	/srv/www/htdocs/
Apache-Account	wwwrun
Apache-Init-Script (Start/Stop)	/etc/init.d/apache2

Tabelle 1.2: Eckdaten der openSUSE-Installation

Installation unter Ubuntu 9.04

Wenn Sie mit Ubuntu arbeiten, installieren Sie mit `apt-get` oder Synaptic die folgenden Pakete. Aufgrund von Paketabhängigkeiten kommen zahllose weitere Pakete hinzu.

- » `apache2-mpm-prefork`
- » `php5`, `php5-gd`, `php5-mcrypt`, `php5-mysql`
- » `mysql-server-5.1`

Beachten Sie, dass Ubuntu 9.04 standardmäßig auf MySQL 5.0 setzt. MySQL 5.1 steht nur im Rahmen der *universe*-Pakete zur Verfügung, für die es keinen offiziellen Support gibt. Erst Ubuntu 9.10 wird MySQL 5.1 voll unterstützen.

Bei Bedarf bieten sich auch noch die folgenden Pakete an:

- » `libzend-framework-php`, `mysql-admin`, `mysql-query-browser`, `phpmyadmin`

Während der Installation müssen Sie das `root`-Passwort für den MySQL-Server angeben. Damit ist der Administrator-Login des MySQL-Servers von Anfang an abgesichert. Bei der Installation von phpMyAdmin müssen Sie zudem angeben, für welchen Webserver das Programm konfiguriert werden soll. Die richtige Option ist `APACHE2`.

Alle Server-Dienste werden sofort gestartet. Es kann aber passieren, dass PHP erst nach Apache installiert wird. Der Webserver erfährt in diesem Fall nicht, dass PHP zur Verfügung steht. Abhilfe: Starten Sie den Webserver neu, bevor Sie PHP ausprobieren:

```
user$ sudo /etc/init.d/apache2 restart
```

Tabelle 1.3 fasst die wichtigen Eckdaten des LAMP-Systems unter Ubuntu zusammen.

ECKDATEN	ORT BZW. ACCOUNT
PHP-Konfiguration	<code>/etc/php5/apache2/php.ini</code>
Apache-Konfiguration	<code>/etc/apache2/httpd.conf</code>
Apache-DocumentRoot-Verzeichnis	<code>/srv/www/htdocs/</code>
Apache-Account	<code>wwwrun</code>
Apache-Init-Script (Start/Stopp)	<code>/etc/init.d/apache2</code>

Tabelle 1.3: Eckdaten der Ubuntu-Installation

1.4 XAMPP-Installation unter Linux

XAMPP ist ein Komplettpaket, das aus Apache, MySQL, PHP, Perl, phpMyAdmin und diversen anderen Komponenten besteht. Die in XAMPP inkludierten Programmversionen sind in der Regel aktueller als die Pakete, die mit Linux-Distributionen mitgeliefert werden. Allerdings ist XAMPP explizit nur für den Entwicklereinsatz konzipiert, nicht für den richtigen Server-Einsatz. XAMPP für Linux besteht aus einem einzigen, sehr großen tar-Archiv, das Sie hier zum Download finden:

<http://www.apachefriends.org/de/xampp-linux.html>

Bevor Sie XAMPP installieren, müssen Sie sicherstellen, dass auf Ihrem Rechner weder Apache noch MySQL installiert ist! Die Installation ist wirklich denkbar einfach – ein einziges tar-Kommando reicht:

```
root# tar xvfvz xampp-linux-n.n.tar.gz -C /opt
```

Anschließend befinden sich sämtliche XAMPP-Komponenten im Verzeichnis `/opt/lampp`. Ein weiteres Kommando startet das System:

```
root# /opt/lampp/lampp start
```

Indem Sie die Seite <http://localhost> mit einem Webbrowser öffnen, können Sie sich davon überzeugen, dass alles funktioniert hat (siehe Abbildung 1.6). Tabelle 1.4 fasst zusammen, wo Sie die Konfigurationsdateien finden.

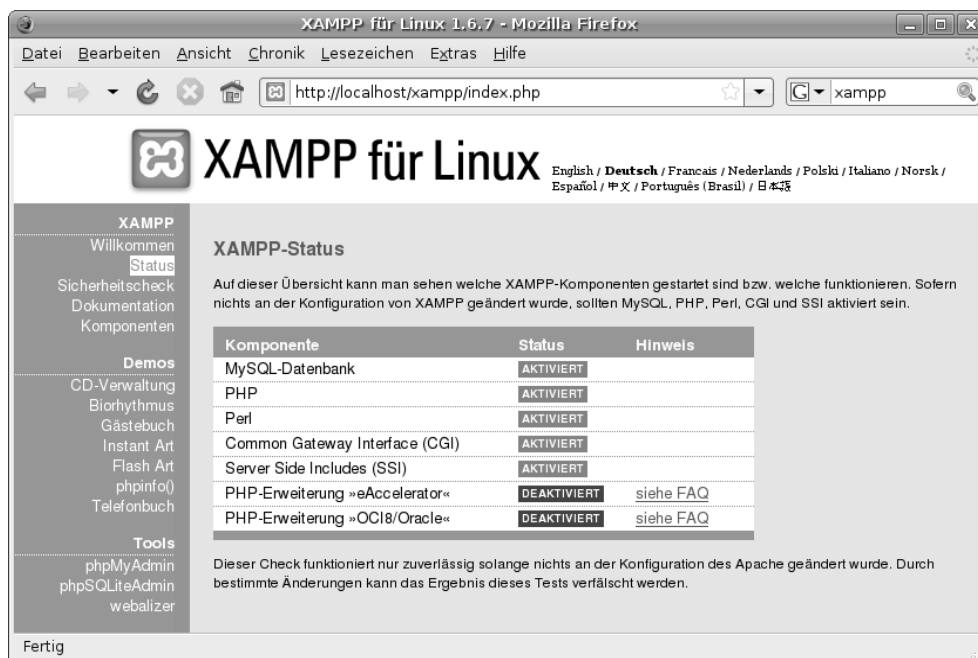


Abbildung 1.6: Die XAMPP-Statusseite

ECKDATEN	ORT BZW. ACCOUNT
PHP-Konfiguration	/opt/lampp/etc/php.ini
Apache-Konfiguration	/opt/lampp/etc/httpd.conf
Apache-DocumentRoot-Verzeichnis	/opt/lampp/htdocs
Apache-Account	nobody
Apache-Init-Script (Start/Stop)	/opt/lampp/lampp

Tabelle 1.4: Eckdaten der XAMPP-Installation

XAMPP absichern

Die Standardinstallation von XAMPP ist nicht durch Passwörter abgesichert und daher unsicher. Das sollten Sie mit dem folgenden Kommando ändern:

```
root# /opt/lampp/lampp security
```

Sie können nun verschiedene Aspekte von XAMPP absichern:

- » **XAMPP-Seiten:** Damit sichern Sie den Zugang zu den von Apache erzeugten Webseiten zur XAMPP-Administration ab. Unabhängig davon, ob die Seiten vom lokalen Rechner oder extern aufgerufen werden, müssen Sie sich nun zuerst mit dem Benutzernamen *lampp* und dem von Ihnen angegebenen Passwort anmelden. (Die Absicherung erfolgt durch zwei `.htaccess`-Dateien in den Verzeichnissen `/opt/lampp/xampp` und `/opt/lampp/phpmyadmin`. Hintergrundinformationen zu `.htaccess` finden Sie in Abschnitt 1.6.)
- » **MySQL-Netzwerkzugriff:** Wenn Sie ausschließlich mit PHP auf den MySQL-Server zugreifen, erfolgt dieser Zugriff über eine sogenannte Socket-Datei. Die Netzwerkfunktionen von MySQL sind daher nicht erforderlich und sollten aus Sicherheitsgründen deaktiviert werden.
- » **phpMyAdmin-Passwort:** Zur Administration des MySQL-Servers hat XAMPP phpMyAdmin installiert (siehe Kapitel 7). Für einige Zusatzfunktionen von phpMyAdmin hat XAMPP einen eigenen MySQL-Benutzer eingerichtet, für den Sie nun ein Passwort definieren können. Dieses Passwort wird im Klartext in die Datei `/opt/lampp/phpmyadmin/config.inc.php` eingetragen (sonst funktioniert phpMyAdmin nicht). Verwenden Sie aus Sicherheitsgründen nicht dasselbe Passwort, mit dem beispielsweise Ihr Login abgesichert ist!
- » **MySQL-root-Passwort:** Die MySQL-Administration erfolgt durch den Benutzer `root`. Sie können nun für `root` ein Passwort angeben. Dieses Passwort gilt auch für phpMyAdmin. Dort müssen Sie sich in Zukunft mit dem Benutzernamen `root` und dem dazu passenden Passwort anmelden.

- » **FTP-Passwort:** Als Teil von XAMPP wurde auch ein FTP-Server eingerichtet. Dieser ist mit dem Passwort `lampp` abgesichert. Wenn Sie ein anderes Passwort benutzen möchten, können Sie es jetzt angeben.

XAMPP verwenden

Nun können Sie mit der Entwicklung Ihrer ersten PHP-Testprogramme beginnen. Ihre eigenen PHP-Dateien speichern Sie beispielsweise in `/opt/lampp/htdocs/meinebeispiele`. Dabei müssen Sie darauf achten, dass die Dateien von Apache gelesen werden können, der unter dem Account `nobody` ausgeführt wird.

```
root# mkdir /opt/lampp/htdocs/meinebeispiele
root# chown benutzername /opt/lampp/htdocs/meinebeispiele
root# chgrp nobody /opt/lampp/htdocs/meinebeispiele
```

Nach diesen Vorbereitungsarbeiten sollten Sie aus Sicherheitsgründen unter Ihrem normalen Account weiterarbeiten.

XAMPP beenden

Bevor Sie den Rechner herunterfahren, sollten Sie XAMPP stoppen. Das ist insbesondere deswegen wichtig, damit der MySQL-Server alle offenen Datenbankdateien ordnungsgemäß schließen kann.

```
root# /opt/lampp/lampp stop
```

XAMPP automatisch starten und stoppen

Jedes Mal, wenn Sie Ihren Rechner starten, müssen Sie auch XAMPP neu starten. Beim Ausschalten müssen Sie daran denken, XAMPP zu beenden. Wenn Sie dazu keine Lust haben, können Sie das auch automatisch erledigen. Dazu müssen Sie in das Init-V-System Ihrer Linux-Distribution zwei Links einfügen. (Das Init-V-System steuert, wann welche Systemdienste gestartet bzw. gestoppt werden.)

Die folgenden Kommandos zeigen die Vorgehensweise für Ubuntu. Bei anderen Distributionen müssen Sie teilweise abweichende Verzeichnisnamen verwenden (z.B. bei Fedora `/etc/rc5.d`, bei openSUSE `/etc/init.d/rc5.d`).

```
root# cd /etc/rc2.d
root# ln -s /opt/lampp/lampp S99lampp
root# ln -s /opt/lampp/lampp K01lampp
```

XAMPP deinstallieren

Die Deinstallation ist ebenso einfach wie die Installation:

```
root# rm -rf /opt/lampp/
```

Beachten Sie, dass Sie dadurch auch alle Ihre PHP-Dateien, MySQL-Datenbanken etc. verlieren. Gegebenenfalls sollten Sie vorher ein Backup machen. Die PHP-Dateien kopieren Sie einfach von `/opt/lampp/htdocs/beispiele/` in ein anderes Verzeichnis (z.B. in Ihr Heimatverzeichnis). MySQL-Datenbanken sichern Sie am bequemsten mit phpMyAdmin (siehe Abschnitt 7.6). Alternativ können Sie auch das Kommando `mysqldump` zu Hilfe nehmen. Das folgende Beispielkommando wurde hier aus Platzgründen auf drei Zeilen verteilt:

```
user$ cp -a /opt/lampp/htdocs/meinebeispiele ~
user$ /opt/lampp/binmysqldump -u root -p \
    meinedatenbank > ~/meinedatenbank.sql
Password: *****
```

1.5 Manuelle Installation unter Linux

Wenn Ihnen auch XAMPP nicht ausreichend aktuell ist, müssen Sie selbst Hand anlegen. Dieser Abschnitt zeigt exemplarisch, wie Sie unter Ubuntu 8.04 eine aktuelle MySQL-Version aus dem von `dev.mysql.com` verfügbaren `*.tar.gz`-Paket installieren und PHP 5.3 selbst kompilieren. Diese Anleitung fasst gleichzeitig zusammen, wie wir die auf der beiliegenden DVD befindliche Testumgebung zusammengefügt haben.

Wenn Sie andere Distributionen einsetzen oder andere Versionen ausprobieren möchten (z.B. eine Snapshot-Version von PHP 6), müssen Sie die Vorgehensweise entsprechend adaptieren. Generell richtet sich dieser Abschnitt an Linux-Profis: Wer noch nie ein Programm kompiliert hat, wird vermutlich auch mit dieser Kurzanleitung scheitern.

Apache 2.2

Keine Probleme bereitet momentan Apache. Alle Distributionen enthalten die aktuelle Version 2.2. Unter Ubuntu bzw. Debian installieren Sie Apache mit dem folgenden Kommando:

```
root# apt-get install apache2-mpm-prefork
```

MySQL 5.4

Da MySQL 5.4 zum Zeitpunkt der Fertigstellung dieses Buchs noch in der Entwicklung war, gab es noch keine fertigen Debian- oder Ubuntu-Pakete: `mysql.com` stellt derartige Pakete generell nicht zur Verfügung (dort bevorzugt man das RPM-Format), und auch die sonst sehr empfehlenswerte Paketquelle `http://www.dotdeb.org/` konnte nicht weiterhelfen. Daher griffen wir auf ein vorkompiliertes `.tar.gz`-Paket von `http://dev.mysql.com/downloads/mysql/5.4.html` zurück. Dieses Paket muss im Verzeichnis `/usr/local` ausgepackt und eingerichtet werden:

```
root# cd /usr/local
root# tar xzf mysql-version.tar.gz
root# ln -s mysql-5.4.n-name mysql
```

```

root# groupadd mysql
root# useradd -g mysql mysql
root# cd mysql
root# chown -R mysql:mysql .
root# scripts/mysql_install_db --user=mysql
root# chown -R root .
root# chown -R mysql data

```

Die Binärdateien der MySQL-Kommandos `mysql`, `mysqldump` etc. befinden sich nun in `/usr/local/mysql/bin`. Damit dieser Pfad nicht jedes Mal angegeben werden muss, haben wir die Datei `/etc/profile.d/mysqlpath.sh` angelegt und dort die `PATH`-Variable erweitert. Beachten Sie, dass diese Einstellungen erst nach einem neuerlichen Login wirksam werden und nur für gewöhnliche Benutzer gelten, nicht aber für `root`!

```

# Datei /etc/profile.d/mysqlpath.sh
PATH=$PATH:/usr/local/mysql/bin

```

Nach diesen Vorbereitungsarbeiten verwenden Sie die mitgelieferten Muster für die Konfigurationsdateien `/etc/my.cnf` und `/etc/init.d/mysql`. Die Datei `my.cnf` müssen Sie anschließend Ihren eigenen Vorstellungen entsprechend anpassen. Insbesondere müssen Sie in `my.cnf` die Kommentarzeichen `#` bei diversen bereits vorgesehenen InnoDB-Zeilen entfernen und die Einstellungen anpassen, damit der MySQL-Server die InnoDB-Engine aktiviert.

```

root# cd /usr/local/mysql
root# cp support-files/mysql.server /etc/init.d/mysql
root# update-rc.d mysql start 90 2 3 4 5 . stop 10 0 1 6 .
root# cp /usr/local/mysql/support-files/my-small.cnf /etc/my.cnf
root# gedit /etc/my.cnf

```

Um zu vermeiden, dass der MySQL-Server auch die alten Konfigurationsdateien in `/etc/mysql` berücksichtigt, müssen Sie dieses Verzeichnis löschen oder umbenennen:

```

root# mv /etc/mysql /etc/mysql.bak

```

Nun können Sie versuchen, den MySQL-Server zu starten:

```

root# /etc/init.d/mysql start

```

Eventuelle Fehlermeldungen finden Sie in `/usr/local/mysql/data/hostname.err`. Weitere Informationen gibt die folgende, detaillierte Installationsanleitung:

<http://dev.mysql.com/doc/refman/5.4/en/installing-binary.html>

Bei der MySQL-Installation auf der virtuellen Maschine der beiliegenden DVD haben wir in `my.cnf` als Datenbankverzeichnis `/var/lib/mysql` eingestellt (und nicht wie vorgesehen `/usr/local/mysql/data`). Auf diese Weise haben wir die von einer früheren MySQL-5.1-Installation vorhandenen MySQL-Datenbanken übernommen. Zur Aktualisierung der internen `mysql.*`-Tabellen haben wir anschließend `mysql_fix_privilege_tables` ausgeführt.

Um Probleme mit Paketabhängigkeiten zu vermeiden, haben wir das Paket `libmysql-client16` installiert. Die darin enthaltene `libmysql`-Bibliothek ist zwar nicht ganz aktuell, aber kompatibel zu MySQL 5.4. Sie ermöglicht es diversen externen Programmen, mit dem MySQL-Server zu kommunizieren. PHP 5.3 nutzt diese Bibliothek übrigens *nicht*, sondern verwendet stattdessen die in PHP integrierte Bibliothek `mysqlnd`.

PHP 5.3

Am meisten Arbeit verursachte PHP 5.3. Im August 2008 gab es hierfür noch keine fertigen Ubuntu-kompatiblen Pakete. Daher haben wir PHP 5.3 selbst kompiliert, was auch keine Hexerei darstellt. Der erste Schritt besteht darin, die Entwicklungswerkzeuge und -bibliotheken zu installieren:

```
root# apt-get install build-essential autoconf apache-threaded-dev \
      libxml2-dev libjpeg62-dev libpng12-dev libtool
```

Den Quellcode für PHP 5.3 laden Sie wahlweise von <http://snaps.php.net/> oder von <http://php.net/downloads.php> herunter, packen ihn mit `tar xzf` aus und benennen das Verzeichnis dann in `php53-code` um. Die Anweisungen zum Kompilieren verpacken Sie am besten in ein kleines Script `comp-php`, beispielsweise so:

```
cd ~/php53-code
./buildconf
./configure \
  --with-apxs2=/usr/bin/apxs2 \
  --with-zlib \
  --with-gd \
  --with-mysql=mysqlnd \
  --with-mysqli=mysqlnd \
  --enable-pdo \
  --with-pdo-mysql=mysqlnd \
  --with-config-file-scan-dir=/etc/php5/apache2 \
  --with-jpeg-dir=/usr/lib \
  --enable-exif \
  --libdir=/usr/lib \
  --enable-mbstring
make clean
make
sudo make install
sudo /etc/init.d/apache2 restart
```

Für `./configure` existieren unzählige weitere Optionen, die Sie verwenden müssen, um diverse PHP-Erweiterungen einzusetzen. Das Kommando `/configure --help` liefert eine kurze Beschreibung aller Optionen. Beachten Sie, dass Sie häufig auch diverse Entwicklerbibliotheken installieren (unter Ubuntu Pakete, deren Namen mit `-dev` enden).

1.6 Apache-Konfiguration

Beinahe alle Einstellungen von Apache werden durch die Datei `httpd.conf` gesteuert: der Servername, die Orte von Webverzeichnissen und Logdateien, die Zugriffsrechte, die Integration von PHP etc. Tabelle 1.5 gibt an, wo sich diese Datei üblicherweise befindet.

INSTALLATION	ORT VON HTTPD.CONF
Windows-Eigeninstallation	<code>C:\apache22\conf\httpd.conf</code>
Linux XAMPP	<code>/opt/lampp/etc/httpd.conf</code>
Red Hat, Fedora	<code>/etc/httpd/conf/*</code>
Debian, SUSE, Ubuntu	<code>/etc/apache2/*</code>

Tabelle 1.5: Ort der Apache-Konfigurationsdatei

Bei den meisten Linux-Distributionen gibt es nicht eine Konfigurationsdatei `httpd.conf`, sondern gleich ein ganzes Bündel. Das macht jede einzelne Datei übersichtlicher und ermöglicht eine modulare Konfiguration. Gleichzeitig erschwert es aber den Überblick.

Bei openSUSE- bzw. Novell-Distributionen kommt hinzu, dass viele Apache-Konfigurationsdateien bei jedem Start neu erstellt werden. Eigene Änderungen sollten in `/etc/apache2/httpd.conf.local` durchgeführt werden. Damit diese Datei berücksichtigt wird, müssen Sie diesen Dateinamen in die Variable `APACHE_CONF_INCLUDE_FILES` in der Datei `/etc/sysconfig/apache2` einfügen.

Generell werden Änderungen an der Konfiguration erst wirksam, wenn Sie Apache neu starten! Unter Windows müssen Sie in allen Apache-Konfigurationsdateien das Zeichen `\` zur Trennung von Verzeichnissen verwenden, nicht wie sonst üblich `/`!

Elementare Einstellungen

Es fehlt hier der Platz, um auf die Details der Apache-Konfiguration einzugehen. Im Folgenden werden daher nur einige ganz elementare Einstellungen aus `httpd.conf` beschrieben.

- » **ServerName:** Diese Einstellung gibt den Netzwerknamen des Rechners an, auf dem Apache läuft. Unter diesem Namen ist der Webserver von anderen Rechnern im Netzwerk aus erreichbar. (Wenn die Einstellung fehlt, versucht Apache, den Netzwerknamen selbst herauszufinden.)
- » **DocumentRoot:** Diese Einstellung gibt das Verzeichnis an, in dem sich die Webdateien (`*.html`, `*.php` etc.) befinden. Tabelle 1.6 fasst den Defaultzustand je nach Installation zusammen.
- » **LoadModule:** Wenn Apache so kompiliert wurde, dass es zur Laufzeit Module laden kann (das ist meistens der Fall), sind hierfür `LoadModule`-Anweisungen erforderlich, die beispielsweise so aussehen:

```
LoadModule php5_module /usr/lib/apache2/libphp5.so    # für Linux
LoadModule php5_module "c:/php53/php5apache2_2.dll"  # für Windows
```

- » **AddType:** AddType-Anweisungen stellen die Zuordnung zwischen Dateikennungen und dem Applikationstyp her. Dank der folgenden Einstellung weiß Apache, dass *.php vom PHP-Interpreter ausgeführt werden soll:

```
AddType application/x-httpd-php .php
```

- » **PHPIniDir:** Diese Einstellung wird an den PHP-Interpreter übergeben und sagt diesem, von welchem Ort er die php.ini-Datei lesen soll. Die Option soll die Konfiguration von PHP erleichtern, die in der Vergangenheit oft an einem ganz trivialen Punkt scheiterte: PHP fand die php.ini-Datei nicht bzw. wertete eine veraltete oder falsche Version dieser Datei aus. PHPIniDir steht ab Apache 2 zur Verfügung und wird nur von der Windows-Version von Apache ausgewertet.

```
PHPIniDir "C:/php53"
```

- » **AddDefaultCharset:** Diese Einstellung gibt an, welchen Zeichensatz die Webdateien haben. Details zur Zeichensatzfrage folgen in Abschnitt 1.9.

- » **<Directory "xxx">:** Dieser aus mehreren Zeilen bestehende Block beschreibt alle Eigenschaften eines Webverzeichnisses. <Directory /> enthält die Defaulteinstellungen, die unter anderem auch für DocumentRoot gelten. Der <Directory>-Block endet mit </Directory>. Die folgenden Zeilen geben ein Beispiel:

```
<Directory /> oder <Directory "verzeichnis ...">
...
    AllowOverride AuthConfig FileInfo
    Option Indexes
</Directory>
```

Innerhalb des <Directory>-Blocks steuern diverse Optionen, was in diesem Verzeichnis erlaubt bzw. verboten ist und wer darauf Zugriff hat. An dieser Stelle werden nur zwei dieser Optionen vorgestellt:

AllowOverride: Oft ist es erwünscht, dass Apache in Webverzeichnissen die Datei .htaccess auswertet und die darin enthaltenen Zusatzeinstellungen auf das Verzeichnis anwendet (siehe auch den nächsten Abschnitt). Damit das funktioniert, müssen Sie mit AllowOverride angeben, welche Veränderungen zulässig sind. Mögliche Einstellungen sind None (verbietet alle Änderungen, gilt oft per Default), All (erlaubt alle Änderungen) oder eine Kombination aus AuthConfig (Authentifizierung), FileInfo (Sprach- und Zeichensatzeinstellungen, Fehlerdatei), Indexes, Limit (Zugriffsrechte) und Options.

Options Indexes: Wenn im Browser die Adresse `http://rechnername/verzeichnis/` angegeben wird, liefert Apache je nach Konfiguration von `DirectoryIndex` automatisch die Seite `index.html`, `index.php` etc. (siehe unten). Sollte keine der `DirectoryIndex`-Dateien existieren, bewirkt `Options Indexes`, dass Apache ein Inhaltsverzeichnis anzeigt. Das ist während der Programmentwicklung oft praktisch, kann aber im Vollbetrieb ein Sicherheitsrisiko darstellen. (Neben `Indexes` gibt es übrigens noch eine Reihe weiterer Schlüsselwörter, die `Options` nachgestellt werden dürfen.)

- » **DirectoryIndex:** Mit dieser Option können Sie eine oder mehrere Dateien angeben, nach denen Apache sucht, wenn im Webbrowser nur ein Verzeichnis angegeben wird. Eine übliche Einstellung lautet `index.html index.php main.php default.php`. Wenn im Browser nun die Adresse `http://rechnername/verzeichnis/` eingegeben wird, sucht Apache für dieses Verzeichnis nach den angegebenen Dateien und liefert die erste passende Datei.

Defaultverzeichnis für Webdateien

Tabelle 1.6 fasst zusammen, in welchem Verzeichnis Webdateien (*.html, *.php etc.) standardmäßig gespeichert werden. In diesem Verzeichnis bzw. in selbst erzeugten Unterverzeichnissen speichern Sie Ihre eigenen PHP-Beispiele.

INSTALLATION	DOCUMENTROOT-ORT
Windows	C:\apache22\htdocs
Linux XAMPP	/opt/lampp/htdocs
Debian, Ubuntu	/var/www
Fedora, Red Hat	/var/www/html
SUSE	/srv/www/htdocs

Tabelle 1.6: Verzeichnis mit den Webdateien

Beachten Sie, dass Apache unter Linux unter einem eigenen Account läuft (z.B. `nobody`, `wwwrun` oder `www-data`). Bei allen Webdateien müssen die Zugriffsrechte so eingestellt werden, dass dieser Benutzer und somit auch Apache sie lesen kann. Zur Veränderung der Zugriffsrechte verwenden Sie die Unix-Kommandos `chown` und `chmod`. Beispielsweise bewirkt `chmod a+x *`, dass jeder alle Dateien lesen darf.

Passwortschutz für Webverzeichnisse (.htaccess-Datei)

Sobald Apache läuft, hat jeder im Netz bzw. im Internet Zugang zu allen Webseiten, die Apache verarbeiten kann. Gerade auf Entwicklungsrechnern ist das nicht immer gewollt. Es bestehen drei Möglichkeiten, den Zugriff auf Webseiten einzuschränken:

- » **Absicherung in httpd.conf:** In httpd.conf können Sie ganz genau einstellen, von welchen Adressen die Webseiten eines bestimmten Verzeichnisses gelesen werden dürfen. Es fehlt hier allerdings der Platz, um diese Mechanismen zu beschreiben.
- » **Absicherung durch eine Firewall:** Wenn auf dem Entwicklungsrechner eine Firewall läuft, kann diese so konfiguriert werden, dass Apache nicht von außen (also über das Netzwerk) kontaktiert wird. Die Webseiten können also nur vom lokalen Rechner betrachtet werden.
- » **Passwortabsicherung einzelner Verzeichnisse:** Hierfür legen Sie im zu schützenden Webverzeichnis eine Datei mit dem Namen .htaccess an und geben dort an, dass dieses Verzeichnis erst nach einer vorherigen Authentifizierung durch Name und Passwort gelesen werden darf. Apache wertet .htaccess aus und zeigt allen Surfern, die Dateien aus diesem Verzeichnis mit ihrem Browser ansehen möchten, einen Login-Dialog.

Im Folgenden wird nur die letzte Variante behandelt. Sie ist auch dann hilfreich, wenn Sie ein im Internet öffentlich zugängliches Verzeichnis gegen fremde Benutzung absichern möchten. Ein typisches Beispiel hierfür ist das PHP-Programm phpMyAdmin, das auf unzähligen Websites zur Administration von MySQL verwendet wird (siehe Kapitel 7).

Passwortdatei anlegen

Der erste Schritt zum Passwortschutz besteht darin, eine Passwortdatei anzulegen. Die Passwortdaten werden aus Sicherheitsgründen nicht in .htaccess gespeichert, sondern in einer separaten Datei. Diese Datei darf sich auch außerhalb der öffentlich zugänglichen Webverzeichnisse befinden. Die Passwortdatei wird im folgenden Beispiel site.pwd genannt.

Die Passwortdatei wird mit dem Apache-Hilfsprogramm htpasswd erzeugt. Dabei geben Sie die Option -c (create), den Namen der Passwortdatei und einen Benutzernamen an. Das Programm fragt Sie anschließend nach dem gewünschten Passwort und erzeugt dann die Passwortdatei, wobei der Benutzername im Klartext, das Passwort aber verschlüsselt gespeichert wird:

```
> htpasswd -c site.pwd michael
New password: *****
Re-type new password: *****
Adding password for user michael
```

Um einer schon existierenden Passwortdatei eine weitere Kombination aus Benutzername und Passwort hinzuzufügen, führen Sie htpasswd nochmals ohne die Option -c aus:

```
> htpasswd site.pwd name2
New password: *****
Re-type new password: *****
Adding password for user name2
```

Die resultierende Passwortdatei sieht so ähnlich wie die folgenden Zeilen aus:

```
michael:$apr1$gp1.....$q1jDszVJ0SCS.oBoPJItSj
name2:$apr1$A22.....$0V01Nc1FcXgNsruT9c6Iq1
```

Um `htpasswd.exe` unter Windows auszuführen, müssen Sie ein Kommandofenster (Eingabeaufforderung) öffnen und in Anführungszeichen den vollständigen Pfad des Programms angeben, also üblicherweise `"C:\apache22\bin\htpasswd.exe"`.

Wenn Sie die Passwortdatei in einem Verzeichnis bei Ihrem Internet Service Provider erstellen oder ergänzen möchten, benötigen Sie einen `ssh`-Zugang zum Rechner des ISPs (um dort `htpasswd` auszuführen). Manche ISPs stellen stattdessen andere Konfigurationshilfen zur Verfügung.

.htaccess-Datei

Wenn sich in einem Webverzeichnis eine Datei mit dem Namen `.htaccess` befindet, wertet Apache die in dieser Datei enthaltenen Konfigurationseinstellungen aus. Dabei gilt dieselbe Syntax wie in `httpd.conf`. Allerdings können nur einige verzeichnisspezifische Einstellungen vorgenommen werden. Mit der folgenden `.htaccess`-Datei wird der Inhalt eines Verzeichnisses (inklusive aller Unterverzeichnisse!) geschützt:

```
AuthType Basic
AuthUserFile "c:/apache22/htdocs/test/site.pwd"
AuthName "myrealm"
Require valid-user
```

`AuthUserFile` gibt den vollständigen Dateinamen der Passwortdatei an. `AuthName` bezeichnet den Bereich (`realm`), für den der Zugriff gültig ist. Der Sinn besteht darin, dass Sie nicht jedes Mal einen Login durchführen müssen, wenn Sie auf durch `.htaccess` geschützte Verzeichnisse zugreifen wollen. Sobald Sie sich mit einer bestimmten `AuthName`-Bezeichnung eingeloggt haben, gilt dieser Login auch für alle anderen Verzeichnisse mit dieser `AuthName`-Bezeichnung.

Für `.htaccess` bedeutet das, dass Sie für Verzeichnisse mit gemeinsamem Login jeweils dieselbe `AuthName`-Zeichenkette angeben. Umgekehrt können Sie für unterschiedliche Verzeichnisse auch unterschiedliche Bereichsnamen angeben – dann ist für jedes Verzeichnis ein neuer Login erforderlich.

`Require valid-user` bedeutet, dass als Login jede gültige Kombination aus Benutzername und Passwort zulässig ist. Alternativ können Sie hier auch angeben, dass ein Login nur für ganz bestimmte Benutzer erlaubt ist:

```
Require user myname name2 name3
```

Weitere Details zur Benutzerauthentifizierung und zur .htaccess-Konfiguration finden Sie z.B. hier:

<http://www.apacheweek.com/features/userauth>

.htaccess-Dateien werden nur wirksam, wenn dies in der Apache-Konfigurationsdatei httpd.conf erlaubt ist! Wenn die Apache-Konfiguration für das betroffene Verzeichnis bzw. für das Defaultverzeichnis / den Eintrag AllowOverride None enthält, bleiben alle .htaccess-Dateien wirkungslos! Wenn in AllowOverride nur einzelne Optionen angegeben werden, in .htaccess aber darüber hinausgehende Einstellungen verändert werden, wird sogar ein Fehler ausgelöst.

Damit .htaccess-Dateien berücksichtigt werden, muss httpd.conf für das betreffende Verzeichnis (also in der entsprechenden <Directory>-Gruppe) die Einstellung AllowOverride All oder AllowOverride AuthConfig enthalten.

Unter Windows ist es mit dem Windows-Explorer nicht möglich, eine Datei mit dem Namen .htaccess zu erzeugen: Der Explorer glaubt, dass dieser Name ungültig ist. Nennen Sie die Datei vorerst htaccess.txt, und verwenden Sie dann ein Eingabeaufforderungsfenster, um die Datei durch das Kommando `REN htaccess.txt .htaccess` umzubenennen. Windows-Anwender werden sich vielleicht wundern, warum der Dateiname überhaupt mit einem Punkt beginnen muss. Das ist die unter Unix/Linux übliche Methode, Dateien als versteckt zu kennzeichnen. Derartige Dateien werden von Linux-Dateimanagern normalerweise nicht angezeigt.

Zugriff auf ein geschütztes Verzeichnis

Wenn Sie versuchen, mit einem Browser eine Datei aus einem geschützten Verzeichnis zu lesen, zeigt der Webbrowser einen Login-Dialog an (Abbildung 1.7).

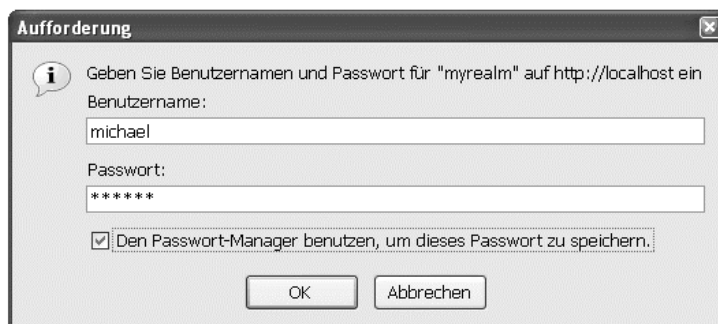


Abbildung 1.7: Zugriff auf ein passwortgeschütztes Verzeichnis

1.7 PHP-Konfiguration

Die Einstellungen und Optionen des PHP-Interpreters werden durch die Datei `php.ini` gesteuert. Tabelle 1.7 fasst zusammen, wo Sie diese Datei üblicherweise finden:

INSTALLATION	ORT VON PHP.INI
Windows-Eigeninstallation	<code>C:\php53\php.ini</code>
Linux XAMPP	<code>/opt/lampp/etc/php.ini</code>
Linux	<code>/etc/php5/apache2/php.ini</code>
Selbstkompiliertes PHP unter Linux	<code>/usr/lib/php.ini</code>

Tabelle 1.7: Ort der PHP-Konfigurationsdatei

Unter Windows wird der Ort von `php.ini` üblicherweise durch `PHPIniDir` in der Apache-Konfigurationsdatei `httpd.conf` eingestellt. Wenn diese Einstellung fehlt, sucht der PHP-Interpreter `php.ini` im Windows-Verzeichnis.

php.ini-Probleme

Wenn der PHP-Interpreter `php.ini` nicht findet bzw. wenn diese Datei gar nicht existiert, verwendet er Standardeinstellungen. Wenn Sie sich nicht sicher sind, ob bzw. von wo der PHP-Interpreter `php.ini` liest, werfen Sie einen Blick auf die PHP-Testseite, die durch die PHP-Funktion `phpinfo()` erzeugt wird (siehe Abbildung 1.5). Veränderungen in `php.ini` werden erst wirksam, wenn Apache neu gestartet wird!

Muster für `php.ini` finden Sie unter Windows im PHP-Installationsverzeichnis (z.B. `C:\php53\`). Als Ausgangsbasis für Ihre eigenen Einstellungen verwenden Sie am besten die mitgelieferte Datei `php.ini-recommended`.

Ab PHP 5.3 besteht die Möglichkeit, einen Teil der PHP-Einstellungen durch verzeichnispezifische Konfigurationsdateien mit dem Namen `.user.ini` zu verändern. Die Idee ist also dieselbe wie bei den `.htaccess`-Dateien für Apache. Diese Funktion ist deaktiviert, wenn die zentrale `php.ini`-Datei die folgende Zeile (ohne Einstellung!) enthält:

```
user_ini.filename =
```

Elementare Einstellungen

Es fehlt hier der Platz, um auf die unzähligen Optionen in `php.ini` einzugehen. Im Folgenden werden daher nur die allerwichtigsten Optionen beschrieben:

- » **display_errors=0n**: Diese Einstellung bewirkt, dass bei Fehlern in PHP-Scripts Fehlermeldungen angezeigt werden. Das ist bei der Programmentwicklung praktisch. Bei Servern, die im Internet stehen, sollte die Einstellung allerdings Off lauten, weil Fehlermeldungen manchmal Rückschlüsse auf den Code zulassen und so zu einer Sicherheitslücke werden können.

Wenn Sie beim Entwickeln einer PHP-Seite statt eines Ergebnisses im Webbrowser einfach nur eine leere Seite sehen, dann ist fast immer die `display_errors`-Einstellung schuld: Es ist ein Fehler aufgetreten, aber `display_error=0ff` verhindert die Anzeige einer Fehlermeldung.

- » **error_reporting=E_ALL**: Diese Einstellung bewirkt, dass alle PHP-Fehler und -Warnungen angezeigt werden.
- » **magic_quotes_gpc=0ff**: Diese Einstellung steuert, ob bei der Übermittlung von Daten zwischen Webseiten (Get/Post/Cookie) Sonderzeichen wie `'`, `"`, `\` oder ein 0-Byte durch `\'`, `\"`, `\\` und `\0` ersetzt werden sollen. Dieser Mechanismus gilt als veraltet (*deprecated*) und soll in PHP 6 entfernt werden. Deswegen lautet die Standardeinstellung in PHP 5.3 `0ff`. Leider verwenden viele Internet Service Provider noch immer die Einstellung `magic_quotes_gpc=0n`. Details zu *Magic Quotes* finden Sie in Abschnitt 5.5.
- » **default_charset**: Diese Option sollte entweder gar nicht gesetzt werden oder für die Beispiele dieses Kapitels auf `"iso-8859-1"` eingestellt werden. `default_charset` steuert, ob der PHP-Interpreter beim Übertragen von HTML-Dokumenten angeben soll, dass die Seiten gemäß eines bestimmten Zeichensatzes codiert sind. Mehr Informationen zum Thema Zeichensatz folgen in Abschnitt 1.9.
- » **max_execution_time=30**: Diese Option gibt an, nach welcher Zeit (in Sekunden) die Ausführung eines PHP-Scripts abgebrochen wird.
- » **max_input_time=60**: Diese Option gibt an, wie lange PHP auf Daten wartet (z.B. bei einem Datei-Upload vom Client zum PHP-Programm).
- » **file_uploads=0n**: Diese Option bewirkt, dass PHP Datei-Uploads verarbeiten kann.
- » **upload_max_filesize=2M**: Diese Option gibt an, wie groß eine Datei bei einem Datei-Upload maximal sein darf.
- » **post_max_size=8M**: Diese Option gibt an, welche Datenmengen maximal durch ein Formular (POST) zum PHP-Programm gesendet werden dürfen.
- » **memory_limit=128M**: Diese Option gibt an, wie viel Speicherplatz ein PHP-Script maximal für sich beanspruchen darf.

PHP-Erweiterungen (Extensions)

PHP-Erweiterungen ermöglichen die Verwendung zusätzlicher Funktionen, die nicht direkt in PHP integriert sind. Welche Funktionen von PHP direkt integriert sind und welche als Extensions realisiert sind, hängt davon ab, wie PHP kompiliert ist.

- » Wenn Sie PHP unter Unix/Linux selbst kompilieren, integrieren Sie die gewünschten Erweiterungen normalerweise direkt in den PHP-Interpreter (durch geeignete `./configure`-Optionen) und müssen sich daher in `php.ini` nicht um Extensions kümmern. Wenn Sie eine Funktion als Erweiterung kompilieren möchten, müssen Sie der entsprechenden `configure`-Option `=shared` hinzufügen, also z. B. `--with-pgsql=shared`.
- » Bei den meisten Linux-Distributionen wird PHP stark modularisiert ausgeliefert: Diverse Erweiterungen können dann einzeln als Paket installiert werden. Um `php.ini` müssen Sie sich in diesem Fall nicht kümmern – diese Arbeit nimmt Ihnen in der Regel das Paketverwaltungssystem ab.
- » Am wahrscheinlichsten kommen Sie mit dem PHP-Erweiterungsmechanismus unter Windows in Kontakt. Die für Windows verfügbare PHP-Version enthält kaum Zusatzfunktionen. Dafür werden alle möglichen Erweiterungen in Form von Zusatzbibliotheken mitgeliefert (*.d11-Dateien im Unterverzeichnis `ext`).

Wenn Sie Erweiterungen nutzen möchten, muss in `php.ini` die Variable `extension_dir` so eingestellt sein, dass sie das Verzeichnis der PHP-Installation angibt:

```
extension_dir = "c:\php53\ext\"
```

Des Weiteren müssen Sie in `php.ini` die gewünschten Erweiterungen durch `extension`-Einstellungen angeben. Die Zeile für das OpenSSL-Modul für Windows würde z. B. so aussehen:

```
extension=php_openssl.d11
```

`php.ini` enthält normalerweise bereits eine Menge solcher `extension`-Anweisungen, denen aber ein Strichpunkt vorangestellt ist. Der Strichpunkt kennzeichnet in `php.ini` einen Kommentar. Um die Erweiterung zu nutzen, müssen Sie also nur den Strichpunkt entfernen. Unter Unix/Linux lautet die Dateiendung für Bibliotheken `.so` statt `.d11`.

Manche PHP-Erweiterungen greifen selbst auf weitere Bibliotheken zurück. Diese müssen so installiert sein, dass der PHP-Interpreter sie findet. Unter Windows heißt das, dass solche Bibliotheken in das Windows-Verzeichnis oder in das Windows-Systemverzeichnis kopiert werden müssen!

E-Mail-Konfiguration für Windows

Abschnitt 5.7 beschreibt, wie Sie mit PHP E-Mails versenden können. Falls Sie dabei die PHP-Funktion `mail` einsetzen (zu der es aber auch Alternativen gibt), müssen Sie in `php.ini` den Rechnernamen des SMTP-Servers sowie dessen Port angeben. SMTP steht für *Simple Mail Transfer Protocol*. Die Aufgabe eines SMTP-Servers ist es, E-Mails an den Empfänger

weiterzuleiten. Zur Übertragung der Daten wird üblicherweise der TCP/IP-Port 25 verwendet. Außerdem sollten Sie die Absenderadresse angeben.

```
; E-Mail-Konfiguration in php.ini (nur für Windows)
SMTP = smtp.firma.de
smtp_port = 25
sendmail_from = "michael@kofler.cc"
```

Tatsächlich funktioniert der E-Mail-Versand mit mail trotz richtiger Konfiguration häufig nicht: Zum einen verlangen fast alle SMTP-Server eine Authentifizierung mit Name und Passwort, wozu PHP aber nicht in der Lage ist. Zum anderen wird die E-Mail möglicherweise vom Empfänger als SPAM klassifiziert und ihre Annahme verweigert – insbesondere dann, wenn der in der E-Mail-Absenderadresse angegebene Domainname nicht verifiziert werden kann. Weitere Informationen zu diesem Thema gibt Abschnitt 5.7. Dort wird auch die Klasse PEAR: :mail vorgestellt, die eine leistungsfähige Alternative zur mail-Funktion bietet.

1.8 MySQL-Konfiguration und -Administration

Konfiguration

Zahlreiche Details und Optionen des MySQL-Servers werden durch die Datei my.cnf (Unix/Linux) bzw. my.ini gesteuert. Tabelle 1.8 fasst zusammen, wo Sie diese Datei finden.

INSTALLATION	MY.CNF/MY.INI-KONFIGURATIONSDATEI
Windows-Standardinstallation	C:\Programme\MySQL\MySQL Server n.n\my.ini
Windows-Eigeninstallation	C:\mysql15n\my.ini
Linux	/etc/my.cnf oder /etc/mysql/my.cnf
Linux mit XAMPP	/opt/lampp/etc/my.cnf

Tabelle 1.8: Ort der MySQL-Konfigurationsdatei

Unter Unix/Linux funktioniert der MySQL-Server auch ohne diese Konfigurationsdatei. MySQL verwendet in diesem Fall Standardeinstellungen, die für die meisten Fälle ausreichen. Unter Windows ist my.ini aber zumeist unumgänglich.

Die Konfigurationsdatei ist in mehrere Abschnitte gegliedert. Die beiden wichtigsten Abschnitte beginnen mit den Zeilen [client] bzw. [mysqld]. Der client-Abschnitt enthält einige Optionen, die für Programme gelten, die auf den MySQL-Server zugreifen. Der mysqld-Abschnitt enthält Optionen, die das Verhalten des Servers bestimmen.

Einige Optionen können in beiden Abschnitten vorkommen. Beispielsweise gibt port die TCP/IP-Portnummer an. Diese Einstellung ist sowohl für die Clients als auch für den Server wichtig – nur wenn sich beide Programme einig sind, ist eine Kommunikation über das TCP/IP-Protokoll möglich.

Daher werden im Folgenden nur die allerwichtigsten Optionen kurz beschrieben. Die zahllosen weiteren Optionen sind vor allem für Datenbankadministratoren wichtig, die die Geschwindigkeit von MySQL optimieren möchten (siehe auch die Abschnitte 15.10 bis 15.12).

- » **port=3306** [client/mysqld]: Diese Option gibt die Port-Nummer für das TCP/IP-Protokoll an.
- » **skip_networking** [mysqld]: Diese Option bewirkt, dass zur Kommunikation mit dem MySQL-Server nur die Socket-Datei (Unix/Linux) bzw. eine sogenannte *Named Pipe* (Windows) verwendet werden kann. Eine Kommunikation über TCP/IP ist unmöglich. `skip_networking` wird manchmal aus Sicherheitsgründen verwendet, wenn ohnedies nur lokale Programme mit dem MySQL-Server kommunizieren sollen. Eine Alternative zu `skip_networking` ist `bind_address=127.0.0.1`. Diese Einstellung erlaubt die Kommunikation über TCP/IP für den lokalen Rechner (localhost), aber nicht für andere Netzwerkadressen.
- » **socket=/var/lib/mysql/mysql.sock** [client/mysqld]: Diese Option gibt die Socket-Datei für die lokale Kommunikation zwischen Client und MySQL-Server an (nur Unix/Linux).
- » **basedir=verzeichnis** [mysqld]: Diese Einstellung gibt an, in welches Verzeichnis der MySQL-Server installiert wurde (z.B. `C:/Programme/MySQL/MySQL Server 5.1` oder `/usr/`). Beachten Sie, dass Sie unter Windows als Verzeichnistrennzeichen `/` verwenden müssen, nicht `\`!
- » **datadir = verzeichnis** [mysqld]: Diese Einstellung gibt an, wo die Datenbankdateien gespeichert werden (z.B. `/var/lib/mysql`).
- » **character-set-server = latin1** [mysqld]: Diese Einstellung gibt den Defaultzeichensatz für Datenbanken, Tabellen und Textspalten an. Die Einstellung gilt nur, wenn beim Erzeugen neuer Datenbanken oder Tabellen nicht explizit ein anderer Zeichensatz angegeben wird.
- » **default_storage_engine = MyISAM** [mysqld]: Diese Einstellung gibt den Defaulttabellentyp an (MyISAM oder InnoDB, siehe auch Abschnitt 8.2).
- » **max_allowed_packet = 1M** [mysqld]: Diese Einstellung gibt an, wie groß ein SQL-Kommando bzw. dessen Ergebnis maximal sein darf. Die Einstellung ist dann wichtig, wenn Sie große binäre Objekte laden oder speichern möchten. Falls Sie keinen Zugriff auf die MySQL-Konfigurationsdatei haben (z.B. bei einer MySQL-Installation beim Webhoster), können Sie die Einstellung `max_allowed_packet` für die aktuelle Verbindung auch durch das SQL-Kommando `SET max_allowed_packet = n` verändern.
- » **old_passwords = 0/1** [mysqld]: Diese Einstellung gibt an, ob der MySQL-Server zur Version 4.0 kompatible Passwörter erzeugen soll. Auf Systemen mit alten Client-Bibliotheken erhöht das die Kompatibilität, auf Neuinstallationen sollte wegen der wesentlich höheren Sicherheit aber unbedingt `old_passwords=0` verwendet werden. Sie können die Option auch einfach weglassen, 0 gilt als Standardeinstellung.

MySQL-Installation absichern

Wenn Sie zur Installation des MySQL-Servers das neue Windows-Installationsprogramm einsetzen, hilft Ihnen dessen Konfigurationsassistent dabei, den Benutzer root durch ein Passwort abzusichern und andere Benutzer gar nicht erst zuzulassen (siehe Abschnitt 1.2). Bei Bedarf können Sie dieses Programm mit `START|PROGRAMME|MYSQL|MYSQL SERVER N.N|MYSQL SERVER CONFIG WIZARD` auch nachträglich ausführen.

Auch unter Debian und Ubuntu wird zumindest der Administrator-Login (root) bereits bei der Installation abgesichert. Bei den meisten anderen Linux-Distributionen müssen Sie sich um die Absicherung aber selbst kümmern. Dazu starten Sie das Programm `mysql` und führen dann die folgenden Kommandos aus, wobei Sie xxx durch das gewünschte Passwort ersetzen.

```
user$ mysql -u root
mysql> UPDATE mysql.user SET password=PASSWORD('xxx') WHERE user='root';
mysql> FLUSH PRIVILEGES;
mysql> exit
```

Eine ausführlichere Erläuterung der Verwaltung der Zugriffsrechte des MySQL-Servers finden Sie in den Abschnitten 7.2 und 15.1. Dort wird auch beschrieben, wie Sie neue MySQL-Benutzer mit eingeschränkten Rechten einrichten können.

mysql

`mysql`, `mysqladmin` und `mysqldump` sind Administrationskommandos, die im Textmodus arbeiten. Unter Linux werden diese Kommandos einfach in einer Konsole ausgeführt. Unter Windows müssen Sie dagegen zuerst ein Eingabeaufforderungsfenster (»DOS-Fenster«) öffnen und dann den vollständigen Pfad der Programme eintippen (z.B. `C:\Programme\MySQL\MySQL Server 5.1\bin\mysql.exe`). Einzig das Programm `mysql` kann komfortabel auch über das `START`-Menü ausgeführt werden.

Die mühsame Pfadangabe kann entfallen, wenn das `MySQL-bin`-Verzeichnis in der Windows-Umgebungsvariablen `Path` enthalten ist. Das MySQL-Konfigurationsprogramm bietet im Dialog `INSTALL AS WINDOWS SERVICE` die Möglichkeit, das zu erledigen, allerdings ist diese Option standardmäßig deaktiviert.

Beim Start der Kommandos müssen Sie normalerweise zwei Optionen angeben: `-u root` bedeutet, dass das Kommando mit den Rechten des MySQL-Benutzers `root` ausgeführt werden soll. (Natürlich können Sie auch einen anderen Benutzernamen angeben, wenn Sie die Zugriffsverwaltung von MySQL bereits entsprechend eingerichtet haben.) `-p` bewirkt, dass Sie nach dem Start des Kommandos das Passwort für den gewünschten Benutzer eingeben können.

mysql wird überwiegend dazu eingesetzt, um SQL-Kommandos auszuprobieren. Als erstes Kommando geben Sie zumeist `use datenbankname` an, um mysql klarzumachen, auf welche Datenbank Sie sich beziehen. Alle Kommandos müssen mit einem Strichpunkt enden. Mit `exit` oder `[Strg]+[D]` (nur Linux) beenden Sie mysql wieder:

```
user$ mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 5.1.25-rc-1 (Debian)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> use mysql;
mysql> SELECT user, host, password FROM user;
+-----+-----+-----+
| user      | host      | password                                     |
+-----+-----+-----+
| root      | localhost | *FACF054603403E6836B8DCFCB1EAC269746E8720 |
| root      | uranus.sol | *FACF054603403E6836B8DCFCB1EAC269746E8720 |
| mylibrary | localhost | *FACF054603403E6836B8DCFCB1EAC269746E8720 |
| pma       | localhost | *FACF054603403E6836B8DCFCB1EAC269746E8720 |
+-----+-----+-----+
4 rows in set (0,00 sec)

mysql> SELECT 1+1;
+-----+
| 1+1 |
+-----+
| 2 |
+-----+
1 row in set (0,00 sec)

mysql> exit;
Bye
```

mysql kann auch dazu verwendet werden, um in einer Textdatei gespeicherte SQL-Kommandos auf eine Datenbank anzuwenden. In der Praxis werden auf diese Weise Backups eingespielt. Das Zeichen `<` bedeutet, dass die Datei `backup.sql` gelesen werden soll. (Das funktioniert sowohl unter Windows als auch unter Linux.) Die angegebene Datenbank muss bereits existieren.

```
user$ mysql -u root -p dbname < backup.sql
Enter password: *****
```

Zeichensatzprobleme mit mysql

Beim interaktiven Einsatz von `mysql` können sowohl unter Windows als auch unter Linux Zeichensatzprobleme auftreten. Alle internationalen Sonderzeichen (also auch `ööüß`) werden dann falsch dargestellt. Am einfachsten ist es, in solchen Fällen ein moderneres Administrationswerkzeug einzusetzen, beispielsweise `phpMyAdmin` oder den `MySQL Query Browser`. Wer den Kommandointerpreter `mysql` vorzieht, kann `mysql` mit ein paar Tricks zur Zusammenarbeit bewegen.

Windows XP: Unter Windows XP besteht die Ursache des Problems darin, dass im Eingabeaufforderungsfenster ein vollkommen veralteter Defaultzeichensatz gilt. Um diesen zu ändern, öffnen Sie das Fenster mit `START|PROGRAMME|ZUBEHÖR` (nicht mit dem von `MySQL` vorgesehenen Menükommando!). Dann klicken Sie die Fensterzeile mit der rechten Maustaste an, wählen `EIGENSCHAFTEN` und stellen die Schriftart von `RASTERSCHRIFTART` auf `LUCIDA KONSOLE` um.

Als Nächstes geben Sie im Fenster das Kommando `chcp 1252` ein, um die Codepage 1252 zu aktivieren. (Eine *Codepage* ist der Windows-Ausdruck für einen Zeichensatz. Die Codepage 1252 entspricht weitestgehend dem `Latin1`-Zeichensatz.) Nach diesen Vorbereitungsarbeiten starten Sie schließlich `mysql` durch die Eingabe des vollständigen Dateinamens.

Windows Vista, 7, 2008: Hier funktioniert die Darstellung internationaler Zeichen auf Anhieb.

Linux: Im Linux-Konsolenfenster gilt bei nahezu allen Distributionen `UTF8` als Zeichensatz. Sie können sich davon mit `echo $LANG` überzeugen. Leider scheint `mysql` die Umgebungsvariable `$LANG` zu ignorieren und führt Ein- und Ausgaben im Zeichensatz `Latin1` durch. Abhilfe: Geben Sie beim Start von `mysql` die Option `--default-character-set=utf8` an, oder führen Sie in `mysql` das Kommando `SET NAMES utf8` aus.

```
user$ mysql -u root -p
mysql> SET NAMES utf8;
mysql> SELECT authName from mylibrary.authors WHERE authName LIKE 'o%';
+-----+
| authName |
+-----+
| Öggl Bernd |
| Orfali Robert |
+-----+
```

mysqldump

Mit `mysqldump` können Sie rasch ein Backup einer Datenbank erstellen. `mysqldump` kann durch unzählige Optionen gesteuert werden (siehe `mysqldump --help`). Für gewöhnliche Backups sind die Standardeinstellungen zumeist ausreichend. Beachten Sie, dass die Ausgabe nun mit `>` nach `backup.sql` umgeleitet wird, d.h., dass das Ergebnis von `mysqldump` in dieser Datei gespeichert wird.

```
user$ mysqldump -u root -p dbname > backup.sql  
Enter password: *****
```

Wenn Sie zusammen mit der Datenbank auch Stored Procedures und Trigger speichern möchten, müssen Sie zusätzlich die Optionen `--routines` und `--triggers` angeben.

mysqladmin

Mit `mysqladmin` können Sie einige einfache Administrationsaufgaben erledigen, beispielsweise neue Datenbanken anlegen oder das Passwort eines Benutzers verändern.

```
user$ mysqladmin -u root -p create dbname  
Enter password: *****
```

MySQL Administrator

Für MySQL gibt es unzählige grafische Benutzeroberflächen, die bei verschiedenen Aspekten der Administration und des Datenbank-Designs helfen. Wir gehen an dieser Stelle nur auf die Werkzeuge ein, die von der Firma MySQL für diesen Zweck entwickelt wurden. Alle im Folgenden vorgestellten Programme können von `dev.mysql.com` kostenlos heruntergeladen werden. Viele Linux-Distributionen enthalten sogar fertige Pakete. Beachten Sie, dass die Programme *MySQL Administrator*, *MySQL Query Browser* und *MySQL Migration Tools* oft in einem einzigen Paket mit dem Namen »MySQL GUI Tools« zusammengefasst werden.

Der MySQL Administrator ist ein sehr vielseitiges Werkzeug, das bei allen möglichen Aufgaben der MySQL-Administration hilft:

- » Server starten und beenden
- » Server konfigurieren
- » Aktuellen Server-Status ermitteln
- » Benutzeradministration durchführen (Verwaltung der Zugriffsrechte)
- » Protokolldateien ansehen
- » Datenbanken sichern und wiederherstellen
- » Datenbanken erzeugen und verwalten

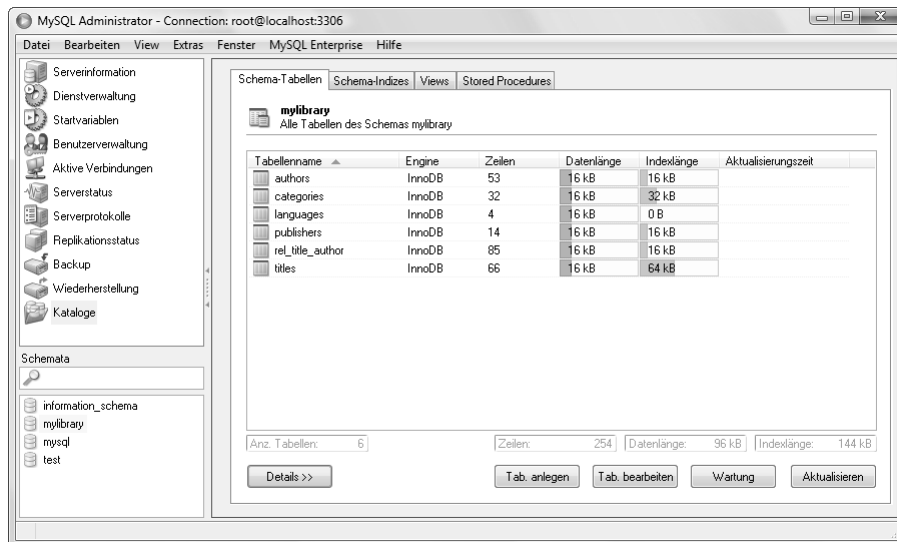


Abbildung 1.8: MySQL Administrator unter Windows

Die Bedienung des Programms ist an sich sehr einfach, setzt aber etwas Grundlagenwissen über die MySQL-Administration voraus. Beim Start des Programms müssen Sie den MySQL-Benutzernamen (meist root), das Passwort und den Rechnernamen des MySQL-Servers angeben (localhost, wenn der Server auf dem lokalen Rechner läuft).

Viele PHP-Entwickler werden den MySQL Administrator vermutlich nur zum Erzeugen und Verwalten von Datenbanken und Tabellen einsetzen (Punkt CATALOGS). Lassen Sie sich nicht davon verwirren, dass das Programm Datenbanken als *Schemata* bezeichnet.

Beachten Sie, dass Sie die MySQL-Konfigurationsdatei `my.ini` bzw. `my.cnf` nur dann verändern können, wenn Sie das Programm mit Administrator- bzw. root-Rechten starten.

MySQL Query Browser

Der MySQL Query Browser hilft dabei, SQL-Kommandos und *Stored Procedures* zu entwickeln und zu testen. Ergebnisse von SQL-Kommandos werden in Tabellen dargestellt. Bei einfachen Abfragen (z.B. `SELECT * FROM tabelle`) können Sie die Datensätze direkt in der Tabelle ändern bzw. neue Datensätze hinzufügen.

Beim Start des Programms müssen Sie eine Verbindung zum Datenbank-Server herstellen. Dabei müssen Sie wie üblich den MySQL-Benutzernamen, das Passwort und den Rechnernamen (oft localhost) angeben. Im Eingabefeld `DEFAULT SCHEMA` geben Sie die Datenbank an, die Sie mit dem Query Browser bearbeiten möchten. Sie können auf diese Eingabe verzichten, müssen dann aber vor der Ausführung mancher Kommandos später die Standarddatenbank durch `FILE|SELECT SCHEMA` einstellen.

Bereits ausgeführte Kommandos können Sie später über den BACK-Button nochmals auswählen. So ersparen Sie sich die neuerliche Eingabe. Außerdem können Sie mehrere Abfragen bzw. Ergebnisse parallel verwalten, wenn Sie dazu mit FILE|NEW QUERY TAB ein neues Dialogblatt öffnen.

Wenn Sie die Syntax eines SQL-Kommandos vergessen haben, reicht ein Doppelklick auf das entsprechende Kommando im SYNTAX-Bereich des Fensters. Der Query Browser öffnet dann ein weiteres Dialogblatt, das den entsprechenden Ausschnitt aus dem MySQL-Handbuch anzeigt. Die Hilfetexte haben allerdings eine im wahrsten Sinne des Wortes erschöpfende Länge. Darin die wenigen Details zu erkennen, die gerade interessant wären, ist eine Kunst für sich.

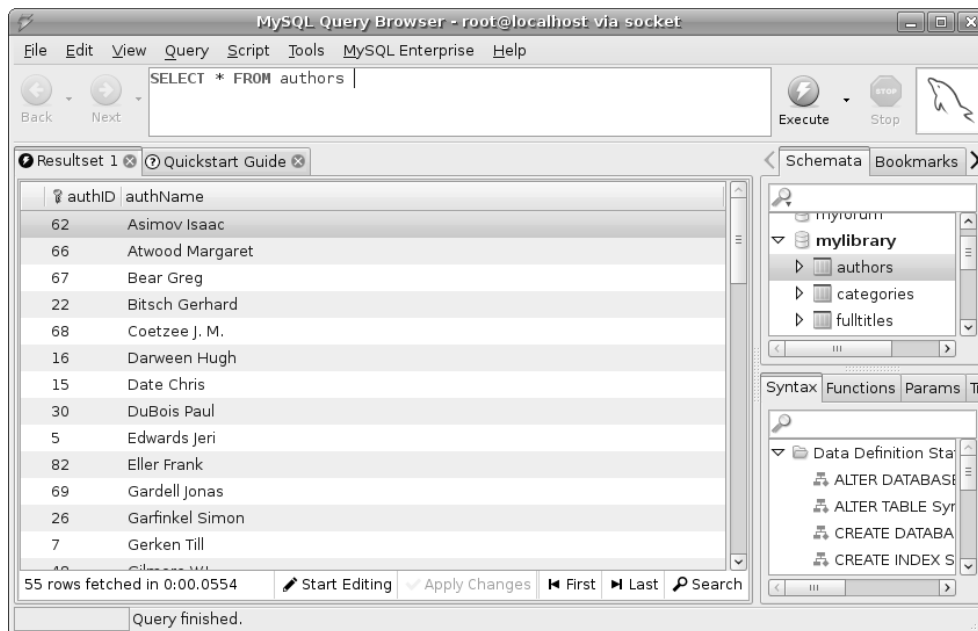


Abbildung 1.9: MySQL Query Browser unter Linux

MySQL Workbench

MySQL Workbench ist ein relativ neues Programm, das beim visuellen Datenbankdesign hilft (siehe Abbildung 1.10). MySQL Workbench steht in zwei Varianten zur Auswahl: in einer etwas abgespeckten OSS-Version (Open-Source-Software) und als Enterprise-Version mit einigen Zusatzfunktionen für zahlende Kunden. Trotz der schönen Benutzeroberfläche ist die Bedienung des Programms alles andere als intuitiv. Ein Studium der umfassenden Dokumentation ist unabdingbar. Eine Menge Informationen zu dem Programm finden Sie unter:

<http://dev.mysql.com/workbench/>

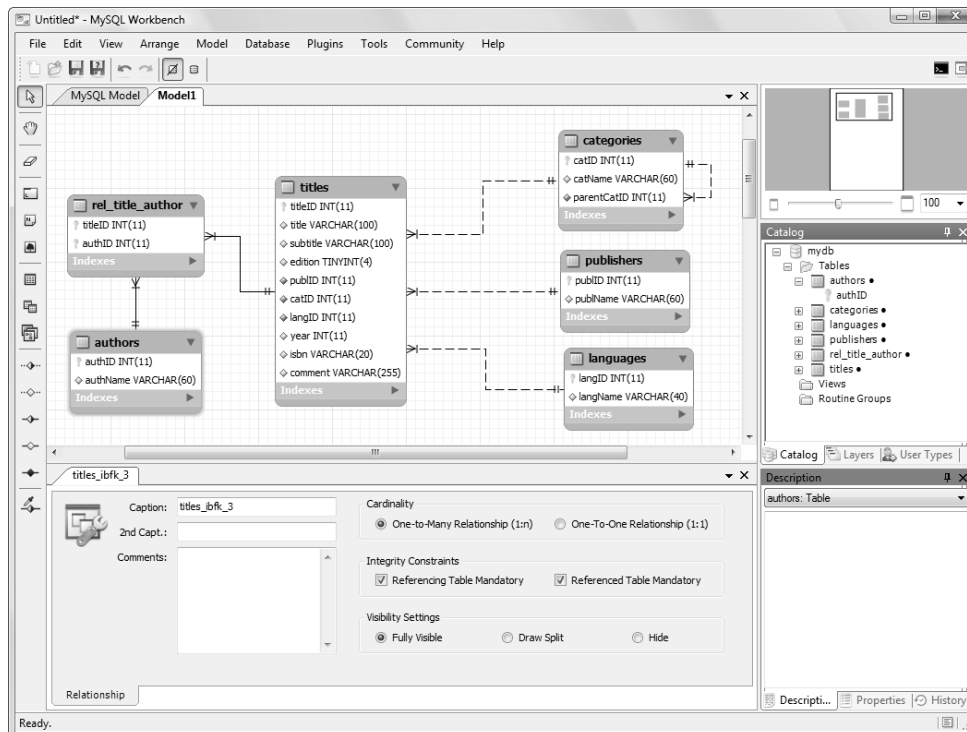


Abbildung 1.10: MySQL Workbench

MySQL Migration Toolkit

Das Migration Toolkit hilft dabei, Datenbanken von fremden Datenbanksystemen in MySQL zu importieren. Das Programm setzt voraus, dass auf dem Rechner Java installiert ist. Es werden grundsätzlich alle Datenbanksysteme unterstützt, für die es Java-Treiber gibt (unter anderem *Microsoft Access*, *Microsoft SQL Server* und *Oracle*). Allerdings können manche Objekte – beispielsweise Stored Procedures – wegen grundlegender Inkompatibilitäten nicht importiert werden.

Auch sonst gibt es Einschränkungen: Beispielsweise kann das Programm nur dann eine Verbindung zum Microsoft SQL Server herstellen, wenn der SQL Server im Modus *mixed security* konfiguriert ist, also einen Login mit Benutzername und Passwort erlaubt. Genau das ist standardmäßig nicht der Fall – der SQL Server verlässt sich in der Regel auf das Windows-Authentifizierungsmodell (*integrated security*). Damit ist der Migrationsversuch bereits gescheitert, denn eine Änderung des Sicherheitsmodus des SQL Servers erfordert dessen Neuinstallation ...

Sind die Voraussetzungen aber erfüllt, funktioniert das Programm hervorragend. Der Import der Access-Beispieldatenbank *Nwind.mdb* gelang beispielsweise innerhalb von zwei Minuten!

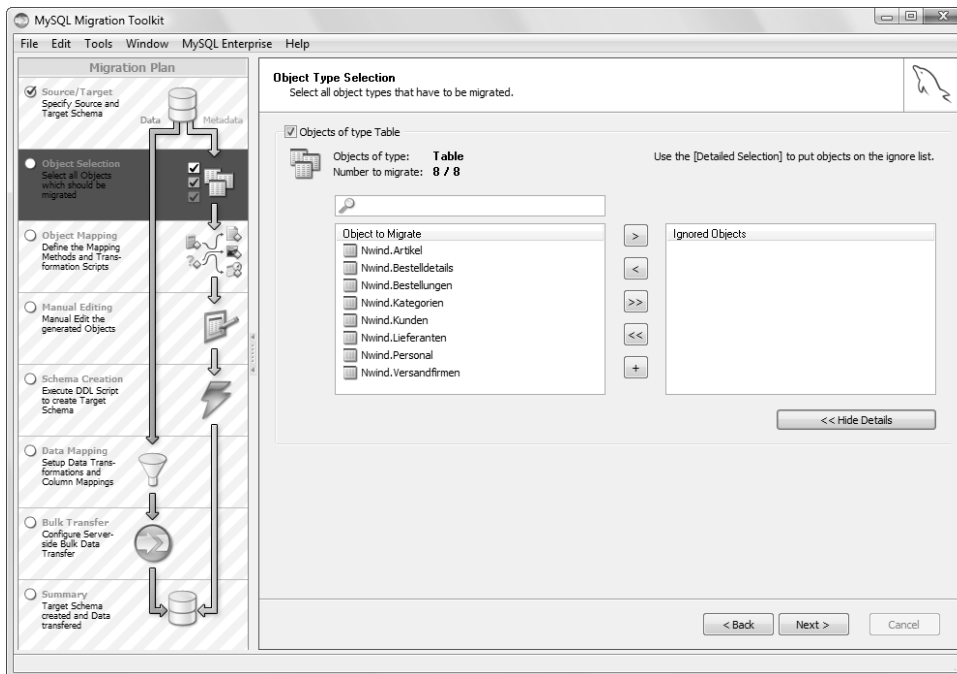


Abbildung 1.11: MySQL Migration Toolkit

Weitere Informationen zu dem Programm finden Sie unter:

<http://dev.mysql.com/doc/migration-toolkit/en/index.html>

phpMyAdmin

Das beliebteste Administrationsprogramm für PHP-Entwickler ist phpMyAdmin. Der Installation und Anwendung dieses Programms ist das gesamte Kapitel 7 gewidmet.

1.9 Zeichensatz (Latin1 oder Unicode UTF-8)

Dieser Abschnitt geht auf die Frage ein, welchen Zeichensatz Sie für die Entwicklung Ihrer Webanwendungen verwenden sollen. Diese Frage ist relativ elementar, und sie sollte beantwortet werden, bevor Sie mit der Entwicklung eines größeren Projekts beginnen. Damit ersparen Sie sich eine mühsame nachträgliche Änderung des Zeichensatzes.

Um das Thema hier nicht ausufern zu lassen, beschränkt sich der Abschnitt auf die beiden in unserem Sprachraum wichtigsten Zeichensätze, nämlich auf Latin1 alias ISO-8859-1 und auf Unicode UTF-8 alias ISO-10646.

- » Wenn für Ihre Anwendung der Latin1-Zeichensatz ausreicht (und das wird für den Großteil der im deutschen Sprachraum entwickelten Anwendungen der Fall sein), verwenden Sie diesen Zeichensatz. Er verursacht bei Weitem die geringsten Probleme. Aus diesem Grund nutzen sämtliche Beispiele dieses Buchs Latin1.
- » Wenn Sie Unicode einsetzen (müssen), gehen Sie dabei möglichst konsequent vor. Nutzen Sie alle zur Verfügung stehenden Unicode-Möglichkeiten sämtlicher Glieder der Entwicklungskette, also Betriebssystem, Webserver, PHP, Datenbank, Editor bzw. Entwicklungswerkzeuge etc.

Theoretisch ist es auch möglich, in unterschiedlichen Ebenen der Entwicklung bzw. Datenverwaltung unterschiedliche Zeichensätze einzusetzen. Die Wahrscheinlichkeit, dass dann an irgendeinem Punkt beim Übergang von Latin1 zu Unicode (oder umgekehrt) Probleme auftreten, beträgt aber nahezu 100 Prozent.

Da der Rest dieses Buches Latin1 als Zeichensatz für alle Aspekte einer Webapplikation voraussetzt, können Sie diesen Abschnitt als Anleitung zur Unicode-Umstellung betrachten. Aus diesem Grund ist der Abschnitt leider recht technisch und setzt zum Teil schon etwas Hintergrundwissen voraus, das im Detail erst in den nachfolgenden Kapiteln vermittelt wird.

Latin-Zeichensätze

Zeichensätze bestimmen, welche Codes zur Darstellung von Zeichen verwendet werden. Bei den 128 US-ASCII-Zeichen sind sich die meisten Zeichensätze einig (z. B. steht Code 65 für den Buchstaben A). Problematischer ist die Darstellung internationaler Zeichen.

In der Vergangenheit wurden je nach Sprachraum verschiedene 1-Byte-Zeichensätze entwickelt, von denen die Latin-Zeichensätze die größte Verbreitung gefunden haben: Latin1 alias ISO-8859-1 enthält alle in Westeuropa üblichen Zeichen (äöüßáâäå etc.), Latin2 alias ISO-8859-2 enthält Zeichen aus Zentral- und Osteuropa etc. Latin0 alias Latin9 alias ISO-8859-15 entspricht Latin1, enthält aber zusätzlich das Euro-Zeichen.

Die Latin-Zeichensätze werden von aktuellen Windows-Versionen unterstützt. Bei älteren Windows-Versionen kann ersatzweise die Codepage 1252 verwendet werden (kurz CP 1252, manchmal auch ANSI-Zeichensatz genannt), die bis auf wenige Abweichungen dem Latin1-Zeichensatz entspricht.

Das Problem bei diesen Zeichensätzen ist offensichtlich: Ihre Anwendung kommt nie mit allen Zeichen aus ganz Europa zurecht, weil jeder Latin-Zeichensatz nur eine Teilmenge der Zeichen enthält.

Unicode-Zeichensätze

Als Lösung wurde der Zeichensatz Unicode entwickelt. Mit anfänglich 65.535 möglichen Zeichen deckt er nicht nur alle Zeichen ganz Europas ab, sondern darüber hinaus auch noch die der meisten asiatischen Sprachen. Ab Version 3.1 unterstützt Unicode sogar mehr als 65.535 Zeichen.

Unicode regelt allerdings nur, welcher Code welchem Zeichen zugeordnet ist, nicht, wie die Codes tatsächlich gespeichert werden. Hierfür bestehen wieder mehrere Varianten, von denen UCS-2 und UTF-8 die beiden wichtigsten sind. (UTF steht für *Unicode Transfer Format*, UCS für *Universal Character Set*.)

- » **UCS-2:** Die einfachste Lösung scheint auf den ersten Blick darin zu bestehen, jedes Zeichen einfach durch 2 Byte (also 16 Bit) darzustellen. Diese Formatierung wird UCS-2 genannt. Fast alle Betriebssystemfunktionen von Microsoft Windows verwenden diese Darstellung.

Sie hat allerdings drei Nachteile: Erstens verdoppelt sich der Speicherbedarf, und zwar auch in solchen Fällen, wo überwiegend europäische Zeichen oder gar nur US-ASCII-Zeichen gespeichert werden sollen. Zweitens tritt der Bytecode 0 an beliebigen Stellen in Unicode-Zeichenketten auf. Bei Texten mit US-ASCII-Zeichen ist sogar jedes zweite Byte 0. Viele C-Programme, E-Mail-Server etc. setzen aber voraus, dass das Byte 0 das Ende einer Zeichenkette markiert. Drittens ist UCS-2 inkompatibel zu aktuellen Unicode-Versionen, die mehr als 65.536 verschiedene Zeichen definieren.

- » **UTF-8:** Die bei Weitem populärste Alternative zu UCS-2 ist UTF-8. Dabei werden die US-ASCII-Zeichen (7 Bit) wie bisher durch ein Byte dargestellt, deren oberstes Bit 0 ist. Alle anderen Unicode-Zeichen werden durch zwei bis vier Byte lange Byte-Ketten dargestellt.

Der offensichtliche Nachteil dieses Formats besteht darin, dass es keinen unmittelbaren Zusammenhang zwischen der Byteanzahl und der Anzahl der Zeichen eines Dokuments gibt. Wegen der größeren Kompatibilität zu existierenden Programmen und einer Reihe anderer Vorteile hat sich UTF-8 unter Unix/Linux und bei den meisten für die Webentwicklung wichtigen Komponenten als Standard etabliert. Wenn in diesem Buch von Unicode die Rede ist, ist in Zukunft immer Unicode im UTF-8-Format gemeint.

Alle gängigen Linux-Distributionen verwenden standardmäßig UTF-8, also beispielsweise zur Ein- und Ausgabe von Text in der Konsole, zur Darstellung der Dateinamen sowie für alle Textdateien.

Unter Windows wird UTF-8 dagegen nur eingeschränkt unterstützt. Einzelne Programme (unter anderem der minimalistische Editor `notepad.exe`!) sind in der Lage, UTF-8-Textdateien zu laden und zu speichern. Standardmäßig kommt UTF-8 aber nicht zum Einsatz.

Trotz der offensichtlichen Vorteile von Unicode – egal, in welcher Darstellung – gibt es zwei Gründe, die gegen den sofortigen Umstieg sprechen: Zum einen ist der Unicode-Zeichensatz inkompatibel mit den bekannten 1-Byte-Zeichensätzen; vorhandene Datenbestände und Codedateien müssen also konvertiert werden.

Zum anderen ist die Unicode-Unterstützung der Komponenten, die bei der Webentwicklung zum Einsatz kommen, noch alles andere als perfekt. (Das schwächste Glied der Kette ist übrigens PHP, und das wird sich voraussichtlich erst mit Version 6 ändern.)

Zeichensatzunterstützung in Apache, PHP und MySQL

Die folgenden Abschnitte beschreiben die Zeichensatzunterstützung in der gesamten Kette der zur Webentwicklung erforderlichen Werkzeuge und Programme, vom Apache-Server bis zum Webbrowser auf dem Client. Der Abschnitt geht auch auf das Format HTML und das Protokoll HTTP sowie auf die Zeichensatzkonfiguration unter Windows und Linux ein. Wenn Sie weitergehende Informationen zu Zeichensätzen suchen, sind die folgenden Seiten empfehlenswert:

<http://de.wikipedia.org/wiki/Unicode>
<http://www.unicode.org/faq/>
<http://www.cl.cam.ac.uk/~mgk25/unicode.html>
<http://www.w3.org/International/0-HTTP-charset>
<http://www.w3.org/International/questions/qa-forms-utf-8.html>

HTML

Laut HTML-Standard gilt für alle Dokumente ohne explizite Zeichensatzangabe der Zeichensatz ISO-8859-1 (also Latin1). Den verwendeten Zeichensatz können Sie am Beginn von HTML-Dokumenten angeben. Im HTML-Code sieht das so aus:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
...
```

Zulässige charset-Einstellungen sind utf-8, iso-8859-1, iso-8859-15 etc. Das Problem besteht allerdings darin, dass viele Webbrowser diese Information schlicht ignorieren und sich stattdessen auf die HTTP-Header-Daten verlassen (siehe unten).

HTML-Codes für Sonderzeichen: Wenn Sie möchten, dass internationale Sonderzeichen auch dann richtig dargestellt werden, wenn der Webbrowser die Zeichensatzinformation nicht oder falsch auswertet, können Sie für viele Sonderzeichen spezielle HTML-Codes einsetzen, beispielsweise `ä`; statt ä. Das macht Texte in HTML-Dokumenten allerdings schwer zu lesen. Bei Ausgaben, die Sie mit PHP durchführen, setzen Sie einfach die Funktion `htmlentities` ein. (`htmlentities('aäb')` liefert die Zeichenkette `aäb`).

Wenn Sie als Zeichensatz Latin1 bzw. ISO-8859-1 nutzen, können Sie zwar Sonderzeichen wie äöüß direkt im HTML-Code verwenden, nicht aber das Euro-Zeichen. Um das Euro-Zeichen dennoch fehlerfrei auszugeben, verwenden Sie einfach den HTML-Code €!

Formulare: Normalerweise werden in ein HTML-Formular eingegebene Texte in dem Zeichensatz versendet, in dem die HTML-Seite dargestellt wird. Hat also beispielsweise die Erkennung des Unicode-Zeichensatzes geklappt, dann sollten auch die eingegebenen Zeichen als Unicode versandt werden.

Was aber, wenn dies bei alten Browsern nicht zuverlässig funktioniert, wenn die Zeichensatzerkennung im Browser fehlschlägt oder im Browser ein bestimmter Zeichensatz fest eingestellt ist? Dann werden die Texteingaben aus dem Formular in einem anderen Zeichensatz zurückgeliefert und in der Folge falsch gespeichert.

HTTP-Protokoll

Das HTTP-Protokoll definiert, wie der Webbrowser und der Webserver miteinander kommunizieren, wie HTML-Dokumente und Formulardaten übertragen werden etc. Dieses Protokoll bietet neben dem HTML-`<meta>`-Tag die zweite Möglichkeit, den Zeichensatz einzustellen. Es wurde bereits erwähnt, dass sich die meisten Webbrowser auf die Information des HTTP-Protokolls verlassen, die Zeichensatzangabe im `<meta>`-Tag aber ignorieren.

Entscheidend für den Zeichensatz ist die Header-Information über den Dokumenttyp. Der HTTP-Header wird vor dem eigentlichen HTML-Dokument übertragen und ist daher nicht Teil des HTML-Codes. Einige Webbrowser bieten aber ein Kommando zur Anzeige der Seiteninformationen oder -eigenschaften an, aus denen auch die Header-Informationen hervorgehen (z.B. `EXTRASISEITENINFORMATIONEN` bei Firefox, siehe Abbildung 1.12).

Woher kommt nun die Header-Information über den Dokumenttyp samt Zeichensatz? Es bestehen folgende Möglichkeiten:

- » Apache-Defaulteinstellung (`AddDefaultCharset utf-8` in `httpd.conf`)
- » Apache-Verzeichniskonfiguration (`AddDefaultCharset` in `.htaccess`)
- » HTML-`<meta>`-Tag (wird von Apache nur dann ausgewertet, wenn `AddDefaultCharset off` gilt!)
- » PHP-Defaulteinstellung (`default_charset="utf-8"` in `php.ini`)
- » PHP-Code der betreffenden Seite:

```
<?php header("Content-Type: text/html; charset=utf-8"); ?>
```

Beachten Sie, dass die Header-Funktion am Beginn der PHP-Seite und vor jeder HTML-Ausgabe erfolgen muss!

Details zu den Konfigurationsmöglichkeiten von Apache und PHP folgen in den weiteren Abschnitten.

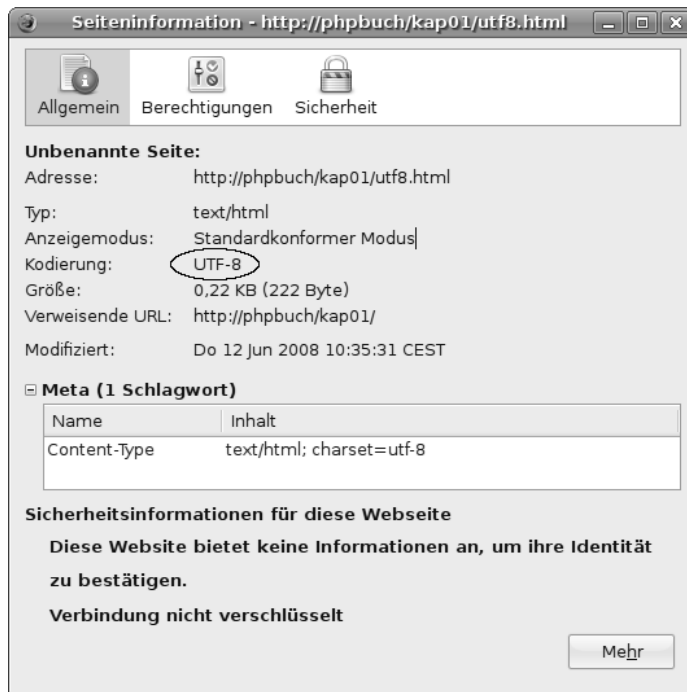


Abbildung 1.12: Seiteninformationsdialog des Webbrowsers Firefox

Webbrowser

Wie bereits erwähnt wurde, ist es die Aufgabe des Webbrowsers, den Zeichensatz des Dokuments zu erkennen und das Dokument entsprechend korrekt anzuzeigen. Der Webbrowser kann dazu zwei Informationen auswerten: die HTTP-Header-Daten und das HTML-`<meta>`-Tag. Die meisten Browser berücksichtigen nur die Header-Informationen und ignorieren das `<meta>`-Tag. Über die korrekte Erkennung des Zeichensatzes hinaus bestimmen noch zwei Faktoren die korrekte Darstellung der Seite:

- » Der Browser muss den Zeichensatz kennen. Das ist keineswegs immer der Fall. Sehr alte Browser haben beispielsweise keine Unicode-Unterstützung. Aber selbst moderne Browser können nicht alle Zeichensätze kennen – dafür gibt es einfach zu viele. (Latin1 alias ISO-8859-1 und UTF-8 bereiten natürlich keinem aktuellen Browser Probleme. Die meisten aktuellen Browser kommen auch mit Latin9 alias ISO-8859-15 zurecht.)
- » Der Browser muss Zugang zu Schriftarten haben, die die gewünschten Zeichen enthalten. Für die in Europa üblichen Zeichen ist das kein Problem. Es gibt aber nur sehr wenige Schriften, die alle in Unicode definierten Zeichen (also auch alle asiatischen Zeichen) enthalten.

Apache

Es bestehen mehrere Möglichkeiten, Apache mitzuteilen, welchen Zeichensatz er in der HTTP-Header-Information an den Browser übermitteln soll:

- » **AddDefaultCharset Zeichensatz** in `httpd.conf`: Apache übermittelt den hier angegebenen Zeichensatz (z.B. `utf-8` oder `iso-8859-1`) für alle Seiten an den Browser. Die Einstellung gilt sowohl für `.html`- als auch für `.php`-Dateien. Das `<meta>`-Tag im HTML-Code wird ignoriert.
- » **AddDefaultCharset off** in `httpd.conf`: Apache wertet das `<meta>`-Tag aus und sendet den dort angegebenen Zeichensatz an den Browser. Es gilt also der Inhalt des `<meta>`-Tags. Diese Einstellung gilt bei den meisten Linux-Distributionen.
- » **AddCharset Zeichensatz .kennung** in `httpd.conf`: Damit wird ein Zeichensatz für Dateien einer bestimmten Kennung eingestellt. `AddCharset utf-8 .utf8` bewirkt also, dass für alle Dateien, deren Name auf `.utf8` endet, als Zeichensatz `utf-8` an den Browser gemeldet wird.
- » **AddDefaultCharset** und **AddCharset** in `.htaccess`: Die beiden Schlüsselwörter sind auch in `.htaccess` zulässig und ermöglichen so eine verzeichnisspezifische Konfiguration. Das ist praktisch, wenn Sie (z.B. bei einem Webhoster) keinen Einfluss auf `httpd.conf` haben.

Beachten Sie aber, dass die Einstellungen in `.htaccess` nur berücksichtigt werden, wenn `httpd.conf` für das Webverzeichnis Veränderungen durch lokale Konfigurationsdateien zulässt (`AllowOverride All` oder `FileInfo` in der betreffenden `<Directory>`-Gruppe). Vorsicht: Die Defaulteinstellung für `AllowOverride` lautet oft `None`. Die `.htaccess`-Datei wird in diesem Fall vollständig ignoriert.

PHP

Falls Sie einen anderen Zeichensatz als `Latin1` einsetzen, ist PHP momentan das schwächste Glied der Kette. Zwar ist PHP prinzipiell in der Lage, HTML-Dokumente in beinahe jedem Zeichensatz zu erzeugen. Funktionen wie `echo` oder `print` geben Zeichenketten einfach unverändert aus und kümmern sich dabei nicht um den Zeichensatz.

Zeichenkettenfunktionen: Das eigentliche Problem besteht aber darin, dass alle PHP-Zeichenkettenfunktionen davon ausgehen, dass die Zeichenketten den `Latin1`-Zeichensatz verwenden. Wenn die Zeichenketten dagegen `UTF-8`-codiert sind,

- » liefert `strlen` die Anzahl der Bytes statt der Anzahl der Zeichen,
- » können Sie sich nicht auf die Ergebnisse der Kleiner- und Größer-Operatoren verlassen,
- » können Sie Zeichenketten-Arrays nicht zuverlässig sortieren,

» können Sie sich auf die Ergebnisse wichtiger Funktionen wie `stripslashes` oder `htmlspecialchars` nicht mehr verlassen.

Ganz ausweglos ist die Situation zum Glück auch nicht: PHP stellt die Funktionen `utf8_encode` und `utf8_decode` zur Verfügung, um Zeichenketten zwischen UTF-8 und ISO-8859-1 zu konvertieren. Falls PHP mit der `iconv`-Erweiterung kompiliert ist bzw. wenn diese Erweiterung als Modul aktiviert wird (unter Windows `extension= php_iconv.dll` in `php.ini`), stehen neben den Standardzeichenkettenfunktionen auch `iconv`-Varianten zur Verfügung. `iconv_strlen($s, "UTF-8")` ermittelt dann die tatsächliche Zeichenanzahl einer UTF-8-Zeichenkette. Noch mehr Zusatzfunktionen stellt die `mbstring`-Erweiterung zur Verfügung. Damit können Sie auch reguläre Ausdrücke auf Multibyte-Zeichenketten anwenden, E-Mails versenden etc.

Zeichensatz der PHP-Dateien: Die PHP-Dateien müssen in einem Zeichensatz vorliegen, der es dem PHP-Interpreter ermöglicht, die PHP-Sprachelemente korrekt zu erkennen. Das setzt voraus, dass zumindest die US-ASCII-Zeichen wie bei Latin1 codiert sind. Diese Voraussetzung ist für UTF-8-Dateien zum Glück erfüllt.

HTTP-Zeichensatz-Header in `php.ini` einstellen: Standardmäßig gibt der PHP-Interpreter keine HTTP-Header-Informationen zum Zeichensatz an Apache weiter und überlässt es somit dem Webserver, ob und welche Header-Zeichensatzdaten dieser an den Browser sendet.

Durch die Einstellung `default_charset="utf-8"` in `php.ini` können Sie dieses Verhalten ändern. Der PHP-Interpreter gibt nun den angegebenen Zeichensatz an Apache weiter. Dieser verändert die Informationen nicht mehr. Das heißt, die von PHP stammenden Informationen haben Vorrang gegenüber denen aus der Apache-Konfiguration!

HTTP-Zeichensatz-Header dynamisch einstellen: Sie können den gewünschten Header-Zeichensatz auch durch die PHP-Funktion `header` erzeugen. Das ermöglicht eine dokumentenspezifische Zeichensatzeinstellung. Die Funktion `header` muss am Beginn der PHP-Seite aufgerufen werden (also vor jeder HTML-Ausgabe!).

```
<?php header("Content-Type: text/html; charset=utf-8"); ?>
```

Die durch die `header`-Funktion gesendeten Daten haben Vorrang gegenüber sowohl der `default_charset`-Einstellung als auch gegenüber der Apache-Konfiguration.

MySQL

Geradezu vorbildlich ist die Zeichensatzunterstützung von MySQL seit Version 4.1. Sie können nicht nur für den Server oder für eine Datenbank den gewünschten Zeichensatz einstellen, sondern auch für jede Tabelle und sogar für jede einzelne Spalte einer Tabelle. Es ist also möglich, in einer Tabelle eine Spalte mit Latin1-Zeichenketten und eine zweite Spalte mit UTF8-Zeichenketten zu erzeugen.

MySQL unterstützt eine ganze Reihe verschiedener Zeichensätze (hier in der MySQL-typischen Schreibweise): `ascii`, `binary`, `cp1250`, `greek`, `hebrew`, `latin1`, `latin2`, `latin5`, `latin7`, `ucs2`, `utf8` etc. In der Liste fehlt allerdings `latin9`. Das Euro-Zeichen können Sie daher nur in `utf8`- oder `ucs2`-Spalten speichern.

UTF8-Spalten weisen gegenüber Latin1-Spalten momentan eine wesentliche Einschränkung auf: Die deutsche Sortierordnung wird nicht unterstützt. Die Sortierordnung `utf8_general_ci` ist zwar für viele deutschsprachige Anwendungen ausreichend, aber wenn Sie gemäß dem DIN-1- oder DIN-2-Standard sortieren möchten, müssen Sie bis auf Weiteres bei Latin1-Spalten bleiben. Details zur Zeichensatzunterstützung im Betrieb von MySQL finden Sie in den folgenden Abschnitten des Buchs:

- Zeichensatz und Sortierordnung für Tabellenspalten: Abschnitt 8.3
- CREATE DATABASE/TABLE: Abschnitt 9.10
- Zeichensatzinformationen in SQL-Kommandos: Abschnitt 10.1

Konfiguration des Standardzeichensatzes: In der MySQL-Konfigurationsdatei `my.cnf` bzw. `my.ini` (Windows) können Sie mit `default-character-set = latin1` bzw. `= utf8` den Defaultzeichensatz für den MySQL-Server einstellen (Optionsgruppe `[mysqld]`). Die Einstellung gilt für neue Datenbanken, Tabellen und Textspalten, soweit beim Erzeugen dieser Objekte nicht explizit ein anderer Zeichensatz angegeben wird.

Zeichensatztransformation durch die Client-Bibliotheken: Was nützt es, wenn der MySQL-Server selbst mit allen erdenklichen Zeichensätzen umgehen kann, der Client (also z. B. Ihr PHP-Programm oder der Kommandozeileninterpreter `mysql`) aber einen ganz bestimmten Zeichensatz voraussetzt? Die MySQL-Client-Bibliotheken wandeln daher automatisch alle Zeichenketten vom Zeichensatz des Clients in den Zeichensatz des Servers um (und umgekehrt).

Für welche Aspekte der Verbindung zwischen Client und Server welche Zeichensätze zum Einsatz kommen, wird durch mehrere MySQL-Systemvariablen gesteuert, die in Tabelle 1.9 zusammengefasst sind.

VARIABLE	BEDEUTUNG
<code>@@character_set_client</code>	Zeichensatz des Clients
<code>@@character_set_server</code>	Defaultzeichensatz des Servers
<code>@@character_set_connection</code>	Zeichensatz für die Verbindung zwischen Client und Server
<code>@@character_set_results</code>	Gewünschter Zeichensatz für SELECT-Ergebnisse
<code>@@character_set_database</code>	Defaultzeichensatz der Datenbank

Tabelle 1.9: MySQL-Zeichensatzvariablen

Einige dieser Variablen können Sie in Ihrem Client-Programm verändern, indem Sie dort die folgenden SQL-Kommandos ausführen:

```
SET @@session.character_set_client = 'utf8'
SET @@session.character_set_results = 'utf8'
SET @@character_set_connection = 'utf8'
```

Alternativ gibt es je nach MySQL-Schnittstelle auch spezielle Funktionen, um den Zeichensatz einzustellen:

```
mysql_set_charset('utf8'); // mysql-Schnittstelle
$mysqli->set_charset("utf8"); // mysqli-Schnittstelle
```

Nach diesen Vorarbeiten können Sie SQL-Kommandos mit UTF8-Zeichenketten ausführen und erhalten SELECT-Ergebnisse ebenfalls im UTF8-Format.

MySQL-Kommandos (mysql, mysqldump)

Der Kommandointerpreter `mysql` versucht selbst zu erkennen, in welchem Zeichensatz Ein- und Ausgaben erfolgen sollen. Das gelingt manchmal, aber nicht immer. Manchmal müssen Sie daher mit der Option `--default-character-set=latin1` oder `=utf8` nachhelfen.

`mysqldump` erzeugt immer UTF8-Dateien. Die Backup-Datei enthält SQL-Anweisungen, die dem MySQL-Server den Zeichensatz mitteilen. Es besteht somit keine Gefahr, dass der Datenbankserver die Datei falsch interpretiert.

MySQL-Administrationswerkzeuge (phpMyAdmin, Query Browser etc.)

Administrationswerkzeuge mit einer eigenen grafischen Benutzeroberfläche kommen erwartungsgemäß gut mit verschiedenen Zeichensätzen zurecht. Überraschenderweise gilt dies auch für `phpMyAdmin`: Das Programm ist gewissermaßen der Beweis dafür, dass man mit PHP bereits jetzt absolut Unicode-taugliche Programme entwickeln kann.

Linux

Wie bereits erwähnt wurde, verwenden alle gängigen Linux-Distributionen mittlerweile den Zeichensatz UTF8. Dieser Zeichensatz gilt aber auch für den Editor, mit dem Sie Ihre PHP-Dateien erstellen. Diese Dateien sollten aber nach Möglichkeit den Latin1-Zeichensatz verwenden – andernfalls riskieren Sie Probleme bei der Bearbeitung von Zeichenketten durch PHP-Funktionen. Die einfachste Lösung besteht darin, in PHP-Dateien generell die Verwendung internationaler Zeichen zu vermeiden. Ist dies nicht möglich, denken Sie daran, die Datei explizit im Latin1-Zeichensatz zu speichern. Nahezu jeder Editor bietet diese Möglichkeit: Die Vorgehensweise ist aber ebenso mühsam wie fehleranfällig.

Wenn Sie den Zeichensatz von Text- oder Codedateien nachträglich ändern möchten, helfen Ihnen dabei die Kommandos `recode` oder `iconv`. Die beiden folgenden Zeilen zeigen zwei alternative Möglichkeiten, um Latin1-Dateien in UTF8-Dateien umzuwandeln:

```
user$ recode latin1..u8 < latin1dat > utf8dat
user$ iconv -f latin1 -t utf-8 latin1dat > utf8dat
```

1.10 Editor und Entwicklungsumgebung

Jetzt brauchen Sie nur noch einen Texteditor zum Schreiben der *.php-Dateien. Editoren oder auch komplette PHP/HTML-Entwicklungsumgebungen (z.B. *Quanta*, *PHP-Eclipse*) finden Sie im Internet wie Sand am Meer, sowohl in kommerziellen als auch in kostenlosen Open-Source-Ausführungen. Tabelle 1.10 gibt einige Empfehlungen für kostenlose Editoren für Windows und Linux.

EDITOR	BESCHREIBUNG
Notepad++	Open-Source-Editor für Windows (http://notepad-plus.sourceforge.net/)
kate	Standardeditor des KDE-Desktops (nur Linux)
gedit	Standardeditor des Gnome-Desktops (nur Linux)
vi/vim	Komplexer Unix/Linux-Editor, auch für Windows verfügbar
emacs/xemacs	Komplexer Unix/Linux-Editor, ebenfalls für Windows verfügbar

Tabelle 1.10: **Texteditoren**

Beachten Sie, dass das minimalistische Windows-Programm `notepad.exe` zur Bearbeitung der Beispieldateien ungeeignet ist: Diese wurden zumeist unter Linux erstellt und enthalten Linux-typische Zeilenumbruchszeichen. `notepad.exe` kommt damit nicht zurecht.

Persönlich haben wir überwiegend unter Linux mit `emacs` und `vi` gearbeitet. Das Funktionsangebot dieser Editoren ist nahezu unbegrenzt. Wer sich einmal an die oft eigenwillige Bedienung gewöhnt hat, setzt nie wieder freiwillig einen anderen Editor ein ...