



acwww - Eine WWW-Schnittstelle für allegro-Datenbanken

Version 2.5
Dierk Höppner

Inhalt

- 1 Einleitung
- 2 Eigenschaften der Schnittstelle
- 3 Notwendige Komponenten
- 4 Installation für den schnellen Einstieg
 - 4.1 Die Zutaten
 - 4.2 Was ist bei der Installation zu beachten?
 - 4.2.1 Der Windows-Rechner
 - 4.2.2 Xitami
 - 4.2.3 *avanti*
 - 4.2.4 Perl
 - 4.2.5 Scripte und andere Dateien der acwww-Schnittstelle
 - 4.3 Anbindung der eigenen Datenbank
 - 4.3.1 Ihre Datenbank benötigt einen symbolischen Namen
 - 4.3.2 Ihre Datenbank bei *avanti* bekannt machen
 - 4.3.3 Kopieren wichtiger Parameterdateien
 - 4.3.4 Bekanntmachen der Datenbank in den Programmen der WWW-Schnittstelle
 - 4.3.4.1 Änderungen in *acwww.pl*
 - 4.3.4.2 Anlegen der datenbankspezifischen *dbinfo.pl*
 - 4.3.5 Testen
 - 4.4 Wie geht's weiter?
- 5 Die Anbindung einer *allegro*-Datenbank an das WWW schematisch
- 6 Grundsätzliche Hinweise zur Installation
 - 6.1 *avanti*
 - 6.1.1 Windows
 - 6.1.2 UNIX
 - 6.2 Perl
 - 6.3 acwww
 - 6.3.1 Windows
 - 6.3.2 UNIX
- 7 acwww25 - die Schnittstelle
 - 7.1 Grundsätzliche Arbeitsweise der Programme
 - 7.2 Die Programme im Einzelnen
 - 7.3 Anpassungen in den Programmen
 - 7.3.1 Allgemeines
 - 7.3.2 Änderungen in *acwww25.pl*
 - 7.3.2.1 Ort und Adresse von *avanti* (*\$AVANTI_HOST* und *\$AVANTI_PORT*)

- 7.3.2.2 Startverzeichnis fuer die Scripte (*\$HTMLScriptPfad*)
- 7.3.2.3 CGI-Pfad absolut (*\$AllegroW3CGIPfad*)
- 7.3.2.4 Relativer Pfad der datenbankspezifischen Scripte (*%dbscripte*)
- 7.3.2.5 Namen der anbietenden Organisation (*%Organisation*)
- 7.3.3 Änderungen in *dbinfo.pl*
 - 7.3.3.1 Konfigurationsbuchstabe *\$kl*
 - 7.3.3.2 Datenbankpfad *\$DBPfad*
 - 7.3.3.3 Datenbankname *\$DBName*
 - 7.3.3.4 *avanti*-Nutzername *\$user* und Passwort *\$passwd*
 - 7.3.3.5 Die Parameterdateien für die Ausgabe der Datensätze (*%Parameterdateien*)
 - 7.3.3.6 Sortierung der Anzeige (*\$SortChoice* und *%Sortierung*)
 - 7.3.3.7 Codierung von Sonderzeichen in der Ausgabe (*%Codierung*)
 - 7.3.3.8 Datenbankname für Überschriften (*\$DatenbankName*)
 - 7.3.3.9 Pfad der datenbankspezifischen Hilfsseiten (*\$HilfsseitenPfad*)
 - 7.3.3.10 Links auf andere Seiten (*%AndereDBs*)
 - 7.3.3.11 Namen der Register für Überschriften (*%RegisterHilfe*)
 - 7.3.3.12 Namen der Register für Anzeigezwecke (*%RegisterAnzeige*)
 - 7.3.3.13 Namen der Register fuer die Registerauswahl in den Suchmasken (*@RegisterPullDown*)
 - 7.3.3.14 Suche über logische Kombinationen (*\$logischeKombinationen*)
 - 7.3.3.15 Name der Variablen fuer die Suchmaske (*@MaskenVariable*)
 - 7.3.3.16 Zuordnung der Variablen zu den Registern der Datenbank (*%VarRegister*)
 - 7.3.3.17 Definition der Eingabefelder der Suchmaske (*%VarInputDef*)
 - 7.3.3.18 Schalter Kurztitelindex (*\$stl*)
- 7.3.4 Unterprogramme in *dbinfo.pl*
 - 7.4 Parametrierung
 - 7.5 Suchmasken ohne *maske.pl*
- 8 Der Standardeinstieg für den WWW-Benutzer
- 9 Abschließende Hinweise

1 Einleitung

Die neue Version der Schnittstelle zur Anbindung einer oder mehrerer *allegro-C*-Datenbanken an das World Wide Web wurde in einigen Punkten erweitert und ergänzt. Diese Schnittstelle bietet von der Oberfläche her nichts neues, dafür hat sich im Inneren einiges getan.

Die größte Änderung bezieht sich auf die Nutzung von *avanti* als Datenbankserver. Da viele darauf warten, wie man *avanti* für diese Zwecke einsetzen kann, habe ich zunächst die schon bestehende Schnittstelle so geändert, dass jetzt nicht mehr die Programme *SRCH* und *QRIX* zur Extraktion und Präsentation der Daten herangezogen werden, sondern es arbeitet jetzt nur einer: *avanti*.

Mit der jetzt vorgestellten Lösung sind aber noch lange nicht alle Möglichkeiten von *avanti* ausgeschöpft. Ich glaube aber dennoch, dass man einen guten Eindruck von den Einsatzmöglichkeiten des Servers bekommt. Selbstverständlich bleibt die Schnittstelle nicht auf dem hier vorgestellten Stand, sondern wird weiterentwickelt, um auch die anderen Fähigkeiten von *avanti* zu nutzen.

2 Eigenschaften der Schnittstelle

über diese Schnittstelle können Sie eine oder mehrere *allegro*-Datenbanken benutzerfreundlich im WWW anbieten. Der Benutzer hat dabei prinzipiell zwei Möglichkeiten der Recherche:

1. kann er einzelne Register ansteuern, die ähnlich aussehen, wie man es von den 'normalen' *allegro*-Programmen *APAC* und *PRESTO* her gewohnt ist. Die Einträge sind dabei als Hyperlinks ausgebildet. Das Anzeigen von Titelaufnahmen Verfolgen von Verweisen geschieht dabei ähnlich wie man es von *allegro* gewohnt ist.
2. kann man in einer selbst definierten Maske mehrere Suchbegriffe eingeben. Das System sucht dann alle passenden Datensätze dazu heraus.
3. Die Schnittstelle ist in Perl realisiert, so dass sie sich ohne Schwierigkeiten auf den unterschiedlichsten Rechner-Plattformen einsetzen läßt.

Die Anzeige ist natürlich über Parameterdateien voll parametrierbar.

3 Notwendige Komponenten

Sie benötigen

avanti

Der *avanti*-Server. Er ist Bestandteil der normalen Lizenz und findet sich auf der *allegro*-CD. Die UNIX-Versionen liegen auf dem FTP-Server.

eine Datenbank

werden Sie schon haben, denn sonst würden Sie sich kaum mit diesem Thema beschäftigen. Für die ersten Schritte benutzen wir aber zunächst die mit *avanti* mit installierte Datenbank *avdemo* *acwww25.exe*

ein selbstentpackendes Archiv, das die Scripte der WWW-Schnittstelle enthält. Sie finden es entweder im Verzeichnis *acwww* unterhalb des Installationsverzeichnisses von *avanti* oder im frei zugänglichen Verzeichnis */ANWENDER* auf unserem FTP-Server.

Perl

am besten die neuste Version. Perl gibt es am frischesten in den sogenannten CPAN-Archiven. In Deutschland ist das <ftp://ftp.rz.ruhr-uni-bochum.de> im Verzeichnis */pub/programming/languages/perl/CPAN/* Dort gibt es Versionen sowohl für UNIX, als auch für Win32 (NT/Win95). Alternativ gibt es von ActiveState eine spezielle Version für Windows.

Einen Webserver

Webserver gibt es in vielen Ausführungen für alle Betriebssysteme; teils kosten sie etwas, in der Mehrzahl sind sie aber kostenlos. Welchen Sie wirklich nehmen ist egal, einzige Voraussetzung ist, dass der Server über eine CGI-Schnittstelle verfügt.

... und

Phantasie und Programmierkenntnisse, wenn Sie die Schnittstelle und deren Ausgaben auf Ihre Verhältnisse anpassen wollen/müssen.

4 Installation für den schnellen Einstieg

Im Folgenden wird eine Installation beschrieben, mit der die Demo-Datenbank, die mit *avanti* mitgeliefert und installiert wird, im WWW oder Intranet über *acwww* zur Verfügung gestellt wird. Wer noch keine Erfahrung mit *allegro/avanti* oder dem Anbinden von Datenbanken an das WWW hat, sollte sich zunächst an das hier beschriebene Vorgehen halten und auch zunächst die beschriebenen Komponenten verwenden; man wird dann am ehesten eine erste funktionstüchtige Installation bekommen. Nachdem man die Zusammenhänge verstanden hat, kann man daran gehen,

die Installation auf seine eigenen Verhältnisse umzubiegen.

In der Beschreibung gehe ich davon aus, dass Sie alles auf einem einzigen Rechner installieren. Dadurch haben Sie alle Komponenten leichter im Griff. Damit spielen Sie sozusagen WWW auf einem Rechner.

4.1 Die Zutaten

Hier werden erstmal nur die Bestandteile aufgelistet. Was bei deren Installation zu beachten ist, folgt danach.

Ein Windows-Rechner.

Es muss mindestens Windows 95 sein.

Ein Webserver

Xitami, ein kleiner und schneller Webserver, der kostenlos ist. Sie finden ihn direkt bei Xitami oder auf der *allegro*-CD.

avanti

avanti gibt es ebenfalls auf der *allegro*-CD.

Webbrowser

Nicht das unwichtigste Programm in diesem System. Ohne einen Browser sehen Sie nichts. Oft ist Internet Explorer von Microsoft vorinstalliert. Sehr gut eignet sich aber auch Netscape. Aber ein Browser wird ja schon installiert sein, denn sonst würden Sie diesen Text nicht lesen können

4.2 Was ist bei der Installation zu beachten?

Sie sollten nicht alle Komponenten auf einmal installieren, sondern sich bei der Installation an der folgende Liste orientieren und schrittweise vorgehen. Die Liste gibt wichtige Hinweise und beschreibt Tests, mit denen Sie die Einzelkomponenten überprüfen können. Eine ähnliche Vorgehensweise empfehle ich Ihnen auch für die spätere Anbindung Ihrer eigenen Datenbank. Beachten Sie dabei aber bitte, dass ich nicht alle Eventualitäten vorhersehen kann. Falls es bei Ihnen an einer Stelle doch nicht klappen sollte, fragen Sie jemanden, der sich mit dem entsprechenden Programm auskennt.

4.2.1 Der Windows-Rechner

Dazu sage ich nicht allzu viel. Ich gehe davon aus, dass Sie über einen funktionierenden Windows-Rechner verfügen. Er ist an ein Netzwerk angeschlossen und das TCP/IP-Netzwerkprotokoll ist installiert.

4.2.2 Xitami

Den Webserver Xitami gibt es in mehreren Varianten für Windows: Als einfaches Konsolenprogramm, als normales Windows-Programm und als NT-Service. Die beiden ersteren laufen unter allen Windows-Versionen, das letzte nur unter NT. Für den Anfang genügt es, das normale Windows-Programm zu nehmen.

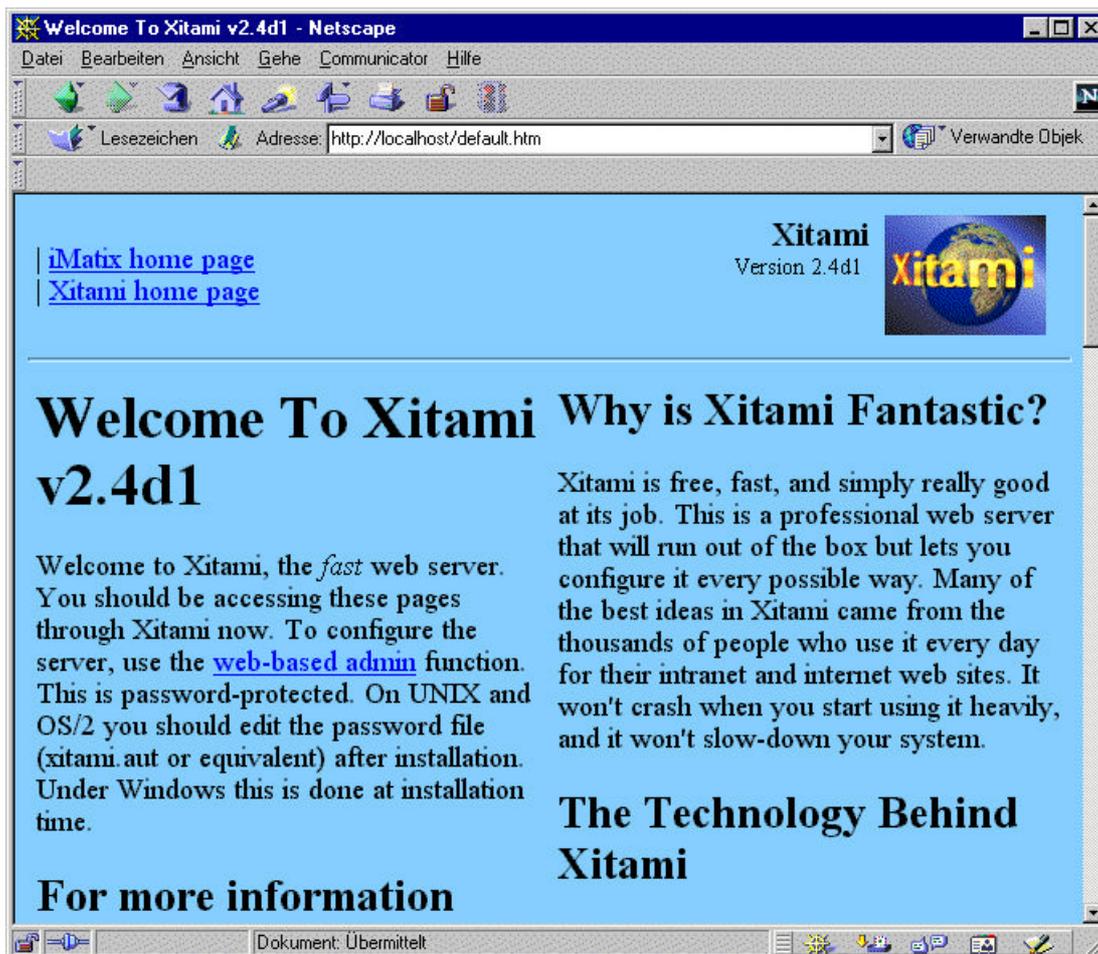
Die Installation ist denkbar einfach: Starten Sie einfach das heruntergeladene Programm. Es enthält ein Installationsprogramm, das alles weitere übernimmt. Folgen Sie den Anweisungen und übernehmen Sie einfach alle Vorgaben. Für das weitere Vorgehen nehme ich an, dass der Server in das Verzeichnis *c:\xitami* installiert wird. Es ist auch sinnvoll, den Server am Ende der Prozedur gleich zu starten. Das Installationsprogramm fragt Sie danach. Wir kommen nämlich dann wahrscheinlich gleich zum ersten Erfolgserlebnis.

Wenn die Installation erfolgreich gelaufen ist, und Xitami erfolgreich gestartet werden konnte, erscheint in der Taskleiste unten rechts das Xitami-Icon ähnlich wie es folgende Abbildung zeigt:



Hat das Xitami-Icon einen grünen Hintergrund ist der Server betriebsbereit und kann Anfragen entgegennehmen. Zeigt er 'Rot', ist er noch beschäftigt und findet wahrscheinlich keine korrekte Netzwerkinstallation vor. Kontrollieren Sie in diesem Fall zunächst Ihre Netzwerkinstallation. Möglich ist auch, dass der Port 80, auf dem er horchen möchte, durch ein anderes Programm belegt ist. Wahrscheinlicher Grund dafür: Sie arbeiten unter NT und der Internet Information Server (IIS) von Microsoft läuft und hat den Port 80 schon belegt. Sie sollten für einen erfolgreichen Test zunächst so verfahren: Deaktivieren Sie in Systemeinstellungen/Dienste den IIS und warten dann auf 'Grün' von Xitami. Wenn das nicht innerhalb einige Sekunden geschieht, stoppen Sie auch Xitami (mit der rechten Maustaste auf das Xitami-Icon klicken und 'Terminate' befehlen) und starten ihn anschließend wieder. Danach sollte er 'Grün' zeigen.

Jetzt können Sie Ihren Webbrowser starten. Im folgenden Bild sehen Sie den Anfangsbildschirm von Xitami. Tippen Sie in Ihrem Browser die im Adressfeld gezeigte Adresse ein und der Anfangsbildschirm erscheint.



Noch ein Hinweis: Sollte die Adresse 'localhost' nicht funktionieren, können Sie stattdessen auch die besondere IP-Adresse '127.0.0.1' eingeben. Damit funktioniert es auf jeden Fall. Auf jedem Rechner. Garantiert. Mit dieser IP-Adresse spricht man nämlich **immer** den eigenen Rechner an.

Wenn Sie den soeben installierten Webserver von einem anderen Rechner in Ihrem Netzwerk ansprechen wollen, müssen Sie die richtige IP-Adresse Ihres Rechners kennen. Die erfahren Sie aus dem Netzwerk-Applet in den Systemeinstellungen. Sollte dort eingestellt sein, dass Ihr Rechner seine IP-Adresse automatisch bezieht, fragen Sie Ihren Systemverwalter danach.

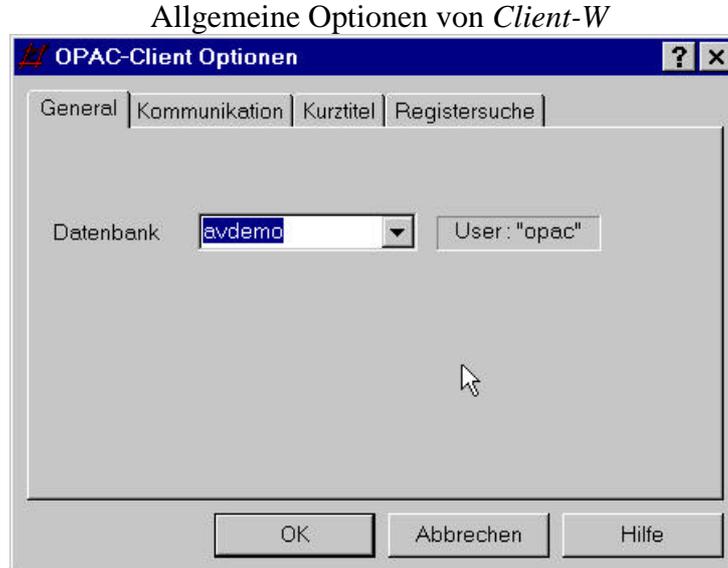
4.2.3 *avanti*

avanti finden Sie auf der *allegro*-CD. Mit unserem speziellen Zugriffsprogramm stellen Sie sich zunächst die Installationsdatei her. Dazu benötigen Sie Ihren Installationsschlüssel, der mit der letzten Rechnung mitgekommen ist. Das Programm, das Sie nach dem Kopiervorgang auf Ihrer Festplatte finden werden, ist ebenfalls ein Installationsprogramm. Sie starten es einfach und übernehmen wieder die vorgeschlagenen Einstellungen.

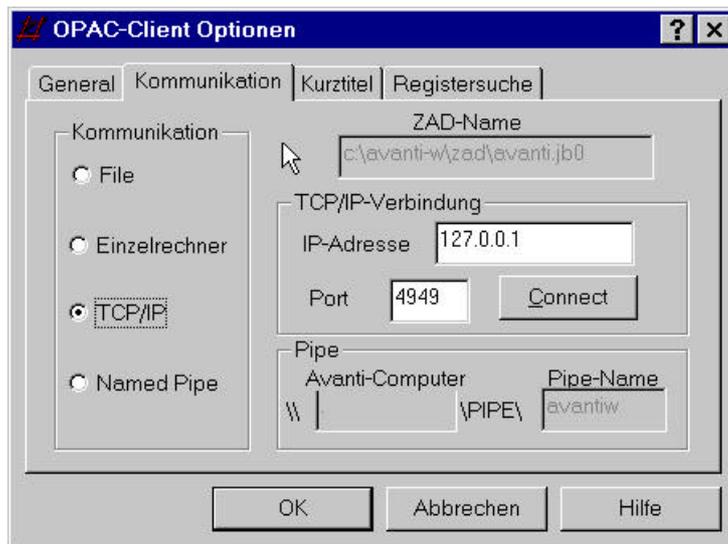
avanti wird nicht automatisch gestartet, sondern Sie müssen es über das Startmenü starten; bei der Installation wird eine Programmgruppe 'allegro-C' eingerichtet; dort finden Sie auch einen Eintrag für *avanti*. In der Taskleiste unten rechts erscheint nach erfolgtem Start das *allegro*-Icon:



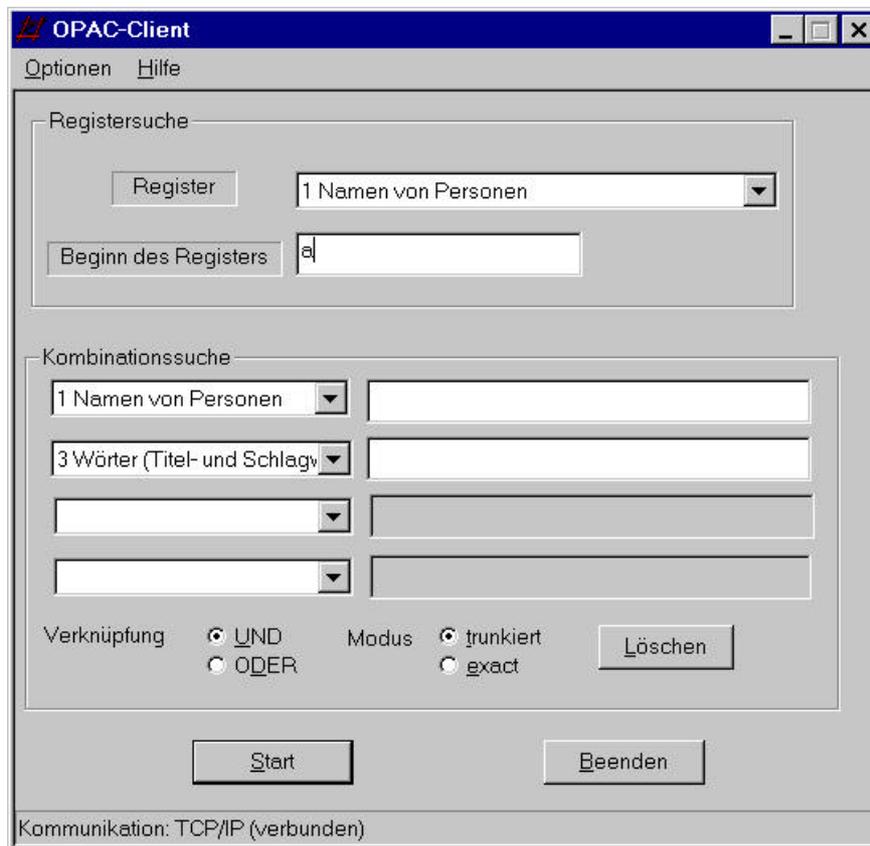
Test, ob *avanti* richtig arbeitet: Mit *avanti* wird ein einfacher Client für *avanti*, *Client-W*, mit installiert. Starten Sie den Client und vergewissern Sie sich zunächst, dass die Einstellungen mit denen auf den folgenden beiden Bildern übereinstimmen:



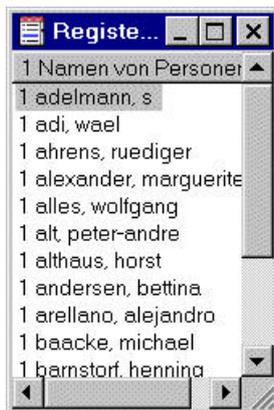
TCP/IP-Kommunikation mit *Client-W*



Zum Schluß schließen Sie den Dialog und tragen im Hauptbildschirm unter 'Registersuche' im Feld 'Beginn des Registers' einfach ein 'a' ein (nächstes Bild)



und drücken dann 'Start'. Es sollte dann ein Registerausschnitt der mit *avanti* mitgelieferten Demo-Datenbank zu sehen sein:



Damit läuft *avanti* korrekt und wartet auf neue Aufgaben.

4.2.4 Perl

Perl wird ebenfalls mittels eines Installationsprogramms installiert. Der genaue Ort der Installation ist eigentlich nebensächlich, im Folgenden setze ich aber eine Standardinstallation nach *c:\perl* voraus. Sie sollten dann noch sicherstellen, dass das *bin*-Verzeichnis von Perl mit im Suchpfad (Umgebungsvariable *PATH*) erscheint. Wenn Sie dieses Verzeichnis nach einem Neustart des Rechners nicht im Suchpfad finden, müssen Sie es von Hand eintragen. Unter Windows 95 und 98 ergänzt man die *PATH*-Angabe in *c:\autoexec.bat*, unter Windows NT stellt man den Pfad im System-Applet in der Systemsteuerung ein.

Test von Perl und *avanti*: Im Verzeichnis *avanti-W\client* gibt es ein minimales Perl-Programm, mit dem man das Zusammenspiel von Perl und *avanti* testen kann. Es heißt *tcp-ip.pl* und zeigt beispielhaft, wie man aus einem Perl-Programm heraus mit *avanti* 'spricht'. (Es eignet sich übrigens sehr gut zum Testen selbst entwickelter *avanti*-Jobs.) Zum Testen öffnen sie eine MS-DOS-Eingabeaufforderung und wechseln nötigenfalls in das Verzeichnis. Den Aufruf des Programms und Ergebnis zeigt das folgende Bild:

```

D:\WINNT\System32\cmd.exe
C:\avanti-w\client>perl tcp-ip.pl
1      adelman, s      797
1      adi, wael      126
1      ahrens, ruediger      73
1      alexander, marguerite      45
1      alles, wolfgang      109
1      alt, peter-andre      830
1      althaus, horst      855
1      andersen, bettina      122
1      arellano, alejandro      123
1      baacke, michael      124

C:\avanti-w\client>

```

Das Ergebnis muss sehr schnell kommen. Sollte es auf sich warten lassen oder gar nicht erscheinen, stimmt entweder Ihre *avanti*-Installation nicht, oder Perl ist nicht korrekt installiert.

4.2.5 Scripte und andere Dateien der acwww-Schnittstelle

Die meisten Webserver stellen bei einer Standardinstallation mehrere Verzeichnisse zur Verfügung, in denen einmal pogrammeigene Informationen zu finden sind, wo aber auch Benutzerdaten untergebracht werden. Xitami ist da keine Ausnahme. Die Verzeichnisse befinden sich unterhalb des Installationsverzeichnisses von Xitami, also in unserem Fall unterhalb von *c:\xitami*. Zwei sind dabei für uns besonders interessant:

1. *cgi-bin*

In diesem Verzeichnis und darunter erwartet der Server Programme, die er über die sogenannte CGI-Schnittstelle startet. (CGI = Common Gateway Interface) Die dort installierten Programme werden bei Bedarf vom Webserver gestartet und erhalten ihre Start-Optionen in bestimmten definierten Umgebungsvariablen. Sie erledigen ihre Arbeit und liegen meistens als Ergebnis eine HTML-Datei an den Server zurück, welcher sie dann an den Browser schickt. Näheres zur Nutzung dieser Schnittstelle finden Sie in den Dokumentationen des Webserver oder in einem der mannigfaltigen Bücher zu diesem Thema.

2. Das Dokumentenverzeichnis

Im Dokumentenverzeichnis und darunter werden Ihre HTML-Seiten abgelegt. Bei Xitami heißt es *webpages*, bei anderen *htdocs* oder *docs*.

In beide Verzeichnisse gehören Teile der acwww-Schnittstelle.

Im Verzeichnis *avanti-w/acwww* finden Sie nach der Installation von *avanti* das Programm

acwww25.exe, ein sich selbst extrahierendes Archiv, und *acwww25.htm*, diese Dokumentation. Verfahren Sie jetzt wie folgt:

- Starten Sie den Windows-Explorer oder ein anderes Dateiverwaltungsprogramm und wechseln in das Verzeichnis *c:\avanti-w\acwww*
- starten Sie mit einem Doppelklick das Programm *acwww25.exe*. Es flackert kurz eine DOS-Box auf, in der Entpackvorgang zu sehen ist. Sollte die DOS-Box nicht von selbst verschwinden, beenden Sie sie es.
- Es entsteht ein Verzeichnis *httpd*, in dem Sie wieder nur zwei Verzeichnisse finden: *cgi-bin* und *htdocs*. Darunter finden Sie die Dateien und weitere Verzeichnisse der *acwww*-Schnittstelle. Kopieren sie nun alles aus *c:\avanti-w\acwww\httpd\cgi-bin* in das CGI-Verzeichnis von Xitami (*c:\xitami\cgi-bin*) und alles aus *c:\avanti-w\acwww\httpd\htdocs* in das Dokumentenverzeichnis von Xitami (*c:\xitami\webpages*). Achten Sie darauf, dass alle Dateien und alle Verzeichnisse mit kopiert werden. Es soll im Xitami-Verzeichnis folgende Verzeichnisstruktur entstehen (weitere Verzeichnisse, die bei der Installation von Xitami entstanden sind, werden in der Übersicht nicht gezeigt.):

```

c:\xitami
|
+-- cgi-bin
|   |
|   +----- acwww25
|           |
|           +-- lib
|           +-- opac
|           +-- texts
|
+-- webpages
|   |
|   +----- acwww25
|           |
|           +-- opac
|
+----- images

```

- Im Verzeichnis *c:\xitami\cgi-bin\acwww25\lib* befinden sich zwei Dateien (*acwww25.pl* und *cgi-lib.pl*), die beide in das Library-Verzeichnis von Perl (*c:\perl\lib*) kopiert werden müssen.

Das war's. Wenn Webserver und *avanti* laufen können Sie Ihren Browser starten und diese Adresse eingeben oder hier klicken:

<http://127.0.0.1/cgi-bin/acwww25/maske.pl?db=avdemo>

Es sollte dann dieses Bild erscheinen:



Damit ist schon mal die Demo-Datenbank im Web zu finden und damit die Hauptarbeit getan. Was noch zu tun bleibt, ist die eigene Datenbank anzubinden.

4.3 Anbindung der eigenen Datenbank

Am einfachsten ist es, wenn Ihre Datenbank auch mit dem Standardschema aufgebaut ist. In der folgenden Beschreibung wird davon ausgegangen, dass Ihre Datenbank auf dem Braunschweiger Standardschema beruht. Wenn Ihre Datenbank auf einem anderen Schema beruht, müssen Sie sich zusätzlich mit den Hinweisen zur Parametrierung beschäftigen. Jetzt erstmal die Anleitung für die einfache Version:

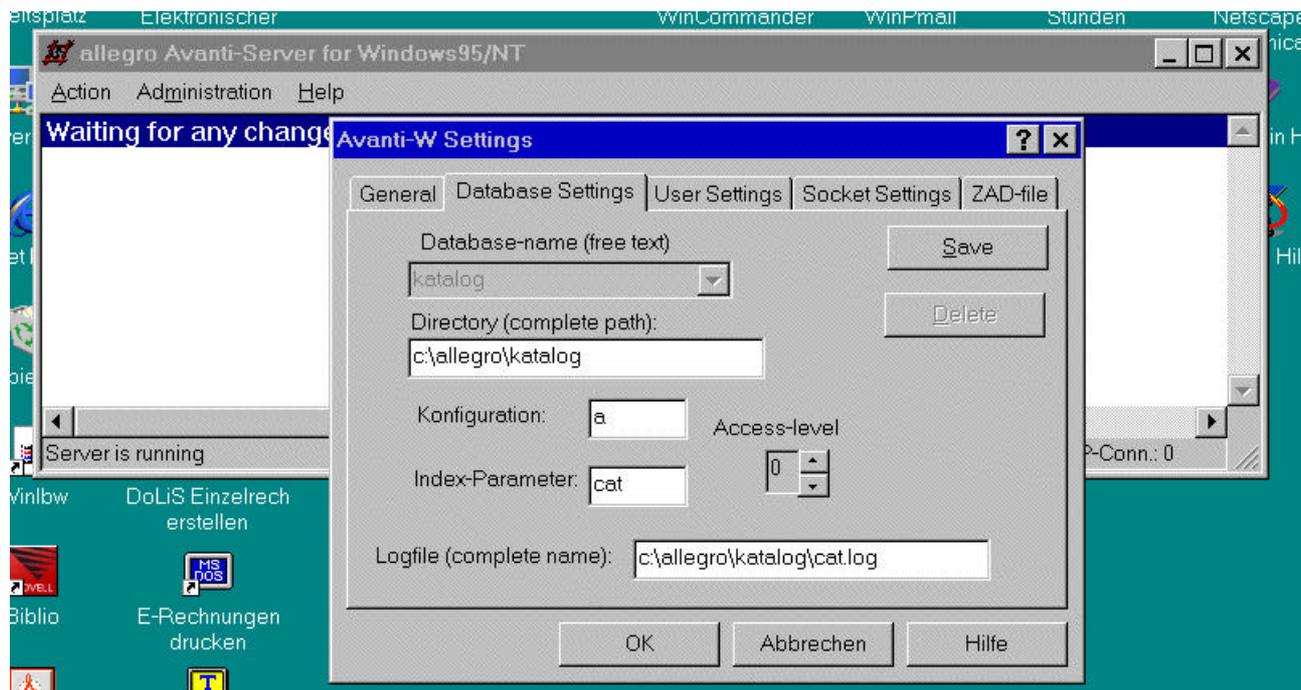
4.3.1 Ihre Datenbank benötigt einen symbolischen Namen

Ihre Datenbank wird über das Netz und auch von *avanti* über einen symbolischen Namen angesprochen. Im Prinzip ist man da frei, nur soll er aus nur einem Wort bestehen und kurz sein. Wir wählen `catalog`

4.3.2 Ihre Datenbank bei *avanti* bekannt machen

Nächster Schritt ist, Ihre Datenbank bei *avanti* bekannt zu machen. Klicken Sie dazu auf das

avanti-Icon und evtl. nochmal in der Taskleiste auf das Programmsymbol um den *avanti*-Bildschirm zur Anzeige zu bringen. Im Dialog unter 'Administration/Settings...' stellen Sie die neue Datenbank ein (das Passwort zum Dialog ist bei einer Neuinstallation von *avanti* immer *allegro*):



Klicken Sie auf 'Save' und verlassen Sie den Dialog durch Klick auf 'OK'. Damit 'kennt' *avanti* Ihre Datenbank. Das dem so ist, können Sie mit Hilfe des *Client-W*-Programms überprüfen: Tragen Sie einfach im Menü 'Optionen' die neue Datenbank ein und machen Sie eine Testrecherche nach einträgen, von denen Sie sicher sind, dass sie nicht in der Demodatenbank vorhanden sein können. Wenn das Ergebnis korrekt ist, ist dieser Punkt erledigt.

4.3.3 Kopieren wichtiger Parameterdateien

Damit bei einer Recherche auch etwas zu sehen ist, benötigen Sie für Ihre Datenbank noch einige spezielle Parameterdateien. Sie finden Sie im Verzeichnis der *avdemo*-Datenbank. Es sind dies:

- *d-html.apr*
- *p-html.apr*
- *p-ansi.apr*
- *p-dos.apr*
- *e-l.apr*

Diese Dateien können Sie zunächst unverändert übernehmen.

4.3.4 Bekanntmachen der Datenbank in den Programmen der WWW-Schnittstelle

Was *avanti* weiß, wissen die Programme der WWW-Schnittstelle noch lange nicht. Um die neue Datenbank über das Netz ansprechen zu können, müssen die Scripte der Schnittstelle diese Datenbank ebenfalls kennen. Dazu sind Änderungen in zwei Dateien notwendig.

4.3.4.1 Änderungen in *acwww.pl*

In der Datei *c:\perl\lib\acwww.pl* muss eine Tabelle ergänzt werden. Sie steckt in der Variablen *%dbscripte*. Es handelt sich dabei um eine sogenannte Hash-Tabelle. Jeder Eintrag besteht aus einem

Schlüssel-Wert-Paar. Tragen Sie die Datenbank `katalog` wie unten angegeben ein:

```
# relativer Pfad der datenbankspezifischen Scripte und
# Parameterdateien. Voreingestellt ist, dass es unterhalb des Verzeichniss
# in dem die Scripte liegen, jeweils pro Datenbank ein Verzeichnis mit den
# spezifischen Scripten und Parameterdateien gibt.
# Datenbankkuerzel      rel. Pfad
%dbscripte = (
    "avdemo",    "opac",
    "opac",      "opac",
    "katalog",   "kat",
);
```

Wichtig ist, dass jeder Eintrag in Anführungszeichen eingeschlossen ist, und die Elemente durch Kommata voneinander getrennt sind. In der ersten Spalte der Tabelle sind die symbolischen Namen der Datenbanken angegeben, die zweite enthält den Namen eines Verzeichnisses. Dieses Verzeichnis muss im nächsten Schritt angelegt werden. Jedoch speichern Sie zunächst die Datei.

4.3.4.2 Anlegen der datenbankspezifischen *dbinfo.pl*

Zu jeder Datenbank muss es eine Datei namens *dbinfo.pl* geben. Sie enthält spezielle Einstellungen zu Überschriften, verwendeten Parameterdateien und anderes. Der Name dieser Datei ist immer derselbe, deshalb muss es bei mehreren Datenbanken für jede Datenbank ein Verzeichnis geben, wo die *dbinfo.pl* zu finden ist. Diese Verzeichnisse müssen unterhalb von `c:\xitami\cgi-bin\acwww25` angelegt werden. Mit der Eintragung in *%dbscripte* für Ihre neue Datenbank haben Sie den Namen dieses Verzeichnisses schon festgelegt. Legen Sie also unterhalb von `c:\xitami\cgi-bin\acwww25` das Verzeichnis *kat* an. Kopieren Sie dann zunächst nur die *dbinfo.pl* aus `c:\xitami\cgi-bin\acwww25\opac` in dieses neue Verzeichnis.

4.3.5 Testen

Damit steht Ihre Datenbank schon im Web zur Verfügung! Machen Sie einen Test mit

```
http://localhost/cgi-bin/acwww25/maske.pl?db=katalog
```

Sie werden beim Erscheinen der Suchmaske zunächst dasselbe Bild wie beim Test der Demo-Datenbank sehen. Das liegt daran, dass die Überschriften und der Datenbankname und andere Dinge noch nicht für Ihre Datenbank eingerichtet sind. Aber recherchieren Sie mal nach einem Titel, der garantiert nur in Ihrer Datenbank sein kann. Sie werden ihn finden.

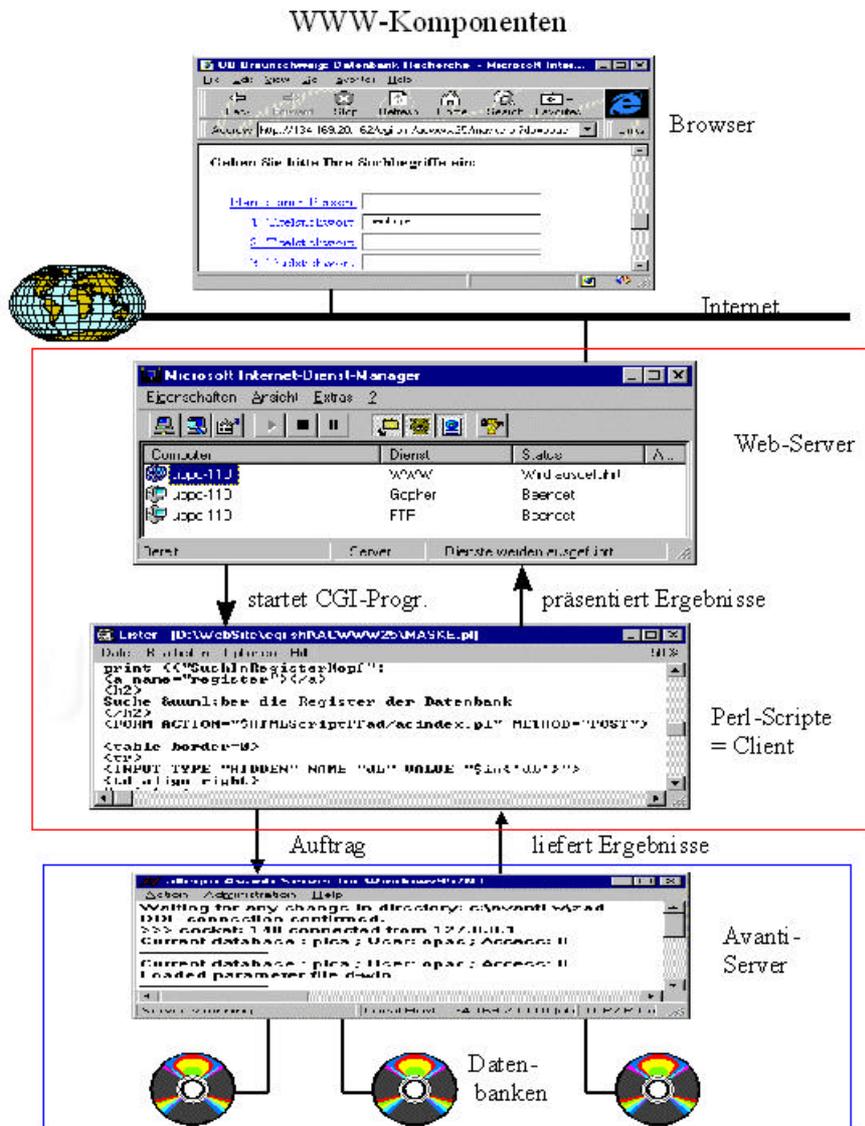
4.4 Wie geht's weiter?

Was jetzt kommt, sind noch Schönheitsarbeiten. Dazu müssen Sie die Datei `c:\xitami\cgi-bin\acwww25\kat\dbinfo.pl` mit einem Editor bearbeiten und an geeigneter Stelle Ihre Werte einsetzen. Was genau wie einzutragen ist, steht in den Kommentaren in der *dbinfo.pl*.

5 Die Anbindung einer *allegro*-Datenbank an das WWW schematisch

Der vorangegangene Abschnitt erläuterte ein einfaches Verfahren, mit dem man eine *avanti*-Datenbank auf einem Windows-Rechner relativ schnell ins Netz stellen kann. Hier werden

jetzt die Zusammenhänge etwas ausführlicher erläutert, so dass man mit diesen Informationen auch andere Konstellationen zusammenstellen kann. Das nächste Bild zeigt die Anbindung einer *allegro*-Datenbank an das WWW schematisch im Überblick:



Der Browser des Benutzers verbindet sich über das Netz mit Ihrem Webserver und fordert eine Informationsseite an. Da der Benutzer in einer Datenbank recherchieren möchte, kann es keine fertige, schon bereit stehende Datei geben, sondern die von ihm gewünschten Informationen müssen erst zusammengestellt und dann als dynamisch erzeugte HTML-Seite präsentiert werden. Anhand der Konstruktion der URL, mit der die Informationen angefordert werden, kann der Webserver erkennen, dass er zunächst ein Programm starten muß, bevor er ein Ergebnis zurück liefert.

Bisher gibt es leider keinen Webserver, der etwas vom Umgang mit einer *avanti*-Datenbank versteht. Daher muss es einen Vermittler zwischen Webserver und *allegro* geben. Das sind die *acwww*-Programme dieser Schnittstelle. Der Webserver verfügt über eine sogenannte CGI-Schnittstelle (Common Gateway Interface), über die er beliebige Programme starten kann, denen er auf definierte Art und Weise Optionen übergeben kann, und über die er das Ergebnis der Programme empfängt. Dieses Ergebnis schickt er dann an den Browser zurück. Die Programme der *acwww*-Schnittstelle 'wissen' ihrerseits, wie sie ihre Optionen vom Webserver bekommen, wie sie sie in eine Datenbankrecherche umsetzen, und wie sie das Ergebnis der Recherche im HTML-Format

dem Webserver wieder übergeben.

Die acwww-Programme könnten so konstruiert sein, dass sie direkt mit einer *allegro*-Datenbank umgehen könnten. Dazu wäre es notwendig, dass die Programme direkten Zugriff auf die Daten hätten. Das hieße, die Datenbank müßte entweder auf demselben Rechner liegen wie die acwww-Programme, oder der Rechner müßte über ein Netzwerk direkten Zugriff auf die Datenbank haben. Diese Lösung wäre aber recht starr. Wir sind einen anderen Weg gegangen: Zusätzlich zu den acwww-Programmen kommt der *avanti*-Server zum Einsatz. *avanti* ist ein universell einsetzbares Programm, das programmgesteuert, also ohne dass ein Mensch es direkt bedient, alles notwendige mit einer *allegro*-Datenbank machen kann. Zudem kann er seine Aufträge über das Internet empfangen und die Ergebnisse auch über das Internet wieder abliefern.

Die Kombination aus Webserver, acwww-Programmen und *avanti* erlaubt somit sehr flexible Lösungen bei der Anbindung einer *allegro*-Datenbank ans WWW: Im Schema oben sehen Sie zwei in rot und blau markiert Bereiche. Diese Bereiche sind jeweils als logische Einheit zu betrachten. Der Ausdruck 'logische Einheit' suggeriert schon, dass das ganze System nicht unbedingt eine physikalische Einheit sein muss. So sind beim Aufbau folgende Kombinationen bei der Anordnung der Programme denkbar:

- **Alles auf einem Rechner.** Sowohl Webserver als auch die acwww-Programme als auch *avanti* als auch die Datenbank(en) befinden sich auf einem Rechner. Dies entspricht der oben beschriebenen Einstiegsconfiguration.
- **Alles auf zwei Rechnern.** Webserver und acwww-Programme auf einem Rechner, *avanti* und die Datenbank auf einem zweiten Rechner. Dazu müssen die beiden Rechner über das lokale Netzwerk miteinander verbunden sein, und auf beiden Rechner muss TCP/IP als ein Netzwerkprotokoll gefahren werden.
- **Alles auf drei oder mehr Rechner verteilt.** Webserver und acwww-Programme auf einem Rechner, *avanti* auf einem zweiten und die Datenbank(en) auf einem dritten (und oder vierten oder ...) im lokalen Netzwerk.

Ich habe bewußt nicht erwähnt, dass bestimmte Komponenten auf bestimmten Rechnerplattformen installiert sein müssen, denn auch auf diesem Gebiet ist man frei. So hat die UB Braunschweig den Webserver und die acwww-Programme auf einer SUN installiert, und *avanti* und die Datenbanken auf einer anderen. Es gibt bei uns aber auch einige Datenbanken für bestimmte interne Aufgaben, wo *avanti* auf einem NT-Server läuft und die Datenbanken noch in unserem lokalen Novell-Netz liegen. Auch damit lässt sich arbeiten.

6 Grundsätzliche Hinweise zur Installation

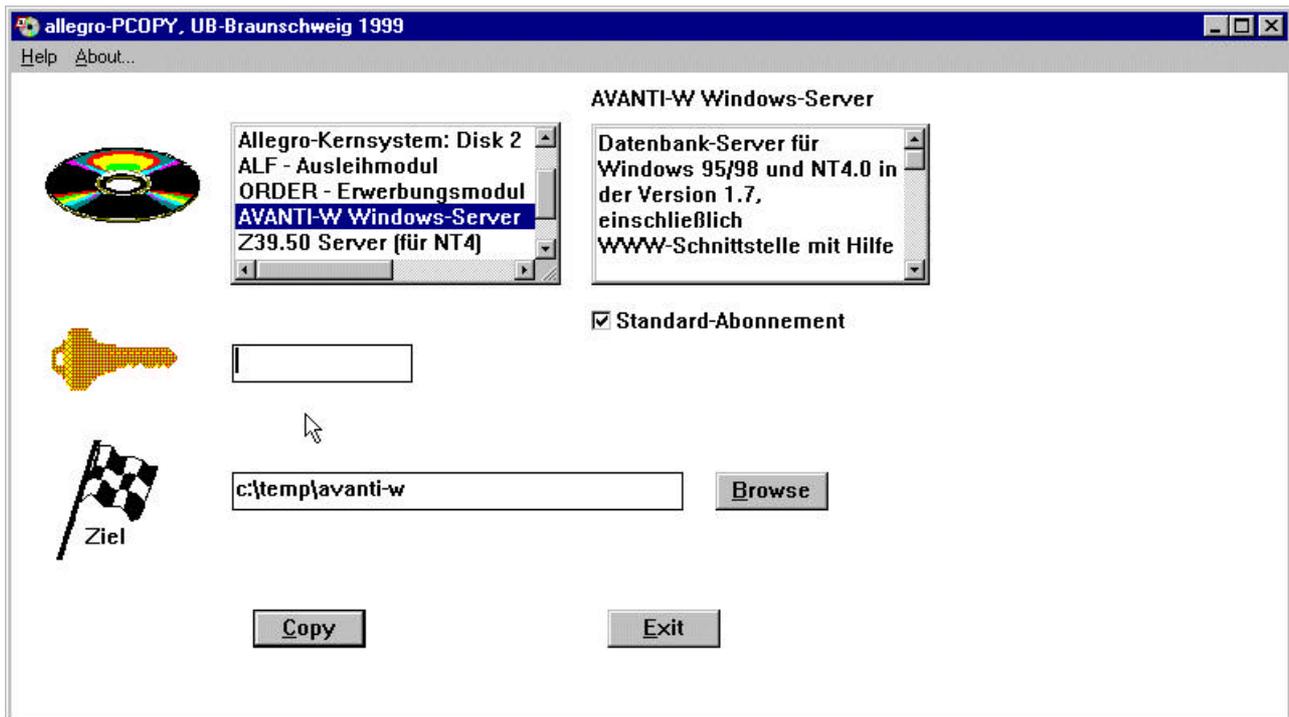
So wie es viele Wege nach Rom gibt, so gibt es nicht **die** Art, eine *allegro*-Datenbank an Windows anzubinden. Durch geschicktes Verknüpfen der beteiligten Komponenten kann man zu sehr flexibel einsetzbaren Lösungen kommen. Sie sind nicht auf den Einsatz eines bestimmten Rechners mit einer bestimmten Ausbaustufe und bestimmten Programmen festgelegt. Es muss beispielsweise nicht unbedingt eine UNIX-Maschine zum Einsatz kommen. Verglichen mit anderen Systemen ist *allegro* ein relatives Leichtgewicht und man kommt meistens mit den schon vorhandenen Rechnern sehr weit.

Ich beschreibe hier die grundsätzlichen Dinge, die bei der Installation der Einzelkomponenten zu beachten sind. Die Zusammenhänge und eine Beispielinstallation sind weiter oben beschrieben worden. Sie finden hier allerdings auch keine genaue Schritt-für-Schritt-Anleitung für jede Komponente, denn dafür sind die einsetzbaren Programmdistributionen zu unterschiedlich. Am ausführlichsten erfahren Sie etwas über *avanti* und die Schnittstellen-Programme.

6.1 avanti

6.1.1 Windows

Unter Windows wird *avanti* mit einem Setup-Programm installiert. Zunächst müssen Sie das *avanti*-Installationsprogramm auf ihre Festplatte von CD-kopieren. Das geht nicht direkt, sondern wie bei den anderen *allegro*-Programmen auch über unser spezielles Zugriffsprogramm auf der CD (es erscheint meistens automatisch, sowie die CD eingelegt wird). Klicken Sie auf den Knopf , um das spezielle Kopierprogramm zu starten. Wählen Sie dort den *avanti*-Server aus, wie das folgende Bild zeigt:



Im Schlüsselfeld tragen Sie Ihren Registrierungsschlüssel ein, der mit der letzten Rechnung mitgekommen ist.

Alternativ können Sie sich das Setup-Programm auch von unserem ftp-Server (Adresse: avanti.biblio.etc.tu-bs.de) herunterladen. Benutzen Sie dazu am besten ein spezielles FTP-Programm, weil der Zugriff auf diese Datei passwortgeschützt ist.

Kopieren Sie das Programm in ein temporäres Verzeichnis und starten Sie es einfach. Der Rest läuft automatisch: Man muss sich nur entscheiden, wo *avanti* liegen soll. Es muss nicht im allgemeinen *allegro*-Programmverzeichnis installiert werden. Sie können ein anderes Verzeichnis wählen.

Wo die anzubindenden Datenbanken genau liegen, spielt keine Rolle. Sie können auf derselben Platte wie *avanti* liegen oder auf einer anderen Platte desselben Rechners oder sogar auf einem anderen Rechner im lokalen Netz. Im allgemeinen können sie dort bleiben, wo sie schon sind.

Zur Hardware ist zu sagen: Je mehr RAM desto besser und je mehr MHz desto besser. Bedenken Sie bitte, dass falls alles auf einem Rechner installiert ist, der Rechner bei einer Datenbankrecherche mehrere Programme fast gleichzeitig ausführen muss. Perl und *avanti* benötigen recht viel Speicher.

Für einen stabilen Serverbetrieb empfehlen wir den Einsatz von Windows-NT, zumal Sie dann auch die NT-Service-Version von *avanti* einsetzen können. In dem Fall ist es nämlich nicht notwendig,

dass ein Benutzer auf dem Rechner eingeloggt ist, um *avanti* zu starten. Bei Einsatz von Windows 95 oder 98 ist es notwendig, dass ein Benutzer eingeloggt ist.

6.1.2 UNIX

Unter UNIX ist mehr Handarbeit zu leisten. Die UNIX-Versionen von *avanti* gibt es nur von unserem FTP-Server. Der Zugang ist passwortgeschützt. Das aktuelle Passwort und den Usernamen entnehmen Sie bitte Ihrer letzten Rechnung.

Die genaue Installationsanleitung finden Sie in einer Textdatei auf dem FTP-Server. Prinzipiell gibt es kein vorgeschriebenes Verzeichnis für die Installation. Richten Sie sich da nach Ihren Vorgaben oder denen des Systemverwalters. Wenn das Programm einmal eingerichtet ist, braucht eigentlich niemand mehr Schreibrechte auf dem Programmverzeichnis. Die Datenbanken hingegen sollten in einem Verzeichnis liegen, wo diejenigen auch Schreibrechte haben, die die Datenbanken erneuern. Aber auch das Nutzerkonto, unter dem *avanti* läuft, sollte dort eventuell Schreibrechte haben, damit *avanti* auch Aufträge mit schreibenden Zugriff auf eine Datenbank erledigen kann.

Die Datenbanken sollten auf einer lokalen Festplatte liegen. Systemübergreifende Anbindung über NFS ginge auch. Ob so eine Anbindung praktikabel ist, muss ein Test erweisen. Meistens ist es aber so, dass die Datenbanken nicht auf dem UNIX-System gepflegt werden, sondern auf einem DOS- oder Windows-System. Um die Datenbanken auf der UNIX-Maschine dann aufzufrischen gibt es mehrere Wege:

- Kopieren per FTP. Das ist der einfachste Weg. Sie kopieren die Datenbanken komplett einfach auf die UNIX-Maschine. Eine Datenreorganisation oder ein Neuaufbau des Index gar sind nicht notwendig.
- oder Sie setzen *avanti* für diesen Zweck ein. Dazu müssen Sie sich aber eigene Programme erstellen, die die neuen Daten direkt über *avanti* einspielen und auffrischen. Das kann im laufenden Betrieb geschehen. Der Zugang zur Recherche muss dafür nicht unterbrochen werden.
- Sie kopieren die neuen Daten als *allegro*-Grunddatei auf den UNIX-Rechner und spielen sie mit UNIX-*update* im laufenden Betrieb in die Datenbank ein.

6.2 Perl

Je nach Geschmack. Die entsprechende Distribution gibt in den div. readmes gute Hinweise. Eingebürgert hat sich unter UNIX die Installation nach `/usr/local/bin` und unter DOS/Windows `c:\perl`. Darauf kommt es aber nicht so genau an. Wichtig nur, dass unter Windows Perl im Suchpfad aufgenommen wird. Unter Win95/NT erzeugt das Installationsscript die notwendigen Registry-Einträge selbst. Eventuell müssen Sie unter Windows noch die Verknüpfung für die Dateiendung `.pl` mit dem Perl-Interpreter einrichten. Wichtig ist nur, dass ein Perl-Programm von überall her aufrufbar ist.

6.3 acwww

Die *allegro*-W3-Schnittstelle besteht aus einigen Programmen in Perl, einigen Textdateien in HTML und einigen Parameterdateien. Für eine Beispieldatenbank der Standardkonfiguration sind die notwendigen Parameter mitgeliefert. Die einzelnen Komponenten werden im nächsten Abschnitt beschrieben. Wer die Schnittstelle für seine eigenen Bedürfnisse anpassen will, muss sich den Abschnitt genauer durchlesen. Hier nur einige grundsätzliche Bemerkungen.

6.3.1 Windows

Wenn Sie *avanti* installiert haben finden Sie im Installationsverzeichnis auch ein Verzeichnis

acwww. Darin befindet sich ein sich selbst extrahierendes Archiv *acwww25.exe*. Starten Sie das Programm und es entpackt alle notwendigen Dateien der WWW-Schnittstelle in zwei Verzeichnissen. Beide Verzeichnisse enthalten wiederum Verzeichnisse. Diese Struktur ist für das Funktionieren der Schnittstelle wichtig. Die Dateien liegen aber noch nicht am richtigen Platz. Verfahren Sie mit der Installation wie in der oben beschriebenen Beispielinstallation. Nur, wo dort konkrete Pfade genannt werden, müssen Sie natürlich Ihre Verhältnisse berücksichtigen.

6.3.2 UNIX

Genau wie unter Windows finden Sie im Installationsverzeichnis von *avanti* ein Verzeichnis *acwww*. Aber in dem Verzeichnis befindet sich nicht ein Archiv, sondern die Dateien selbst in einer Verzeichnisstruktur, wie sie auf unserer SUN besteht. Unser Webserver ist in *home/ns-home* installiert. Das wird bei Ihnen anders aussehen. Sie sollten aber die Verzeichnisstruktur, die unterhalb von *ns-home* kommt, übernehmen. Das Verzeichnis *cgi-bin* gibt es eigentlich unter jedem Webserver, das Verzeichnis *docs*, in dem die Dokumente liegen, heißt bei Ihnen wahrscheinlich anders. Mit *docs* meine ich das sogenannte Document-Root-Verzeichnis des Webserver.

Beachten Sie bitte: Alle Hauptprogramme (*acindex.pl*, *allegro.pl*, *maske.pl* und *regsrch.pl*) müssen auf *executable* gesetzt werden.

In allen Hauptprogrammen muss die **erste** Zeile den korrekten Aufruf von Perl enthalten! Voreingestellt ist `#!/usr/local/bin/perl`. Wenn Perl auf Ihrem System in einem anderen Verzeichnis installiert ist, müssen Sie den Aufruf entsprechend anpassen.

7 acwww25 - die Schnittstelle

Die Schnittstelle besteht aus vielen Dateien, die im Folgenden näher erläutert werden sollen. Bevor wir aber in die Einzelheiten abtauchen, etwas zur grundsätzlichen Arbeitsweise der Programme.

7.1 Grundsätzliche Arbeitsweise der Programme

Jedes Teilprogramm fängt zunächst ähnlich an:

1. Einlesen der Dateien *acwww25.pl* und *cgi-lib.pl*. Beide Dateien enthalten Unterprogramme und Voreinstellungen, die für alle wichtig sind.
2. Grundinitialisierung durch die Routine `&Init` (in *acwww25.pl*). Es werden einige Voreinstellungen vorgenommen und die Verbindung zum *avanti*-Server hergestellt.
3. Einlesen und Aufbereiten der übergebenen Programmoptionen.
4. Eine Option, die immer übergeben wird, ist das Kürzel der Datenbank, mit der gearbeitet werden soll. Aus dem Kürzel und der Variablen `%dbscripte` in *acwww25.pl* wird das Verzeichnis ermittelt, wo das Programm die zur Datenbank gehörige *dbinfo.pl* finden kann.
5. Einlesen der datenbankspezifischen Voreinstellungen aus der datenbankeigenen *dbinfo.pl*. **Ohne** diese Datei läuft nichts!
6. Beginn der eigentlichen Arbeit, wobei Aufbereitung der Ergebnisseite und Ermittlung des Ergebnisses miteinander verflochten sind. Für Änderungen in diesem Prozess muss man sich in Perl, HTML und der *avanti*-Sprache auskennen.

Die eigentliche Arbeit sieht dann meistens so aus: Zuerst wird der Anfang der Ergebnisseite aus *texts/scriptname.kpf* ausgegeben. Achten Sie darauf, dass hier der Anfang einer HTML-Seite korrekt

eingeleitet wird, sonst sieht man im Browser u.U. nichts.

Dann wird der Auftrag für den *avanti*-Server zusammengestellt. Der Auftrag wird an den Server übergeben und das Script wartet geduldig, bis die Antwort zurückgegeben wird.

Das Ergebnis wird ausgewertet und präsentiert.

Zuletzt wird die Datei *texts/scriptname.end* ausgegeben. Sie enthält den Schluß des HTML-Dokumentes. Hier kann man eintragen, was man am Ende der Seite erscheinen lassen will. Wichtig sind die abschließenden `</body>`- und `</html>`-Tags.

7.2 Die Programme im Einzelnen

Sie werden in `.../cgi-bin/acwww25` und darunter diese Dateien finden:

<i>acindex.pl</i>	Script zum Erzeugen eines Registerausschnittes. Die Registereinträge sind 'verlinkt', d.h. anklickbar. Entweder löst man mit einem Mausklick eine Titelanzeige oder bei einem Verweis eine weitere Registerproduktion aus. Das Register kann erweitert werden, d. h. zu jedem Registereintag können die Kurztitel angezeigt werden.
<i>acindex.net</i>	Alternative zu <i>acindex.pl</i> . Hauptunterschied ist, dass der Zugriff auf die Kurztitel für das erweiterte Register nicht direkt erfolgt, sondern über <i>avanti</i> -Jobs geregelt wird. Dieses Programm müssen Sie einsetzen, wenn <i>avanti</i> auf einer anderen Maschine als diese Programme eingesetzt wird. Sie haben dann nämlich keine Möglichkeit, direkt auf die Kurztiteldatei zugreifen zu können! Um dieses Programm einsetzen zu können, müssen Sie es in <i>acindex.pl</i> umbenennen.
<i>allegro.pl</i>	verarbeitet die Suchanfrage mit logischen Kombinationen und erzeugt aus der Ergebnismenge eine Kurztitelanzeige, aus der heraus dann die gewünschten Titel zur Anzeige gebracht werden können. Übersteigt die Ergebnismenge die Anzahl der anzuzeigenden Zeilen, gibt es eine Blätter-Funktion.
<i>maske.pl</i>	erzeugt die Suchmasken für die Datenbanken. Dies ist der eigentliche Einstieg für die Recherche einer Datenbank. Die Verwendung dieses Programms als Einstieg ist nicht zwingend vorgeschrieben. Es sind nur Besonderheiten in den anderen Programmen zu beachten.
<i>regsrch.pl</i>	Das Programm zur Erzeugung der Titelanzeige.
<i>acwww25/texts</i>	Verzeichnis, in dem sich Dateien mit HTML-Dateifragmenten finden. Für jedes Script zwei. <code>' .kpf'</code> enthält den Anfang eines HTML-Dokumentes, <code>' .end'</code> das Ende. Jedes Script gibt am Anfang seiner Ausgabe das <code>.kpf</code> -Fragment aus, dann den eigentlichen Inhalt und zum Schluß das <code>.end</code> -Fragment. Sinn dieser Konstruktion: Sie können das generelle Aussehen der produzierten HTML-Seiten einfacher beeinflussen. Um Änderungen anzubringen, sind unbedingt HTML-Kenntnisse notwendig.
<i>acwww25/lib</i>	enthält die Dateien <i>acwww25.pl</i> und <i>cgi-lib.pl</i> . Sie gehören unbedingt in das Library-Verzeichnis von Perl! Für Windows ist das wahrscheinlich <code>c:\perl5\lib</code> und unter UNIX oft <code>/usr/lib/perl5</code> . Der genaue Ort hängt aber von Ihrer Perl-Installation ab!
<i>acwww25/opac</i>	Ort der Konfigurationsdatei <i>dbinfo.pl</i> für die Datenbank mit der Kennung 'opac' und 'avdemo'. Mehr dazu weiter unten.

7.3 Anpassungen in den Programmen

7.3.1 Allgemeines

Es gibt notwendige Anpassungen, die Sie in jedem Fall vornehmen müssen und solche, die nicht unbedingt notwendig sind. So werden Sie wahrscheinlich nicht damit zufrieden sein, wenn Ihr OPAC immer unter der Braunschweiger Überschrift zu recherchieren ist. Dann wollen Sie u.U. noch Anpassungen in den Parameterdateien für die Datensatzanzeige vornehmen. Nicht unbedingt notwendig sind Anpassungen, die das gesamte Erscheinungsbild der erzeugten Seiten betreffen. Wenn Sie das aber vorhaben, müssen Sie sich etwas mit Perl vertraut machen, da das Layout der Seiten im Programmcode enthalten ist.

In den Hauptprogrammen *acindex.pl*, *allegro.pl*, *maske.pl* und *regsrch.pl* sind zunächst mal keine Anpassungen notwendig -- es sei denn, Sie setzen Sie unter UNIX ein. Dann müssen Sie zumindest den Programmaufruf in der ersten Zeile jedes der Hauptprogramme überprüfen. Weiter oben ist beschrieben, was zu tun ist.

Hingegen müssen Sie in den Dateien *acwww25.pl* im Perl-Library-Verzeichnis und in **jeder** *dbinfo.pl* einer anzubindenden Datenbank Änderungen vornehmen.

7.3.2 Änderungen in *acwww25.pl*

Die Datei *acwww25.pl* enthält die grundlegenden Einstellungen, die in allen Programmen der Schnittstelle benötigt werden. Außerdem finden Sie hier noch einige wichtige Unterprogramme. Anpassungen sind i. a. nur in den globalen Variablen am Anfang der Datei notwendig. Die Unterprogramme brauchen eigentlich nicht geändert zu werden. Im folgenden erläutere ich die Variablen im Einzelnen. Die Reihenfolge der Beschreibungen stimmt mit der in der Datei *acwww25.pl* überein und ist ganz bestimmt kein Indiz dafür, wie wichtig die jeweilige Einstellung ist.

7.3.2.1 Ort und Adresse von *avanti* (*\$AVANTI_HOST* und *\$AVANTI_PORT*)

Es wurde schon weiter oben erwähnt, dass es nicht notwendig ist, die Programme der Schnittstelle und *avanti* auf ein und derselben Maschine laufen zu lassen. Daher müssen die Programme natürlich wissen, wo sie *avanti* finden. Mit diesen beiden Variablen gibt man die Adresse der Maschine an, auf der *avanti* läuft:

\$AVANTI_HOST

Hier können Sie entweder die IP-Adresse des Rechners (z.B. "198.20.1.3" oder einen symbolischen Namen wie z.B. "mein_server.meine_domain.de" angeben. Wenn *avanti* auf derselben Maschine laufen sollen wie die Schnittstellenprogramme muss als Adresse entweder "localhost" oder "127.0.0.1" eingetragen sein.

\$AVANTI_PORT

Portnummer, auf der *avanti* 'horcht'. Dieser Wert muss eine Zahl sein, also ohne einschließende Anführungszeichen angegeben werden. Welche Portnummer hier einzutragen ist, hängt von der *avanti*-Installation ab.

Voreinstellung:

```
$AVANTI_PORT = 4949;
$AVANTI_HOST = "localhost";
```

7.3.2.2 Startverzeichnis fuer die Scripte (*\$HTMLScriptPfad*)

An einigen Stellen setzen die Programme Links in den erzeugten HTML-Seiten auf andere

Programme der Schnittstelle. Diese Links müssen einen Pfad enthalten. Welcher Pfad dort eingesetzt wird, bestimmt diese Variable. Sie sollte in HTML-Notation auf das CGI-Verzeichnis dieser Schnittstelle verweisen.

Achtung: Hier darf nicht der absolute wirkliche Pfad angegeben werden. Erstens würden die Links in den HTML-Seiten damit *nicht funktionieren* und zweitens wäre dies ein Sicherheitsloch, denn es offenbart etwas über die Struktur des Webservers.

Voreinstellung:

```
$HTMLScriptPfad = "/cgi-bin/acwww25";
```

7.3.2.3 CGI-Pfad absolut (*\$AllegroW3CGIPfad*)

Hier muss der einzig wahre, real existierende Pfad der Schnittstellenprogramme eingestellt werden. Jedes Programm wechselt am Anfang zu Sicherheit in dieses Verzeichnis. Denn abhängig von Einstellungen im Server, werden die Programme evtl. aus einem anderen Verzeichnis heraus gestartet. Für das Einlesen der datenbankeigenen *dbinfo.pl* muss das Script aber das eigene Verzeichnis als Arbeitsverzeichnis haben. Wenn sie aus einem anderen Verzeichnis gestartet würden, könnten sie *dbinfo.pl* nicht finden. Voreinstellung:

```
$AllegroW3CGIPfad = "c:/xitami/cgi-bin/acwww25";
```

7.3.2.4 Relativer Pfad der datenbankspezifischen Scripte (*%dbscripte*)

Dies ist die wirklich wichtige Variable, die immer dann erweitert werden muss, wenn Sie eine neue Datenbank anbinden wollen. Es handelt sich hierbei um ein sogenanntes Hash-Array, auch assoziatives Array oder *dictionary* genannt. Die Einträge bestehen immer aus einem Schlüssel-Werte-Paar. Der Schlüssel ist der symbolische Name der Datenbank (für die WWW-Schnittstelle, aber auch für *avanti*), und der Wert ist das Verzeichnis, wo die Programme die zur Datenbank gehörige *dbinfo.pl* finden können. Die Angabe **muss** relativ zum Verzeichnis *cgi-bin/acwww25* sein. Meistens wird das Verzeichnis genauso genannt wie die Datenbank. Voreingestellt ist:

```
%dbscripte = (
    "avdemo",    "opac",
    "opac",     "opac",
);
```

Achten sie bitte genau auf die korrekte Syntax: Sowohl Schlüssel als auch Wert sind in Anführungszeichen einzuschließen, und jedes Element wird mit Komma abgeschlossen!

7.3.2.5 Namen der anbietenden Organisation (*%Organisation*)

Die Werte in dieser Variablen werden Sie wahrscheinlich so schnell wie möglich ändern wollen. *%Organisation* ist auch ein dictionary, enthält aber nur zwei Einträge: Den Namen der Bibliothek in einer Kurz- und in einer Langform. Beide sind in HTML-Schreibweise anzugeben.

Voreinstellung:

```
# Namen der anbietenden Organisation in Kurzform und Langform
%Organisation = (
    "Kurzform", "UB Braunschweig",
    "Langform", "Universitätsbibliothek Braunschweig",
);
```

7.3.3 Änderungen in *dbinfo.pl*

Diese Datei enthält Einstellungen speziell für eine Datenbank. Damit sollte deutlich sein, dass in dieser Datei so ziemlich alles angepasst werden muss. Für die Einstellungen in dieser Datei gibt es prinzipiell keine Voreinstellungen. Ich erkläre die einzelnen Elemente daher am Beispiel der mitgelieferten *dbinfo.pl* für die Demo-Datenbank. Sie finden sie im Verzeichnis *.../cgi-bin/acwww25/opac*.

7.3.3.1 Konfigurationsbuchstabe *\$k1*

Diese Variable spielt keine Rolle mehr. Der Wert wird nicht mehr in den Programmen benötigt.

7.3.3.2 Datenbankpfad *\$DBPfad*

Tragen Sie hier den vollständigen Pfad zu Ihrer Datenbank ein. Er wird von *acindex.pl* für den Zugriff auf die Kurztiteldatei benötigt. Beachten Sie, dass unter Windows die Pfadtrenner `\` doppelt angegeben werden müssen.

Falls Sie jedoch *acindex.net* einsetzen, spielt diese Variable keine Rolle mehr, da der Zugriff auf die Kurztitel dort über *avanti*-Aufträge geregelt wird.

Vorgabe:

```
$DBPfad="c:\\avanti-w\\avdemo";
```

7.3.3.3 Datenbankname *\$DBName*

Tragen Sie hier den wirklichen Namen Ihrer Datenbank (= Vorname der benutzten Indexparameterdatei) ein. Er wird ebenfalls von *acindex.pl* für den Zugriff auf die Kurztiteldatei benötigt.

Falls Sie jedoch *acindex.net* einsetzen, spielt diese Variable keine Rolle mehr, da der Zugriff auf die Kurztitel dort über *avanti*-Aufträge geregelt wird.

Vorgabe (Name der Demodatenbank *avdemo*):

```
$DBName="cat";
```

7.3.3.4 *avanti*-Nutzername *\$user* und Passwort *\$passwd*

Jeder *avanti*-Auftrag muss mit Nutzernamen und Passwort gekennzeichnet werden. Für jede Datenbank, die unter Windows bei *avanti* angemeldet wird, gibt es einen voreingestellten Benutzer *opac* mit dem Passwort *OPAC*. Ihm ist nur der lesende Zugriff auf die Datenbank erlaubt. Für unsere Zwecke genügt das ebenfalls. Deshalb die Voreinstellung:

```
$user="opac";           # Nutzername
$passwd="opac";        # Passwort
```

7.3.3.5 Die Parameterdateien für die Ausgabe der Datensätze (*%Parameterdateien*)

Die Zahl der möglichen Ausgabeformate ist nur durch die Zahl der bereitgestellten Parameterdateien begrenzt. Jedoch muss es mindestens eine Parameterdatei geben. Bei den Angaben zu den Parameterdateien muss man sich an bestimmte Regeln halten:

- Die Voreinstellung für die Anzeige bekommt den Schlüssel `DEFAULT`. Alle weiteren Schlüssel sollen in Kleinbuchstaben geschrieben sein. Sie können aber beliebig gewählt werden. Der Grund für die Regel ist, das das Anzeigeprogramm *regsrch.pl* die Liste der möglichen Ausgabeformate in sortierter Reihenfolge ausgibt, wobei die Schlüssel von *%Parameterdateien* alphabetisch sortiert werden. Es gilt dabei die ASCII-Wertigkeit der Buchstaben. Die Voreinstellung erscheint zuerst, weil Großbuchstaben vor Kleinbuchstaben sortiert werden.
- Die Werte bestehen aus zwei Teilen, die untereinander durch '|' (das Pipe-Zeichen) zu trennen sind. Der erste Teil ist eine kurze Umschreibung für das was die Anzeigeparameterdatei leistet. Er wird von *regsrch.pl* in der Auswahlliste ausgegeben. Der zweite Teil ist der vollständige Name der zu verwendenden Parameterdatei.

Aus diesen Angaben erzeugt dann *regsrch.pl* ggf. eine Auswahl für andere Anzeigformate.

Vorgabe:

```
%Parameterdateien = (
    'DEFAULT', 'Standardanzeige|d-html.apr',
    'allegro', 'allegro-Internformat (a.cfg)|e-1.apr',
);
```

7.3.3.6 Sortierung der Anzeige (*\$SortChoice* und *%Sortierung*)

Wenn das Suchergebnis aus mehreren Datensätzen besteht, sollte sie für die Anzeige nach einem sinnvollen Kriterium sortiert sein. Prinzipiell gibt es dafür zwei Möglichkeiten:

1. Sortierung durch *avanti* anhand der Kurztitel, die zu den Sätzen gehören (*\$SortChoice="kurztitel"*;) und
2. Vorbereitung der Sortierung nach beliebigen Kriterien, die durch eine Sortierparameterdatei zusammengestellt werden (*\$SortChoice="sortierfile"*;))

Man kann für eine Datenbank nur eine von beiden Methoden benutzen.

Die erste Möglichkeit ist in der Reaktionszeit schneller und damit generell zu empfehlen. Sie muss sogar eingesetzt werden, wenn Schnittstellenprogramme und *avanti* nicht auf derselben Maschine laufen. Wenn die Kurztitelzeile geschickt aufgebaut ist, kann man schon sehr viele Sortierungen anbieten. So gibt es bei der Demodatenbank, die ja auf dem Standardschema beruht, die Möglichkeiten der Sortierung nach

- Titel
- erstem Verfasser
- Erscheinungsjahr
- Signatur

Die Sortierung lässt sich jeweils auf- oder absteigend einstellen. Das Zusammenstellen der auszugebenden Titel, Sortieren und Erzeugen der Ausgabe kann *avanti* in einem Auftrag erledigen.

Die zweite Möglichkeit bietet u.U. mehr Flexibilität bei der Sortierung, ist aber komplizierter einzurichten. Sie dauert in der Ausführung länger, weil zwei *avanti*-Aufträge notwendig sind, und eine Zwischendatei geschrieben werden muss. Zudem funktioniert sie nur, wenn *avanti* und Schnittstellenprogramme auf demselben Rechner arbeiten oder auf dasselbe Verzeichnis zugreifen können.

Zunächst müssen Sie die notwendigen Sortierparameterdateien erstellen. Das Ergebnis der Ausgabe

mit so einer Datei muss die Datensätze in einem Format liefern, dass sie sortiert werden können. Dieses Format ist dasselbe, was bei der Standardlistenproduktion eingesetzt wird. Nähere Informationen dazu finden Sie im Systemhandbuch unter dem Thema Listenproduktion und am Ende des Kapitels über die Parametrierung. Die erzeugten Daten werden vom Programm *regsrch.pl* sortiert und müssen dann in eine Datei geschrieben werden, auf die *avanti* lesend zugreifen kann. Insbesondere unter UNIX muss bei dieser Methode gewährleistet sein, dass zum einen *regsrch.pl* unter einem Nutzerkonto läuft, das unterhalb von *.../cgi-bin/acwww25* Schreibrechte hat, weil dort nämlich die Zwischendateien angelegt werden. Zum anderen muß gesichert sein, dass *avanti* diese Dateien auch lesen darf. In einem zweiten Auftrag liest *avanti* dann die sortierten Daten ein und gibt das für die Ausgabe formatierte Ergebnis an *regsrch.pl* zurück.

Nicht nur von der Methodik her ist diese Variante langsamer, sondern sie erfordert auch mehr Programmieraufwand.

```
$SortChoice="kurztitel";
if($SortChoice eq "kurztitel")
{
    %Sortierung = (
        'DEFAULT',          'nach Erscheinungsjahr (absteigend)|d 58',
        'erschjahr',       'nach Erscheinungsjahr (aufsteigend)|a 58',
        'verfauf',         'nach Verfasser (aufsteigend)|a 45',
        'verfab',         'nach Verfasser (absteigend)|d 45',
        'titelauf',        'nach Titel (aufsteigend)|a 1',
        'titelab',         'nach Titel (absteigend)|d 1',
        'znosort',         'keine Sortierung|a -1',
    );
}
else{
    %Sortierung = (
        'DEFAULT',          'nach Verfasser / Jahr|s-verfj.apr',
        'erschjahr',       'nach Erscheinungsjahr|s-ejahr.apr',
        'verftitel',       'nach Verfasser / Titel|s-pt.apr',
        'titel',           'nach Titel|s-titel.apr',
        'znosort',         'keine Sortierung|s-nosort.apr',
    );
}
```

7.3.3.7 Codierung von Sonderzeichen in der Ausgabe (%Codierung)

Wenn in den Ergebnissen Sonderzeichen wie Umlaute etc. zu erwarten sind, kann man das Ergebnis mit unterschiedlichen Codierungen anbieten. Wie bei den Parameterdateien gilt aber auch hier, dass es mindestens eine Parameterdatei geben muss. Die Regeln für den Aufbau der Variablen sind dieselben wie bei *%Parameterdateien*.

Vorgabe:

```
%Codierung = (
    'DEFAULT',          'HTML-Codierung|p-html.apt',
    'ansi',             'ANSI-Codierung|p-ansi.apt',
    'dos',              'DOS-Codierung|p-dos.apt',
);
```

7.3.3.8 Datenbankname für Überschriften (\$DatenbankName)

Hier sollten Sie eine kurze, aber sprechende Bezeichnung eingeben. Dieser Wert wird auf den einzelnen Bildschirmen als Überschrift ausgegeben. Daher müssen Sonderzeichen in HTML-Codierung eingegeben werden. Sie haben hier auch die Möglichkeit, den Namen als Link zu gestalten, bei dessen Aktivierung eine andere Seite erscheint.

Vorgabe:

```
$DatenbankName = "Demo" ;
```

7.3.3.9 Pfad der datenbankspezifischen Hilfsseiten (*\$HilfsseitenPfad*)

Einige Links auf Hilfeseiten werden vom Programm automatisch erzeugt. Der Pfad zu diesen Seiten variiert von Datenbank zu Datenbank oder sollte es zumindest. In *\$HilfsseitenPfad* trägt man diesen Pfad ein. Er ist relativ zum Hauptverzeichnis für die normalen HTML-Seiten anzugeben. Dort müssen dann aber auch die entsprechenden Seiten zu finden sein.

Zur Zeit wird diese Angabe nur von *maske.pl* benutzt, um einen Link auf die Hilfeseite *maske.htm* zu erzeugen.

Vorgabe:

```
$HilfsseitenPfad = '/acwww25/opac' ;
```

7.3.3.10 Links auf andere Seiten (*%AndereDBs*)

Trotz des Namens der Variablen, kann man hier Links angeben, die auf beliebige andere Seiten verweisen. Wenn diese Variable nicht leer ist, erscheinen die aufgeführten Verweise am Ende jedes Bildschirms. Wie aus der Vorgabe unten zu ersehen ist, muss ein Link vollständig angegeben werden.

Vorgabe:

```
%AndereDBs = (
    'zk',    '<a href="/cgi-bin/maske.pl?db=zk">[Katalog der Institute]</A>'
    'zd',    '<a href="/cgi-bin/maske.pl?db=zd">[Zeitschriftendienst des De
);
```

7.3.3.11 Namen der Register für Überschriften (*%RegisterHilfe*)

Jeder Bildschirm mit einem Ausschnitt aus einem Register erhält als Überschrift den Namen dieses Registers, wobei er als Link ausgebildet sein sollte. Dieser Link verzweigt bei Aktivierung auf eine für das Register spezifische Hilfe. Diese Variable sollte für jedes angebotene Register einen Eintrag enthalten.

Die Variable ist ein dictionary mit bestimmten Werten als Schlüssel:

- für die normalen Register gelten die Werte '1' bis '9'; sie sind jeweils als Strings anzugeben.
- *allg* verweist auf eine allgemeine Hilfeseite, die einen Überblick über den prinzipiellen Aufbau der Register geben sollte.
- *benutzung* verweist auf eine Stelle, die erläutert, wie die Register zu benutzen sind.
- *expandiert* enthält einen Link auf die Hilfe zum expandierten Register. Dieser erscheint nur, wenn ein Register expandiert angezeigt wird.

Generell gilt hier, dass es egal ist, ob man jeden Link auf verschiedene Seiten zeigen lässt, oder auf unterschiedliche Abschnitte ein und derselben Datei. Historisch gewachsen ist die Aufteilung in verschiedene Seiten, wobei es für jedes Register jeweils eine Seite und für die allgemeinen Bemerkungen eine eigene Seite gibt. Vorbild waren die Hilfsbildschirme der DOS-Distribution.

Vorgabe:

```
%RegisterHilfe = (
  '1', '<A HREF="/acwww25/opac/hreg1.html">Namen von Personen</A>',
  '2', '<A HREF="/acwww25/opac/hreg2.html">Körperschaften</A>',
  '3', '<A HREF="/acwww25/opac/hreg3.html">Wörter (Titel- und Schlagwört
  '4', '<A HREF="/acwww25/opac/hreg4.html">Buchtitel (Sonderabteilung K
  '5', '<A HREF="/acwww25/opac/hreg5.html">Zeitschriften und Reihen (Ser
  '6', '<A HREF="/acwww25/opac/hreg6.html">Verlage (Name,Erscheinungsjah
  '7', '<A HREF="/acwww25/opac/hreg7.html">Sachgruppen (mit Sonderabteil
  '8', '<A HREF="/acwww25/opac/hreg8.html">Signaturen (insbes. Lesesaal-
  '9', '<A HREF="/acwww25/opac/hreg9.html">ISBN (mit \'i\' davor, also
  'allg', '<A HREF="/acwww25/opac/hreg.html">[Allgemeine Informationen z
  'benutzung', '<A HREF="/acwww25/opac/hreg.html#ALLGINFO">[Wie benutze
  'expandiert', '<A HREF="/acwww25/opac/hreg.html#EXPANDIERT">[Wie benut
);
```

7.3.3.12 Namen der Register für Anzeigezwecke (%RegisterAnzeige)

Der Aufbau dieser Variablen ist ähnlich wie der von %RegisterHilfe. Hier werden nur die Registerüberschriften nochmal für einfache Anzeigezwecke angegeben. Die Hinweise auf allgemeine Abschnitte fehlen.

Vorgabe:

```
%RegisterAnzeige = (
  '1', 'Namen von Personen',
  '2', 'Körperschaften',
  '3', 'Wörter (Titel- und Schlagwörter)',
  '4', 'Buchtitel (Sonderabteilung K : Kongresse)',
  '5', 'Zeitschriften und Reihen (Serientitel ; Bandnummer)',
  '6', 'Verlage (Name,Erscheinungsjahr)',
  '7', 'Sachgruppen (mit Sonderabteilungen G und H)',
  '8', 'Signaturen (insbes. Lesesaal-Standorte)',
  '9', 'ISBN (mit \'i\' davor, also z.B. i3-512-02341)',
);
```

7.3.3.13 Namen der Register fuer die Registerauswahl in den Suchmasken (@RegisterPullDown)

In der Suchmaske wählt man das anzuzeigende Register über ein Dropdown-Menü aus. Der Inhalt dieses Menüs wird aus dieser Variablen erzeugt. Sie ist ein einfache Liste.

Vorgabe:

```
@RegisterPullDown = (
  '1 Namen von Personen',
  '2 Körperschaften',
  '3 Wörter (Titel- und Schlagwörter)',
  '4 Buchtitel (Sonderabteilung K : Kongresse)',
  '5 Zeitschriften und Reihen (Serientitel ; Bandnummer)',
  '6 Verlage (Name,Erscheinungsjahr)',
  '7 Sachgruppen (mit Sonderabteilungen G und H)',
  '8 Signaturen (insbes. Lesesaal-Standorte)',
  '9 ISBN (mit \'i\' davor, also z.B. i3-512-02341)',
);
```

7.3.3.14 Suche über logische Kombinationen (\$logischeKombinationen)

Wenn die durch *maske.pl* automatisch generierte Suchmaske auch Eingabefelder für die Suche nach mehreren Aspekten anbieten soll, muss diese Variable auf 1 gesetzt werden. Wenn das der Fall ist, müssen auch die im folgenden aufgeführten Variablen belegt sein.

Vorgabe:

```
$logischeKombinationen = 1;
```

7.3.3.15 Name der Variablen fuer die Suchmaske (@MaskenVariable)

Jedes Feld der Suchmaske muss einen eigenen Namen haben. Die Schreibweise ist dabei egal. Auch die Benennung ist beliebig. Empfohlen wird jedoch, die Namen 'sprechend' zu wählen. In den nachfolgend beschriebenen Variablen *%VarRegister* und *%VarInputDef* müssen die Schlüssel dann jeweils diesen Namen entsprechen. Auch darf es dort nur soviele Einträge geben, wie hier Variablennamen definiert werden!

Vorgabe:

```
@MaskenVariable = (
    'pers',      # fuer Suche nach Person in Register 1
    'tit1',     # erstes Stichwort (Register 3)
    'tit2',     # zweites Stichwort (Register 3)
    'tit3',     # drittes Stichwort (Register 3)
);
```

7.3.3.16 Zuordnung der Variablen zu den Registern der Datenbank (%VarRegister)

Aus den Eingaben in den Feldern der Suchmaske wird für *avanti* ein find-Befehl zusammengesetzt. Dazu muss das Programm wissen, in welchem Register der Datenbank es mit welchem Suchbegriff suchen soll. Die Zuordnung der Eingabefelder zu einzelnen Registern wird in *%VarRegister* festgelegt. Die Kürzel für die Register entnimmt man der Indexparameterdatei der Datenbank. Es ist die sogenannte I-Tabelle, die man meistens am Ende der Datei findet. Sollten die Registerdefinition dort fehlen, muss man sie anlegen. Näheres dazu finden Sie im Systemhandbuch Kap. 10.2.1.3.

Die Schlüssel dieses dictionary müssen mit den Vorgaben in *@MaskenVariable* übereinstimmen.

Vorgabe:

```
%VarRegister = (
    'pers', 'PER',
    'tit1', 'TIT',
    'tit2', 'TIT',
    'tit3', 'TIT',
);
```

7.3.3.17 Definition der Eingabefelder der Suchmaske (%VarInputDef)

Die Felder der Suchmaske werden automatisch generiert. Sie haben alle denselben Aufbau: Sie benötigen einen Prompt, damit der Benutzer weiß, was er wo einzutragen hat, ein Eingabefeld, dessen Länge festgelegt werden muss und eine Angabe über eine Hilfeseite zu diesem Feld. Der

Prompt wird als Link aufgebaut, so dass man bei einem Klick darauf auf eine Hilfeseite zu diesem Eingabefeld kommt.

Der Wert zu jedem Schlüssel besteht aus einer Zeichenkette. Diese ist in Unterfelder unterteilt, wobei die Felder durch das Pipe-Symbol (|) voneinander getrennt werden. Es muss immer dieselbe Anzahl felder geben:

1. Länge des sichtbaren Eingabefeldes in Zeichen. Eine ganze Zahl
2. Maximale Eingabelänge des Feldes. Eine ganze Zahl. Dieser Wert kann größer sein als die Länge des sichtbaren Feldes.
3. Pfad und Name der Hilfeseite zu diesem Feld relativ zum Dokumentenhauptverzeichnis (*document root*) des Webservers. Man kann für jedes Feld auf dieselbe Datei verweisen und innerhalb dieser Datei ein Lesezeichen anspringen.
4. Prompttext. Bezeichnung des Feldes, damit der Benutzer weiß, was er in das Feld eintragen kann.

Die Schlüssel dieses dictionary müssen mit den Vorgaben in *@MaskenVariable* übereinstimmen.

Vorgabe:

```
%VarInputDef = (
    'pers', '30|50|/acwww25/opac/acautor.html|Name einer Person',
    'tit1', '30|50|/acwww25/opac/actitel.html|1. Titelstichwort',
    'tit2', '30|50|/acwww25/opac/actitel.html|2. Titelstichwort',
    'tit3', '30|50|/acwww25/opac/actitel.html|3. Titelstichwort',
);
```

7.3.3.18 Schalter Kurztitelindex (\$stl)

Tragen Sie hier die Länge der Kurztitelzeilen ein, falls die Datenbank über ein Kurztitelregister verfügt. Ansonsten tragen Sie bitte 0 (Null) ein. Abhängig von diesem Wert kann man die Registeranzeige expandieren oder nicht. Die Länge der Kurztitelzeile finden Sie in der Indexparameterdatei als Befehl `i0`

Vorgabe:

```
$stl = 72;
```

7.3.4 Unterprogramme in *dbinfo.pl*

Es gibt in der Datei *dbinfo.pl* nur ein Unterprogramm: `sub PrintErgebnis`. Diese Routine gibt das Ergebnis einer Recherche für die Titelanzeige aus. In der Vorgabe ist sie ganz einfach gehalten:

```
sub PrintErgebnis
{
    local ( $ErgebnisDatei ) = @_;      # Parameter uebernehmen
    print "<pre>\n";
    print $ErgebnisDatei;
    print "</pre>\n";
    return 1;
}
```

Dieser Routine wird nur ein Parameter übergeben, nämlich das Downloadergebnis in **einer** Zeichenkette. Wie das genau aussieht hängt natürlich von der zur Anzeige benutzten Parameterdatei ab. Die Vorgabe *d-html.apr* erzeugt aber eine recht einfache Ausgabe, die der der DOS-Programme entspricht. Alle Zeilen sind durch die Codes CR LF (ASCII 13 10) voneinander getrennt. Die

Routine `PrintErgebnis` macht nun nichts weiter als vor der Titelausgabe und danach das `Preformatted`-Tag auszugeben.

Sie können diese Routine natürlich nach Belieben erweitern und verändern. Sie benötigen dazu aber solide Perl- und ggf. noch Parametrierkenntnisse.

Selbstverständlich ist es möglich, in der Datei `dbinfo.pl` weitere Variablen und Unterprogramme unterzubringen, wenn sie für die Anbindung einer Datenbank benötigt werden. Sie müssen die Verwendung dieser Ergänzungen aber in den Hauptprogrammen selbst hinzufügen.

7.4 Parametrierung

Hier kann man sich nach Herzenslust austoben. Mitgeliefert wird eine für die Anzeige geeignete Parameterdatei `d-html.apr`. Sie gehört zum Standardschema und ist eine leicht abgewandelte `d-1.apr`. Sie finden sie im Datenbankverzeichnis `avanti-w/avdemo` der Demodatenbank.

Änderungen zur `d-1.apr`:

- Die Köpfe (`ak`-Zeilen) sind deaktiviert, weil diese Datei nur eine Anzeige pro Titelsatz erzeugen soll. Daher geht es gleich bei `#-@ los`. Von dort wird zu den einzelnen Satztypen verzweigt. Vor jedem Satz wird direkt das `<hr>`-Tag (horizontal rule) ausgegeben.
- Die Zeichenumcodierung sollte mit `dx=1` eingeschaltet werden, weil die Umlaute und Sonderzeichen in den wenigsten Fällen schon in korrekter HTML-Codierung in der Datenbank vorliegen dürften. Für Umcodierungszwecke gibt es z.B. die Datei `p-html.apr`, die geeignete Tabellen enthält.
- Die Zahl der Zeilen pro Karte wird mit `f1=0` auf 'unendlich' gesetzt.
- Umlaute in indirekten Prä- und Postfixen müssen in HTML-Ersatzdarstellungen umgewandelt werden, weil die nicht der Umcodierung durch die `p-` oder `q-` Tabellen unterliegen. Dasselbe gilt für die Zwischenteile.
- Wenn Sie verschiedene Codierungen für Ihre Ausgabe anbieten wollen, darf das Einbinden einer Umcodiertabelle (`.apr`-Datei) nicht mit dem `t`-Befehl in der Parameterdatei direkt geschehen. Das Nachladen dieser Dateien geschieht über den `avanti`-Befehl `xport t` in der Jobdatei abhängig von den eingestellten Optionen beim Aufruf von `regsrch.pl`.

7.5 Suchmasken ohne `maske.pl`

In der Standardversion der Schnittstelle sorgt das Programm `maske.pl` für die Erzeugung der Suchmaske. Es ist aber nicht zwingend notwendig, die Suchmaske damit zu erzeugen. Statische HTML-Seiten sind genauso geeignet. Von denen aus werden dann die anderen Programme des Paketes angesprochen. In der Gestaltung dieser Seiten sind Sie vollständig frei, nur die an die Programme übergebenen Parameter müssen stimmen. Das beste ist, Sie erzeugen sich zunächst mit `maske.pl` eine Referenzseite und speichern sie ab. Am HTML-Code können Sie dann studieren, welche Variablen und Parameter die einzelnen Programme benötigen.

8 Der Standardeinstieg für den WWW-Benutzer

Der Einstieg für die Recherche in einer der angebotenen Datenbanken kann beliebig gestaltet werden. Ein Beispiel, wie man es machen kann, zeigt <http://www.biblio.tu-bs.de/Kataloge.html>.

Der Aufbau der Seite ist beliebig. Wichtig ist nur die Gestaltung des Links für die Erzeugung der

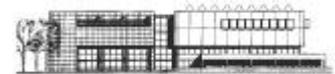
Suchmaske. Er muss in HTML-Notation ungefähr so aussehen:

```
<a href="www.myserver.de/cgi-bin/maske.pl?db=katalog">Recherche</a>
```

Der Ausdruck '/cgi-bin/maske.pl?db=katalog' bewirkt, dass das Script *maske.pl* auf dem Server 'www.myserver.de' die Suchmaske für die Datenbank mit der Kennung 'katalog' erzeugt. Alles weitere folgt dann von der erzeugten Seite aus.

9 Abschließende Hinweise

Die Scripte und Parameterdateien können frei verwendet und den eigenen Bedürfnissen angepaßt werden. Eine Erwähnung der Herkunft aber wäre sehr freundlich.



Copyright © 1999 UB Braunschweig
1999-09-29, Dierk Höppner