



Inhaltsverzeichnis

1. **Inhaltsverzeichnis**
2. **WAMP Umgebungen**
3. **Der Uniform Server**
4. **Die ODBC-Verbindung 1**
5. **Die ODBC-Verbindung 2**
6. **Namo - Listen Formular 1**
7. **Namo - Listen Formular 2**
8. **Namo - Listen Formular 3**
9. **Namo - Listen Formular 4**
10. **Namo - Listen Formular 5**
11. **Namo - Listen Formular 6**
12. **Datenbankrechte 1**
13. **Datenbankrechte 2**
14. **Datenbankrechte 3**
15. **Namo - Eingabeformular 1**
16. **Namo - Eingabeformular 2**
17. **Database Wizard Ende**





Der Database Wizard

Eine sehr mächtige aber bislang nur wenig genutzte Fähigkeit des Namo Webeditors ist es Webseiten mit Datenbankanbindung zu erzeugen.

Zum einen liegt es daran, das nicht jeder über Webspaces mit Datenbank und serverseitiger Programmiersprache verfügt; ein ander Grund ist die nicht ganz einfache Einrichtung der offline Entwicklungsumgebung.

Exemplarisch will ich hier zeigen, wie man eine solche Umgebung auf einem Windows PC einrichtet.

Da der für Datenbankanwendungen erforderliche Webspaces überwiegend in einer Linux - Apache - MySQL - PHP (L A M P) Kombination angeboten wird, soll diese für die offline Entwicklung auf dem Windows PC nachgebildet werden. Dazu muss also unter Windows noch Apache, MySQL und PHP installiert werden (W A M P). Das manuell selbst zu machen, ist leider kein einfaches Unterfangen und an meinen eigenen, fehlgeschlagenen Versuch vor langer Zeit denke ich nur ungern zurück.

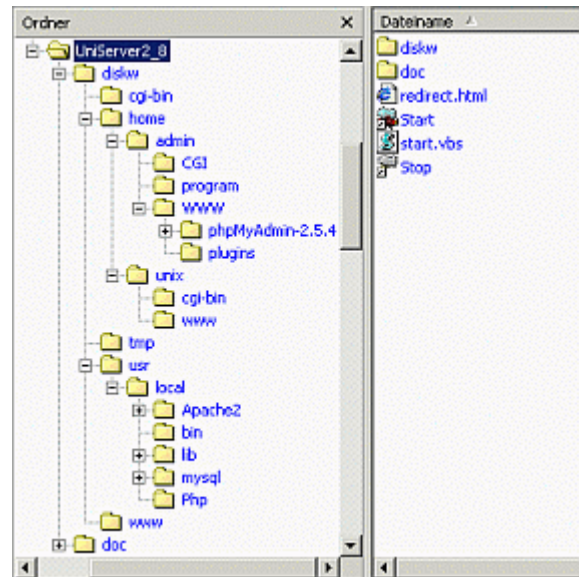
Glücklicherweise gibt es im Internet einige freie Softwarepakete, die einem diese Arbeit abnehmen. Die Bekanntesten sind **XAMPP**, **PHPTriad** und **FoxServe**. Das Umfangreichste ist XAMPP, welches nahezu jeglichen Bedarf stillen kann. Leider ist es auch sehr ressourcenhungrig. Die Festplatte wird schnell mit über 150 MB in Beschlag genommen und nach dem Starten der Anwendung sind 80 MB RAM Speicher verschwunden. Im übrigen dauert der Start auf meinem alten Laptop (166 MHz, 128MB RAM) ca. 3 Minuten.

Beim Wühlen in sourceforge.net bin ich dann durch Zufall auf **UniServ** gestoßen. Das Paket benötigt nur knapp 10 MB auf der Festplatte, belegt nach dem Starten 30 MB RAM und steht auf meinem Laptop nach 8 Sekunden zur Verfügung. So wie auch bei XAMPP wird das Softwarepaket einfach auf der Festplatte entpackt. Eine Installation wird nicht vorgenommen. Zum Deinstallieren reicht das Löschen des entsprechenden Verzeichnisses.



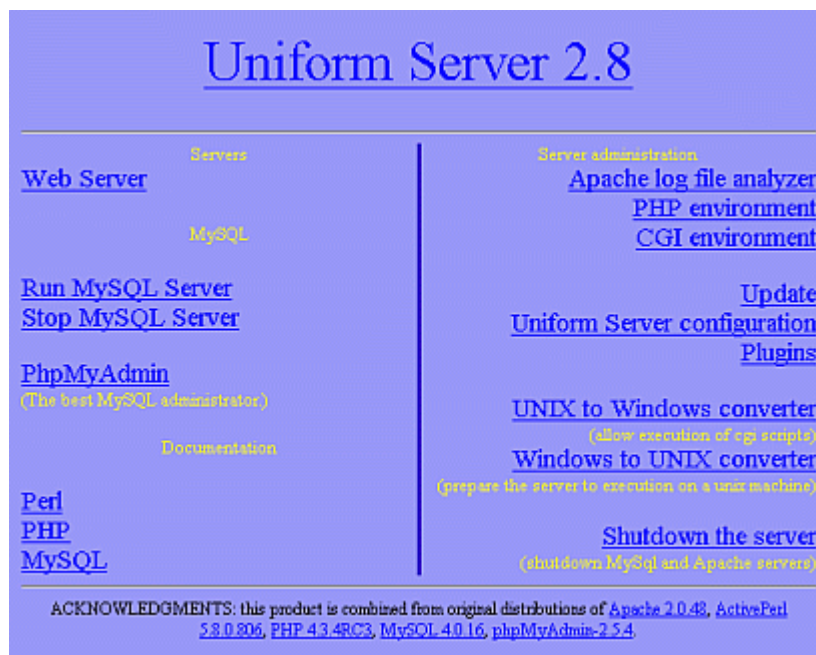


Nach dem Download von **UniServ** entpackt man das Paket z.B. nach "C:Program Files". Dort entsteht nun das Verzeichnis "Uniserver2_8" mit folgender Struktur:



Um UniServ zu starten macht man einen Doppelklick auf die "Start" Verknüpfung im "UniServ2_8" Hauptverzeichnis. Das DOS-Fenster anschließend minimieren - nicht schließen! Nun ist der Inhalt des Verzeichnisses "diskw" als neues Festplattenlaufwerk "W:" verfügbar.

Nun macht man eine Doppelklick auf "redirect.html". Daraufhin startet der Browser mit der Webseite. Apache läuft zu diesem Zeitpunkt bereits.



Nun klickt man noch auf den Link "Run MySQL Server" und anschließend auf "Back".

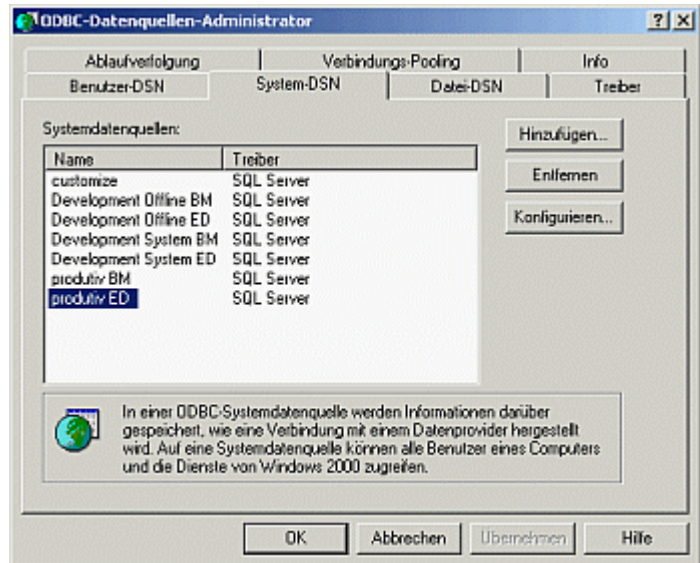
Ein Klick auf den Link "PhpMyAdmin" zeigt einem, das der Apache Server läuft, PHP ausgeführt wird und sich auch die Datenbank nutzen läßt. Jetzt sind alle notwendigen Module gestartet.



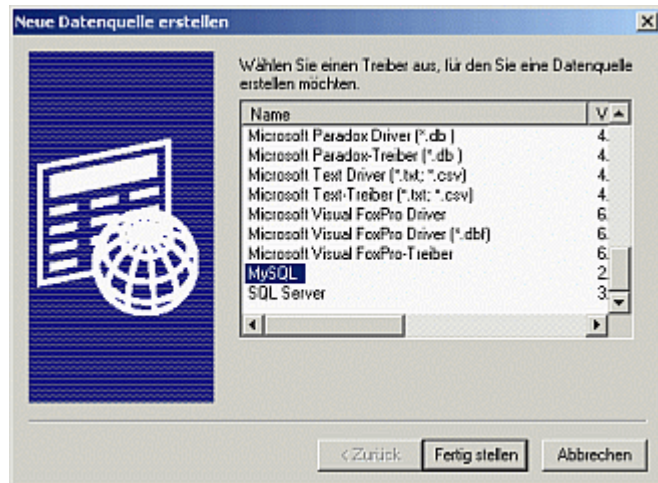


Der nächste wichtige Schritt ist, auch Namoden Zugriff auf die Datenbank zu ermöglichen. Leider unterstützt Namoden Datenbankzugriff nur per ODBC (Open DataBase Connectivity), MySQL benutzt aber TCP/IP als Übertragungsprotokoll. Das ist aber kein großes Problem. Man kann in Windows einen Treiber installieren, der diese beiden Protokolle miteinander verbindet. Dazu lädt man von www.mysql.com den für sein Betriebssystem passenden Treiber (Connector/ODBC 3.51 - MySQL ODBC driver -) und installiert ihn.

Als nächstes muss man in Windows die benötigten Verbindungen anlegen. Dazu öffnet man unter "Start - Einstellungen - Systemsteuerung - Verwaltung" den Punkt "Datenquellen (ODBC)".



Dort wählt man den Reiter "System_DSN", klickt auf den Button "Hinzufügen" und wählt MySQL.





In der nun erscheinender Maske trägt man unter "Data Source Name:" und "Database Name:" den Namen der gewünschten Datenbank ein. "Description:" kann einen erklärenden Eintrag enthalten.

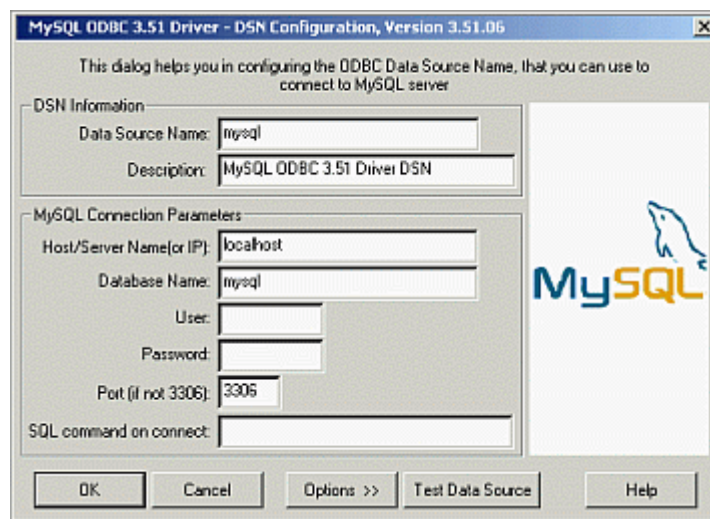
"Host/Server Name (or IP)" erhält "localhost".

"User:" und "Passwort:" lassen wir zunächst offen.

Den Eintrag "Port (if not 3306):" kann man offen lassen. Ich trage aber immer "3306" ein.

Durch einen Klick auf "OK" speichert man die Angaben.

Nun ist diese Verbindungsschnittstelle zwischen MySQL und dem ODBC Treiber definiert und kann genutzt werden.

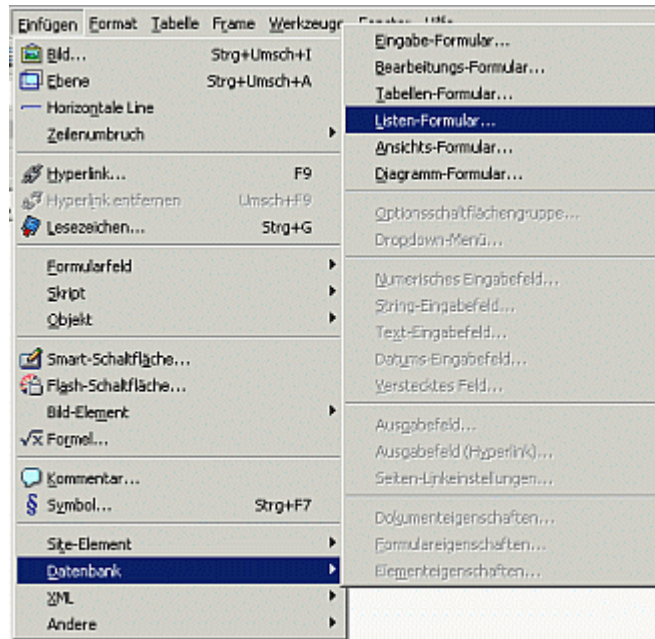




Das testen wir in Namo.

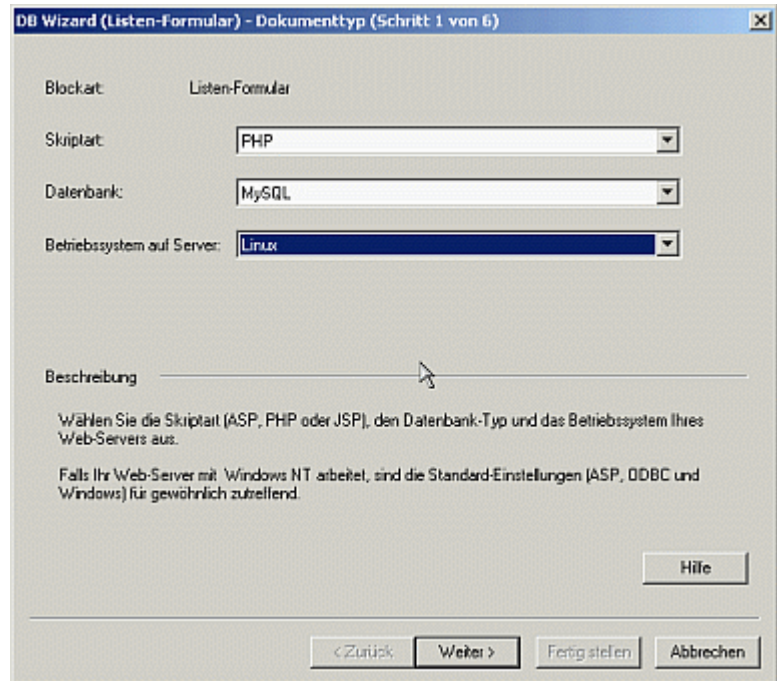
Dazu öffnen wir Namo und legen eine neue, leere Seite an. Nun fügen wir ein Listen-Formular ein.

Dieses Formular fragt die Datenbank nur ab und zeigt die gewünschten Werte an.



In der folgenden Dialogbox werden nun die Parameter für den **Webserver** abgefragt, auf dem die Seite später zum Einsatz kommen soll.

Das wird meistens PHP, MySQL und Linux sein.





In der nächsten Dialogbox werden wieder Parameter für den **Webserver** abgefragt.

Wenn der Provider eine Standardinstallation gewählt hat sind folgende Werte einzutragen:
 "MySQL Host:" "localhost:3306" und
 "MySQL DB Name:" der Name der Datenbank.
 Unter "ODBC DSN für Vorschau:" wählen wir den zuvor festgelegten Namen für diese Verbindung.
 Hier "mysql".
 "Benutzer-ID:" wird "root".
 "Passwort:" lassen wir zunächst offen.

Nach einem Klick auf "Weiter>" erscheint nun zum ersten mal ein Datenbankinhalt.

Hier der Inhalt der Datenbank "mysql", die man ansonsten unangetastet lassen sollte, da dort alle Verwaltungsinformationen der Datenbank abgelegt werden.

Verändert man die Inhalte der Tabellen, kann es sein, das alle Datenbanken, die durch MySQL verwaltet werden zerstört werden!

Darum greifen wir nur lesend darauf zu.





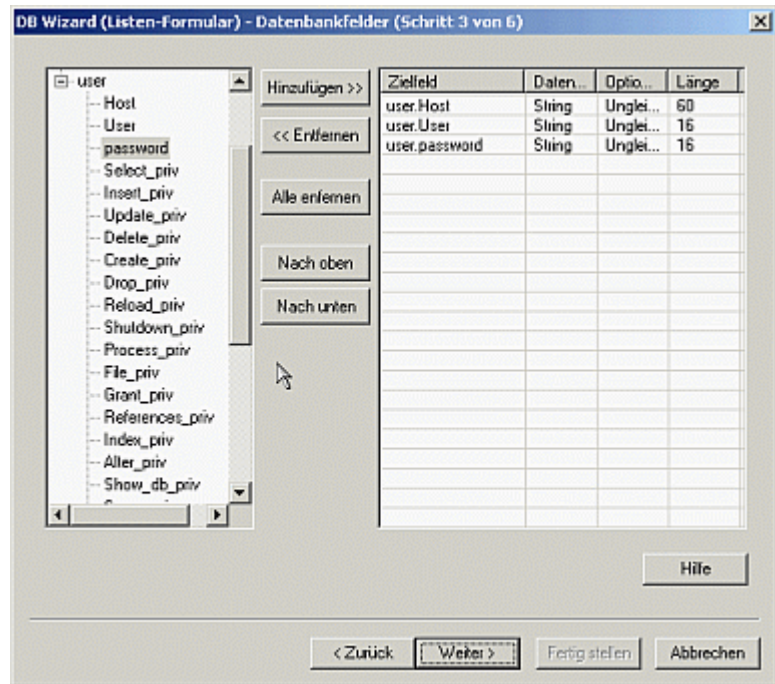
Hier sehen wir nun alle Tabellen, die in der Datenbank "mysql" verfügbar sind.

Für Namo muss man aber einzelne Felder aus den Tabellen für die Anzeige auswählen.

Dazu weitet man z.B. die Tabelle "User" aus und bringt hier die ersten 3 Felder durch einzelne Klicks auf "Hinzufügen>>" in die Auswahl.

Die Buttons "Nach oben" und "Nach unten" bieten sich zum Umsortieren der Liste an, aber das kann man sich sparen.

Die einzelnen Felder lassen sich später im Webeditor markieren und nach Bedarf verschieben.



Durch "Weiter>" übernehmen wir alles und kommen nun zum einzig komplizierten Teil der Datenabfragedefinition.



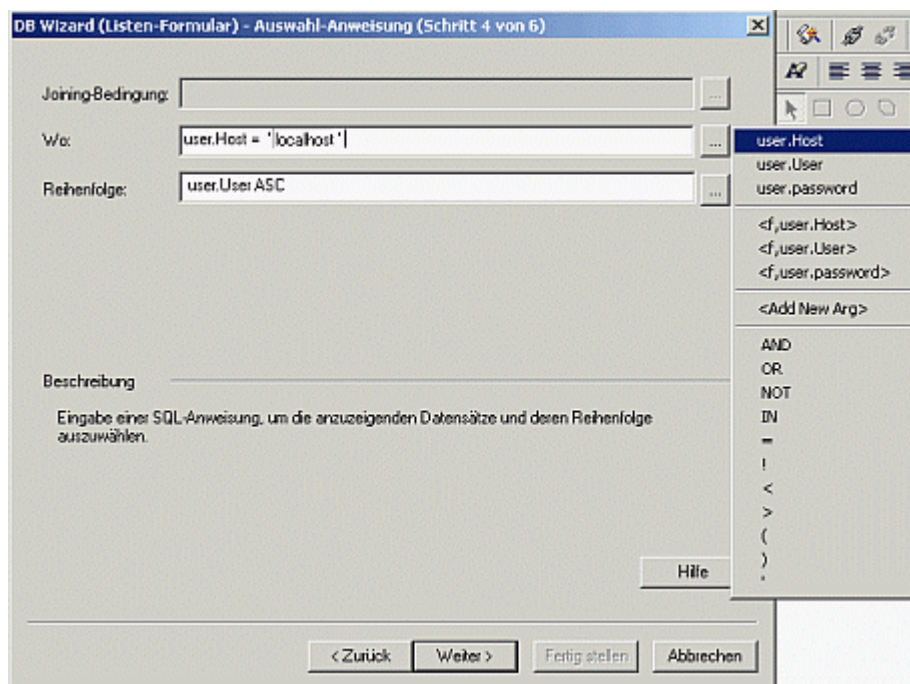


Um diese Felder richtig verstehen zu können, sollte man Datenbankwissen haben und idealerweise auch SQL-Queries definieren können. Einfache Sachen kann man aber auch mit Hilfe von Namo erledigen.

Die "Joining-Bedingung:" ist in diesem Beispiel nicht definierbar. Sie wird nur benötigt, wenn mehrere Tabellen anhand von Bedingungen verknüpft werden sollen. Wird nur eine Tabelle verwendet (hier "user") bleibt sie leer.

Das "Wo:" Feld entspricht der "Where"-Klausel in SQL-Queries und ist meiner Meinung nach schlecht übersetzt. Man hätte "Where" lassen sollen. Hier kann man die Datenauswahl durch Bedingungen einschränken. Man könnte hier z.B. festlegen, das nur die User angezeigt werden, die für den Host "localhost" definiert sind (andere existieren im Moment sowieso noch nicht). Dazu klickt man auf den "..." Button rechts neben dem Eingabefeld und klickt sich die Definition zusammen. Wer sicher genug ist, kann die Klausel auch direkt per Hand schreiben. Manche erweiterte Möglichkeiten lassen sich nur per Hand eingeben. Immerhin steht der gesamte Befehlsumfang von MySQL zur Verfügung.

Unter "Reihenfolge:" kann man die Anzeige der Werte sortieren. Dazu wählt man zunächst ein Feld aus und legt dann "ASC" (aufsteigend) oder "DESC" (absteigend) fest.



Für unsere erste Seite lassen wir aber alle Felder leer, da es nur einen Datensatz gibt.





Nun kann man endlich auf "Fertigstellen" klicken und hat damit seine erste Datenbankabfrage erzeugt. Sie ist nicht sonderlich schön, aber für das Design ist man schließlich selbst zuständig.

user.Host	[val]
user.User	[val]
user.password	[val]
[<<] [1 2 3 4 5 6 7 8 9 10] [>>]	

Auf der linken Seite der Tabelle befinden sich nun die ausgewählten Feldnamen. Das ist lediglich statischer Text, der bedenkenlos geändert werden kann. In dem rechts dazu gehörendem Feld befindet sich die Variable, die später durch die Datenbankwerte ersetzt wird. Diese Tabelle kann nun nach Belieben formatiert werden. Es können Zellen gelöscht oder hinzugefügt werden. Auch die Variablenfelder können verschoben werden ABER dürfen diese Tabelle NICHT verlassen!

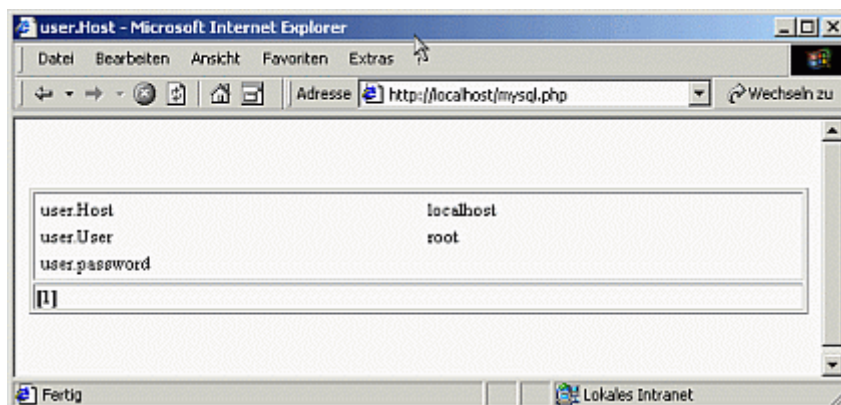
Eingerahmt wird diese Tabelle durch eine weitere (blaue Box), die die Seitennavigation erzeugt. Die innere Tabelle darf diese auch nicht verlassen! Ansonsten kann man sich, was das Design angeht, frei entfalten.

Nun will man sich das Ergebnis natürlich auch ansehen und kontrollieren, ob es arbeitet.

Dazu ist es wichtig zu wissen, dass diese Seite nun im Einzugsbereich des Apache-Servers abgespeichert werden muss. Im Fall von UniServ ist das "W:\www...". Dieses Verzeichnis entspricht "localhost". Nur dort lassen sich die von Namo erzeugten PHP-Skripte ausführen. Also speichern wir die Seite als "mysql.php" unter "W:\www" ab. Die Dateiendung ".php" ist sehr wichtig, weil sie den Apache-Server anweist die Seite zu parsen und die PHP-Skripte Serverseitig auszuführen.

Bei XAMPP liegt dieses Verzeichnis unter "C:\wampp2\htdocs".

Nach dem Abspeichern kann man die Seite nun im Browser aufrufen. Das kann aber nicht durch einen Doppelklick im Explorer erfolgen. Dieser Aufruf läuft am Apache-Server vorbei. Man muss zunächst den Browser öffnen. Nun gibt man in der Adresszeile "http://localhost/" ein und drückt "Enter". Die angezeigte Seite wird nun schon von Apache geliefert. "http://localhost/mysql.php" bringt unsere erste Seite zur Anzeige.





Nachdem nun alles zufriedenstellend arbeitet, muss noch etwas Finetuning vorgenommen werden. Der Connect zur Datenbank erfolgt nämlich als "root" ohne Passwort und das kann nicht so bleiben. Auf dem Webserver herrschen später andere Bedingungen. Dort lautet der Username für die Datenbank z.B. "meier" und das Passwort ist zwingend notwendig und lautet z.B. "huskyschnauze". Würden wir diese Parameter in Namo eingeben, würden wir aber von Namo aus keinen Zugriff auf die Datenbank erhalten.

Beim späteren Hochladen auf den Webserver wäre es notwendig in allen Quelltexten mit einem Texteditor die Accountdaten anzupassen.

```
wed_read_list_process ("localhost:3306", "mysql", "root", "", $w_sqlstr, $w_record,
$w_disprovs, $a_startrow, $a_maxrows, $w_rows);
```

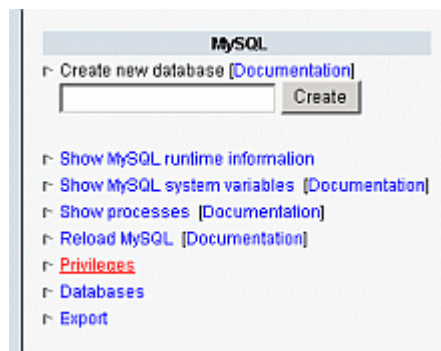
müsste auf

```
wed_read_list_process ("localhost:3306", "mysql", "meier", "huskyschnauze", $w_sqlstr,
$w_record, $w_disprovs, $a_startrow, $a_maxrows, $w_rows);
```

geändert werden. Das ist lästig und Fehler anfällig.

Also passen wir die MySQL Installation an. Dazu öffnen wir noch mal "http://localhost/" und klicken dort auf den Link "administration page", anschließend auf "PhpMyAdmin".

Mit phpMyAdmin wird nun unter "Privileges" für den Benutzer "root@localhost" ein Passwort vergeben. Z.B. "masterpass" (Klick auf "Edit").



Nun funktioniert PhpMyAdmin nicht mehr, da sich PHP ohne Passwort angemeldet hat.

```
Error
MySQL said:
#1045 - Access denied for user: 'root@localhost' (Using password: NO)
```





Deshalb muss man in der Datei "C:\Program Files\UniServer2_8\diskw\home\admin\WWW\phpMyAdmin-2.5.4\config.inc.php" in der Zeile 81 das Passwort für "root" eingetragen werden.

```

$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'masterpass';
    
```

Anschließend kann man wieder auf den "Home" Link auf der linken Seite klicken und PhpMyAdmin arbeitet wieder.

Jetzt kann man auf der Startseite von phpMyAdmin eine neue User Datenbank anlegen. Als Beispiel eine Datenbank für ein Gästebuch.

The image shows two screenshots from the phpMyAdmin interface. The left screenshot shows the 'MySQL' section with a 'Create new database' form where 'gaestebuch' is entered. The right screenshot shows the 'Structure' tab for the 'gaestebuch' database, where a new table 'eintraege' is being created with 5 fields.

Database gaestebuch - Table eintraege running on localhost

Field	Type [Documentation]	Length/Values*	Attributes	Null	Default**	Extra	Collation
id	INT			not null		auto_increment	
name	VARCHAR	30		not null			
email	VARCHAR	30		null			
homepage	VARCHAR	50		null			
text	TINYTEXT			not null			

Table comments :
 Table type :

Die Parameter wie abgebildet einstellen. Zusätzlich, nicht im Bild zu sehen, für das erste Feld ganz rechts "Primary" aktivieren. Damit wird das Feld "id" als Primärschlüssel festgelegt. Das die Felder "email" und "homepage" "null" sein dürfen ist wichtig. Überall wo "not null" gewählt wird, muss in Eingabefeldern zwingend ein Wert eingegeben werden. Ansonsten verweigern die von Namu erzeugten Skripte die Weiterverarbeitung. So dürfen die Felder "email" und "homepage" leer bleiben. Ein Klick auf "Save" legt dann die gewünschten Felder an.





Database gaestebuch - Table eintraege running on localhost

Table eintraege has been created.

SQL-query: [\[Edit\]](#) [\[Create PHP Code\]](#)

```
CREATE TABLE `eintraege` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(30) NOT NULL,
  `email` VARCHAR(30),
  `homepage` VARCHAR(50),
  `text` TINYTEXT NOT NULL,
  PRIMARY KEY (`id`)
);
```

Structure Browse SQL Search Insert Export Operations Empty

Field	Type	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> id	int(11)		No		auto_increment	
<input type="checkbox"/> name	varchar(30)		No			
<input type="checkbox"/> email	varchar(30)		Yes	NULL		
<input type="checkbox"/> homepage	varchar(50)		Yes	NULL		
<input type="checkbox"/> text	tinytext		No			

Check All / Uncheck All With selected:

Nun legen wir den für den Webserverzugriff notwendigen Useraccount an. Dazu rufen wir in PhpMyAdmin erneut links "Home" und dann mittig "Privileges" auf, klicken auf "Add a new User" und geben die Werte für "User name:" = "meier", "Host:" = "localhost" und "Password:" = "huskyschnauze" ein. Ganz unten nun auf "GO" klicken.

Nun wählt man die gewünschte Datenbank aus (Home - Pulldownmenü -> gaestebuch) und öffnen mit einem Klick auf den Reiter "SQL" das Eingabefenster. Dort vergibt man nun die Rechte auf die Datenbank durch Eingabe von:

```
grant all privileges on gaestebuch.* to meier@localhost identified by 'huskyschnauze';
```

Ein Klick auf "GO" richtet alles ein und beendet die Prozedur.

Nun kann man über den lokalen Webserver genauso wie über den Richtigen arbeiten und in Namu mit den richtigen Accountdaten entwickeln.

Unsere "mysql.php" arbeitet nun nicht mehr, wie ein testweiser Aufruf im Browser zeigt, weil dem Script das notwendige Password fehlt. Das korrigieren wir nun nachträglich.

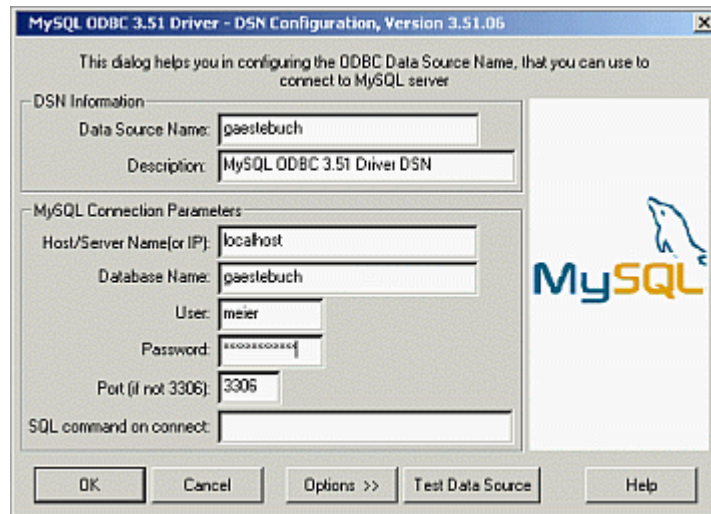
Wir öffnen in Namu "mysql.php". Mit einem Rechtsklick in die Datentabelle und "Datenbank - Formulareigenschaften" öffnen sich die Eigenschaften der Listentabelle. Der Reiter "Datenquelle" zeigt die Verbindungsdaten an, die wir nun um das Passwort "masterpass" ergänzen. Nach dem Abspeichern arbeitet "mysql.php" wieder ordentlich und zeigt uns auch den neuen Useraccount "meier" an.





Als nächstes erstellen wir nun ein Eingabeformular, um Daten in die Datenbank eingeben zu können.

Zuerst muss wieder unter "Start - Einstellungen - Systemsteuerung - Verwaltung - Datenquellen (ODBC)" ein neuer Verbindungseintrag gemacht werden. "System DSN - Hinzufügen - MySQL"



Jetzt öffnen wir in Namo wieder eine neue, leere Seite und klicken im Menü auf "Einfügen - Datenbank - Eingabeformular".

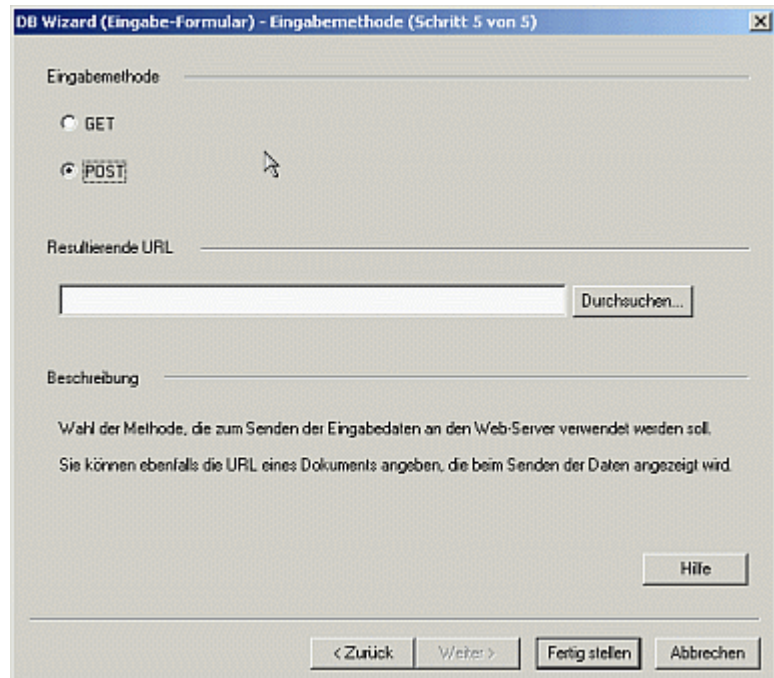
Nun wieder "PHP - MySQL - Linux" "Weiter>"



Nun alle Felder der Tabelle hinzufügen und zwei mal weiter klicken.

Dort nun "POST" auswählen und "Resultierende URL". Mit der URL ist die Seite gemeint, die nach dem Abschicken der Daten angezeigt wird. Hier sollte die Gästebuchausgabe stehen. Da die Seite noch nicht existiert, lassen wir das Feld leer.

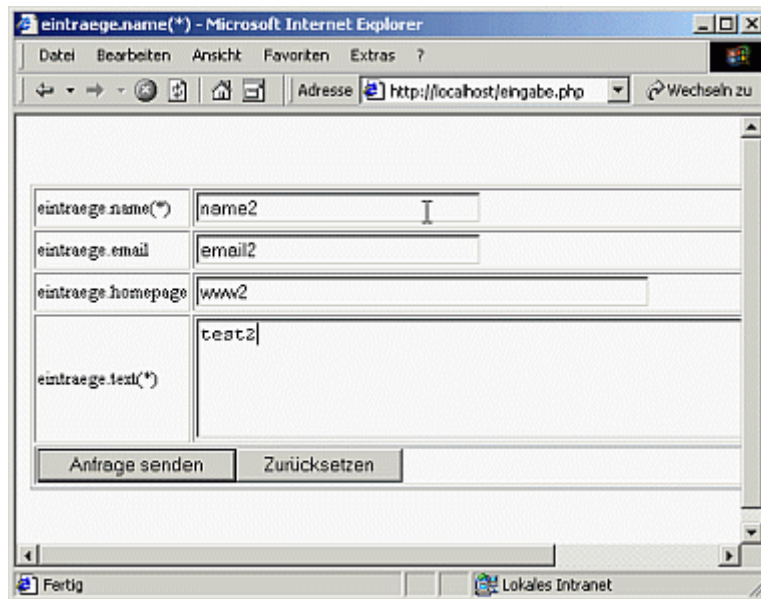
Mit einem Rechtsklick in die Eingabetabelle "Datenbank - Formulareigenschaften - Eingabe-Einstellungen" kann das später nachgeholt werden.



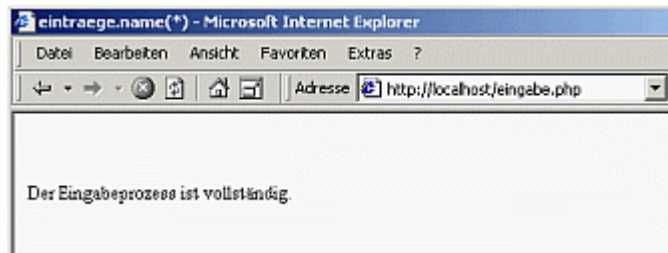


Eine wichtige Bemerkung ist noch zu machen. Da das Feld "id" als Primärschlüssel mit "auto_increment" festgelegt wurde, sollte man das Feld aus der Eingabemaske löschen, damit dort keine fehlerhaften Werte eingegeben werden können.

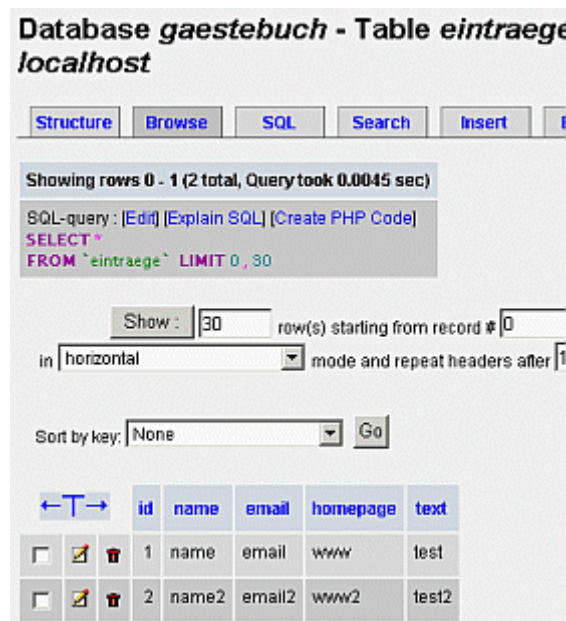
Nun noch unter "W:\www" als "eingabe.php" abspeichern, im Browser aufrufen und ausfüllen.



Da noch keine "Resultierende URL" angegeben wurde, erscheint im Browserfenster folgendes:



Nach dem "Anfrage senden" kann man den erfolgreichen Eintrag in die Datenbank mit PhpMyAdmin prüfen.





Damit wären die grundlegendsten Funktionen und Einstellungen für die Arbeit mit dem Datenbankwizard erklärt und ich wünsche frohes Schaffen.

Um UniServ zu beenden stoppt man Apache und MySQL über das Webinterface von Uniserv rechts unten "Shutdown the server". Der Browser kann nun geschlossen werden.

Uniform Server 2.8

Servers

- [Web Server](#)
- [MySQL](#)
- [Run MySQL Server](#)
- [Stop MySQL Server](#)
- [PhpMyAdmin](#)
(The best MySQL administrator)
- [Documentation](#)
- [Perl](#)
- [PHP](#)
- [MySQL](#)

Server administration

- [Apache log file analyzer](#)
- [PHP environment](#)
- [CGI environment](#)
- [Update](#)
- [Uniform Server configuration](#)
- [Plugins](#)
- [UNIX to Windows converter](#)
(allow execution of cgi scripts)
- [Windows to UNIX converter](#)
(prepare the server to execution on a unix machine)
- [Shutdown the server](#)
(shutdown MySql and Apache servers)

ACKNOWLEDGMENTS: this product is combined from original distributions of Apache 2.0.48, ActivePerl 5.8.0.806, PHP 4.3.4RC3, MySQL 4.0.16, phpMyAdmin-2.5.4.

Fragen zur Arbeit mit diesem Thema, bitte ich im Forum zu stellen.

