



[www.biblio.tu-bs.de/allegro/](http://www.biblio.tu-bs.de/allegro/)

## V15

### Schillers Räuber gefaßt!

*The number forty-six is out!  
What's new, what is it all about?  
Do we need it?  
Must we read it?  
Is it what we're waiting for?  
Even nicer, even better, even more!*

Eines der letzten Grundsatzprobleme, die es im *allegro*-Kernsystem noch zu lösen galt, war das "Schiller-Räuber-Problem". Die "Räuber" waren nicht zu fassen, wenn sie sich in einem verknüpften Untersatz versteckten, während "Schiller" sich nur im Hauptsatz der "Gesammelten Werke" aufhielt, wo er ja auch hingehört. Die Überschrift verrät es schon: jetzt gibt es die Rasterfahndung, die "satzübergreifende Suche", der die "Räuber" nicht mehr entgehen. Darüber hinaus gab es noch zwei andere Grundsatzprobleme, die einer Lösung harren: das Restriktionenproblem (Beschränkung einer Suche nach Erscheinungszeitraum, Sprache o.ä.) und das Problem der zu großen Zwischenergebnisse (eine UND-Verknüpfung konnte nicht ausgeführt werden, wenn die Summe der Zwischenergebnisse zu groß war, auch wenn die Ergebnismenge selbst sehr klein wäre). Welches der Probleme sollte für V15 gelöst werden? Alle drei waren dringend, die Entscheidung war deswegen schwieriger, als die Lösung aller drei Probleme auf einmal. Daher wurde dieser Weg eingeschlagen. Die Zeit reichte dann aber nicht, auch noch alles für die *news* Nr. 45 zu beschreiben. Diese Ausgabe holt das Versäumte nach. Die Programme auf der V15.0-Diskette enthalten die neuen Funktionen noch nicht komplett. Da mit V15 eine Vervollständigung des Kernsystems eingetreten ist, wollen wir in einem kurzen Überblick der Versionen ab V8 nur die Meilensteine aufzählen, um die Neuerungen in den Zusammenhang zu stellen:

1989	8	Erste Version mit Index	
1990	9	Mehrplatzbetrieb; UNIX-Urversion; Ergebnismengen	
1991	10	Mehrere Register	
1992	11.1	<i>CockPit</i>	
	11.2	Nachladetechnik	
1993	12.1	Erste Versionen von <i>aLF</i> und <i>ORDER</i> ; Nummerengenerierung	
	12.2	Kurzanzeige	
1994	13	Neue CFG: 3- und 4-stellige Kategorien; Teilfelder; Konsolidiertes Format	
1995	14	Stammsatz-Verknüpfungen (Mehrdateisystem) DOS-UNIX-Synchronisierung	
1996	14	Klassenbibliothek C++ (plattformunabhängige Kernfunktionen auf Basis V14) Darauf aufsetzend: <ul style="list-style-type: none"><li>· VP (Ergebnismengen-Ordnung)</li><li>· Windows-PRESTO</li><li>· <i>avanti</i>-Server, Web-Anwendungen in Client/Server-Struktur</li><li>· Z39.50-Schnittstelle (Ende 1997)</li></ul>	
1997	15	Vervollständigung der Kernfunktionen (für alle Plattformen): Satzübergreifende Suche ("Schiller-Räuber-Problem") Restriktionen (Einschränkung der Ergebnismenge) Ergebnismengengröße (siehe <i>news</i> 45, letzte Seite)	Seite 6-9 Seite 2-5

Mit der Lösung dieser letzten drei Probleme sind in der Tat Schlußsteine gesetzt, jedenfalls was das Kernsystem anbelangt. Auf der konventionellen Schiene (DOS und UNIX) sind wir nah an einer Endstation angelangt: hier sind nur noch marginale und kosmetische Verbesserungen erreichbar, aber mehr auch kaum noch nötig. Es ist sichergestellt, daß die Datenbanken weiterhin von den konventionellen Plattformen aus benutzbar sind, z.B. auch mit *aLF* und *ORDER*, auch wenn jetzt auf den Windows- und WWW-Plattformen neue Oberflächen hinzukommen. Vom kleinen Einzelplatzsystem auf einem billigen, alten PC bis hinauf zu einer Datenbank mit Millionen von Sätzen unter UNIX und Zugriff per WWW besteht nahtlose Kontinuität in der Struktur. V15.0 erschien im Februar, die definitive V15 auf CD soll im Juli herauskommen (siehe S. 12).

Ein Problem ist nur, daß das V15-Handbuch nun Lücken hat. Redaktionsschluß war im November '96 - die neuen Lösungen kamen später. Die nachfolgenden acht Seiten stellen eine wichtige Ergänzung des Handbuchs dar, welches nicht kurzfristig neu aufgelegt werden kann. Außerdem wird ca. Ende Juni die aktualisierte und erweiterte *Ouvertüre* mitsamt großem *Glossar* erscheinen. Frau Dr. Tews in Leipzig hat ganze Arbeit geleistet! (Preis: DM 25; Ringheftung)

## Restriktionen

Diese Neuerung ist bei weitem die wichtigste. Daher soll sie zuerst behandelt werden, erst danach die "Räuber". Kataloge wachsen mit den Beständen, und so wachsen zwangsläufig auch die Ergebnismengen, die man beim Zugriff erhält. Der häufigste Wunsch ist es dann vermutlich, den Zugriff auf die neueste Literatur zu begrenzen. Genau das konnte man bisher mit *allegro* kaum machen. Zwar kann man die Erscheinungsjahre indexieren (bei CAT.API geschieht dies im Register 6), aber eine Begrenzung der Ergebnisse mit Bedingungen wie "nach 1990" oder "vor 1850" ist damit nicht erreichbar. Weitere Wünsche sind Einschränkungen nach Sprache oder Publikationstyp, falls man diese erfaßt hat. Auch solche Elemente kann man indexieren, aber die Ergebnismengen unter solchen Einträgen werden zu groß, sobald man eine Datenbank mit einigen -zigtausend Datensätzen hat. Andere Systeme kennen Suchbeschränkungen schon lange. An der Endbenutzer-Oberfläche haben die Systeme in der Regel zusätzliche Schalter, Auswahllisten oder Eingabefelder für Sprache, Typ oder Erscheinungsjahr. Es gibt keine allgemein übliche Bezeichnung für solche Funktionen. Z.B. wird von "Sekundäraspekten" gesprochen, im Englischen nennt man derlei Elemente meistens "qualifier" oder "limitation", manchmal findet man auch recht unanschauliche Namen wie z. B. bei Pica "ADI" (Additional Discriminative Information). Um einen möglichst knappen, prägnanten und unverwechselbaren Begriff zu haben, wurde für *allegro* der Terminus **Restriktionen** ausgewählt.

Um aber Restriktionen endlich auch unter *allegro* möglich zu machen, mußte ein neues Konzept her, denn in die bisherige Indextechnik lassen sich solche Funktionen nicht effizient einbauen. Das neue Konzept sollte natürlich einerseits flexibel sein, also unter jeder Konfiguration für jeden gewünschten Zweck parametrierbar, andererseits sollte der Aufwand für den Anwender so gering wie möglich sein.

Die unter diesen Vorgaben nunmehr realisierte Lösung ist eine Ergänzung, keine Änderung der Indexierungstechnik. An bestehenden Indexparametern braucht nichts verändert zu werden, bestehende Indexdateien funktionieren unverändert weiter. Es kommen nur ein paar neue Befehle und eine neue Datei hinzu. Diese erhält den Typ .RES (für **Restriktionen**), also z.B. CAT.RES. Im Prinzip wird sie genauso erzeugt wie die Kurztiteldatei vom Typ .STL.

Die Grundidee ist diese: man hinterlegt in der Datei .RES für jeden Datensatz ein paar Bytes für die Restriktionen, die es geben soll. Zum Beispiel: 4 Bytes für das Erscheinungsjahr, 3 Bytes für die Sprache, 1 Byte für den Publikationstyp. (Die Anzahl Bytes multipliziert mit der Anzahl Datensätze ergibt die Größe der Datei.) Das Programm (APAC, PRESTO, AVANTI etc.) kann hernach diese Bytes benutzen, um für jeden Satz festzustellen, ob er zur Ergebnismenge gehören soll oder nicht. Es gibt damit einen sehr erwünschten Zusatzeffekt: wenn die Restriktion bereits eingeschaltet wird, bevor eine Ergebnismenge gebildet wird, dann wird diese von vornherein kleiner. Somit kann eine Suche auch dann noch Erfolg haben, wenn es im Index z.B. unter einem Stichwort viele -zigtausend Einträge gibt. Allerdings kann die Bildung der Ergebnismenge in solchen Fällen etliche Sekunden dauern, weil ja für jeden Eintrag zusätzliche Vergleiche nötig sind. Ohne eingeschaltete Restriktion ändert sich am Zeitverhalten der Zugriffe nichts.

### Was muß man tun?

Wie immer kommt nichts von selbst, sondern man muß es parametrieren. Wie also produziert man die Einträge für die .RES-Datei? Ganz genau wie für die .STL-Datei muß man bestimmte Zusätze in die **Indexparameter** einbauen:

**ir=k** Länge des Restriktionsschlüssels (entsprechend i0 bei .STL)  
Nur wenn **ir** gesetzt ist, wird Datei *dbn.RES* angelegt, d.h. **ir** darf nicht vergessen werden!

**ak=zz+/** ak-Befehl für den "Restriktionsschlüssel" (entspr. ak=zz+0 bei .STL)

**#- /** Abschnitt für die Erstellung des Restriktionsschlüssels. Dieser muß das Präfix "|/" erhalten.  
**...** (entspr. #-0 und Präfix "|0" bei .STL)

Für die Benutzung (APAC, *avanti*) kommen noch Hilfszeilen in folgender Form hinzu (ebenfalls in der .API):

```
R XXX mp "Text zu dieser Restriktion?Hilfszeile dazu"
  XXX                               symbolischer Name der Restriktion, z.B. ERJ für Ersch.Jahr
  m                                 Modus r, R, s oder S (d.h. die zugehörigen Werte sind in .RES bzw. .STL)
  p                                 Position 1..72 (in .RES bzw. .STL), wo die Werte stehen
  "Text?Hilfe"                       wird z.B. von APAC für das Menü benutzt, die Hilfszeile erscheint unten
```

Beispiel:

```
R ERJ r1 "Erscheinungsjahr?Einschränkung auf einen Zeitraum"
```

Das bedeutet: man hat im .RES-Schlüssel auf Position 1 das Erscheinungsjahr, symbolischer Name soll ERJ sein. Die R-Befehle werden in APAC zu einem Auswahlmenü verarbeitet (s.u.). Dem *avanti*-Server ermöglichen sie das Abarbeiten von Befehlen wie `find ... erj >1990` (s.u.), denn dem Server muß ja gesagt werden, was "erj" bedeutet. Für sich genommen (und für PRESTO) bewirken die R-Zeilen aber gar nichts: die Struktur der .RES (und evtl. .STL) muß damit übereinstimmen, sonst kann die Sache nicht funktionieren. Hauptsächlich kommt es also auf den Abschnitt unter #-/ an. Unter PRESTO kann der Eingeweihte auch Restriktionen nutzen, für die es keine R-Zeilen gibt. Die Modusbuchstaben 'R' und 'S' haben eine etwas andere Wirkung als 'r' und 's', siehe unten.

Aber es gibt sogar die **Alternative**, sich keine *dbn*.RES zu leisten, sondern nur Teile innerhalb der bestehenden .STL für diesen Zweck zu nutzen. Dann sind keine zusätzlichen Maßnahmen nötig, nur die R-Zeilen, und zwar mit einem 's' oder 'S' statt 'r' oder 'R' an der Position m. Die .STL muß dann natürlich geeignete Bestandteile haben, die immer an derselben Position stehen. Jede Position von 1 bis 72 ist dafür nutzbar.

*Nachteil:* Zugriffe auf *dbn*.STL sind viel langsamer auf *dbn*.RES. (bis ca. Faktor 10)

Beides ist kombinierbar, d.h. es kann R-Zeilen mit 'r' oder 'R' und andere mit 's' oder 'S' geben. Der OPAC-Benutzer muß den Zusammenhang und die Positionen nicht kennen - dafür sind die R-Zeilen ja gedacht.

## Neue Funktion für INDEX

Der Aufruf `INDEX -fx ...` produziert die .RES-Datei (analog zu `-fs` für die .STL)

Wenn der Index erneuert wird (mit `-fi`, `-f7` oder `-fn`) wird die .RES automatisch mit erzeugt, genau wie auch die .STL. (Leider konnte man nicht `-fr` nehmen, weil dies schon für das Entlüften belegt ist.)

Im *CockPit* von V15 gibt es einen eigenen Menüpunkt "x = Restriktionsdatei erneuern" unter "organisieren". Hat man die oben beschriebenen Vorkehrungen getroffen, braucht man nur noch diesen Punkt anzuwählen.

## Implementierung in *avanti* (s.a. Dokumentation zum *avanti*-Server 1.2)

Innerhalb eines *avanti*-find-Befehls kann etwa dieses stehen (mit Bezug auf das oben angegebene Beispiel):

```
find ... and erj >1992
```

Dann wird die Ergebnismenge entsprechend dieser Restriktion verkleinert.

Sagt man jedoch einfach nur:

```
find erj >1992
```

so wird die aktuelle (vorher gebildete) Ergebnismenge mit dieser Restriktion verkleinert.

Dies kann man z.B. folgendermaßen nutzen:

```
find tit energie?           // bilde Ergebnismenge
if less 50 jump ausgabe     // sind es weniger als 50 ? dann -> ausgabe
find erj >1990              // sonst Einschränkung auf Ersch.Jahr >1990
:ausgabe
order al                    // aufsteigend ordnen nach Titel (beginnt Pos. 1 in .STL)
list num cod                // Kurzliste numeriert und nach ANSI codiert ausgeben
```

Wenn aber vorher (im selben Auftrag) noch kein find-Befehl vorkam, also noch keine Ergebnismenge vorliegt, wird die Restriktion fest eingestellt, d.h. nachfolgende Suchbefehle (im selben Auftrag) werden so ausgeführt, daß die Restriktion gleich mit berücksichtigt wird. Es ist manchmal ratsam, wenn man mit großen Ergebnismengen rechnet, zuerst (vor dem eigentlichen find-Befehl) die Restriktion zu setzen. Dann werden diese gar nicht erst zu groß.

Zum Beispiel liefern die beiden Befehle

```
find tit deutsch? and erj >1992   bzw.   find erj >1992 and tit deutsch?
```

im Prinzip dasselbe Resultat, aber es kann sein, daß beim ersten Befehl die Menge der Einträge unter "deutsch?" schon zu groß wird! Dann funktioniert der gesamte Befehl nicht. Beim zweiten ist die Chance größer.

Ganz deutlich: Der Befehl `find erj >1992` allein liefert nicht etwa die Menge aller nach 1992 erschienenen Titel! Er verkleinert vielmehr eine bestehende Ergebnismenge, aber wenn noch keine besteht, stellt er für nachfolgende `find`-Befehle diese Restriktion ein.

Will man im selben Auftrag noch einen weiteren, neuen `find`-Befehl ohne Restriktion geben, kann man mit

```
find new
```

die Ergebnismenge **und** die eingestellte Restriktion wieder aufheben. Der nächste `find`-Befehl erzeugt dann eine neue Ergebnismenge.

## Implementierung in PRESTO

Im Gegensatz zu APAC hat PRESTO keine Menüunterstützung. Die Nutzung der Restriktionen unter PRESTO ist daher nicht benutzerfreundlich, jedoch möglich. Man muß die Positionen der `.RES` oder `.STL` und ihre Bedeutung kennen. Auf dem Registerbildschirm drückt man :

**F9** Dann wird die gewünschte Restriktion abgefragt. Die Frage lautet  
`Pos >=< Value?`  
 Man muß den Wunsch dann explizit eingeben, d.h. eine Position, einen Operator und einen Wert, z.B.

```
r1>1990   Vergleich ab Position 1 in .RES, Wert soll >1990 sein
s58<1950  Vergleich ab Pos. 58 in .STL, Wert soll < 1950 sein
s58=1997  Auf Position 58 in der .STL soll 1997 stehen-
s68!XYZ   Auf Position 68 in der .STL soll nicht XYZ stehen
0         Rücksetzen (Restriktion aufheben)
```

Der versierte PRESTO-Anwender kann hier auch Positionen nutzen, die nicht in einer R-Zeile definiert sind!

Der durchgeführte Vergleich ist immer ein *Zeichenvergleich*, kein *Zahlenvergleich*, auch wenn auf der fraglichen Position Zahlen stehen. Die Anzahl der zu vergleichenden Zeichen ergibt sich aus der Eingabe, z.B. 4 wenn man `r1>1990` eingegeben hat, d.h. nachfolgende Zeichen in der `.RES` oder `.STL` haben keinen Einfluß.

Nachdem mit **F9** eine Restriktion gesetzt wurde, wirkt sie sich auf alle nachfolgenden Funktionen (+ / -) zur Ergebnismengenbildung aus. Wenn man also `'/'` drückt, werden nur diejenigen Sätze in die Ergebnismenge genommen, die der Restriktion genügen. Für den Anzeigebildschirm gilt das nicht, und zwar aus gutem Grund: wenn man einen einzelnen Satz auf dem Schirm hat, entscheidet man sich wohl ganz bewußt dafür, ihn in die Ergebnismenge aufzunehmen, und das sollte das Programm dann nicht verhindern.

Zu einem Zeitpunkt kann immer nur eine Restriktion gesetzt sein.

Wenn nichts gesetzt oder die Setzung mit `'0'` wieder aufgehoben ist, funktioniert alles normal wie gewohnt.

Wenn man eine Ergebnismenge gebildet hat, kann nachträglich eine Restriktion angewendet werden. Sie muß dann, nach der Setzung mit **F9**, durch **Strg+F9** auf die bestehende Ergebnismenge angewendet werden. So kann man nacheinander die Ergebnismenge durch zwei oder mehr verschiedene Restriktionen verkleinern, d.h. jeweils mit **F9** und **Strg+F9** unterschiedliche Aktionen durchführen.

Hilfestellung: In PRESTO erhält man auf dem Anzeigeschirm mit `Alt+F7` die vorhandenen Restriktionen angezeigt: es erscheinen die R-Zeilen aus den Indexparametern.

### Und wie sieht es im OPAC aus?

Das Programm APAC macht alles viel einfacher: es erscheint ein neuer Menüpunkt "Einschränkungen" im Indexmenü (Achtung: die Dateien UIFA... wurden dafür um einige Zeilen ergänzt). Man bekommt eine Eingabemaske mit drei Zeilen: Auswahl der Restriktion, des Operators, und Eingabe des gewünschten Wertes. Die ersten zwei erlauben per Hilfsmenü nur die sinnvollen Werte (die sich aus den R-Befehlen ergeben).

Angenommen, wir haben vier Restriktionen und dafür in der CAT.API diese Hilfszeilen:

```
R ERJ r1 "Erscheinungsjahr?Einschränkung auf einen Zeitraum"
R TYP r5 "Typ?z.B. ho=Dissertation af=Aufsatzsammlung"
R FGR r7 "Fachgruppe?z.B. ma=Mathematik, py=Physik"
R TAB s1 "Titel-Anfangsbuchstabe?z.B. G, wenn die Titel mit G anfangen sollen"
```

APAC liefert uns dann auf dem Registerbildschirm über den neuen Menüpunkt "Suche einschränken" dieses Bild:

```

Register 1 : Namen von Personen
1  schwarze, wolfgang
30==>shakespeare, william
1  shakespeare, william #
43  shakespeare, william *
4   shakespeare, william - DRAMATISCHE
4   shakespeare, william / bibliographi
7   shakespeare, william / biographie
11  shakespeare, william / drama
1   shakespeare, william / drittes reic
8   shakespeare, william / festschrift
1   shakespeare, william / henry iv
1   shakespeare, william / henry v
2   shakespeare, william / imagery
2   sha
1   sha
4   sha
2   sha
1   sha
1   shakespeare, william / n
5   shakespeare, william / r
1   shakespeare, william / r

```

INDEX-Funktionen

Zugriff auf Titel  
 Register-Erweiterung ein/aus  
 Kurzliste der Ergebnisse ein/aus  
 Andere Stelle im selben Register  
 H : Hilfe zum Register  
 1 : Namensregister  
 2 : Körp.Reg.  
 3 : Wortregister  
 4 : Titelregister  
 5 : Zeitschriften- und Serientitel  
 6 : Verlage/Orte  
 7 : Sachgruppen

**Einschränkung** [F10] = ok

**Suche einschränken**  
 und zwar:  
 gewünschter Wert

1 : Erscheinungsjahr  
 2 : Typ  
 3 : Fachgruppe  
 4 : Titel-Anfangsbuchstaben

lten  
 nge hinzufügen  
 #lden  
 Menge rausnehmen  
 setzen

Andere Stelle? Suchwort eintippen. <Sh+ Verknüpfungen berücksichtigen  
 Einschränkung auf einen Zeitraum **Suche einschränken**  
 x : Ausstieg F10 = Einschr. aus

In der Menüzeile "und zwar:" hat man die Wahl zwischen den vier Vergleichsfunktionen < > ! und =. Wenn man "Erscheinungsjahr" gewählt hat, sieht das dann so aus:

```

2   sha
1   sha
4   sha
2   sha
1   sha
1   shakespeare, william / n
5   shakespeare, william / r
1   shakespeare, william / r
4   shakespeare, william / s

```

**Einschränkung** [F10] = ok

**Suche einschränken** 1 : Erscheinungsjahr

**und zwar:**

**gewünschter Wert**

> **Größer**  
 < **Kleiner**  
 = **Gleich**  
 ! **Ungleich**

lten  
 nge hinzufügen  
 #lden  
 Titel aus Erg.Menge rausnehmen  
 bnmengene rücksetzen  
 nüpfnungen berücksichtigen  
 e einschränken

Unter "gewünschter Wert" gibt man schließlich den Begriff ein, der für die Restriktion verwendet werden soll, also z.B. eine Jahreszahl. Mit [F10] wird die Restriktion dann dem Programm übergeben. Wenn der Leuchtbalken auf "gewünschter Wert" steht, sieht man unten einen Hinweis " >>> F1 = Hilfe <<<". Damit aber Hilfe kommt, muß man eine Hilfsseite haben. Diese Seiten müssen HRES1, HRES2 usw. heißen, d.h. für jede der Restriktionen kann man eine spezifische Hilfsseite einrichten. Am besten legt man diese Seiten in das Datenverzeichnis, denn in der Regel werden sie datenbankspezifisch sein.

**Besonderheiten**

Wenn zum Zeitpunkt dieser Menüauswahl schon eine Ergebnismenge besteht, wird diese gleich anschließend der gewählten Restriktion unterzogen, also verkleinert. Vermutlich ist dies genau das, was der OPAC-Benutzer erwartet. Der PRESTO-Benutzer muß Strg+F9 drücken, um die Restriktion ausführen zu lassen. Die eingeschaltete Restriktion wirkt sich dann aber auf alle nachfolgende Operationen aus, bis man eine andere einstellt oder sie vom Indexmenü mit [F10] wieder aufhebt. In der vorletzten Zeile unten rechts wird die eingestellte Restriktion angezeigt, auch unter PRESTO.

Der Systemverwalter hat die Möglichkeit, die Restriktionen auf unterschiedliche Art in APAC zu implementieren: Die Restriktionen mit 'R' oder 'S' (statt 'r' oder 's') haben eine etwas andere Wirkung: VOR Anwendung der Restriktion wird die Ergebnismenge expandiert (siehe Satzübergreifende Suche, nächste Seite), d.h. als ob man zuerst '&' drücken würde. Außerdem wird die Restriktion nach einmaliger Anwendung wieder abgeschaltet. Beim OPAC muß man ja bedenken, daß ein Benutzer jederzeit das Gerät verlassen kann. Der nächste kommt und bemerkt gar nicht die eingeschaltete Restriktion, aber wundert sich dann natürlich über die Ergebnisse, die er bekommt oder eben nicht bekommt.

**Abschalten der Restriktion**

Auf dem ESC-Menü des Registerschirms gibt es den Hinweis: F10 : Einschr. aus. Zusätzlich kann die Restriktion auch mit der Rücktaste [←] abgeschaltet werden; siehe unten bei APAC (Seite 10).

## Satzübergreifende Suche : Das "Schiller-Räuber"-Problem

Seit es die verknüpften Sätze für mehrbändige Werke gibt (1993, V11.2), ist immer wieder einmal ein Disput über das "Schiller-Räuber-Problem" aufgeflammt. Dabei handelt es sich nicht um ein *allegro*-spezifisches Problem, sondern alle bekannten Systeme kranken daran (außer MARC-Systemen, weil sie keine Satzverknüpfungen kennen). Keine ganz geringe Herausforderung also, und es war lange Zeit unklar, wie dieses Problem am besten anzupacken wäre. Schließlich und endlich wurde für *allegro* eine sinnvolle Lösung gefunden und realisiert. Die folgenden, zur Lösung hinführenden Gedanken sollen zunächst das Problemverständnis fördern und mit instruktiven Beispielen den Lösungsweg begründen.

### Was ist überhaupt das Schiller-Räuber-Problem?

Wenn ein mehrteiliges Werk "Schillers Dramen" heißt, der Name "Schiller" aber nur im Hauptsatz und der Titel "Räuber" in einem **verknüpften** Untersatz steht, kann man nichts finden mit einer logischen UND-Verknüpfung zwischen `schiller` und `raeuber`.

*Warum nicht?* Zur Bildung von Ergebnismengen (und das gilt nicht nur für *allegro*) werden aus dem Index nur die Satznummern entnommen und verglichen, die beiden Suchwörter stehen aber in getrennten Sätzen. Das logische UND sieht also zwei ungleiche Nummern und erfaßt sie deshalb nicht.

Eine gängige Abhilfe besteht darin, den Namen des Verfassers auch in jeden Untersatz mit einzutragen. Das erzeugt erstens unerwünschte Redundanz, löst aber zweitens nur einen Teil der Probleme dieses Typs, denn auch für andere Suchbegriffe, nicht nur für Verfassernamen, wird u.U. eine **satzübergreifende** Suche gewünscht oder stillschweigend erwartet, z.B. für Schlagwörter.

Hier ein anderes Beispiel, damit es noch deutlicher wird (nicht erfunden, dieses Werk gibt es!)

Wir lassen alles weg, was mit dem Problem nichts zu tun hat.

---

Hauptsatz: Vollmer, Gerhard:  
 Nr.A Was können wir wissen?  
*Schlagwort:* Erkenntnistheorie

---

Untersatz: Band 1  
 Nr.B Die Natur der Erkenntnis : Beiträge zur Evolutionären  
 Erkenntnistheorie.  
*Mitverf.:* Lorenz, Konrad.

---

Untersatz: Band 2  
 Nr.C Die Erkenntnis der Natur : Beiträge zur modernen  
 Naturphilosophie.  
*Schlagwort:* Naturphilosophie

---

Die Nummern A, B und C sollen die internen Satznummern der drei Datensätze sein, nicht die Primärschlüssel. Die Ergebnismengen werden stets aus diesen Nummern gebildet, nicht aus Bestandteilen der Datensätze!

Gegenwärtige Systeme erbringen in der Regel folgende Resultate, manchmal eben weniger als erwünscht:

(Der Leser überlege für sich selbst, was wohl sinnvoll ist. Wir geben hier bewußt ziemlich viele Beispiele, damit deutlich wird, daß nicht jedes denkbare Ergebnis unbedingt erwünscht ist.)

Suchfrage:	Resultat:	erwünscht:
1. vollmer	A	A, B, C ???
2. wissen	A	A, B, C ???
3. erkenntnis	B, C	

4a. erkenntnis?	A, B, C		
4b. naturphilosophie	C	A, C	???
4c. erkenntnisth? AND naturphil?	nichts	C	
5. vollmer AND wissen	A	A, B, C	???
6. wissen AND natur	nichts	B, C	???
7. vollmer AND natur	nichts	B, C	(entspricht schiller-raeuber)
8. vollmer AND naturphilosophie	nichts	C	
vollmer AND lorenz	nichts	B	
9. lorenz AND erkenntnis	B		
10. lorenz AND wissen	nichts	B	???

(Die Suchfragen sind hier vereinfacht dargestellt. Je nach Syntax des Systems sieh der eigentliche Befehl anders aus, z.B. bei Pica: "find per vollmer? and tit natur" heißen. Das '?' ist auch bei Pica das Trunkierungssymbol.)

In den Fällen 1,2,4b,5,6 und 10 kann man wohl nicht generell sagen, daß die erweiterten Ergebnisse in jedem Fall erwünscht sind, sicher trifft das nur für die Fälle 4c, 7 und 8 zu. Es gibt natürlich keinen Algorithmus, der den tatsächlichen Wunsch erschließen kann und in jedem Fall korrekt eine erweiterte Suche (satzübergreifende Suche) durchführt oder nicht. Folglich wird man dem Benutzer auf irgendeine Art eine neue Option "erweiterte Suche" zur Wahl stellen müssen. Das eigentliche Problem tritt nur bei den UND-Verknüpfungen auf. Die Fälle 1 bis 3 gehören nicht zum Problembereich. Wenn A gefunden wird, hat man implizit auch B und C, da diese dann ohnehin mit angezeigt werden können. Die nachfolgend skizzierte Lösungsidee wird zwar auch die logische NICHT-Verknüpfung mit einbeziehen (siehe unten Beispiele 5,6,10), aber der Leser mag sich selbst überlegen, wie sinnvoll das ist.

**Abstrakte Lösungsidee**

Es wird ein Steuerzeichen eingeführt, sagen wir '&'. Dieses kann vor jeden Suchbegriff einer kombinierten Suche gesetzt werden.

Also z.B. &vollmer statt vollmer.

Ein so gekennzeichnete Begriff soll "erweitert" (satzübergreifend) gesucht werden. (Man könnte auch von einer "Plus-Suche" sprechen.) Das bedeutet: alle Untersätze werden zu den Ergebnissen hinzugenommen. (Nur die UNTERsätze, nicht die ÜBERgeordneten!) Dann würde also im Beispiel &vollmer A,B,C herauskommen statt nur schlicht A. Mit &lorenz würde aber unverändert nur B herauskommen. Das erscheint sinnvoll, denn Vollmer hat viel mit B und C zu tun, Lorenz dagegen nicht viel mit A und womöglich gar nichts mit B. Diese Überlegung trifft wohl auch für andere Suchkriterien zu, denn schließlich erfaßt man ja ganz bewußt im Hauptsatz diejenigen Elemente, die auf das Gesamtwerk zutreffen (also AUCH auf die Untersätze), in den Untersätzen aber diejenigen, die NUR für den jeweiligen Untersatz gelten.

Also nochmal ganz klar: die "erweiterte Suche" bezieht zu jedem Datensatz alle **unter**geordneten Sätze mit ein, und dies muß auf jeden Term einer kombinierten Suche einzeln anwendbar sein, sonst können durchaus unerwünschte Ergebnisse entstehen.

Die nachfolgenden Beispiele sind nicht alle sinnvoll, d.h. man würde sie in der Praxis nicht alle anwenden. Das heißt, es wäre wohl nicht sinnvoll, die Erweiterung generell dauerhaft einzuschalten, also als ob automatisch vor jedes Suchwort ein '&' gesetzt würde.

Suchfrage:	Ergebnisse:
1. vollmer &vollmer	A A, B, C
2. wissen &wissen	A A, B, C
3. erkenntnis &erkenntnis	B, C B, C

4.	erkenntnis?	A,B,C
	&erkenntnis?	A,B,C
	erkenntnis? AND naturphil?	C
	erkenntnisth? AND naturphil?	nichts
	&erkenntnisth? AND &naturphil?	C
5.	vollmer AND wissen	A
	&vollmer AND wissen	A
	vollmer AND &wissen	A
	&vollmer AND &wissen	A,B,C
	vollmer NOT wissen	nichts
	&vollmer NOT wissen	B,C
	&vollmer NOT &wissen	nichts
6.	vollmer AND natur	nichts
	&vollmer AND natur	B,C
	vollmer AND &natur	nichts
	&vollmer AND &natur	B,C
	vollmer NOT natur	A
	&vollmer NOT natur	A
7.	wissen AND natur	nichts
	&wissen AND natur	B,C
	wissen AND &natur	nichts
	&wissen AND &natur	B,C
8.	vollmer AND naturphilosophie	nichts
	&vollmer AND naturphil?	C
	vollmer and lorenz	nichts
	&vollmer AND &lorenz	B
9.	lorenz AND erkenntnis	B
	&lorenz AND erkenntnis	B
	lorenz AND &erkenntnis	B
	&lorenz AND &erkenntnis	B
10.	lorenz AND wissen	nichts
	&lorenz AND wissen	nichts
	lorenz AND &wissen	B
	&lorenz AND &wissen	B
	lorenz NOT wissen	B
	lorenz NOT &wissen	nichts

Es wird hieraus deutlich, daß mit der "Plus-Suche" jede gewünschte Erweiterung des Zugriffs differenziert durchgeführt werden *kann*. Voraussetzung: man muß sich klar sein, was man will, und die Plus-Suche *bewußt* einsetzen.

*Ob* dies verstanden und ausgenutzt werden wird und *wie oft*, das ist eine ganz andere Frage. Man kann auch die Meinung hören, daß die Booleschen Verknüpfungen insgesamt von kaum einem Endbenutzer verstanden werden, wie auch immer man sie verpackt... Andererseits kann man sicher sein, daß die Wünsche nicht verstummen, nachdem dieses Verfahren realisiert ist. Mindestens wird die Frage kommen, wie es denn ist, wenn einer der Beispielsätze, oder alle drei, noch einem anderen oder mehreren anderen Werken untergeordnet sind, und wenn es mehrere Hierarchiestufen gäbe...

Wahr ist aber auch: je größer unsere Datenbanken werden, umso mehr brauchen wir differenzierte Suchmöglichkeiten.

### Was muß man parametrieren?

Zum Glück wenig. Es genügen ein paar für Kenner sehr einfache Maßnahmen in den Indexparametern.

#### 1. Neuer Befehl i7

Mit dem neuen Befehl **i7** wird zum einen das Register festgelegt, wo die notwendigen Hilfsschlüssel anzusiedeln sind, zum andern wird damit dem System signalisiert, **daß** die Datenbank eine Plus-Suche gestattet.

**i7=k**      Nummer des SR-Hilfsregisters (1 <= k <= 11)    Empfehlung: k= 9 oder 10.

## 2. SR-Hilfsschlüssel: neuer Manipulationsbefehl X

Jeder Untersatz, der bei einer Plus-Suche auffindbar sein soll, muß einen oder mehrere zusätzliche Schlüssel bekommen. Ein solcher "SR-Hilfsschlüssel" muß aus dem Primärschlüssel eines anderen Satzes bestehen (i.d.R. ist das der Hauptsatz eines mehrbändigen Werkes). Es kann zu einem Untersatz mehr als einen Hilfsschlüssel geben, d.h. die Unterordnung unter zwei oder mehr Gesamtwerke kann adäquat berücksichtigt werden.

Ein solcher Hilfsschlüssel wird mit Hilfe des neuen Manipulationsbefehls X angelegt. Wie das geht, zeigen wir an einem *Beispiel*: Wir gehen vom Konsolidierten Format aus, wo der Primärschlüssel aus #00 gebildet wird und Untersätze in der #00 die IdNummer des Hauptsatzes mit angehängtem '+' enthalten. Die Primärschlüssel stehen im Register 9. Die Hilfsschlüssel sollen aber im Register 8 stehen. Es folgen die notwendigen Zeilen:

(Eine reservierte Sprungmarke gibt es für diesen Zweck übrigens nicht.)

```
i7=8           Die SR-Hilfsschlüssel sollen ins Register 8
...
ak=zz+U       Für jeden Satz wird der Abschnitt #-U angesprochen ...
...
#-U           ... aber nur bei Untersätzen passiert hier wirklich etwas:
#00 +#00 c"+" e0 Kommt '+' vor, d.h. ist es ein Untersatz?
#+-           wenn nicht, dann keine Aktion
#00 y0 e"+" x9 SR-Hilfsschlüssel aus Inhalt von #00 per Reg. 9 bilden.
#+#           (Für Register 10 muß es X: heißen; dann aber ':' aus Befehl i2 herausnehmen!)
```

### Und was passiert dann?

Das braucht man nicht unbedingt zu wissen, aber nützlich zum Verständnis dürfte es sein:

Das Programm INDEX bildet zunächst nur einen temporären Schlüssel im Register 1. Wenn man z.B. #00 55555+17 in einem Untersatz hat, sieht der temporäre Schlüssel so aus: \ | 955555 (Mit F7 in PRESTO sieht man das). Im zweiten Durchlauf macht INDEX daraus den eigentlichen Hilfsschlüssel. Wenn der Hauptsatz (der mit dem Primärschlüssel 55555 im Register 9) die interne Nummer 123 hat, dann lautet der SR-Hilfsschlüssel | 123 und steht im Register 8. Das besagt also: der Untersatz gehört zum Satz mit der internen Nummer 123.

Die Hilfsschlüssel dürfen auch im selben Register stehen wie die Primärschlüssel.

Wenn die Plus-Suche dann ausgelöst wird, spielt sich folgendes ab:

Das Programm nimmt sich die aktuelle Ergebnismenge vor. Diese besteht für das Programm immer nur aus einer Liste von internen Satznummern. (Wohlgemerkt: nur diese sind in dem Moment greifbar, nicht die Primärschlüssel.) Für jede dieser Nummern wird im Register i7 (im Beispiel also Register 8) nachgesehen, ob es Hilfeinträge unter dieser Nummer gibt. Wenn ja, werden die zugehörigen internen Nummern zur Ergebnismenge hinzugenommen.

Daraus ergibt sich: die Plus-Suche muß auf einen Suchbegriff angewendet werden, **bevor** die logische UND-Verknüpfung ausgeführt wird. Nochmals: was ist dafür zu tun?

In PRESTO und APAC kann jeweils nur die zum ersten Suchbegriff gehörige Ergebnismenge erweitern. In den Beispielen entspricht das dem Fall &vollmer AND natur, also genau dem originalen Schiller-Räuber-Problem.

Man bilde also zuerst die Ergebnismenge zu demjenigen Begriff, der tatsächlich erweitert werden soll (d.h. der möglicherweise in einem Hauptsatz vorkommt, z.B. schiller oder vollmer). Danach Plus-Suche auslösen:

In PRESTO wird '&' gedrückt (auf dem Registerbildschirm).

In APAC ist dafür auf dem ESC-Menü die neue Funktion "Verknüpfungen berücksichtigen" zu finden.

Die Ergebnismenge wird dann etwas größer, aber nur, wenn es zu mindestens einem der Sätze verknüpfte Untersätze gibt, für die ein SR-Hilfsschlüssel gebildet wurde. Nun führt man normal die Verknüpfung mit dem zweiten Begriff aus (also raeuber bzw. natur).

Für PRESTO und APAC gibt es auch den "Hauptschalter": wenn mit der neuen Option +- gestartet wird, ist die Plus-Suche fest eingeschaltet. Man überlege, ob dann nicht in vielen Fällen zuviel kommt.

### Oberflächenfragen

Ein paar Worte noch zur Präsentation an der Benutzeroberfläche. In der Praxis würde man natürlich nicht vom Endbenutzer verlangen, ein '&' vor jeden zu erweiternden Begriff zu setzen. Das wäre archaisch! (Man würde ja dem Benutzer ein Wissen oder eine Erkenntnis abverlangen, das war noch nie gefragt und wäre heute vollends unverkäuflich.) Man würde in einer Web-Suchmaske zu jedem Suchbegriff einen Schalter setzen, der die Erweiterung für diesen Begriff einschalten würde. Wie für die Restriktionen gilt auch hier: innerhalb der Programme PRESTO und APAC ist eine wirklich benutzerfreundliche, also glasklar verständliche, Implementierung nicht möglich. Für die WWW-Schnittstelle auf der Grundlage des *avanti*-Servers soll in Bälde versucht werden, eine Beispiel-Implementierung zu machen.

In der *avanti*-Sprache braucht man jedenfalls nur das '&' vor jeden Begriff zu setzen, der satzübergreifend gesucht werden soll (siehe oben unter "Abstrakte Lösungsidee"). Anders als bei PRESTO/APAC kann dies jeder der verknüpften Suchbegriffe sein.

Beispiele: (s.o. die Beispiele 5 und 8)

```
find per &vollmer? and tit &wissen
find per &vollmer? and &lorenz?
```

Die korrekte Behandlung der Indexparameter für die Hilfsschlüssel ist in die Programme PRESTO, MENUED, ALFA, INDEX und UPDATE schon in der ausgelieferten V15.0 eingebaut, desgleichen in den *avanti*-Server.

## APAC

Dieses Programm wurde über die V15.0 hinaus noch in ein paar wichtigen Punkten verbessert. Vorgestellt wurde im Kapitel "Restriktionen" (S. 2ff) schon das Untermenü, das man mit Hilfe der R-Zeilen einrichten kann. Daneben gibt es noch weitere Kleinigkeiten, die aber auch für PRESTO etc. gelten:

### Die Rücktaste kann jetzt mehr

Die Rücktaste (Backspace) hat bislang nur die Funktion gehabt, die Ergebnismenge auf 0 zu setzen. Das bleibt auch so. Wenn aber die Taste mehrfach hintereinander betätigt wird, können noch weitere Zustände auf den Normalwert zurückgesetzt werden, womit dann im Endeffekt sehr schnell das Programm in den Anfangszustand versetzt werden kann. Wichtig ist das besonders beim OPAC, da oftmals ein Benutzer mit gefundener Signatur spornstreichs davoneilt und das Programm mit diversen Einstellungen zurückläßt, die den nachfolgenden Benutzer heillos verwirren. Es handelt sich um folgende vier Zustände, die durch die Rücktaste nacheinander aufgehoben werden, und zwar in folgender Reihenfolge, die auch logisch plausibel sein dürfte:

1. Kurzanzeige abschalten
2. Ergebnismenge auf 0 setzen
3. eingestellte Restriktion aufheben
4. eingestellte Trunkierung aufheben

Wenn z.B. die Zustände 1 und 3 ohnehin schon auf normal stehen, wird der erste Druck auf Backspace die Ergebnismenge auf 0 setzen, der zweite die Trunkierung aufheben.

### Hilfsseiten für APAC und PRESTO

Die neueren Funktionen wurden in die Hilfsseiten H10 (für den Anzeigeschirm) und H11 (Registerschirm) eingebaut, und zwar in die deutschen und englischen Versionen. Wenn man diese selber modifiziert hat, sollte man sich die neuen Versionen anschauen, um daraus die neuen Teile zu übernehmen.

Die Hilfsseiten HRES1 usw. für APAC zu den Restriktionen wurden auch erst nach Lieferung der V15.0 ermöglicht.

## PRESTO

### Fehler in Abfrageliste: Typwahl nicht möglich

Peinlich ist ein Fehler in der Abfrageliste der ausgelieferten \$A.CFG. Wenn man mit 'I' eine Neueingabe beginnen will, erscheint zwar die Typabfrage Typ 1 = selbständig (Buch) 2 = unselbständig ..., aber dann kommt sofort die Abfrage von #40, d.h. man kann keinen Typ wählen, sondern es beginnt sofort der Abschnitt für Typ 1.

Ursache des Fehlers: die Typzeile ist genau ein Zeichen zu lang geraten.

*Abhilfe:* Per *CockPit* über Optionen/Konfiguration die Datei \$A.CFG ansteuern, in die Bearbeitung gehen (mit 'e'), dann folgende Tasten hintereinander betätigen (wenn Sie mit dem X-Editor nicht vertraut sind, geben Sie ganz genau das ein, was hier gezeigt wird):

```
f(Buch) [Enter] [Entf] q s
```

Dann haben Sie ein überzähliges Leerzeichen beseitigt, und es funktioniert.

### Stichwörter im Register 1 ???

Wenn Sie darüber nach dem Neu-Indexieren erschrecken: keine Panik, Ihre Daten sind völlig in Ordnung, nur die Parameter nicht. Die Datei CAT.API muß korrigiert werden: in dem Abschnitt unterhalb #-@ sind mehrere Zeilen, die verlagert werden müssen. Die erste Zeile beginnt mit #nr dk1 dk2 ..., die letzte mit #00 I4,z p"|3" e2 dty...

Verschieben Sie diese Zeilen in den Abschnitt, der auf die Zeile #-0 folgt, und zwar unter die Zeile #hi +- .

Die korrigierte Datei liegt auch auf dem Verzeichnis PARAM des FTP-Servers bereit.

**"Trossinger Rösselsprung"**

So haben Kollegen an der Staatlichen Hochschule für Musik und der Bundesakademie für Jugendmusik in Trossingen ihr Verfahren genannt, in zusammengesetzten Registern mit länglichen Einträgen schnell an eine andere Stelle springen zu können. Dazu wird die Umcodiererroutine in den Indexparametern ausgenutzt. Bei der *bolero*-Parametrierung stehen im Register 2 Einträge der Form "Komponist: Einheitstitel". Steht man dort auf einer Zeile mit "bach, johann sebastian: ..." und gibt ein "viol+", so landet man bei "bach, johann sebastian: viol...". Wie funktioniert das? Nun, in den Indexparametern steht dieses:

```
#-2      Umcodierung der Benutzereingabe für Reg.2
#u1 +#uxa c"+" e0      Enthält die Benutzereingabe ein + ?
!u1      Nein, dann normal nach Tabelle 2 umcodieren
#+#
!uxa e": " P": "      Ja, dann aktuelle Zeile nehmen, hinter ": " abschneiden
!u1 F"+"      und Benutzereingabe anhängen
#+#
```

Als Ergebnis dieser Umcodierung hat man "bach, johann sebastian: viol", und dahin wird dann gesprungen. *Aber:* es klappt nur, wenn man vor Eingabe von "viol+" kurz zweimal Enter drückt, denn dann ist #uxa mit "bach..." belegt. Natürlich ist das etwas für Insider, einem "normalen" OPAC-Benutzer kann man es kaum klarmachen. Alternativ kann man [Shift+F6] drücken, dann hat man im Eingabefenster die aktuelle Zeile und kann sie ändern.

## UPDATE

**Datei .RES wurde nicht aktualisiert**

Die mit V15.0 ausgelieferte UPDATE.EXE hatte den Fehler, die neue Datei *dbn.RES* nicht zu aktualisieren. Dieser wurde kurz darauf behoben. Per FTP ist das korrigierte UPDATE.EXE erhältlich und wird natürlich auf der CD mitgeliefert.

**Option -N: Fehler im Handbuch**

Leider ganz falsch beschrieben wird im Handbuch V15 die neue Option -N (S. 274, Kap. 12.1). Korrekt ist dieses:

- N0 Leersätze nicht zum Speichern nutzen
- N1 Alle Leersätze nutzen, egal in welcher Datei sie stehen
- N2 (default) Nur die Leersätze in der Datei, die durch Option -n angegeben ist

Das alles betrifft nur neue Datensätze, nicht korrigierte. Diese werden auf jeden Fall in derselben Datei abgelegt, wo sie vorher standen, auch wenn sie verlängert wurden.

Am schnellsten läuft UPDATE bei -N0, aber man sollte dann gelegentlich entlüften, weil sich immer mehr Leersätze ansammeln. Auch -N1 geht recht schnell, aber wenn man sich das Arrangement der Dateinummern gut überlegt hat, bringt diese Option die Daten durcheinander. (Das schadet eigentlich den Datensätzen nicht, nur ist es nicht im Sinne des Datenbankbetreibers.) Mit -N2 läuft UPDATE nur dann langsamer, wenn es viele gelöschte Sätze gibt und mehr als eine Datenbankdatei.

## INDEX / QRIX

**Zwischendateien**

Die von INDEX erstellten und von QRIX zusammenzumischenden Zwischendateien wurden mit den Namen *i\$1*, *i\$2*, usw. bezeichnet. Unter UNIX ist das Zeichen '\$' in Dateinamen problematisch. Da es ohnehin keine Funktion hat, wurden diese Hilfsnamen geändert: jetzt heißen die Dateien *ii1*, *ii2*, ...

Im Normalfall wird man das nicht bemerken, weil die Dateien nach getaner Arbeit von QRIX gelöscht werden. Das einzige Problem ist, daß das neue INDEX nicht mit einem alten QRIX zusammenarbeiten kann und umgekehrt. Es gibt aber keinen Grund, warum man dies würde machen wollen, und bei der V15-Installation werden ja beide Programme durch die neuen ersetzt. Die sonstigen Funktionen von QRIX bleiben unberührt.

**V14-Steuerzeichen**

INDEX von V15.0 hatte in dem Durchlauf mit -@2 noch eine Ungenauigkeit: wenn man das Steuerzeichen *i5* (normalerweise '\_') auch hinter die Kürzel setzt, wurde das hintere Zeichen nicht eliminiert, d.h. es blieb hinter dem Klartext stehen, der für das Kürzel eingesetzt wurde. Dieses Problem ist inzwischen bereinigt.

## Export

### Schaltbefehl #<^M

Dieser schaltet wieder "hoch" zu einem vorher schon nachgeladenen Satz, wenn man zwischenzeitlich mit #< vom nachgeladenen zum Ausgangssatz zurückgeschaltet hatte. Der Sprung zur Marke #-M soll nur dann erfolgen, wenn tatsächlich ein solcher Satz vorhanden war, das Schalten also ausgeführt werden konnte. Bis V15.0 wurde aber in jedem Fall gesprungen. Dies wurde korrigiert. Das Handbuch ist damit nun auch in diesem Punkt korrekt.

### Indexparameter

Die Werte des Parameters i2 (10.2.1.3, S.176) sollen nur gelten für echte Registereinträge, nicht für die Kurzzeile und die Restriktionsdaten (Präfixe |0 und |/). V15.0 war hier noch unkorrekt und wurde nachgebessert.

## V15a? V15b? Nein, V15cd

Es wird keine V15a und keine V15b geben, aber eine V15 auf CD. Es ist geplant, diese im Juli '97 herauszubringen, also in knapp zwei Monaten. Weitgehend steht der Lieferumfang jetzt fest. Die Programme sind seit Auslieferung der V15.0 in Kleinigkeiten verbessert und korrigiert worden, worüber in dieser Ausgabe berichtet wird. Größere funktionale Veränderungen gegenüber V15.0 gibt es nicht, so daß die Zeit bis zum Pressen der CD als Testperiode ausreichend sein sollte. Was die DOS-Schiene angeht, wird die V15cd dann schon sehr nah an einer definitiven, abschließenden Version liegen.

Was aber wird alles drauf sein auf der CD? Hier eine vorläufige Inhaltsübersicht, zu der wahrscheinlich noch einige Dinge hinzukommen, da noch genügend Platz bleibt:

0. Ein Installationsprogramm, das dem Windows-Standard entspricht. Jeder Abonnent erhält eine 13stellige Kennzahl und kann die ihm zustehenden Programme nur mit Hilfe dieser Zahl installieren. Bis auf die Programme ist jedoch alles frei zugänglich, insbesondere die Datenbanken.
1. V15-Kernprogramme (Kopien der Lieferdisketten incl. Demo-Datenbank). Bei Bedarf kann das Installationsprogramm daraus echte Disketten für das Installieren auf anderen PCs erstellen, die kein CD-Laufwerk haben.
2. Programme der "neuen Generation: Windows-Presto; VPW, *avanti* für Windows 3.1, *avanti-W* für Windows'95 und Windows-NT (s. S. 13)
3. *aLF* und *ORDER* auf dem Level von V15
4. Pakete: Allers-Lernsystem, Glossardatenbank, EXPEX, QUEX, QUANT, PRONTO, OPUS.
5. WWW-Skripte (HTML und Perl) mit Dokumentation
6. Datenbanken: (auch für Demo-Zwecke gedacht)  
OPAC der UB Braunschweig (600.000 Sätze, ca. 270 MB)  
Braunschweiger Forschungsbibliographie (55.000 Sätze)  
USMARC-Musikdatenbank (44.000 Sätze)  
*bolero*-Demodatenbank (ca. 2.500 Sätze)  
Formate-Datenbank auf aktuellem Stand (incl. MAB2)
7. Texte verschiedener Art: *allegro news* ab Nr. 20 (WordPerfect, PostScript und HTML), Konsolidiertes Format, Archivdateien der E-mail-Liste; HTML-Texte vom Web-Server der UB.

Damit die Datenbanken frei genutzt werden können, wird APACG als ausführbares Programm vorhanden sein. Die CD soll nämlich auch als Demo-Disk verwendet sowie an Benutzer der UB preisgünstig abgegeben werden. Deshalb gibt es den Schutzmechanismus mit den Kennzahlen für die Abonnenten: nur damit lassen sich die Programme entpacken, und die Kennzahl enthält jeweils die IdNummer des Anwenders. Über diese könnten Raubkopien einwandfrei identifiziert werden. Jeder wird einsehen, daß wir eine Schutzmaßnahme ergreifen müssen. Die Unbequemlichkeit für den Anwender ist minimal. Wenn die Nummer verlorengelht, können wir sie auf Anfrage sofort mitteilen.

## E-Mail Liste

Um sich in die Liste einzutragen, senden Sie an die Adresse [maiser@buch.biblio.etc.tu-bs.de](mailto:maiser@buch.biblio.etc.tu-bs.de) nur diese Message: **subscribe allegro**. Und zum Abmelden: `unsubscribe allegro`

## Termin

Das Einsteigerseminar soll stattfinden am Dienstag/Mittwoch, 23./24.9. in der UB. Noch sind Plätze zu haben, jedoch muß die Teilnehmerzahl auf ca. 24 begrenzt werden. Melden Sie sich umgehend schriftlich an (FAX 0531-391-5836). Teilnahmegebühr DM 50. Darin enthalten ist die neue *Ouverture*.

## WWW-OPAC mit *avanti-W* auf Win'NT 4.0 oder Win'95 Was braucht man, was ist wichtig?

Die weltweite Vernetzung von Computern ist eine explosive Entwicklung, die vor den Toren einer Bibliothek nicht haltmacht. Auch mittlere und kleine Bibliotheken sehen die Nachfrage, den Benutzern eine orts- wie zeitunabhängige Recherchemöglichkeit zu bieten und so die Attraktivität Ihres Dienstleistungsangebotes zu erhöhen. Die Entwicklungsabteilung hat diesen Trend bekanntlich nicht ignoriert, sondern Mittel und Wege gefunden, wie Sie Ihre *allegro*-Datenbank für die Recherche im world-wide-web (WWW) öffnen können (vgl. *allegro news* Nr.44). Diese Arbeiten stehen im Zusammenhang mit dem Projekt "Z39.50". Dessen umfassende Realisierung wird noch eine Weile brauchen, aber als Vorstufe sind die jetzt vorliegenden Ergebnisse schon jetzt anwendbar und an einigen Stellen schon im Einsatz.

Als technische Voraussetzung benötigen Sie selbstverständlich einen Internetanschluß. Auch wenn der Zugang bei der Mehrzahl der *allegro*-Anwender heute noch nicht vorhanden ist, so scheint es doch, daß die Bereitschaft, Finanzmittel bereitzustellen, um Bibliotheken an das Internet anzuschließen, in Zukunft weiter wächst.

Ist diese technische Hürde genommen, stellt sich als nächstes die Frage, welche Computer-Hardware und welches Betriebssystem nötig ist. Die Antwort lautet: Sie können Ihre bestehende Ausstattung in der Regel weiterverwenden, denn es sind Programme für alle gängigen Unix-Plattformen und für die aktuellen 32 Bit Windows-Betriebssysteme vorhanden. Windows 3.1 läuft für eine Serveranwendung allerdings nicht stabil genug. Strebt man eine sparsame Lösung an, wird man einen Standard-PC (z.B. Pentium 133, mindestens 16MB Ram) mit dem Betriebssystem Windows' 95 oder besser Windows NT 4.0 einsetzen. Auch das als sehr stabil geltende Linux (= Unix für Intel-PC's) ist eine erprobte Alternative. Linux besitzt den Vorteil, daß es im Internet kostenlos bereitliegt. Allerdings fallen die Pflegearbeiten an einer *allegro*-Datenbank unter Unix weniger komfortabel aus als dies unter einem Windows-Betriebssystem möglich ist, bei dem man mit einem Mausklick in's *Cockpit* steigen kann. Aber es kann ja durchaus auf DOS gearbeitet und dann eine Kopie der Datenbank unter UNIX für den Zugriff bereitgestellt werden. Die Einrichtung der für die Internetanbindung notwendigen Software gestaltet sich auf einem mit Windows'95 oder NT 4.0 ausgerüsteten Rechner zur Zeit noch einfacher als auf einer Unix-Plattform. Auch deshalb, weil Neueinsteiger hier in der Regel ein größeres Wissen mitbringen.

Der Kern der *allegro*-Internetanbindung ist der bekannte *avanti*-Server. Dieses Programm recherchiert eine Anfrage in der Datenbank und stellt die Ergebnisse bereit. In einer speziell für Windows 95 und NT 4.0 geschriebenen Version nennt sich der Datenbankserver *avanti-W*. Dieser Server verwaltet mehrere Datenbanken auch unterschiedlicher Konfiguration, bearbeitet gleichzeitig eintreffende Anfragen (quasi-)parallel, verbraucht während der Wartephase keine Rechenzeit und kann Benutzern abweichende Rechte auf einer Datenbank zuteilen.

Der *avanti-W* Server erhält die Anfragen von einem Client-Programm, das der Anwender über seinen Web-Browser startet, ohne daß ihm dieser Vorgang bewußt wird. Die Ergebnisse der Recherche werden wieder an das Client-Programm weitergereicht und im Browser präsentiert. Das Client-Programm stellen wir Ihnen im Perl-Quellcode zur Verfügung, damit Sie die vom Anwender wahrnehmbare Oberfläche mit geringem Aufwand an Ihre Wünsche und Erfordernisse anpassen können. Das Grundkonzept ist keine ganz neue Entwicklung sondern gehörte bereits Anfang 1996 zum *allegro*-Angebot. In jüngster Zeit ergänzt wurde der Datenaustausch zwischen CGI-Skript und *avanti-W* über TCP/IP-Sockets und Named-Pipes. Um Perl-Programme auf dem Server-Computer ausführen zu können, benötigen Sie einen Perl-Interpreter, den es im Internet kostenlos gibt. Als letztes brauchen Sie noch eine Software, die für jede aus dem Internet eintreffende Anfrage einen Client-Prozeß startet und die Ergebnisse wieder an die richtige Adresse schickt: einen Web-Server wie z.B. "WebSite". Auch solche Programme können öffentliche Einrichtungen gratis beziehen. Auf dem Braunschweiger ftp-Server finden Sie im Verzeichnis *ac14/windows* das selbstentpackende Archiv "*avantiw.exe*", das bis auf den Perl-Interpreter und den Web-Server alle notwendigen Komponenten enthält. Selbstverständlich wird das Paket auch auf der im Sommer erscheinenden CD enthalten sein. Eine ausführliche Installationsanleitung (Datei *INSTALL.DOC*) nennt Ihnen die Bezugsquellen für die fehlenden Programme und erläutert Ihnen die Einrichtung der Internetschnittstelle im Detail. Der Unterschied zwischen Windows 95 und NT 4.0 betrifft lediglich die Kommunikation zwischen dem Perl-Client und dem *avanti-W* Server. Nutzen Sie Windows 95, haben Sie nicht die Möglichkeit, den Datenaustausch mit Hilfe von "Named Pipes" durchzuführen, sondern müssen TCP/IP-Sockets wählen. Windows NT zeigt hier ein etwas schnelleres Antwortverhalten und gilt außerdem als das wesentlich stabilere System.

Aus der Anwendersicht können Sie eine solche Schnittstelle beispielsweise auf den Internetseiten der UB Braunschweig beurteilen ([www.biblio.tu-bs.de/acrecher.html](http://www.biblio.tu-bs.de/acrecher.html)). Zunächst zu Testzwecken wurde vor einigen Wochen die WWW-Recherche der UB von einer Unix-Workstation auf einen Windows-NT Computer verlegt. Ein einzelner *avanti-W* Server macht dort fünf Datenbanken zugänglich. Die Installation läuft seit der Umstellung ohne nennenswerte Beanstandung. Auf der Seite [www.biblio.tu-bs.de/allegro/ac-dbs.html](http://www.biblio.tu-bs.de/allegro/ac-dbs.html) finden Sie die Adressen von zur Zeit vierzehn weiteren Institutionen, welche die *allegro*-Internetschnittstelle in ähnlicher oder auch stark veränderter Gestalt anbieten. (ve)

## Wie geht es weiter?

Mit der Fertigstellung der konventionellen Programme plus *avanti*-Server auf dem Niveau V15 für alle Plattformen erhebt sich nun die Frage, auf welche Schwerpunkte sich die Entwicklungsarbeit in der Folgezeit konzentrieren soll.

Um die Weichen bedarfsgerecht stellen zu können, müssen wir uns einen etwas besseren Überblick verschaffen, was denn in der Praxis kurz- und mittelfristig gebraucht wird.

Nachfrage besteht auf jeden Fall nach Windows-Software, soviel ist klar. Mit PRESTO-W ist zwar der Einstieg in ein Programm mit graphischer Oberfläche gemacht, doch gibt es mehrere Entwicklungslinien, die nun verfolgt werden können. Unterscheiden muß man Programme von vier verschiedenen Typen:

### A. Monolithische Programme (Unterschiede zwischen den Plattformen)

#### A.1 Konventionelle Programme (DOS oder UNIX) V15-Kernsystem

*Basis:* C-Quellcodes; *Einsatz:* Einzelplatz-PC, Mehrplatz auf LAN (Novell o.a.) oder UNIX (telnet)

*Beispiele:* V15-Kernsystem (DOS und UNIX), *aLF*, *ORDER* (nur DOS)

#### A.2 Windows-Programme

*Basis:* C++-Klassenbibliothek

*Beispiele:* PRESTO-W, VPW (gleichfalls mehrplatzfähig)

### B. Client/Server-Programme (plattformunabhängig)

*Basis:* *avanti*-Server (muß im Hintergrund laufen); Verbindung per TCP/IP "Sockets" oder "Named Pipes"

#### B.1 Eigenständige Client-Programme

*Basis:* jede geeignete Programmiersprache mit Socket- oder Pipe-Routinen, u.a. JAVA

Klassenbibliothek verwendbar, aber nicht Bedingung.

*Beispiel:* CLIENT-W für WIN'95 und NT

#### B.2 Browser-basierte Programme (WWW-Anwendungen)

*Basis:* HTML, JavaScript; CGI-Programme (vorwiegend Perl-Skripte).

*Beispiele:* Paket ACWWW25 für Web-OPAC; "Forschungsbibliographie"

Nur der Typ A.1 ist mit V15 zu einem gewissen Abschluß gelangt! Programme der Typen B.1 und B.2 können durchaus ohne Rückgriff auf die C- oder C++-Quellen entwickelt werden, also auch durch andere Stellen ohne Kontakt mit der Entwicklungsabteilung, sofern nur der *avanti*-Server und das Kernsystem V15 vorhanden sind (u.a. zum Indexieren und Konvertieren). Das bedeutet, daß neue Anwendungen und Oberflächen nun losgelöst und unabhängig vom Kernsystem entwickelt werden können. Die Schnittstelle ist dabei die *avanti*-Sprache und auf unterer Ebene TCP/IP (Sockets) oder Named Pipes. Verbesserungen am *avanti*-Server werden natürlich so vorgenommen, daß die Sprache dabei erweitert, nicht aber strukturell verändert wird. Eine einmal z.B. in JAVA entwickelte Anwendung muß dann nicht angepaßt werden, um einen verbesserten Server einsetzen zu können.

**Wichtig:** Programme aller vier Typen können zur gleichen Zeit auf ein und derselben Datenbank arbeiten.

Kommen wir nun zurück auf den Bedarf. **Dringend** müssen wir vor allem wissen, ob Windows 3.1 oder 3.11 bei Ihnen noch länger als ein oder zwei Jahre eine wichtige Plattform sein wird. Wenn es so ist, teilen Sie uns wenigstens dieses umgehend mit, entweder mit dem unten stehenden Formular oder per E-mail ([allegro@tu-bs.de](mailto:allegro@tu-bs.de)).

Wir würden es sehr begrüßen, wenn Sie sich eine Minute Zeit nähmen, um das folgende Formular auszufüllen und uns per FAX (0531-391-5836 "allegro") oder Post zuzusenden:

Bitte schreiben Sie **Ziffern** vor die für Sie zutreffenden Punkte der folgenden Liste, und zwar

**1 = sehr wichtig, 2 = auch wichtig, 3 = nicht wichtig**

#### Mittelfristig wichtig:

(mind. noch 2 Jahre)     DOS         Win'3         Win'95         Win'NT         Browser: \_\_\_\_\_

Linux         SUN         DEC-Alpha     AIX         X-Terminal

#### Wir brauchen:

(jetzt oder später)     *aLF*         *aLFA*         *ORDER*         *REF*         anderes: \_\_\_\_\_

*avanti-X*     *avanti-W*     *PRESTO-W*     WWW-Anwendungen