

# 0 Das *allegro*-Konzept

## 0.0 Zentrale Begriffe

Wer ein **Datenbanksystem** benutzt, interessiert sich in der Regel erheblich mehr für die eigenen **Daten** als für das **System**. Von **Programmen** und ihren internen Funktionen oder gar von **Datenstrukturen** möchte nicht jeder viel wissen müssen. Weil Computer vorerst die Umgangssprache weder sprechen noch verstehen, ist ein Minimum an DV-Kenntnissen wichtig, auch um zu einer realistischen Einschätzung der Möglichkeiten und Grenzen des Computereinsatzes zu gelangen. Soweit es um *allegro* geht, soll dieses Handbuch dabei helfen, obwohl es nicht zugleich ein Lehrbuch sein kann: das Kapitel 0 und in den übrigen Kapiteln jeweils der Abschnitt 0 sollen die Begriffe und Konzepte erklären und die Einarbeitung erleichtern.

Wenn Sie gerade erst mit *allegro* anfangen wollen:

Der "Leitfaden zur Einarbeitung" (hinter dem Vorwort) weist den Weg!

Jedes Rechnersystem speichert Daten in **Dateien**. Die Vorgänge des Speicherns und Lesens von Daten sowie jede Art des Umgangs mit den Daten besorgen **Programme**.

Jedes Programm folgt dem bekannten "E-V-A-Schema":

**Eingabe** (Daten) → **Verarbeitung** (Programm) → **Ausgabe** (Daten)

wobei allerdings mehrere Ein- und Ausgabevorgänge in dauerndem Wechselspiel ablaufen können.

**Eingabedaten** kommen entweder

- über die Tastatur (vom Anwender); es kann sich um "richtige" Daten, aber auch um Funktionstasten-Betätigungen, Mausclicks oder eingetippte Befehle handeln,
- oder werden aus Dateien eingelesen. Dabei kommt es immer sehr auf deren Struktur an (siehe unten).

**Ausgabedaten** als Ergebnis einer Verarbeitung

- erscheinen auf dem Bildschirm oder Drucker
- oder werden wieder gespeichert, d.h. in Dateien geschrieben.

Als Ausgabe können zwar auch Katalogzettel herauskommen, doch ist dies mehr ein Nebenprodukt. Eingegeben und gespeichert werden natürlich keine Katalogkarten, sondern **Datensätze**, meistens ist aber ein Datensatz gleichwertig mit einer **Titelaufnahme**. Deswegen steht in diesem Buch oft "Aufnahme" statt "Datensatz". Aus gespeicherten Datensätzen kann das System sehr vieles machen: Karten, Listen, Bildschirmanzeigen, Auswertungen (auch mit Berechnungen), alphabetische Register. Alles das ist nicht fest eingebaut, sondern kann variiert, angepaßt und ausgebaut werden. Um eigene Wünsche zu verwirklichen, muß man freilich dem System geeignete Vorschriften machen. Solche Vorschriften werden **Parameterdateien** genannt. Sie bestehen oft aus sehr vielen einzelnen Befehlen. Viele fertige Parameterdateien für Karten, Listen usw. werden mitgeliefert. Ist man damit nicht zufrieden, muß man Änderungen machen oder ganz neue Vorschriften erarbeiten und die Befehle dafür zusammenstellen. Die Befehle werden, wie gesagt, nicht in natürlicher Umgangssprache formuliert. Man muß eine besondere Sprache dazu benutzen, die sog. **Exportsprache** (⇒ Kap.10). Will man Fremddaten in *allegro*-Daten umwandeln, kommt noch eine Sprache hinzu, die **Importsprache** (⇒ Kap.11). Der Wortschatz dieser Sprachen ist viel kleiner, die Grammatik viel einfacher als bei natürlichen Sprachen. Dennoch gehört das Schreiben von Export- und Importvorschriften zu den schwierigsten Aufgaben im *allegro*-System, denn man muß dabei sehr präzise und in ungewohnt kleinen Schritten denken.

Der Rest dieses Kapitels ist nicht nur zum Lesen, sondern auch zum Nachschlagen gedacht. Ein alphabetischer Überblick zu den wichtigsten Eingreifmöglichkeiten findet sich im Abschnitt 0.9.

## Die wichtigsten Begriffe

**Zehn Grundbegriffe** sind zunächst einmal wichtig, wenn man zu einem eigenständigen Umgang mit *allegro* finden will. Aber auch "Nur-AnwenderInnen" werden sich im System besser zurechtfinden, wenn sie sich eine ungefähre Vorstellung von diesen Grundbegriffen aneignen. Versuchen Sie bei Ihrer weiteren Beschäftigung mit *allegro*, sich immer an diesem Gerüst zu orientieren. (Siehe auch Diagramm 2 in Anhang C.) Die Grundbegriffe bleiben auch dann gültig, d.h. man muß nicht umlernen, wenn ab Version 15 zunehmend neue Programme herauskommen, die unter Windows laufen. Die internen Strukturen des Systems haben sich bewährt und bleiben bestehen.

(Mit dem Zeichen  $\Rightarrow$  wird jeweils auf einen der anderen Begriffe oder auf andere Kapitel verwiesen.)

*Was bedeutet  $\mu$ ?* Am rechten Rand sehen Sie jeweils, wie man vom *CockPit* aus an die betreffenden Dateien oder Programmteile herankommt. **Das Zeichen 'μ' steht für "Menü"**, der Buchstabe dahinter ist der Anfangsbuchstabe des Menüs. Also heißt z.B.  $\mu o k$  soviel wie "Menü 'Optionen', Punkt k". Erreichbar ist dieser Menüpunkt mit der Tastenfolge  $\langle \text{Alt} \rangle + o k$  (*CockPit*  $\Rightarrow$  0.11).

*Hinweis für UNIX:* Alle Dateinamen sind unter UNIX komplett in Kleinbuchstaben zu schreiben.

### 1. Konfiguration ( $\Rightarrow$ Anhang A.)

$\mu o k$

*allegro* hat kein fest eingebautes System von Datenfeldern und keine festgelegten Eingabemasken. Sie können selber entscheiden, welches **Kategorienschema** Sie benutzen wollen, welche **Datensatztypen** es gibt, was alles in der **Erfassungs-Abfrageliste** vorkommen soll. Auch die **Bildschirmfarben**, die **Sprache** der Menüs und andere Einzelheiten sind variabel. Alle diese Angaben müssen in einer **Konfigurationsdatei** (Typ .CFG) stehen. Mitgeliefert werden die Konfigurationen **\$A.CFG**, **D.CFG** (MAB1), **M.CFG** (MAB-ähnlich), **\$P.CFG** (Pica), **\$U.CFG** (USMARC). Sie können mehrere Datenbanken mit derselben Konfiguration betreiben oder mit unterschiedlichen. Änderungen an der Konfiguration führt man mit einem Bearbeitungsprogramm aus (man nennt so ein Programm "Texteditor",  $\Rightarrow$  3b). Die Konfiguration ist allerdings nur die eine Hälfte einer Datenbankdefinition, die andere ist die **Index-Parameterdatei**:

### 2. Parameterdateien = Vorschriften für's System

$\mu d e \dots$

Die Konfiguration enthält nur Angaben zur Struktur der Daten und einige Grundeinstellungen für die Funktion der Programme. Damit ist noch nicht festgelegt, wie die Daten z.B. am Bildschirm erscheinen, wie sie sortiert, umgewandelt oder ausgedruckt werden können, oder über welche Kriterien man wie zugreifen kann. In diesen Punkten bleibt viel Raum für Kreativität, und das Werkzeug dazu sind eben die **Parameterdateien**. Sie enthalten so etwas wie Vorschriften und sind geschrieben in einer Art Programmiersprache, auch **Datenmanipulationssprache** genannt, die speziell für *allegro* geschaffen wurde. Es gibt sogar zwei solche Sprachen:

a) Die **Importsprache** ( $\Rightarrow$  Kap.11) : damit beschreibt man, wie "Fremddaten" aussehen und wie sie in *allegro*-Daten umgewandelt werden sollen. Damit wird nicht jeder Anwender konfrontiert. Import-Parameterdateien haben die Typkennung .AIM  $\mu d imp$

b) Die **Exportsprache** ( $\Rightarrow$  Kap.10) : in dieser Sprache beschreibt man und teilt man dem System mit: (in Klammern stehen Namen von geeigneten Standard-Parameterdateien)

- wie die Daten am Bildschirm angezeigt werden sollen (z.B. D-1.APR)  $\mu d e$
- wie eine gedruckte Liste aussehen soll (z.B. P-NORMAL.APR)
- wie eine Datenmenge zu sortieren ist (z.B. S-PT.APR)
- in welcher Form Daten in eine andere Datei zu schreiben sind (z.B. E-1.APR).
- wie der  $\Rightarrow$  Index einer Datenbank aussehen soll (z.B. CAT.API)  $\mu d i$

Die Exportsprache hat Ausdrucksmittel für die Anordnung der Elemente, für deren Vorbehandlung (Zerlegen, Zusammenfügen, Ersetzen), für Interpunktion und Umbruchgestaltung sowie Zeichen-Umcodierung. Außerdem gibt es Sprungbefehle, Bedingungsprüfungen und Unterprogramme.

Export-Parameterdateien haben die Typkennungen .APR, .API und .APT.

Auch Parameterdateien sind Textdateien, die man mit einem  $\Rightarrow$  Texteditor (siehe 3b.) erstellt und verändert. Das *CockPit* ruft automatisch einen solchen Editor auf ( $\Rightarrow$  0.11 und Anh.D).

### 3. Editor

Damit ist ein Programm oder Programmteil gemeint, mit dem man Datensätze oder Dateien am Bildschirm bearbeiten kann. Es gibt viele solche Programme, aber keins eignet sich für alle Daten. Es hängt von der Struktur der Daten ab, ob man sie mit einem bestimmten Editor bearbeiten kann oder nicht. Der *allegro*-Anwender braucht zwei Editoren:

- a) den **integrierten Editor** (Funktion 'E' zum Bearbeiten, => Kap.1.5.4 und Kap.3),  
Damit kann man keine anderen, sondern **nur allegro**-Daten bearbeiten, und zwar immer zu einem Zeitpunkt nur einen Datensatz: wenn er auf dem Anzeigeschirm steht, gibt man 'E';
- b) einen **Texteditor**. Damit lassen sich Textdateien des einfachsten Typs, sog. "ASCII-Dateien", bearbeiten. Man muß wissen: Textdateien bestehen aus *Zeilen*, Datensätze aber aus *Datenfeldern*, und ein Datenfeld kann viel länger als eine Bildschirmzeile sein. Zu diesem Typ gehören die => Parameter- und Konfigurationsdateien. Deshalb braucht der *allegro*-Anwender einen Texteditor. DOS enthält ab 5.0 mit EDIT einen sehr brauchbaren Texteditor. Die luxuriösesten Texteditoren sind WORD, WordPerfect und verwandte Produkte. Integrierte Texteditoren gibt es in Hilfssystemen wie "Norton Commander" und "PCTools". Wenn man vom *CockPit* aus eine Datei zur Bearbeitung anfordert, kann das mit jedem Texteditor geschehen. Es wird zwar einer namens X mitgeliefert (=> Anhang D.), aber Sie können auch z.B. EDIT benutzen (wegen der Einbindung ins *CockPit* => 0.11.6). Unter UNIX verwendet man in der Regel den Editor namens vi. Nochmals: *allegro*-Datenbankdateien kann man **nicht** mit einem Texteditor bearbeiten, denn es sind keine ASCII-Dateien.

### 4. Datenbank

Wie die Datensätze aussehen, das hängt ab von der => Konfiguration (Kategorienschema), die Register aber werden beschrieben in der => Index-Parameterdatei. Der Name dieser Datei ist zugleich der Name der Datenbank. Die vom Anwender erfaßten Datensätze stehen in Dateien vom Typ .ALD (oder z.B. .PLD, wenn man mit einer anderen Konfiguration P.CFG arbeitet!) Deshalb müssen diese Dateien unbedingt und vor allen anderen gesichert werden. Außerdem gehören zur Datenbank der => Index (Typ .ADX bzw. .PDX) und noch eine Satztable (vom Typ .TBL). Beide können erneuert werden, indem man die Datenbank aus den .ALD-Dateien neu aufbauen läßt. Benutzt wird die Datenbank mit dem Programm PRESTO. Der darin => integrierte Editor ermöglicht Neueingabe und Korrektur. Auch das Volltext-Suchprogramm SRCH (=> Kap.4) kann die Datenbank lesen: es durchsucht die Daten zeichenweise, kann auch => exportieren, ermöglicht aber keine Korrekturen und Neueingaben. Die Programme INDEX und UPDATE können neue Daten, die man z.B. aus fremden Quellen **importiert** hat (=> Kap.5), einer Datenbank einverleiben. Eine neue Datenbank legt man per *CockPit* an ( $\mu r$  a) oder mit dem Hilfsprogramm PRONTO. Sind schon Daten vorhanden, läßt man vom Programm INDEX eine Datenbank daraus machen (=> Kap.7).

In => Kap.0.8 wird zusammengefaßt, welche Dateien zu einer Datenbank gehören.

### 5. Index (=> Kap.1.4, Kap.7)

Jede Datenbank braucht einen Index. Wenn man sich die Datenbank als Nachschlagewerk vorstellt, enthält der Index das oder die alphabetischen Register. Bis zu 11 solche Register kann ein Index umfassen. Man blättert darin, indem man (ohne Befehl) ein gesuchtes Wort oder dessen Anfang eingibt. Mit Druck auf die Taste <Alt> und eine Ziffer schaltet man auf eines der anderen Register um. Die Indexdatei entsteht und wächst automatisch mit der Datenbank. Wenn man mit dem => Editor an den Daten etwas ändert, wird der Index automatisch auch geändert. Wie die Register aussehen, das steht in der zur Datenbank gehörigen Index-=> Parameterdatei. Diese muß man ändern (mit einem => Texteditor), wenn man grundsätzliche Änderungen am Index vornehmen möchte. Anschließend muß man aber den Index erneuern lassen - mit *CockPit* kein Problem (=> Kap.0.11.2, Kap.7).

$\mu r$  o i

- 6. Export = Ausgabe = Formatierung** (=> Kap.6 und Kap.10) µf 4  
 bedeutet: Daten in eine Datei schreiben oder auf ein Gerät (Drucker oder Bildschirm) ausgeben. In welcher Form das geschehen soll, das beschreibt eine Export-=>Parameterdatei (siehe 2b.). In die Programme PRESTO, SRCH, INDEX und IMPORT ist eine **Exportfunktion** eingebaut, d.h. jedes dieser Programme kann exportieren. Es nimmt dazu eine Parameterdatei her und wendet die darin stehenden Vorschriften auf die anstehenden Daten an.  
 Über sog. **Optionen** (=> Kap.12) kann man den Programmen beim Aufruf mitteilen, welche Parameterdatei auf welche Daten anzuwenden ist (*CockPit* macht das automatisch). Das Ergebnis ist eine Liste, eine Bildschirmanzeige, ein Index, etc.
- 7. Ergebnismenge** (=> Kap.1.4 und 1.5)  
 Wenn man auf eine => Datenbank über einen => Index zugreift, kann man die zu bestimmten Registereinträgen gehörigen Datensätze mit den logischen Kombinationstasten '+' (UND), '/' (ODER) bzw. '-' (NICHT) zu Teilmengen der Datenbank zusammenfassen (bis zu 16.000 Sätzen, Begrenzung möglich mit einem Befehl **mr** in der Konfiguration). Eine Ergebnismenge kann man durchblättern, exportieren (z.B. drucken), aber auch global verändern oder löschen.
- 8. Datensicherung** (=> Kap.0.7) µr s / w  
 Hardware- oder Softwareschäden (Viren!) können Datenbanken ruinieren. Wer gewissenhaft Sicherungskopien (auf Disketten oder Bandkassetten) macht, hat wenig zu befürchten: *allegro* zeichnet alles auf, was an der Datenbank gemacht wird. Wenn der GAU kommt, hat man die Sicherungskopie und die sog. **LOG-Datei**, in der die Änderungen protokolliert sind. Ein eigenes Programm, **UPDATE** (=> Kap.9), fügt beides wieder zusammen und stellt so den Zustand vor dem Fiasko wieder her.
- 9. Programmaufrufe** (=> Kap.12) µm  
*allegro* besteht aus mehreren Programmen mit unterschiedlichen Aufgaben (=> Kap.0.4). Diese werden entweder vom *CockPit* automatisch gestartet und wissen dann, was sie tun sollen. Oder man gibt ihnen die nötigen Angaben schon beim Aufruf mit. Der folgende Befehl  

```
srch -d abc -e PRINT/liste -s0 -f4 -v0
```

 bewirkt z.B. einen Aufruf des Volltext-Suchprogramms SRCH (=> Kap.4). Alle sog. "Optionen" beginnen mit einem Bindestrich, das Zeichen hinter dem Bindestrich ist eine Befehlskennung (=> Kap.12.2). Bearbeitet wird die Datei abc.alg. Erzeugt wird eine Datei namens LISTE. In der Parameterdatei PRINT.APR ist beschrieben, wie LISTE aussehen soll. -s0 bedeutet: jeder Satz kommt auf die Liste, -v0 heißt: keine Anzeige der Daten während des Durchlaufs. Solche Programmaufrufe kann man in **Batch-Dateien** zu umfangreichen Abläufen zusammensetzen. Batchdateien (auch **Stapeldateien** oder bei UNIX **Shell Scripts** genannt) erstellt und bearbeitet man mit einem => Texteditor. PR-LIST.BAT und QUEX.BAT sind Beispiele dafür; diese enthalten mehrere Programmaufrufe, unter anderem auch Sortiervorgänge.
- 10. Benutzeroberfläche**  
 Für diese wurde ein hohes Maß an Flexibilität angestrebt. Menüzeilen und Hilfsseiten sind Textdateien und können mit einem => Texteditor verändert oder sogar übersetzt werden. Es gibt eine englische (wird mitgeliefert) sowie eine spanische und eine niederländische Version. Eigene Menüs können auf verschiedene Art erstellt werden. Dazu kann die MS-DOS-Batchprogrammierung (unter UNIX die Shell-Skriptsprache) benutzt werden (=> Kap.12), evtl. angereichert durch zusätzliche Software-Tools.  
 Für das DOS-System gibt es das *CockPit* (=> Kap.0.11). Dieses ermöglicht die Vorbereitung und Steuerung aller Vorgänge über Menüs in einer auch von anderer Software gewohnten Manier. Standardprozeduren wie Neu-Indexieren, Sichern und Wiederherstellen von Datenbanken, Produktion von sortierten Listen u.a. lassen sich dann durch wenige Tasten starten, man kann aber auch eigene Programme und Prozeduren in die Menüs einbeziehen.

## 0.1 Allgemeines über Dateien

Dieser Abschnitt soll besonders Einsteigern eine genauere Vorstellung davon geben, was Dateien eigentlich sind, und wie insbesondere die Titeldaten (ehemals Katalogzettel) gespeichert sind.

Ein Computerspeicher ist nicht einfach ein Behälter voller Daten, in denen sich der Rechner dann schon "irgendwie" zurechtfindet und stets die richtigen Sachen auf den Schirm oder zu Papier bringt. Vielmehr ist so ein Speicher so etwas wie ein Aktenschrank, in dem viele Ordner für verschiedenste Zwecke stehen, und in jedem Ordner herrscht eine zweckmäßige Untergliederung. Es gibt somit bei jedem Rechnersystem genau festgelegte, unterschiedliche Arten von Daten, und dafür muß es jeweils eigene Ordner geben, **Dateien** genannt, die sich in Struktur und Inhalt unterscheiden. Jede Datei hat ihren Namen. Damit man schon am Namen einer Datei erkennen kann, welche Art Daten sie enthält, hängt man an den Namen eine **Typbezeichnung** ("**extension**") aus 1 - 3 Zeichen und nennt diese auch kurz den **Typ** einer Datei. *Wichtig:* In diesem Buch werden Dateinamen oft mit Großbuchstaben geschrieben, weil sie dann optisch besser auffallen. Unter UNIX sind jedoch alle *allegro*-Dateinamen vollständig in Kleinbuchstaben zu schreiben.

### Datensätze

Was früher auf Karteikarten gestanden hätte, speichert *allegro* in Dateien vom Typ ".ALD" bzw. ".ALG". An diesen Typen erkennt man also auf einer Diskette oder Platte die *allegro*-Dateien, wenn man sich das Inhaltsverzeichnis ("directory") ansieht. Beide Dateitypen sind so etwas wie Karteischubladen, aber was ist der Unterschied?

.ALD - Dateien sind **Datenbank**-Dateien, die für den direkten Schnellzugriff über einen alphabetisch sortierten Index aufbereitet sind. Für diesen Zugriff sind zwei weitere Dateien nötig (siehe unten: Indexdatei vom Typ .ADX und Adressentabelle, Typ .TBL), für deren automatische Erstellung wird eine Index-Parameterdatei des Typs .API gebraucht.

.ALG - Dateien sind dagegen **Grunddateien** ohne alphabetischen Schnellzugriff über Index. Deshalb wird, etwas unpräzise, auch von "offline"-Dateien gesprochen. Solche Dateien können als Ergebnis (= Ausgabedatei) einer Volltextsuche (Programme 4 und 1) oder eines Importvorgangs (Programm 5) entstehen, man kann aber auch mit dem Programm 2/3 solche Dateien schreiben und bearbeiten. Mit Hilfe der Programme 7 (INDEX) und 9 (UPDATE) können solche Grunddateien dann in Datenbank-Dateien umgewandelt oder in eine vorhandene Datenbank eingemischt werden. Sehr wichtig: Grunddateien können sortiert werden (=> Kap.8)

Der Buchstabe hinter dem Punkt, das 'A' von .ALD und .ALG, wird **Konfigurationsbuchstabe** genannt und bezieht sich auf das verwendete Categoriesystem. 'A' steht für das *allegro*-Standardschema (=> Anh.B), das vom ehemaligen Schema des Niedersächsischen Katalogisierungsverbundes (sog. NMN-Schema, bis 1992) abgeleitet ist. Wenn man mit einem anderen Schema (einer anderen Konfiguration) arbeitet, z.B. mit 'U' für "USMARC", lautet der Dateityp ".ULG". Wenn in diesem Handbuch von **.ALG-Dateien**, **.APR-Dateien** usw. die Rede ist, gilt entsprechendes immer auch für .ULG-Dateien, .UPR-Dateien etc. Deshalb wird meistens, wenn es nicht speziell um ein bestimmtes Schema geht, 'c' und ".cLD", ".cLG", ".cPR" usw. geschrieben statt ".ALD", ".ALG", ".APR"... Denken Sie sich also für das 'c' jeweils ein 'A' oder Ihren eigenen Konfigurationsbuchstaben.

### Struktur der Datensätze

Jede *allegro*-Datei (Typ .cLD oder .cLG) kann eine größere Anzahl **Datensätze** enthalten. Ein Datensatz, oft einfach nur **Satz** oder auch **Aufnahme** genannt, entspricht bei Buchdaten einer Titelaufnahme, also einem Katalogzettel.

Bei "mehrbändigen Werken" kann so ein Satz aus einem **Hauptsatz** und mehreren **Untersätzen** (Bandaufführungen) bestehen. Letztere können wiederum hierarchisch untergliedert sein, d.h. jeder Untersatz kann seinerseits Untersätze zweiter Stufe enthalten usf. bis zur Stufe 6. Diese Möglichkeit ist eine Besonderheit des Systems *allegro*. (Ein einfaches Beispiel dafür finden Sie im nächsten Abschnitt unter 0.2.2, wo die Struktur genauer dargestellt ist.)

Ab Version 11.2 können Haupt- und Untersätze auch getrennt gespeichert und über **Satzverknüpfungen** miteinander gekoppelt sein. Verknüpfungen können auch zwischen Sätzen verschiedener Typen bestehen, z.B. zwischen Titelsätzen und Körperschafts-Stammsätzen. Somit ist eine **Mehrdateistruktur** realisierbar (=> 10.2.6.7).

Jede Aufnahme hat eine variable Anzahl **Datenfelder** (meist **Kategorien** genannt).

Jede Kategorie kann innerhalb einer Aufnahme mehrfach auftreten und jeweils bis zu 3000 **Zeichen** (auch **Bytes** genannt) enthalten. Wenn sie nichts enthält, also in einer Aufnahme nicht mit Inhalt gefüllt wird, beansprucht sie auch keinen Speicherplatz, und zwar noch nicht einmal ein einziges Byte.

Weitere Einzelheiten zur Struktur der Titeldaten im nächsten Abschnitt.

Damit man überhaupt mit den Daten arbeiten kann, werden gewisse zusätzliche Dateien benötigt:

- die wichtigste von allen ist die **Konfigurationsdatei**, in der die Datenfelder (das **Kategorienschema**) definiert und andere grundlegende Angaben festgehalten sind; A.CFG ist die Standard-Konfigurationsdatei,
- **Parameterdateien** für Indexdefinition, Bildschirmanzeige, Druckerausgabe, und evtl. noch andere Zwecke,
- zu den .ALD-Dateien einer Datenbank gehören eine **Indexdatei** und eine **Satztabelle**: beide werden automatisch erzeugt und verwaltet, erstere ist jedoch durch den Anwender weitgehend beeinflussbar ("parametrierbar"),
- eine **Kurztiteldatei** kann das Suchen im Index erleichtern: statt größere Treffermengen einzeln durchzublätern, gibt die Kurztiteldatei, wenn man eine angelegt hat, eine schnelle Übersicht: mit einer Zeile je Titel gehen bis zu 22 Datensätze auf einen Bildschirm. Jedoch funktioniert eine Datenbank auch ohne die Kurztiteldatei.
- eine **Restriktionsdatei** ermöglicht das Einschränken von Suchergebnissen: mit Kriterien wie Erscheinungsjahr, Sprache oder Publikationstyp (sofern man solche Angaben erfaßt hat), kann man große Ergebnismengen of sehr wirkungsvoll verkleinern, bevor man sich die Titel ansieht. Für das Funktionieren der normalen Zugriffe ist auch diese Datei nicht notwendig.
- Dateien mit den **Menütexten** (UIF-Dateien, z.B. UIF1GER) und Erklärungen zu den Funktionen (H-Dateien oder "HELP-Texte", z.B. H10).

Es muß dabei immer erkennbar sein, auf welchem Kategorienschema eine Datenbankdatei oder eine Parameterdatei beruht. Deshalb gibt es den schon erwähnten **Konfigurationsbuchstaben** (Standard 'A') als Bestandteil des Dateityps.

Diese und weitere Dateitypen werden im folgenden näher beschrieben ( $\Rightarrow$  0.3), zunächst wird aber versucht, das *allegro*-System als einen Schichtenbau zu veranschaulichen, auf dessen Ebenen sich gleichartige Objekte befinden, die sich jeweils zu Objekten der nächsthöheren Schicht zusammensetzen. Hierarchische Schichtenmodelle ermöglichen es in der Informatik, auch komplexe Zusammenhänge noch übersichtlich darzustellen.

Das wohl bekannteste Beispiel für ein hierarchisches Modell ist das OSI-7-Schichtenmodell der Datenkommunikation.

In den Kapiteln 10 und 11, darauf sei hier schon hingewiesen, wird ebenfalls mit einem hierarchischen Schichtenmodell versucht, die umfangreichen Export- und Importparametersysteme überschaubar zu machen.

### Bemerkungen zu den Dateien für MS-DOS-Kenner

Die Grunddateien vom Typ .cLG dürfen (im Gegensatz zu den .cLD-Datenbankdateien) mit den normalen Befehlen des Betriebssystems kopiert und verkettet (COPY), gelöscht (ERASE) und umbenannt (RENAME) werden. Ändern Sie den Typ .cLG jedoch nicht, weil SRCH und UPDATE daran die Dateien erkennen!

Mit den MS-DOS Befehlen SORT und FIND kann der Systemkenner durchaus auch gewisse Sortierungen und Auswertungen an .cLG-Dateien vornehmen, nicht aber an .cLD-Dateien.

Auch TYPE zum Betrachten der Daten auf dem Bildschirm ist möglich, wobei aber die Anzeige ohne Zeilentrennung erfolgt und daher unübersichtlich ist. Besser benutzt man z.B. das Programm SRCH, wobei man als Suchbegriff einfach nur die Ziffer '0' eingibt (die kommt in jedem Datensatz vor); dann laufen die Aufnahmen am Bildschirm durch.

Direkte Bearbeitung der .ALG- bzw. .ALD-Dateien mit anderer Software (z.B. Textprogrammen) ist zwar nicht möglich, da es keine übliche Zeilenstruktur gibt (Datenfelder sind unterbrechungslose Zeichenketten!), aber es ist auch nicht nötig, denn die Exportfunktion ( $\Rightarrow$  Kap. 4 u. 6) kann Dateien von beliebiger Struktur herstellen. Auch solche, die einem Textprogramm zugänglich sind. Konfigurations- und Parameterdateien sowie Menü- und Hilfstexte können dagegen mit jedem beliebigen Textprogramm oder ASCII-Editor manipuliert werden ( $\Rightarrow$  0.3).

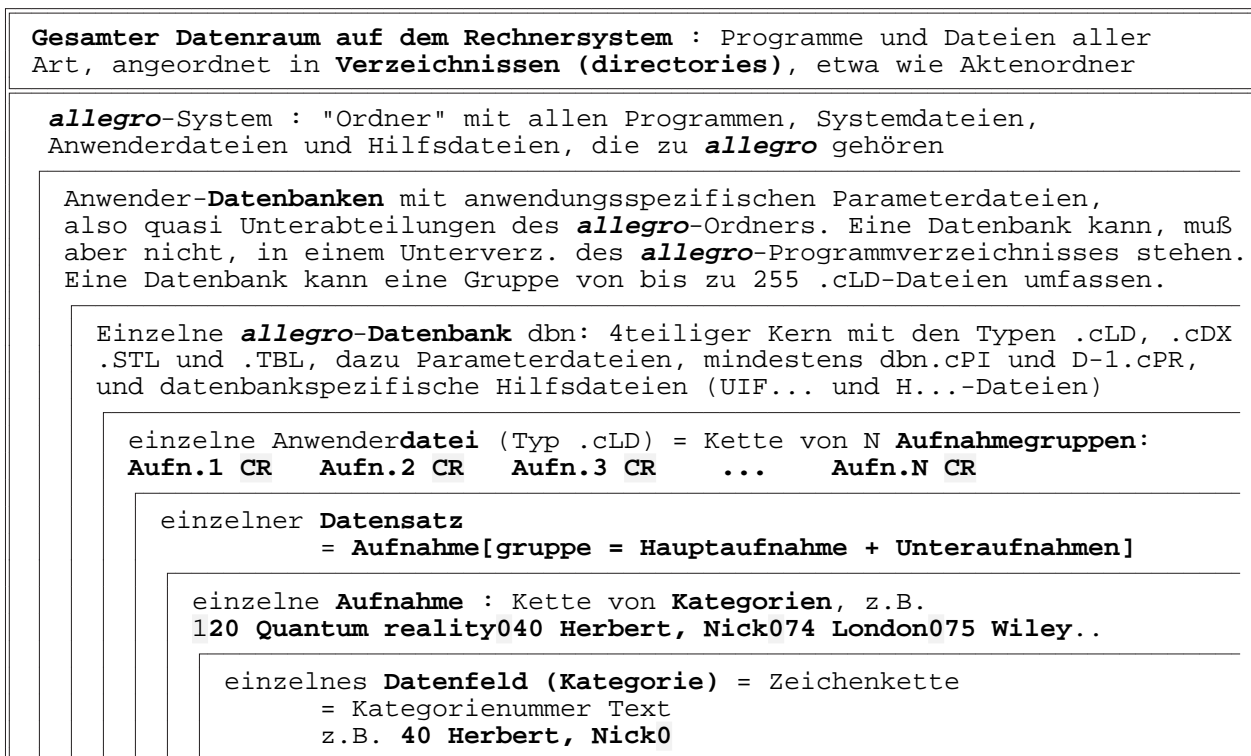
## 0.2 Mehr über Datenspeicherung und -struktur

Bevor nun alle Dateitypen im einzelnen genau erklärt werden, folgt hier die Darstellung des erwähnten Schichtenmodells.

Wie behält man in einem Massenspeicher die Übersicht? Daß es baumartig angeordnete **Verzeichnisse (directories)** gibt, weiß jeder auch nur einigermaßen eingeweihte MS-DOS- oder UNIX-Anwender. Was aber steht alles in einem Verzeichnis bzw. dem dazugehörigen Plattenbereich? Mit dem Befehl **dir** (MS-DOS) bzw. **ls** (UNIX) sieht man nur eine Liste von Namen mit gewissen anderen Angaben, unter denen man Datum und Uhrzeit noch erkennt, den Rest aber ohne Vorkenntnisse nicht zu enträtseln vermag. Relativ leicht lernt man bei DOS, wie Programme und Dateien auseinanderzuhalten sind (die Typen .EXE, .COM und .BAT sind Programme), aber dann hört die Anschaulichkeit auf: irgendwelche Hinweise über Inhalt und Funktion der Dateien und deren Zusammenhänge und Abhängigkeiten untereinander sind den nackten Namen nicht anzusehen, schon gar nicht, wie es drinnen aussieht in den Dateien, wie sie also in sich strukturiert sind.

### 0.2.1 Dateien-Hierarchie

Es ist hilfreich, wenn man das ganze Dateisystem mit dem Umfeld, das zur *allegro*-Software gehört, in eine Anzahl von Schichten oder Ebenen gliedert, die inhaltlich und funktional hierarchisch aufeinander aufgebaut sind. Das sieht folgendermaßen aus:



CR ist ein Steuerzeichen (Carriage Return) und trennt zwei Datensätze. Es gibt noch weitere Steuerzeichen, die der Anwender ebenfalls nicht kennen muß, z.B. Feldendezeichen, hier als 0 dargestellt, und Satzanfang/Hierarchie-Kennung, hier als 1 dargestellt. Jeder Datensatz in einer Datei des Typs .ALD hat ausserdem, hier nicht dargestellt, eine interne Satznummer. Diese besteht aus vier Bytes und steht direkt hinter dem Code 1. Grunddateien (Type .ALG) sehen genauso aus, nur ohne die interne Satznummer. Auf der nächsten Seite wird dies noch genauer gezeigt.

Der Anwender kann Gruppen von bis zu 255 *allegro*-Dateien zu einer Datenbank zusammenfassen. Jede Datei kann bis zu 16 MB wachsen. Bei der Erfassung kann man bei jedem neuen Datensatz die Datei auswählen, in die er geschrieben werden soll (Kap. 1.5.4, Befehl F). Jeder Datensatz kann beliebig viele Kategorien und Unteraufnahmen haben und insgesamt bis zu 40K groß sein.

## 0.2.2 *allegro*-Datenstruktur

Dieser Abschnitt beschreibt noch etwas genauer die Struktur der "eigentlichen" Dateien, d.h. der Typen .ALD und .ALG, in denen die Anwenderdaten stehen.

Wer braucht diese Kenntnisse? Wer "nur" katalogisiert oder einen *allegro*-Katalog benutzt, kann diesen Abschnitt übergehen. Dafür interessieren muß sich aber, wer Fremddaten importieren, d.h. in *allegro*-Daten umwandeln will (Kap. 5 und 11) und wer *allegro*-Daten mit eigener Software erstellen will, als sog. "ASCII"-Datei, um sie dann mit dem Programm UPDATE oder mit dem *avanti*-Server in eine Datenbank einzubringen.

Normal erfaßte Datensätze stehen als zusammenhängende Zeichenkette in einer Datei des Typs .ALD. (Das Wort "zusammenhängend" ist wichtig: der Datensatz ist nicht in logische oder physikalische Teile oder Blöcke oder Tabellenspalten o.ä. zerlegt, sondern bildet eine einzige Zeichenkette!) Und wenn ein Satz beim Bearbeiten verlängert wird? Dann merkt das die Software (PRESTO und UPDATE) und verlagert den gesamten Satz an eine andere, passende Stelle; wenn es innerhalb der Datei keine gibt, an das Ende der Datei. Der "alte", kürzere Datensatz wird vorn mit dem ASCII-Code 09 statt 01 gekennzeichnet und erhält einen Indexeintrag im Register 1 unter "/" mit seiner Länge (in Bytes) als fünfstelliger Zahl dahinter, z.B. //00249 für einen Datensatz, der 249 Byte lang ist. Diese Schlüssel braucht das Programm, um geeignete Stellen zum Speichern neuer oder verlängerter Sätze zu finden.

Und so sieht ein Datensatz in einer Grunddatei aus: (Typ .ALG)

```
100 654321020 Was können wir wissen?031 Evolutionäre Erkenntnistheorie040 Vollmer,
Gerhard074 Stuttgart075 Hirzel076 1985077 Bd.1.2.0
201 1=Bd. 1020 ¬Die¬ Natur der Erkenntnis : Beiträge zur Evolutionären Erkenntnisth-
eorie025 Mit einem Geleitw. v. Konrad Lorenz077 337 S.087 3-7776-0403-8090 2647-36110
201 2=Bd. 2020 ¬Die¬ Erkenntnis der Natur : Beiträge zur modernen Naturphilosophie077
350 S.087 3-7776-0404-6090 2650-97050CR
```

Ganz korrekt ist dieser Eindruck nicht, denn es gibt in dieser Datei keine Zeilentrennungen: alles steht in einer langen Zeichenkette hintereinander, nur ganz am Ende stehen die ASCII-Codes 13 10 (CR). Die Codes 0, 1 und 2 sind Steuerzeichen, und zwar die ASCII-Codes 00, 01 und 02, nicht die Ziffern '0', '1' und '2'. Sie bezeichnen das Feldende, den Satzanfang und den Untersatz-Anfang. Wir haben hier nämlich einen hierarchischen Satz mit zwei Untersätzen vor uns. Es kann weitere Unterstufen geben, die dann mit 302 ..., 403 ... etc. beginnen.

Derselbe Satz kann für die neuere Software (*avanti*-Server, Windows-Programme) auch so aussehen: (aber nur zum externen Bearbeiten! Innerhalb der Datenbankdatei bleibt es bei der oben gezeigten Struktur.)

```
#00 654321
#20 Was können wir wissen?
#31 Evolutionäre Erkenntnistheorie
#40 Vollmer, Gerhard
#74 Stuttgart
#75 Hirzel
#76 1985
#77 Bd.1.2.
#01 1=Bd. 1
#20 ¬Die¬ Natur der Erkenntnis : Beiträge zur Evolutionären
  Erkenntnistheorie
#25 Mit einem Geleitw. v. Konrad Lorenz
#77 337 S.
#87 3-7776-0403-8
#90 2647-3611
#01 2=Bd. 2
#20 ¬Die¬ Erkenntnis der Natur : Beiträge zur modernen Naturphilosophie
#77 350 S.
#87 3-7776-0404-6
#90 2650-9705
```

Jede Kategorie muß hier auf einer neuen Zeile, mit einem '#' auf der ersten Position, beginnen. Zeilenumbrüche sind nicht nötig, d.h. jede Kategorie kann eine lange, ununterbrochene Zeile sein. Wenn jedoch Umbrüche auftreten, müssen diese bei einem Leerzeichen stattfinden, und die Folgezeile muß mit diesem Leerzeichen beginnen. Wenn man diese Form mit anderer Software aus Fremddaten herstellen kann, kann das *allegro*-System sie einlesen.

Die Programme merken beim Einlesen selbst, und zwar an den Kategorien #00 und #01, wo ein Satz bzw. Untersatz beginnt. (In einem anderen Schema, wie z.B. \$P.CFG, sind dies andere Nummern, z.B. #0100 und #70XX.)

## 0.3 Dateitypen

### Wozu dieser Abschnitt?

Alle Dateitypen sind hier zusammengestellt und kommentiert, mit denen *allegro* arbeitet. Das ist wichtig für Systemverwalter und Entwickler. "Normalanwender" werden diese Kenntnisse nicht so dringend brauchen.

*Nochmals der Hinweis:* unter UNIX sind alle Namen und Typen vollständig klein zu schreiben, also z.B. `cat.api` statt `CAT.API`.

Sie finden hier Angaben zu den Aspekten (und Antworten auf die Fragen):

<b>Inhalt</b>	der Dateien ("Was steht drin?"),
<b>Struktur</b>	("Wie sind die Dateien aufgebaut?"),
<b>Bezug</b>	und <i>Standort</i> ("Womit hängt das zusammen? Wo stehen diese Dateien?"),
<b>Funktion</b>	("Wozu wird das gebraucht?"),
<b>Entstehung</b>	("Woher kommt die Datei, wie entsteht sie oder wer legt sie an?"),
<b>Zugriff</b>	("Wer oder welches Programm kann, darf, muß daran etwas machen?"). Hauptsächlich geht es dabei um den <b>Schreibzugriff</b> .

Nicht alle Angaben sind für das Arbeiten mit *allegro* zum Verständnis notwendig, das Handbuch soll aber jedem an Details Interessierten eine möglichst vollständige Dokumentation bieten.

Ein anschauliches Bild der wichtigsten Zusammenhänge zeigt Anhang C. Eine weitere Übersicht in 0.8 zeigt, welche Dateien zu einer konkreten Datenbank gehören.

Für andere Programme zugänglich, vor allem Textprogramme, sind nur die Typen, bei denen als Struktur "ASCII-Textdatei" angegeben ist. Unter *CockPit* ist angegeben, wie man bequem an die Dateien herankommt (=> 0.11).

Typ	Bedeutung/Inhalt:	Zugang über <i>CockPit</i>
-----	-------------------	----------------------------

### **.CFG Konfigurationsdatei** (=> Anhang A) µo k

**Inhalt:** **Kategorienschema** (z.B. A.CFG als Standard, D.CFG für "DB-MAB"), Angaben zur **Hardware, Bildschirmfarben, Datumskategorien, Dialogsprache, Abfrageliste** ("Erfassungsmaske") für die Dateneingabe., **Nummerngenerator**.

**Struktur:** ASCII-Textdatei

**Bezug:** Wichtig ist der erste Buchstabe des Namens, der **Konfigurationsbuchstabe** genannt wird. Alle von einem Kategorienschema c.CFG abhängigen Dateien haben den Konfigurationsbuchstaben 'c' als ersten Buchstaben ihrer Typenbezeichnung. Z.B. arbeiten die Parameterdateien der Typen .APR oder .API nur mit der Konfiguration A.CFG zusammen (siehe unten). In Netzwerken ist es günstig, jedem Mitarbeiter "seine" .CFG-Datei auf sein eigenes Verzeichnis zu legen. Die Programme suchen die .CFG nämlich zuerst auf dem Datenverzeichnis, dann auf dem Verzeichnis, von dem gestartet wurde, und wenn sie dort auch nicht steht, auf dem Programmverzeichnis.

**Funktion:** Alle Programme mit Ausnahme von ASORT und QRIX suchen und lesen die Konfigurationsdatei beim Start, um zu erfahren, wie die Anwenderdaten strukturiert sind. Als Standard nehmen alle Programme A.CFG an. Mit dem DOS-Befehl `set -k=p` kann man z.B. P.CFG als Standard vorgeben. Die Option `-k` (=> Kap.12.1) ermöglicht die Wahl einer spezifischen Konfiguration bei jedem Programmaufruf. Es kann unterschiedliche Versionen einer Konfiguration geben: wenn z.B. mehrere Bearbeiter mit verschiedenen Abfragelisten und Änderungsberechtigungen arbeiten sollen, erstellt man dafür je eine Konfiguration. Nur der Anfangsbuchstabe des Namens muß derselbe sein, also z.B. A1.CFG, A17.CFG oder ADELHEID.CFG. Alle drei beziehen sich auf dasselbe Kategorienschema A.CFG und können mit denselben Parameterdateien der Typen .API und .APR betrieben werden.

**Entstehung:** Mehrere werden mitgeliefert (A, D, M, PICA, U) und können beliebig verändert werden. Eigene, neue Konfigurationen können erstellt werden (=> Anh.A).

**Zugriff:** mit jedem Texteditor zu erstellen und zu bearbeiten.

Typ	Bedeutung/Inhalt:	Zugang über <b>CockPit</b>
-----	-------------------	----------------------------

<b>.cLD</b>	<b>allegro-Datenbank-Datei</b>	<b>µr b</b>
-------------	--------------------------------	-------------

- Inhalt:** In solchen Dateien stehen die "echten" Daten des Anwenders.
- Struktur:** Datensätze sind sequentiell angeordnet, intern in der Reihenfolge der Kategorienummern, wie durch die .CFG-Datei festgelegt. Jeder Satz beginnt mit der automatisch vergebenen Satznummer (4 stellig, hexadezimal). Ansonsten stimmt die Dateistruktur mit derjenigen der Grunddateien (siehe .cLG) überein.  
Die Einzelheiten der Struktur sind schon in 0.2 beschrieben.
- Bezug:** Zum Zugriff auf die Daten werden die .cDX- und die .TBL-Datei gebraucht. Zu einer Datenbank *dbn* kann es bis zu 255 .cLD-Dateien geben. Die Dateinamen setzen sich aus *dbn* und der Nummer zusammen: z.B. buch\_17.ALD. Der Name *dbn* darf deshalb nicht länger als 4 Zeichen sein.  
*Standort:* Diese Dateien stehen zusammen mit der .TBL-Datei auf einem eigenen Verzeichnis, dem Datenbankverzeichnis des Anwenders. Hier stehen in der Regel auch die .ADX- und die .STL-Dateien, jedoch können diese auch anderswo gelagert sein.
- Funktion:** Datenspeicherung. Der Platz wird vom Programm 1 (= PRESTO), mit dem man die Daten bearbeitet, automatisch verwaltet, insbesondere dann, wenn bei der Bearbeitung der Satz länger geworden ist.
- Entstehung:** Entsteht beim Aufbau einer Datenbank mittels Programm INDEX und wird bei der Datenerfassung im Programm PRESTO automatisch erweitert.
- Zugriff:** für Bearbeitung und evtl. Export: Programm 1 (PRESTO) und 9 (UPDATE); lesend mit oder ohne Export auch per Volltextsuche (Programm SRCH); zum Aufbau und Ergänzung einer Datenbank mit Programm INDEX.

<b>.cDX</b>	<b>Index-Datei</b> (=> Kap.1.4, 7, 10)	<b>µr o i/c</b>
-------------	--	-----------------

- Inhalt:** Suchbegriffe (Zugriffs-Schlüssel) zu einer Datenbank, bis zu 11 Register.
- Struktur:** "Präfix-B\*-Baum" : das ist eine heute in der Informatik speziell für Indexdateien bevorzugte Struktur. Hier ist sie realisiert mit variabler Schlüssellänge und "Frontkomprimierung" (spart Platz: je größer die Datenbank, umso mehr); zu jedem Schlüssel sind die internen Nummern der zugehörigen Datensätze gespeichert.
- Bezug:** arbeitet nur mit einem Schema c.CFG zusammen. Zu jeder Datenbank *dbn* gehört eine Index-Datei *dbn.cDX*, eine Index-Parameterdatei *dbn.cPI* und eine Satztable *dbn.TBL*, ferner als Hauptteil eine oder mehrere Dateien *dbn\_k.cLD*. Dabei ist *k* eine Zahl zwischen 1 und 255.  
*Standort:* Die Datei muß auf dem Datenverzeichnis stehen, mit Option -y kann ein anderes vorgegeben werden (siehe Kap.12).
- Funktion:** Sie wird für den Datenbank-Schnellzugriff benötigt (Kap.1) und kann inhaltlich nach Wunsch des Anwenders individuell gestaltet werden (Kap.7 und 10). Dazu dient die Parameterdatei des Typs .cPI (s.u.).
- Entstehung:** wird durch INDEX erstellt und durch PRESTO und UPDATE während des Zugriffs automatisch verwaltet.
- Zugriff:** nur automatisch per PRESTO, UPDATE, INDEX, SRCH und QRIX. Das letztgenannte ist einerseits ein Hilfsprogramm, das von INDEX aufgerufen wird, es kann aber auch einige nützliche Funktionen an einem Index selbständig ausführen (=> Kap.7.5).  
Von eigenmächtigen, schreibenden Zugriffen mit Editoren, Software-Tools oder selbstgeschriebenen Programmen auf eine .ADX-Datei ist abzuraten.

Typ	Bedeutung/Inhalt:	Zugang über <b>CockPit</b>
-----	-------------------	----------------------------

## **.TBL**      **Satztabelle**

μr o t

- Inhalt:** 4 Byte je Datensatz für die Zuordnung der Satznummer zur realen Adresse. Die ersten zwei Bytes am Anfang der Datei dienen Steuerzwecken. Derzeit wird nur das erste Byte verwendet: normalerweise ist es 0, während eines Schreibvorgangs wird es auf 1 gesetzt, damit niemand gleichzeitig in die Datei schreiben kann. Per **CockPit** kann man es verändern: μr o s setzt es auf 1 und sperrt damit die Datenbank für Schreibzugriffe, μr o e (entsperren) gibt sie wieder frei.
- Struktur:** sequentiell in aufsteigender Reihenfolge der Satznummern. Dabei ist das erste Byte die Dateinummer (1..255), die anderen 3 Byte geben die Position des Satzes in der Datei an. Wenn N eine Satznummer ist, findet man an der Position N\*4-2 die Adresse des Satzes.
- Bezug:** Sie bildet eine funktionale Gesamtheit mit .cDX und .cLD! Von der Konfiguration hängt die Tabelle nicht ab - sie würde bei jeder Konfiguration gleich aussehen. Daher kommt 'c' nicht in der Typbezeichnung vor. *Standort:* siehe .cLD .
- Funktion:** dbn.TBL ist Bindeglied zwischen dbn.cDX und den Datenbank-Dateien dbn\_k.cLD, indem sie jeder Satznummer die reale Adresse zuordnet. Im Index findet PRESTO die Satznummer, und ein Zugriff auf die .TBL-Datei liefert dann die Adresse.
- Entstehung:** wird automatisch mit der Datenbank aufgebaut (per Programm 7 = INDEX) und wird, wo nötig, automatisch aktualisiert. Erneuerung möglich durch **CockPit** μr o t .
- Zugriff:** nur automatisch aus den Programmen 1, 4, 7 und 9.  
Von jedem direkten Eingriff durch den Anwender wird abgeraten.

## **.STL**      **Kurztitelregister**

μr o k

- Inhalt:** eine bis zu 72 Zeichen lange Zeile für jeden Datensatz.
- Struktur:** sequentiell in aufsteigender Reihenfolge der Satznummern, ohne Steuerzeichen. Die Position der Kurzzeile zu einem Datensatz ergibt sich aus dessen Satznummer multipliziert mit der für den Kurztitel vorgegebenen Länge (Parameter i0 in der Index-Parameterdatei). Daher kann es je Datensatz nur genau eine Kurzzeile geben. Die Zusammensetzung der Kurzzeile ist mit den Mitteln der Parametrierung frei gestaltbar (⇒ Kap.10). Für die Beispieldatenbank wurde die Konstruktion "Sachtitel /Verfasser Jahr Signatur" gewählt.
- Bezug:** Die Datei bildet eine funktionale Gesamtheit mit .cDX, .cLD und .TBL. Im Gegensatz zu diesen ist sie jedoch entbehrlich: wenn sie fehlt, gibt es eben keine Kurzanzeige.  
*Standort:* siehe .cDX (und Kap.12, Option -y).
- Funktion:** Die Kurzzeile wird angezeigt, wenn die Kurzanzeige per <Shift+F8> oder <Shift+F9> verlangt wird (⇒ 1.4).  
Eine Datenbank funktioniert auch ohne .STL-Datei (bis Version 11.2 gab es keine).
- Entstehung:** entsteht automatisch mit der Datenbank (per Programm 7 = INDEX) und wird, wo nötig, automatisch aktualisiert. Allerdings müssen in der Index-Parameterdatei dafür besondere Vorkehrungen getroffen sein (⇒ 10.2.6.3): es müssen die Befehle i0=72 und ak=zz+0 vorhanden sein und in der Kategorielliste ein Abschnitt mit der Sprungmarke #-0 . (Sonst hat man keine Kurzanzeige)  
Erneuerung jederzeit möglich durch **CockPit** μr o k .
- Zugriff:** nur automatisch aus den Programmen 1, 7 und 9.  
Direkte Eingriffe sind nicht sinnvoll, können allerdings keinen Schaden an den Daten anrichten, auch keine Folgeschäden außer fehlerhaften Kurzzeilen.

Typ	Bedeutung/Inhalt:	Zugang über <b>CockPit</b>
-----	-------------------	----------------------------

<b>.RES</b>	<b>Restriktionsdatei</b>	$\mu r \ o \ x$
-------------	--------------------------	-----------------

- Inhalt:** für jeden Datensatz eine feste Anzahl von Zeichen.
- Struktur:** wie .STL, nur mit entsprechend anderem Inhalt.
- Bezug:** Die Datei bildet eine funktionale Gesamtheit mit .CDX, .cLD und .TBL. Im Gegensatz zu diesen ist sie jedoch entbehrlich: wenn sie fehlt, gibt es eben keine Restriktionen.  
*Standort:* siehe .CDX .
- Funktion:** Die Angaben in dieser Datei werden gebraucht, wenn eine Suche eingeschränkt werden soll, z.B. nach Kriterien wie Erscheinungszeitraum oder Sprache ( $\Rightarrow$  1.4.6).
- Entstehung:** entsteht automatisch mit der Datenbank (per Programm 7 = INDEX) und wird, wo nötig, automatisch aktualisiert. Allerdings müssen in der Index-Parameterdatei dafür besondere Vorkehrungen getroffen sein ( $\Rightarrow$  10.2.9): es müssen die Befehle  $ir=k$  und  $ak=zz+/$  vorhanden sein und in der Kategorieliste ein Abschnitt mit der Sprungmarke #-/ .  
Erneuerung: jederzeit möglich durch **CockPit**  $\mu r \ o \ x$  .
- Zugriff:** nur automatisch aus den Programmen 1, 7 und 9.  
Direkte Eingriffe sind nicht sinnvoll, können allerdings keinen Schaden an den Daten anrichten, auch keine Folgeschäden außer fehlerhaften Kurzzeilen.

<b>.cLG</b>	<b>allegro-Grunddatei</b> ( $\Rightarrow$ Kap. 2, 3)	$\mu f \ 2/3$
-------------	--	---------------

- Inhalt:** Anwender-Datensätze, strukturiert gemäß Konfiguration c.CFG
- Struktur:** Der Aufbau ist derselbe wie bei der .cLD-Datei, es fehlen aber die Satznummern. (Einzelheiten sind in 11.0.1 beschrieben.)
- Bezug:** Mit Hilfe des Programms INDEX kann man .cLG-Dateien in eine Datenbank umwandeln bzw. in eine vorhandene Datenbank einspeisen. Diese Dateien haben keinen festen Standort., weil sie nicht zu einer Datenbank gehören.
- Funktion:** .cLG-Dateien gehören nicht zu einer Datenbank. Sie sind deshalb in der Regel ein Zwischenprodukt für die Weiterverarbeitung mit SRCH oder INDEX ("Export"). Grunddateien können sortiert werden (mit dem Programm ASORT, siehe 0.4). Die meisten Verfahren der Listenproduktion erzeugen zuerst sortierfähige Grunddateien und exportieren diese nach dem Sortieren erneut mit SRCH. *Beispiele:* PR-LIST.BAT oder QUEX.BAT.
- Entstehung:** als "download" (Funktion F4) aus einer Datenbank, als Ergebnis einer Volltextsuche oder eines Imports (Ausgabe-Parameter S-\*.cPR oder I-1.cPR), oder auch durch manuelle Eingabe und Bearbeitung (Hauptfunktion 2 = Programm ALLEGRO, "offline"-Erfassung).
- Zugriff:** Lesend mit SRCH (Volltextsuche) und INDEX (Datenbankaufbau);  
Schreibende Bearbeitung nur sequentiell ("offline") mit dem Programm ALLEGRO.  
Mit IMPORT können strukturelle Umwandlungen, etwa Änderungen an den Kategorien, vorgenommen werden ( $\Rightarrow$  11.2.3.1, Pauschal-Import). Das bedeutet, **allegro**-Daten können für grundlegende Änderungen quasi re-importiert werden:  
UPDATE kann eine .cLG-Datei lesen und in eine Datenbank einmischen, wobei vorhandene Datensätze durch neue ersetzt oder die neuen einfach hinzugefügt werden (siehe Kap. 9).

<b>.cIM</b>	<b>Import-Parameter-Dateien</b> ( $\Rightarrow$ Kap.5 und 11)	$\mu d \ i..$
-------------	---	---------------

- Inhalt:** Beschreibung eines fremden Datenformates und genaue Anweisungen, wie dieses in das **allegro**-Format c.CFG umzuwandeln ist. Z.B. enthält OCLC.UIM die Beschreibung der Umwandlung von OCLC-Daten in das Schema U.CFG (= USMARC-Standardschema).
- Struktur:** ASCII-Textdatei, mit jedem Texteditor zu erstellen und zu bearbeiten.
- Bezug:** *Standort:* gesucht werden diese Dateien zuerst auf dem Quellverzeichnis (wo die einzulesenden Fremddaten stehen), dann auf dem Programmverzeichnis.
- Funktion:** Steuerung der Datenumwandlung im Programm 5 (= IMPORT).
- Entstehung:** einige werden mitgeliefert (DBDISK.AIM, OCLC.AIM, ...), weitere können vom Anwender beliebig erstellt werden.
- Zugriff:** mit jedem Texteditor zu erstellen und zu bearbeiten. Neben den Befehlen können beliebige Kommentare vorkommen.

Typ	Bedeutung/Inhalt:	Zugang über <b>CockPit</b>
-----	-------------------	----------------------------

<b>.cPx</b>	<b>Export-Parameter-Dateien</b> (=> Kap. 10)	µd e ..
-------------	--	---------

**Inhalt:** Genaue Vorschriften über die Form einer Datenausgabe (sog. "Export"), formuliert in der *allegro*-Exportsprache.

**Struktur:** ASCII-Textdatei.

**Bezug:** Jedes Programm außer dem Sortierprogramm (Nr. 8) benötigt für die Ausgabe von Daten Parameterdateien.

*Standort:* Die Programme suchen jede solche Datei zuerst auf dem Datenverzeichnis (Quellverzeichnis), wenn sie dort nicht ist, dann auf dem Startverzeichnis (wo man das Programm aufgerufen hatte), und zuletzt noch auf dem Programmverzeichnis.

**Funktion:** Jede Ausgabe, ob Bildschirmanzeige, Papierausdruck oder "download" (auch Indexdatei!), wird durch Parameter gesteuert. Auswahl und Reihenfolge der Datenfelder, Interpunktion, Zeichenumwandlungen, Zeilen- und Seitenumbruch u.v.m. können gesteuert werden.

**Entstehung:** einige werden mitgeliefert (D-1.APR, I-1.APR, E-1.APR S-\*.APR, P-\*.APR, Prototypen x-0.APR), weitere können auf dieser Grundlage vom Anwender beliebig erstellt werden.

**Zugriff:** mit jedem Texteditor zu erstellen und zu bearbeiten. Neben den Befehlen können beliebige Kommentare vorkommen.

Das 'x' steht für die 3 Arten R, T und I dieser Dateien: (=> Kap.10 und 6/7)

**.cPR** eigentliche **Produktions**-Parameterdatei.

Die Programme 1 - 3 benutzen automatisch die Parameterdatei D-1.cPR für die formatierte Anzeige auf dem Bildschirm. Wenn Sie diese Datei verändern, bekommen Sie eine andere Standardanzeige. I-1.cPR produziert eine Ausgabe im Original-*allegro*-Format, S-\*.cPR erstellen sortierfähige Dateien. Wenn I-1 zur Erstellung einer Exportdatei benutzt wird, bekommt diese automatisch den Typ .cLG, da es sich um Dateien mit der *allegro*-Grundstruktur handelt, die sich für eine Weiterverarbeitung z.B. mit dem Programm SRCH oder zur Einspeisung in eine andere Datenbank mittels INDEX eignen.

**.cPT** **Tabelle** : kann solche Teile enthalten, die in mehreren Parameterdateien identisch sind (z.B. eine Zeichentabelle für die Druckersteuerung (=> 10.2.4) oder eine Stopwortliste (=> Kap.10.2, Ladebefehl t ). Grundsätzlich kann eine .cPT-Datei alles enthalten, was in einer .cPR- oder .cPI-Datei stehen kann. Die sog. **Druckertreiber** werden mit .cPT-Dateien realisiert. Diejenigen .APT-Dateien auf der Lieferdiskette, deren Namen mit **P-** beginnen, sind Beispiele für Druckertreiber (=> 0.11.5, 10.2). Sie sind unabhängig von der Konfiguration, d.h. man kann z.B. P-EPSON.APT in U-EPSON-PPT umbenennen und hat den geeigneten Druckertreiber für das U-Schema.

**.cPI** **Index-Parameterdatei** (=> Kap.7 u. 10) : könnte auch **Datenbank-Definition** genannt werden; dient zur Produktion von Indexdateien und für den Schnellaufzug zur Aktualisierung der Indexdatei.

*Prototyp:* INDX.API. Diese können Sie als Muster verwenden. Sie enthält alle für eine Index-Parameterdatei wichtigen Teile und ist ausführlich kommentiert.

### Prototypen

Für die wichtigste Unterabteilung, die .cPR-Dateien, wurde eine Klassifizierung in 10 Typen mit festgelegten Namen entworfen, z.B. SORT für die Aufgabe der Sortieraufbereitung, DISPLAY für die Bildschirmanzeige. Am Anfang von Kapitel 10 finden Sie diese tabellarisch aufgeführt. Zur Konfiguration A wurde sodann für jeden Typ eine Parameterdatei geschrieben, die alle charakteristischen Merkmale hat und auch eine genau beschreibbare, nachvollziehbare Aufgabe ausführt. Diese Dateien werden als **Prototypen** bezeichnet, weil man sie als Muster für eigene Entwürfe benutzen kann. Die Namen dieser Dateien bestehen aus dem Anfangsbuchstaben des Typs und "-0", also z.B. **S-0.APR** und **D-0.APR**. Wenn man im *CockPit*-Menü "Dateien / Exportparameter" die Funktion "Neue Datei vom selben Typ erstellen" wählt, bekommt man eine Kopie des Prototyps und hat damit schon die wesentlichen Teile beisammen, die man für die jeweilige Aufgabe braucht.



Typ	Bedeutung/Inhalt:	Zugang über <i>CockPit</i>
-----	-------------------	----------------------------

**UIF...**     **User Interface File = Menütexe** (ohne Typbezeichnung => 0.6)      $\mu d m$   
 Diese Dateien enthalten die Texte der Menüs, Fehlermeldungen, Fragen und Instruktionen. Jeder Anwender kann daran Änderungen vornehmen, jedoch nur an den Texten, nicht an den als Zahlen und Sonderzeichen erscheinenden Steuerzeichen (z.B: %s oder %d). UIF1ENG enthält z.B. den englischen Text zum Programm 1 = PRESTO. Die Datei UIFA... ist für APAC und ALFA, UIFC... für *CockPit*.  
*Bearbeitung* als Textdatei mit jedem Texteditor.  
*Standort*: wie Parameterdateien, d.h. datenbankspezifische Dateien legt man ins Datenverzeichnis.

**IIInnn**     **Zwischendateien (temporär) bei der Indexproduktion:**      $\mu f q$   
 Der Anwender bekommt sie normalerweise nicht zu sehen, da sie nach erfolgreicher Indexierung verschwinden. Wenn aber eine Indexierung nicht korrekt durchgelaufen ist oder INDEX mit manueller Unterbrechung gestartet wurde, sind evtl. diese Dateien vorhanden, nicht aber die erwünschte Indexdatei von Typ .CDX. Das Programm QRIX mit Option -fq kann die II-Dateien dann zu der .CDX verarbeiten (=> 7.5).  
*Standort*: Datenverzeichnis.

**.BAT**     **Stapeldatei:**      $\mu m b$   
 kein *allegro*-, sondern ein MS-DOS-Dateityp, auch "Batch File" oder "Stapeldatei" genannt. Eine solche Datei, mit jedem Texteditor zu erstellen und zu bearbeiten, kann beliebige MS-DOS-Befehle und Programmaufrufe enthalten. Die *allegro*-Programme (=> 0.4) können in solche Makros eingebettet werden. Alle Variationen (Optionen) der Programmaufrufe werden in Kap.12 genau beschrieben. Makros bieten eine Möglichkeit, eigene Menüsysteme oder Arbeitsumgebungen zu schaffen, indem man häufig benötigte Abläufe (z.B. Listenproduktionen, Auswertungsprogramme, Importvorgänge) und Datenbankaufrufe zusammenfaßt.  
*Standort*: Im Prinzip beliebig, meistens auf dem Programmverzeichnis.

## CP.OPT / DEFAULT.OPT

**Vorgaben für *CockPit*** (d.h. für das Programm ACP.EXE):      $\mu o v$   
 Darin stehen alle anwenderseitig änderbaren Einstellungen des *CockPit*. Auch dies ist eine ASCII-Textdatei. Die mitgelieferte CP.OPT enthält Kommentare (abgedruckt in => 0.11.6).  
*Standort*: Diese Datei wird auf dem **Startverzeichnis** gesucht, d.h. dort, wo man ACP.EXE oder CP.BAT gestartet hat. Hängt man die Aufrufoption **-oxyz** an, nimmt *CockPit* die Datei xyz (Name völlig beliebig) als Vorgabendatei, sonst wird nach CP.OPT gesucht, wird sie nicht gefunden, nach DEFAULT.OPT.  
 Man startet das Programm ACP.EXE normalerweise mit Hilfe von CP.BAT, d.h. man gibt den Befehl **cp**. Darin steht ein Aufruf `acp . . . -oCP.OPT . . .`.  
 Alle anderen Programme, z.B. PRESTO, werten diese Datei nicht aus, nur ACP.EXE.

**CP.PRE**     **Vorgaben für den nächsten *CockPit*-Start**      $\mu r e 4/5$   
 In dieser Datei bewahrt sich *CockPit* die aktuellen Einstellungen (Datenverzeichnis, Datenbankname, Konfiguration) auf, wenn man aussteigt. Wenn ACP dann wieder gestartet wird, liest es CP.PRE, so daß man da fortfahren kann, wo man zuletzt gearbeitet hatte. Kann ohne Bedenken gelöscht werden.

**LISTE**     **Standard-Ausgabedatei**      $\mu r e 4/5$   
 Wenn man per *CockPit* Listen produziert (=> Kap.6), entsteht jedesmal eine Datei dieses Namens, d.h. bei jeder erneuten Produktion wird sie überschrieben. Form und Inhalt hängen von den gewählten Parametern ab.  
*Standort*: Hängt von der Stapeldatei ab, von der sie erzeugt wird (Option -e )  
 Will man eine LISTE länger aufbewahren, sollte man sie umbenennen.

## Tabelle der Dateien

*dbn* steht für den Datenbanknamen des Anwenders (bis zu 4 Zeichen!). Wenn nicht mit A.CFG gearbeitet wird, sondern z.B. mit D.CFG, muß es .DPR statt .APR heißen, usw. (K = Kernsystem)

DATEI.TYP	K	Bedeutung und Funktion
<b>A.CFG</b> <b>\$.CFG</b>	x	Standardkonfiguration; enthält die Definition des Kategorienschemas, außerdem die Abfrageliste. Ab Version 13 (\$.CFG) erheblich erweiterte Möglichkeiten.
ARTIKEL.CFG	x	wird automatisch geladen, wenn in .CFG keine Artikelliste steht.
CCC.BAT		wird vom <i>CockPit</i> automatisch erstellt und startet den gewünschten Prozeß
CP.OPT	x	Vorgaben für das <i>CockPit</i> , anwenderspezifische Einstellungen (früher DEFAULT.OPT)
H...	x	Hilfsbildschirme; vom Anwender erweiterbar
D-1.APR	x	Standard-Anzeige-Parameterdatei für PRESTO etc.
(P1.APR)		Anzeigeparameter (wird genommen, wenn D-1.APR nicht existiert)
PR-LIST.BAT	x	Stapeldatei für Listenproduktion. Wird von <i>CockPit</i> gestartet, wenn die Routine "exportieren/1" gewählt wird. Die mit F4 exportierten Daten werden dann verarbeitet und unter dem Namen LISTE ausgegeben.
SR-LIST.BAT	x	funktioniert wie PR-LIST.BAT, nur mit Volltextsuche statt Schnellzugriff
P-*.APR	x	Parameterdateien für Druckformatierung; Auswahl per <i>CockPit</i> , gewählte Datei wird kopiert auf PRINT.APR
P-*.APT	x	Druckertreiber; Auswahl per <i>CockPit</i> "Optionen/Drucker", gewählte Datei wird kopiert auf PRINTER.APT
S-*.APR	x	Parameterdatei für die Sortierung von Listen. Auswahl per <i>CockPit</i> , gewählte Datei wird kopiert auf SORT.APR
S-*.APT	x	Parameterdatei für die Vollständigkeit (Kategorieumfang) einer Liste. Auswahl per <i>CockPit</i> , gewählte Datei wird kopiert auf SELECT.APT
PRINT.APR, PRINTER.APT, SORT.APR, SELECT.APT		Diese Dateien sind Kopien von Dateien der Typen P-*.APR, P-*.APT (Druckertreiber), S-*.APR und S-*.APT. Diese Kopien werden vom <i>CockPit</i> angelegt und von SR-LIST.BAT und PR-LIST.BAT verwendet.
UIFilng	x	"User Interface Files" = Textdateien mit den Menütexen für die einzelnen Programme. Vom Anwender veränderbar. i=Programmnummer, lng=Sprachcode
*.AIM	x	Import-Parameterdateien (ASCII-Text) für die Umwandlung von Fremddaten
<i>dbn</i> .API	x	Index-Parameterdatei für die Datenbank <i>dbn</i> (z.B. CAT.API)
<i>dbn_n</i> .ALD		Datendatei, enthält die vom Anw. erfaßten Datensätze, Felddefinitionen in A.CFG
<i>dbn</i> .ADX		Index, enthält alle Register, die in der .API des Anwenders definiert sind
<i>dbn</i> .TBL		Tabelle d. Satzadressen, stellt für Zugriff Verbindung zw. .ADX und .ALD her
<i>dbn</i> .STL		Kurztitel-Register. Definition in .API
<i>dbn</i> .RES		Restriktionen-Datei (ab V15). siehe 10.2.9
<i>dbn</i> .APT		Überschriften der Register. Wichtig bei Datenbankwechsel mit Alt+A (10.2.1.3)
<i>dbn</i> .LOG		aufgezeichnete Transaktionen; nötig zur Rekonstruktion nach Absturz
*.ALG		Grunddatei (wie .ALD, aber ohne Satznummern); entsteht z.B. bei Import
S1.ASP	x	Tabelle der Zeichenumwandlung für die Volltextsuche (Prog.4 = SRCH.EXE)

## 0.4 Programme

### Wozu dieser Abschnitt?

*allegro* ist kein einzelnes großes Programm, sondern es ist ein **Softwarepaket**, das aus etlichen Programmen und vielen verschiedenen Dateien (⇒ 0.3) besteht. Das Kernsystem von etwa 10 Programmen wird durch einige Spezialprogramme ergänzt. Dieser Abschnitt stellt die Kernprogramme vor. Jedes wird in einem der Kapitel 1 bis 9 genau beschrieben. Hier nicht erwähnt sind die Programme der "neuen Generation": VPW.EXE, AVANTLEXE, PRESTOW.EXE etc.

Die Programme können von Fall zu Fall auf unterschiedliche Art aktiviert werden:

- Entweder man startet das *CockPit* mit dem Befehl **cp** (⇒ 0.11), dann erscheint das Menü "Routinen"; mit <Alt>+f schaltet man ins Menü "Funktionen" und ruft jedes Programm mit seiner Ziffer auf; dann sind vor dem Start in einem Untermenü einige Felder auszufüllen, µf
- oder man ruft nur das gerade benötigte Programm von MS-DOS aus mit seinem Namen auf und beantwortet die Fragen, die es stellt,
- oder man stellt Stapeldateien (UNIX: Shell-Skripte) zusammen (⇒ Kap.12), in die man die Programmaufrufe mitsamt den für die eigene Anwendung nötigen Optionen einbettet. Dies kann bis zur Gestaltung eigener Menüoberflächen gehen. Per *CockPit* kann man die Stapel erstellen, bearbeiten und starten. µm

Die Beschreibung eines Programms mit seinem Ablauf und seinen Funktionen finden Sie im entsprechenden **Kapitel mit derselben Nummer**. Die Hauptfunktionen 2 und 3 werden vom selben Programm (ALLEGRO) ausgeführt. Dieses wird in Kapitel 2 beschrieben, welches auch in den Gebrauch des integrierten Editors einführt. Dieser ist auch in den Programmen 1, 4 und 5 enthalten. Seine Funktionen sind vollständig in Kap.3 beschrieben.

**0**      **ACP**      *CockPit* (⇒ Kap.0.11) [nicht für UNIX]  
Die "Benutzeroberfläche" für die Steuerung aller Vorgänge einer *allegro*-Umgebung

**Funktionen:** **Einstellen** und Bearbeiten von Optionen und Parametern  
**Auswahl** von Datenbanken  
**Start** von Programmen und Prozessen

**1**      **PRESTO Schnellzugriff** [UNIX: nur presto]  
**MENUED ... mit Menü-Erweiterung**  
**APAC OPAC-Programm**  
**ALFA ... mit Einfach-Ausleihe**

Das Hauptprogramm für **Zugriff, Erfassung und Bearbeitung**;  
APAC ist eine Version nur für den lesenden Zugriff, geeignet als OPAC-Programm.  
Von PRESTO und APAC gibt es grafikfähige Versionen: PRESTOG und APACG  
(⇒ 10.2.7), aus denen beliebige andere Programme aufgerufen werden können.  
Wichtig für Netze: APAC benötigt keine Berechtigung für Schreibzugriff.

**Zugriff auf:** Datenbankdateien (Typ .cLD) in Kombination mit Indexdatei (Typ .cDX und .STL) und Satztable (Typ .TBL)

**Zugriffsart:** Schlüssel (Namen, Stichwörter, Titel, Schlagwörter...) mit logischer Kombination durch UND ODER NICHT

**Funktionen:** **Retrieval** mit **Browsing** (die eingeschränkte Version APAC enthält nur diese Funktionen)  
**Bearbeitung** und formatierte **Ausgabe** gefundener Datensätze  
**Erfassung** neuer Daten mit sofortiger Index-Aktualisierung

**2/3**      **ALLEGRO : "offline"-Erfassung und "Redigieren" von Grunddateien**  
(⇒ Kap.2.3/4; veraltet; Empfehlung: nur noch PRESTO benutzen!) [nicht für UNIX]

**Zugriff:** sequentiell auf einzelne *allegro*-Grunddateien (Typ .cLG)

**Funktionen:** 2 : Dateneingabe und Speicherung als .cLG-Datei  
3 : Bearbeiten vorhandener .cLG-Dateien (z.B. Export- oder Importdaten)  
Ausgabe per Exportfunktion möglich

## 4 SRCH Volltext-Suche und Export

**Zugriff auf:** Dateigruppen

**Zugriffsart:** sequentiell (auf .cLG- und .cLD-Dateien)

**Funktionen:** **Volltext-Suche** nach Zeichenketten, mit logischer Kombination. Beliebige Ausgabe (Export) der Ergebnisse, fungiert deshalb zugleich als **EXPORT-Programm** mit oder ohne Selektion. Ein eigenes Programm namens EXPORT gibt es deshalb nicht.

## 5 IMPORT Fremddaten-Konvertierung

**Zugriff auf:** sequentiell auf beliebig strukturierte Fremddaten

**Funktionen:** Umwandlung fremder Daten in *allegro*-Dateien (Typ .cLG) mit Hilfe von Import-Parameterdateien (Typ .cIM). IMPORT schreibt nicht in die Datenbank! Zum Einordnen in eine Datenbank (Typ .cLD) ist anschließend Programm 7 = INDEX zu benutzen (s. Kap.7) oder UPDATE (s. Kap.9).

## 7 INDEX Index-Produktion

## q QRIX Index-Management

**Zugriff auf:** Dateigruppen (Typen .cLG und .cLD)

**Zugriffsart:** sequentiell

**Funktionen:** Generierung einer Datenbank mit Index-Datei (Typ .cDX), Satztable (Typ .TBL), Kurztitelregister (Typ .STL) sowie Datenbankdateien (Typ .cLD). QRIX wird von INDEX bei Bedarf automatisch aufgerufen, hat aber auch eigene Funktionen. Teilfunktionen:

- Einarbeitung neuer Dateien in eine Datenbank (Option -f7);
- Neu-Indexierung einer vorhandenen Datenbank (mit oder ohne Beibehaltung der alten Satznummern: Optionen -f7, -fn, -fi, -ft, -fs);
- Wiedergewinnung von ungenutztem Platz in .cLD-Dateien (-fr)
- Index kompaktieren, einzelne Register herausnehmen (QRIX)

## 8 ASORT Sortierprogramm

**Zugriff auf:** einzelne Dateien (auch andere als *allegro*-Dateien!)

**Funktion:** alphanumerisches Sortieren von beliebig großen Dateien mit variabel langen Sätzen. Das Programm kann durch andere, z.B. GNU-Sort, ersetzt werden.

## 9 UPDATE Datenbank-Management

**Zugriff auf:** .cLD- .cDX- und .TBL-Dateien mit Ein-/Ausgabe, .LOG- bzw. .cLG-Dateien als Eingabe

**Zugriffsart:** Satznummern bzw. "Primärschlüssel"

**Funktionen:**

- a) **Playback** : Datenbank-Regenerierung nach "Absturz"
- b) **Mischen** : Einspeichern neuer Daten (.cLG) mit Ersetzung von vorhandenen Sätzen durch neue (sonst => 7).
- c) **Check** : "Dublekkenkontrolle"

Auf den folgenden zwei Seiten werden weitere Programme erwähnt, die nicht alle zum Kernsystem gehören. Die UB Braunschweig führte 1992 für die **Sacherschließung** ein Programm namens REF ein, das die komfortable Bearbeitung aller für den sachlichen Zugriff relevanten Kategorien ermöglicht, ferner Datenübernahme aus der SWD. 1992 erstmals ausgeliefert wurde das Programm ORDER, mit dem man für die **Monographienerwerbung** die Funktionen Bestellkatalogisierung und Akzession automatisieren kann. Für die Ausleihe gibt es *aLF* (= *allegro*-Leih-Funktionen). Diese Programme sind Erweiterungen des Programms 1, haben also dieselben Zugriffsmöglichkeiten.

**Zusatzprogramme**

(bis auf LOG2ALG nur für DOS verfügbar)

**aLF****Ausleihprogramm** (Hierzu existiert eine eigene Dokumentation)

**Zugriff auf:** Datenbank *dbn* wie PRESTO  
**Zugriffsart:** wie PRESTO  
**Funktionen:** Ausleihvorgänge, Vormerkungen, Benutzerdatenverwaltung, Kalenderfunktionen, Exemplar-  
datenverwaltung, Fristüberwachung, Gebührenfunktionen.

**ORDER****Erwerbungsprogramm** (hierzu existiert eine eigene Dokumentation)

**Zugriff auf:** Datenbank *dbn* wie PRESTO  
**Zugriffsart:** wie PRESTO  
**Funktionen:** Vorakzession, Bestellvormerkung, Bestellungen, Akzession, Nummern- und Signaturvergabe,  
Kontingentverwaltung, Lieferantendaten, Exemplardatenverwaltung, Fristüberwachung,  
Währungsumrechnungen

**ALFX / ORDERX Ausleih- bzw. Erwerbungs-Systemdaten pflegen**

Menügeführte Erfassung und Pflege aller Systemdaten

**ALFA****Einfach-Ausleihprogramm** (nur Ausleihe/Rücknahme)

Arbeitet wie APAC, wird genauso gestartet. Mit TAB wird die Ausleihfunktion ausgelöst. Mit  
<ALT>+F10 schaltet man zwischen Ausleihe und Rücknahme um. Benutzer und Datum  
werden einfach an Kategorie #90 angehängt. Wenn es eine andere sein soll, z.B. #91: Start  
mit -C#91. Benötigt wird UIFAGER (Darin auch Kommentare)

**REF****Referentenprogramm** (hierzu existiert eine eigene Dokumentation)

**Zugriff auf:** Datenbank *dbn* wie PRESTO (wird genauso gestartet)  
**Zugriffsart:** wie PRESTO  
**Funktionen:** a) Erfassung und Pflege von Sacherschließungs-Stammdaten: Schlagwort-Normsätze,  
Klassifikation, Aufstellungssystematik, Sprach- und Regionalschlüssel.  
b) Eingabe und Bearbeitung aller Sacherschließungselemente zu Titelaufnahmen.  
Dabei kann z.B. die SWD als Ressource benutzt werden.  
Benötigt wird UIFRGER (Darin auch Kommentare für Anpassungen)

**INVENT****Inventarisierung von Exemplaren und Bänden**

**Zugriff auf:** Datenbank *dbn* wie PRESTO  
**Zugriffsart:** wie PRESTO  
**Funktionen:** Erfassung und Bearbeitung von Exemplarsätzen und Zeitschriften-Einzelbänden, vor allem  
für das Ausleihsystem, aber auch von Bestandssätzen für Zeitschriften. Die Datensätze  
werden als getrennte Sätze gespeichert, die Verknüpfung zum Titelsatz geschieht über die  
Titel-Identnummer. Die Sonderfunktionen werden immer mit der TAB-Taste ausgelöst.

**LOG2ALG****.LOG-Datei in .ALG-Datei umwandeln**

**Zugriff auf:** Die Protokolldatei *dbn.LOG* der Datenbank *dbn*  
**Zugriffsart:** sequentiell  
**Funktionen:** Die Datensätze, die als neue, geänderte und gelöschte Sätze während der Arbeit an einer  
Datenbank in der Datei *dbn.LOG* protokolliert wurden, werden in eine Grunddatei *dbn.ALG*  
überführt. Dabei verschwindet die interne Satznummer, die nur vom Programm UPDATE für  
die Wiederherstellung nach einer Zerstörung gebraucht wird. Gelöschte Sätze erhalten die  
Zusatzkategorie #u1 @@@@, neue Sätze werden mit #u1 #####nnn gekennzeichnet (nnn  
= Dateinummer, in der der Satz gespeichert wurde). Wenn es einen Primärschlüssel gibt,  
kann man *dbn.ALG* mit `update -fm11` in eine andere Datenbank einmischen lassen, um  
dort dieselben Vorgänge nachzuvollziehen. *dbn.ALG* hat außerdem den Vorteil, daß man es  
mit dem Volltextsuchprogramm SRCH durchsuchen kann.

## Hilfsprogramme (bis auf RDWR nur für DOS verfügbar)

**AW**                    **Zeichensatz-Tabelle einblenden**  
 Residentes Programm, das mit Alt+w jederzeit aktiviert werden kann. Es erscheint die Tabelle des aktuellen Zeichensatzes. Mit dem Cursor kann man jedes Zeichen auswählen, mit <Enter> holt man es in den aktuellen Bildschirm.

**FONTLOAD** **Anderen Zeichensatz laden**  
 Die Datei OSTWEST.FON für den erweiterten DIN 31628-Zeichensatz lädt man mit dem Befehl  
`fontload <ostwest.fon`  
 OSTWEST.FON ist eine ASCII-Datei, man kann sie bearbeiten.

**GK**                    **Get Key (Tastendruck abwarten)**  
 In jeder Stapeldatei einsetzbar. Aufruf z.B.  
`echo Weitermachen? j / n`  
`gk /G jn`  
 Nur mit 'j' oder 'n' geht es weiter. ERRORLEVEL ist dann 1 bzw. 2.

**JUMPS**                **Sprungmarken auflisten** (von Heinrich Allers)  
 Dieses Programm liest .CFG-Dateien sowie Parameterdateien (Typen .cPR und .cPI) und stellt die darin vorkommenden Sprungbefehle und Sprungmarken fest. Ergebnis ist eine Tabelle der vorkommenden Sprungmarken mit Fehlerprotokoll (Sprünge ohne Ziel, Marken ohne Sprung, ...)

**RDWR**                **Lesen + Schreiben**  
 Dieses Programm liest eine sequentielle Datei und schreibt ihren Inhalt ab einer vorgegebenen Zeile und schreibt dann den Inhalt bis zu einer anderen vorgegebenen Zeile in eine neue Datei. Der Aufruf erfolgt mit

`rdwr FROM TO INFILE OUTFILE [Hn] [0]`

Die Zeilen *FROM* bis *TO* werden aus *INFILE* in die Datei *OUTFILE* kopiert. Will man .ALD oder .ALG-Dateien in ASCII-Dateien überführen, gibt man zusätzlich H5 bzw H1 an, um die Steuerzeichen am Satzanfang zu übergehen, sowie 0, um einen Zeilenvorschub am Kategorieende zu bekommen. Als Satzende gilt 13 10 oder 10 (UNIX), d.h. in .ALD- und .ALG-Dateien gilt ein ganzer Datensatz als "Zeile", nicht eine Kategorie.

**ACREPLY**            **Benutzereingabe abwarten.** (Beispiele siehe in Kap. 12.2 und z.B. in PRONTO.BAT)  
 In jeder Stapeldatei einsetzbar. Die Eingabe des Benutzers (Zeichenkette, abgeschlossen mit <Enter>) wird in eine Batchdatei namens `acantw.bat` geschrieben. Wenn man diese dann aufruft mit `call acantw.bat`, wird die Antwort in die Environment-Variable `REPLY` kopiert.

**SNIFFER**            **Konsistenz einer Datenbank untersuchen**  
**Zugriff auf:**        Die Dateien `dbn.TBL`, `dbn_*.cLD`, `c.CFG`  
**Zugriffsart:**        sequentiell  
**Funktionen:**        Das Programm kann prüfen, ob die in der .TBL stehenden Adressen der Datensätze stimmen, ob in den .cLD-Dateien die Satzanfänge korrekt sind, und ob die Datensätze keine unerlaubten Kategorien enthalten. Entdeckte Fehler werden protokolliert. Außerdem kann SNIFFER gesperrte Sätze finden und wieder freigeben.

**FREE**                **Gesperrte Datenbank wieder freigeben**  
 Wenn endlos die Meldung "bitte warten" nach F10 kommt und man nicht mehr speichern kann.

## Tabelle der Programme

Die folgende Seite ist als Kurzreferenz gedacht und faßt alle Programme nochmals tabellarisch zusammen, und zwar mit Angabe ihrer Funktion und der zugehörigen Dateien. Außerdem wird angegeben, was zum Kernsystem gehört (K), und welche Programme derzeit für UNIX verfügbar sind (X).

Die mit (P) markierten Programme bauen auf dem Hauptprogramm PRESTO.EXE auf, d.h. sie enthalten alle Zugriffsfunktionen, aber nicht immer die Bearbeitungsfunktionen (Editor). Diese Programme brauchen dieselben Dateien wie PRESTO (.CFG, D-1.APR, .API, UIF0x, UIF1x; x steht für die "Sprachbezeichnung", also GER oder ENG etc.).

## Tabelle der Programme

(x = Sprachcode, z.B. GER)

Name	K	X	Funktion	dazu gehörige Dateien
<b>ACP.EXE</b> <b>CP.BAT</b>	x x		<i>CockPit</i> -Programm Damit startet man es	CP.OPT, UIFCx,
<b>ALF.EXE</b> <b>ALFA.EXE</b>	x		Ausleih-Programm (aufwendig) Einfach-Ausleihe (nur Ausgabe/Rücknahme)	UIFLx, (P) UIFAx
ALFX.EXE			Hilfsprogramm: Bearbeitung der <i>aLF</i> -Systemdaten	
<b>ALLEGRO.EXE</b>	x		Offline-Bearbeitungsprog. für .ALG-Dateien	UIF0/2x D-1.APR, .CFG
<b>APAC.EXE</b>	x		<b>allegro</b> -OPAC-Programm	UIFAx, (P)
<b>ASORT.EXE</b>	x	x	Sortierprogramm für .ALG- u. ASCII-Dateien	
CFGCONV.EXE	x		Umwandlung alte x.CFG -> neue \$x.CFG	
<b>IMPORT.EXE</b>	x	x	Umwandlung von Fremddaten in .ALG-Daten (Einmischen macht UPDATE oder INDEX!)	UIF0x, UIF5x, .CFG, .AIM, PA.APR o.a.
<b>INDEX.EXE</b>	x	x	Aufbau einer Datenbank aus .ALG-Dateien, Erneuerung des Index einer Datenbank	UIF0x, UIF7x .CFG, .API
<b>INVENT.EXE</b>			Inventarisierung (Bestand/Exemplare erfassen)	wie PRESTO (P)
<b>LOG2ALG.EXE</b>	x	x	.LOG-Datei in .ALG-Datei wandeln	
<b>MENUED.EXE</b>			PRESTO plus einige Menüfunktionen	wie PRESTO + UIFMx (P)
<b>ORDER.EXE</b>			Monographien-Erwerbung; dazu gehört:	UIF0x (P)
ORDERX.EXE			Hilfsprogramm: Bearbeitung der Systemdaten	
<b>PRESTO.EXE</b> <b>REF.EXE</b>	x	x	Datenbank-Zugriff und -Bearbeitung MENUED = Overlay-Version (bei Speichermangel)	UIF0x, UIF1x, .CFG, D-1.APR, .API, UIFR..
<b>QRIX.EXE</b>	x	x	Hilfsprogramm für INDEX.EXE, außerdem: Kompaktieren / Auslisten eines Index	UIFQx, .API
<b>SRCH.EXE</b>	x	x	Volltextsuche und Export	UIF4x, .CFG, (.APR), S1.ASP
<b>UPDATE.EXE</b>	x	x	Neue Daten oder .LOG-Datei einmischen	UIF0/9x, .CFG, .API
Hilfsprogramme:				
AW.EXE	x		Residenten Hilfsprogramm, Aktivierung: Alt+w	
FONTLOAD.COM	x		Anderen Zeichensatz laden (z.B. OSTWEST.FON)	
GK.COM	x		Tastenabfrage in Stapeldatei (ERRORLEVEL)	
JUMPS.EXE	x		Sprungbefehle und -marken untersuchen	
RDWR.EXE	x	x	Sequentielle Dateien lesen + schreiben	
ACREPLY.COM	x		zum Einsatz in Stapeldateien (Benutzereingabe abwarten, in Environment-Variable speichern)	
SNIFFER.EXE	x		Untersuchen von Datenbanken auf Inkonsistenzen	SNIFFER.INI
V.COM	x		Programm zum Anschauen von ASCII-Dateien	
X.EXE	x		Editor für ASCII-Dateien (Makros in X.MAC)	

## 0.5 Auswahl-Listen

Wann immer es notwendig ist, aus einer Anzahl von Dateien eine oder mehrere auszuwählen, geht dies mit demselben Verfahren. (Nur im *CockPit* geht die Dateiauswahl etwas anders, und zwar mit einem "Leuchtbalken", => 0.11.4).

1. Oft erscheint eine solche Liste automatisch, wenn das Programm schon "weiß", wo die Dateien stehen. Weiter geht es dann bei Punkt 2.  
Wenn es nicht klar ist, auf welchem Laufwerk und/oder Verzeichnis sich die Daten befinden, mit denen zu arbeiten ist, wird die Frage gestellt:

(Laufw. : ) (Pfad \ ) Name ?

Dabei sind mehrere Arten von Antworten möglich:

Wenn die gewünschten Daten auf demselben Laufwerk und Pfad sind, von dem aus man das Programm gestartet hatte, drückt man nur Enter. Wenn sie auf einem Unterverzeichnis des aktuellen Verzeichnisses stehen, braucht man nur dessen Namen einzugeben, nicht die komplette Pfadbezeichnung.

Beispiele:

**<Enter>** Die Dateien auf dem aktuellen Verzeichnis werden angezeigt,

- xyz**
- wenn es auf dem aktuellen Verzeichnis Dateien des gesuchten Typs gibt, deren Namen mit "xyz" beginnen, werden diese gezeigt
  - sonst wird geprüft, ob es ein Unterverzeichnis "xyz" innerhalb des aktuellen Verzeichnisses gibt; dann werden die darin befindlichen Dateien genommen

- xyz\**
- wenn beides zutrifft und man möchte gezielt das Unterverzeichnis ansprechen, muß man ausdrücklich "xyz\" eingeben.

**e:\xyz** So kann auch ein Pfad auf einem anderen Laufwerk ausgewählt werden

"xyz" kann auch ein mehrfach gestuftes Unterverzeichnis sein, und es kann mit einer Zeichenfolge enden, die mit dem Anfang bestimmter Dateinamen übereinstimmt, wobei man '?' als Maskierungszeichen verwenden kann. Also würde man mit "bd\??k" alle Dateien des Unterverzeichnisses BD auswählen, deren Name an vierter Stelle das Zeichen 'k' hat. Somit kann im Bedarfsfall sehr differenziert ausgewählt werden. Man vergleiche dazu auch die Möglichkeiten der automatischen Auswahl, wenn man die Programme als Makros benutzt (=> Kap.12.1, Option -d\*).

2. Es erscheint ein Menü mit den verfügbaren Dateien des relevanten Typs, die das Programm auf dem Datenverzeichnis findet.

Die Namen werden in bis zu fünf Spalten nebeneinander angezeigt, und zwar alphanumerisch sortiert. Die Auswahl wird getroffen, indem man mit den Cursortasten zu den gewünschten Dateinamen fährt, diese mit '+' markiert und <Enter> drückt, nachdem man den oder die betreffenden Namen markiert hat. Die Funktion der Tasten Home, End, 'a' (=alle), 'c' (=clear) und PgUp / PgDn (wenn die 5 Spalten nicht reichen!) erklärt sich von selbst.

Mit **F5** ist als Orientierungshilfe die erste Zeile der Datei abrufbar, wenn es sich um eine Textdatei (z.B. Parameterdatei) handelt. Die Zeile erscheint dann unterhalb der Liste.

Mit **F2** (nicht bei **-a0**) schaltet man um auf die entsprechenden Dateien desselben Typs (z.B. .APR) auf dem **Start**verzeichnis, nochmaliges **F2** schaltet zum **Programm**verzeichnis. Erneutes **F2** schaltet wieder zum **Datenbank**verzeichnis, usw. Ferner: F1 = Hilfe, F10 = Bearbeitung der Datei. (s.a. 10.3.3)

Mit **F10** wird die angewählte Datei in den X-Editor geladen (=> Anh.D) für sofortige Bearbeitung.

Wenn man gelegentlich "nicht an die Daten rankommt", obwohl man genau weiß, daß sie da sind, prüfe man, ob die Hintergrundvariablen -D und/oder -K anders besetzt sind. Man stellt dies fest, indem man auf MS-DOS-Ebene den Befehl **set** gibt. Dann werden diese (und andere) Variablen angezeigt. Mit **set -D=** und **set -K=** beseitigt man sie.

## 0.6 Menütexte, Sprache

Die Menütexte, Fragen und Fehlermeldungen der Programme sind in Textdateien ("ASCII Files") abgelegt und somit für Anwender zugänglich. Die Namen dieser Textdateien beginnen mit UIF (für "User Interface"), dann kommt eine Ziffer (= Programmnummer), und daran hängt dann noch (ohne Punkt!) die Sprachbezeichnung: GER für Deutsch und ENG für Englisch werden mitgeliefert, SPA für Spanisch und DUT für Niederländisch wurden von Anwendern erstellt.

UIF0s	Texte für den Programmkern (wird von allen Programmen außer QRIX und <i>CockPit</i> benötigt)		
UIF1s	Texte für Programm 1 = Schnellzugriff		
UIF2s	Texte für Programm 2 = Offline-Erfassung		
....			
UIFCs	Texte für <i>CockPit</i>	UIFAs	Texte für APAC und ALFA
UIFQs	Texte für QRIX	UIFRs	Texte für REF

Beispiele (aus UIF1GER):

```

152 "ist bereits gelöscht"
156 " Datenbank: %s   Datei#: %d   Satz#: %ld"
158 "Satzlänge: %d" 13 10
186 "Online-Katalog der Bibliothek ...   "
```

In der Zeile 186 können Sie also den Namen Ihrer Bibliothek einsetzen. Diese Zeile erscheint als Kopfzeile des Funktionsmenüs im Programm APAC und bei PRESTO -a0 .

Wenn Sie Veränderungen vornehmen wollen, sind folgende Punkte zu beachten:

1. Jede Zeile beginnt mit einer Nummer, die auf keinen Fall verändert werden darf. Die Reihenfolge der Zeilen ist allerdings beliebig. Eine Zeile steht für jeweils eine Meldung. Die Länge der Zeile kann man mit einem geeigneten Texteditor auch über 80 Zeichen ausdehnen. Fortsetzungszeilen gibt es nicht!
2. Die in Anführungszeichen stehenden Texte sind im Prinzip beliebig veränderbar. Zu achten ist evtl. auf die Länge. Verkürzungen sind unkritisch, Verlängerungen mit Vorsicht durchzuführen, da in manchen Fällen andere Bereiche des Bildschirms dann überlagert werden könnten. **Nicht verändern** darf man diejenigen Teile innerhalb von Anführungszeichen, die mit dem Zeichen '%' beginnen, z.B. Angaben wie %ld oder %s. An solchen Stellen erscheinen dann aktuelle Werte des Programms.
3. Außerhalb der Anführungszeichen stehen Steuerzeichen in Form von Dezimalzahlen, getrennt durch genau ein Leerzeichen. *Vorsicht:* wenn man mehr als ein Leerzeichen macht, fällt der Rest der Zeile weg (wie in den Parameterdateien). Prinzipiell dürfen auch hier Veränderungen und Ergänzungen vorgenommen werden, z.B. Farbumschaltungen nach ANSI-Standard, jedoch muß man dabei genau wissen, was man tut. Auskunft gibt jedes bessere MS-DOS-Handbuch. Die Werte 13 10, die sehr häufig vorkommen, stehen für Wagenrücklauf/Zeilen-vorschub, die Zahl 27 steht für das ESCAPE-Zeichen und leitet immer eine Befehlsfolge für die Bildschirmsteuerung ein (sog. ANSI-Escape-Sequenz). Diese Befehle ermöglichen es z.B. auch, eine bestimmte Meldung auf eine bestimmte Bildschirmposition zu setzen.

Die Programme laden die Textdateien in der per Option '-1' vorgewählten Sprache (=> Anhang A, .CFG-Datei), standardmäßig Deutsch (= GER). Gesucht wird zuerst auf dem Daten-, dann auf dem Programmverzeichnis. Wenn UIFxGER nicht existiert, wird auf denselben Verzeichnissen nach UIFx gesucht, also ohne Sprachkennung.

Nur von der Satzanzeige des Schnellzugriffs aus (=> Kap.1.5) kann die Sprache auch während des Programmlaufs umgeschaltet werden, indem man **F9** und dann die Sprachbezeichnung eingibt. Wenn die betreffenden UIF-Dateien nicht existieren, erhält man eine Fehlermeldung.

Für die Hilfsseiten (=> 0.3) gelten ähnliche Regelungen bzgl. der Sprache. Wenn ENG als Sprache eingestellt ist, sucht das Programm z.B. bei F1 auf Anzeigeschirm (1.5) nach H10ENG; wenn es dies nicht findet, wird H10 genommen.

## 0.7 Datensicherung: Konzept und Methoden

Bedenken Sie: Ihre Daten sind die weiche Frucht harter Arbeit - Sicherung tut not. Die unbedingt zu sichernden Daten stehen nur in den .cLD-Dateien - die anderen lassen sich rekonstruieren. Veränderungen an diesen Dateien werden nur durch PRESTO (und dessen Varianten) und UPDATE in einer Datei CAT.LOG protokolliert, wenn CAT der Name Ihrer Datenbank ist. Nehmen wir als Beispiel an, diese Datenbank befindet sich auf C:\ALLEGRO\KATALOG.

Um im Katastrophenfall die Datenbank **CAT** rekonstruieren zu können, geht man folgendermaßen vor. Hier ist der "Handbetrieb" beschrieben, um den interessierten *Allegrologen* ein detailliertes Verständnis zu vermitteln.

Der *CockPit*-Pilot hat es im Normalfall viel leichter: er aktiviert nur `µr s` bzw. `µr w` (⇒ 0.11.2).

1. Anfertigen einer Sicherungskopie (DOS-Standardmethode) µr s  
Eine genügende Anzahl formatierter Disketten muß bereitliegen.  
Man wählt mit dem Befehl CD das Verzeichnis KATALOG an, auf dem sich die Datenbank befindet.  
Dann gibt man die Befehle

```
rename cat.log xxx.log  (.LOG-Datei darf nicht mit gesichert werden!)
backup c:cat*.* a:      (o.a., z.B. TAPE für die Sicherung auf einem "streamer tape")
del xxx.log             (jetzt .LOG-Datei löschen)
```

wenn A: das zur Sicherung zu benutzende Diskettenlaufwerk ist und die Daten auf dem Laufwerk C: liegen (sonst statt c: das aktuelle Laufwerk angeben). MS-DOS fordert weitere Disketten an, wenn eine nicht reicht. Konsequenz: feuersichere Aufbewahrung der Kopie an einem vom Rechner entfernten Ort. Es ist **sehr** wichtig, die .LOG-Datei **nicht** mit zu sichern, sondern gleich nach dem Kopieren zu löschen, bevor man wieder an der Datenbank arbeitet. (Auch wenn Ihnen dies nicht einleuchtet, machen Sie es unbedingt trotzdem so!) Bei der nächsten Bearbeitung wird automatisch eine neue .LOG-Datei begonnen. **Diese** braucht man zum Rekonstruieren, wenn es denn mal dazu kommt!

2. Man arbeitet mit der Datenbank. In gewissen, frei bestimmbar Abständen wiederholt man den Vorgang 1. **oder** kopiert die .LOG-Datei ihrerseits auf eine Diskette (siehe unten).
3. Wenn irgendwann durch Plattenfehler, Stromausfall oder andere Störungen die Datenbank "abgestürzt" und nicht mehr brauchbar ist (keine Registeranzeige oder offensichtliche Fehler), löscht man die unbrauchbar gewordenen Dateien, **aber nicht** die .LOG-Datei, und speist die Sicherungskopie ein. Das geht so: man gibt nach Anwählen des Datenbank-Verzeichnisses die folgenden Befehle:

```
del cat*.cLD
del cat.tbl
del cat.cDX
restore a: c:      (bzw. entspr. Befehl für ein "streamer tape")
```

und führt danach die Schritte 4. bis 6. aus.

Alternativ, oder wenn man keine Sicherungskopie gemacht hat und wenn die .cLD-Dateien noch da sind, kann man auch so vorgehen: man löscht CAT.CDX und CAT.TBL und regeneriert dann mit dem Programm 7 (= INDEX) die Datenbank aus den noch vorhandenen .cLD-Dateien (⇒ Kap.7). Dazu nutzt man die Funktion **-fi** des Programms INDEX, dabei bleiben auch die Satznummern erhalten! Mit diesem Schritt wäre die Rekonstruktion abgeschlossen, d.h. die folgenden Schritte entfallen dann.

4. Danach kopiert man die .LOG-Datei, falls sie beim Zusammenbruch verschwunden war (das ist sehr unwahrscheinlich), von ihrer Sicherungsdiskette auf das Datenverzeichnis. Sie muß CAT.LOG heißen.
5. Schließlich gibt man, vom \ALLEGRO-Programmverzeichnis aus, den Befehl
 

```
update -fp -dkatalog\cat -ucat.log -nl
```

µr w

Hier muß man **-s** ergänzen, wenn man auf einem **Einzelplatz**system arbeitet. Daraus kann man sich ein Makro namens PLAYBACK machen. Oder (bequemer!) man setzt *CockPit* ein (⇒ 0.11.2).

6. Nach Ablauf dieses Programms wird die Datenbank wieder in dem Zustand sein, den sie vor dem Zusammenbruch (nach der letzten noch fehlerfreien Speicherung) hatte.
7. Gelegentlich beginnt man wieder neu bei 1, wenn die .LOG-Datei eine gewisse Größe erreicht hat. Eine vorgeschriebene Grenze gibt es nicht - zu empfehlen sind ca. 500 KB bis 1 MB. *CockPit* macht Sie ab 1.2 MB darauf aufmerksam, daß es an der Zeit wäre, eine Sicherung vorzunehmen.

Das Beste ist, so albern Sie es vielleicht auch empfinden, die Datensicherung zu einer Art rituellen Handlung zu kultivieren, die Sie ausführen, komme was wolle.

Wenn man LOG-Dateien auf Disketten oder auf ein anderes Laufwerk bzw. Dateisystem (UNIX) kopiert, muß man über die Vorgänge sorgfältig Buch führen und sich für eine von drei Methoden entscheiden:

- A** man kopiert jeweils die gesamte LOG-Datei und löscht sie **nur** zum Zeitpunkt einer Gesamtsicherung (Vorgang 1.) Diese einfachste Methode empfiehlt sich, wenn man genügend Platz auf der Platte hat. Man beachte aber die unten folgende Anmerkung.
- B** man numeriert die LOG-Dateien (cat1.LOG, cat2.LOG, ...) und **löscht sie nach dem Kopieren**. Dann muß man die Vorgänge 4. und 5. im Bedarfsfall mehrfach ausführen, zuerst mit cat1.log ...
- C** man kopiert die jeweils neu entstandene LOG-Datei an das Ende der gesicherten LOG-Datei (z.B. mit `copy /b A:cat.LOG + C:cat.LOG`) und **löscht sie nach dem Kopieren**. Dann brauchen 4. und 5. nur einmal ausgeführt zu werden. So geht es mit **CockPit**: Gesamtsicherung mit `µr s b`, für die **erste** .LOG-Datei `µr s L`, dann immer `µr s +` ("additiv") bis zur nächsten Gesamtsicherung. Man beachte aber:

#### Wichtige Anmerkung:

Nur Methode B kommt in Frage, wenn man mit INDEX zwischendurch neue Dateien einspielt. Die Regenerierung einer Datenbank muß dann so erfolgen, daß die LOG-Dateien in chronologischer Folge eingespielt werden und die neuen Dateien dazwischen, ebenfalls in korrekter Folge, erneut mit dem Programm INDEX einsortiert werden. Man führe also Vorgang B jeweils vor einem INDEX-Lauf durch, kopiere auch die neuen Dateien auf eine Sicherungsdiskette und nummeriere sie ebenfalls chronologisch.

Es wäre zu riskant, das INDEX-Programm ebenfalls seine Vorgänge in der LOG-Datei protokollieren zu lassen. Ein Absturz während eines INDEX-Laufes würde eine anschließende Regenerierung sehr schwierig gestalten und nicht immer zuverlässig ermöglichen. Wenn ein solcher Absturz passiert, führe man das PLAYBACK-Verfahren durch und wiederhole den INDEX-Lauf.

Wenn man mit UPDATE statt mit INDEX neue Daten einspeist, ist keine besondere Aktion vonnöten, denn UPDATE protokolliert wie PRESTO alles in der .LOG-Datei. Einen abgestürzten UPDATE-Lauf können Sie wiederholen - wenn der Index danach noch in Ordnung ist, denn die schon vorhandenen Sätze werden nicht verdoppelt.

Der Platzbedarf für die .LOG-Datei ergibt sich daraus, daß jede Aufnahme bei jedem Bearbeitungsvorgang, den man mit dem Speicherbefehl (F10 oder #r) abschließt, vollständig kopiert wird. Besonders dann, wenn man mit dem Platz sparsam umgehen muß, überprüfe man nach mehreren längeren Bearbeitungssitzungen die Größe der .LOG-Datei und wende nicht die Methode A an, weil dabei die .LOG-Datei auf der Platte bleibt und ständig wächst. (Wie oben schon erwähnt: **CockPit** paßt auf und macht Sie ab einer einstellbaren Dateigröße darauf aufmerksam (=> 0.11.6, Befehl L).

Eine **völlig andere Sicherungsmethode** ist diese:

Wenn man zwei oder mehr getrennte (nicht vernetzte) Rechner hat, auf denen dieselbe Datenbank installiert ist, gehe man so vor:

- Bearbeitung findet nur auf Rechner 1 statt, auf Rechner 2 etc. wird die Datenbank nur lesend benutzt. Rechner 2 kann z.B. auch ein UNIX-System sein.
- Zu gewissen Zeiten kopiert man die .LOG-Datei von Rechner 1 auf eine Diskette und speist sie mit dem UPDATE-Programm in die Datenbank auf Rechner 2 ein. Dazu genügt ein Befehl:

```
update -d dbp\dbn -fp -u a:cat.log
```

den man auf Rechner 2 von C:\ALLEGRO aus gibt, wobei dbp der Datenbankpfad und dbn der Datenbankname ist. Die Diskette mit cat.LOG muß zu dem Zeitpunkt in Laufwerk A: sein. Natürlich wird man für einen Routinebetrieb diesen Befehl in einem Makro, z.B. mit dem Namen NEWDATA.BAT verstecken.

- Auf Rechner 1 löscht man anschließend die .LOG-Datei. Die Datenbank auf Rechner 2 stellt nun quasi die Sicherungskopie dar!

Auf einem lokalen Netz mit Mehrplatzzugriff liegt diese Situation natürlich nicht vor, denn dort hat man nur ein Exemplar der Datenbank, an dem von mehreren Plätzen aus gearbeitet werden kann. Es kann aber sein, daß einer der angeschlossenen Rechner (der nicht im selben Raum steht!) eine genügend große Platte hat, die man zur Sicherung benutzen kann. Das ist dann schneller und bequemer als alles andere.

Im Kap. 9.2 wird erklärt, wie man zwei Datenbanken **gegenseitig** aktualisieren kann, wenn in beiden katalogisiert wird.

## 0.8 Was gehört alles zu einer Datenbank?

Hier ist zusammengestellt, welche Dateien zum Kern und Umfeld einer *allegro*-Datenbank gehören. Wenn man z.B. eine Datenbank mit allem Zubehör sichern, auf ein anderes System kopieren oder auch verkaufen will ("Database Publishing"), ist diese Aufstellung als Checkliste brauchbar. (Eingeklammert: die nur in Nichtstandardfällen notwendigen Dateien). Bei den einzelnen Dateien ist angegeben, ob und wie sie im Katastrophenfall **regenerierbar** (d.h. automatisch wiederherstellbar) oder (manuell-intellektuell) **rekonstruierbar** sind, oder ob nicht.

Nicht aufgeführt sind die Programme (Typ .EXE). Soll eine Datenbank auf einen anderen Rechner kopiert und dort nur zur Recherche benutzt werden, genügt das Programm APAC.EXE. Ansonsten müßte auf dem Zielrechner das normal erworbene und komplett installierte *allegro*-System vorhanden sein.

*Nochmals der Hinweis:* Unter **UNIX** sind alle Namen vollständig mit **Kleinbuchstaben** zu schreiben.

**Nehmen wir mal an**, die Datenbank heißt DAT und benutzt die Konfiguration K.CFG. Es ist hier unerheblich, auf welchem Unterverzeichnis die Dateien angesiedelt ist. Empfehlung: alle datenbankspezifischen Dateien auf das Datenverzeichnis bringen, alles andere auf das Programmverzeichnis. (Ersetzen Sie DAT durch Ihren eigenen Datenbanknamen und evtl. K durch Ihren Konfigurationsbuchstaben, z.B. A oder P.)

### Kernbestandteile

<b>K.CFG</b>	die <i>Konfiguration</i> : Kategorienschema und Abfrageliste; regenerierbar (=> Anh.A)
<b>DAT_n.KLD</b>	die <i>Datenbankdateien</i> (n = 1...255); vollständig regenerierbar, aber <b>nur</b> aus einer Sicherungskopie mit Hilfe von UPDATE und der aktuellen .LOG-Datei (=> 0.7)
<b>DAT.KDX</b>	<i>Indexdatei</i> ; zusammen mit DAT.TBL regenerierbar : benötigt werden die .KLD-Dateien, DAT.KPI und das Programm INDEX (Optionen -fi und -fn, => Kap.7), dabei wird auch DAT.STL erneuert und DAT.TBL, wenn sie nicht mehr da ist.
<b>DAT.TBL</b>	<i>Satztable</i> ; regenerierbar durch das Programm INDEX mit Option -ft. Ausführung über <i>CockPit</i> $\mu x \circ t$ )
<b>DAT.STL</b>	<i>Kurztitelregister</i> ; regenerierbar durch das Programm INDEX, und zwar mit der Funktion "Kurztiteldatei erneuern" (Option -fs : <i>CockPit</i> $\mu x \circ k$ ). Datenbank funktioniert ohne.
<b>DAT.RES</b>	<i>Restriktionsdatei</i> ; regenerierbar durch Programm INDEX mit Funktion "Restriktionsdatei erneuern" (Option -fx : <i>CockPit</i> $\mu x \circ x$ ). Datenbank funktioniert ohne.
<b>DAT.KPI</b>	<i>Index-Parameterdatei</i> ; bei genauer Kenntnis der Datenbank und der Parametersprache rekonstruierbar (=> 10)
<b>DAT.LOG</b>	<i>Sicherungsdatei</i> (=> 0.7); nicht regenerierbar. Wird nur benötigt, wenn eine beschädigte Datenbank wiederherzustellen ist. Nicht mit kopieren.
<b>D-1.KPR</b>	<i>Anzeigeparameter</i> . (=> Kap.10); rekonstruierbar

### Peripherie

(diese Dateien sind bei hinreichender Dokumentation alle rekonstruierbar, außer den UIF-Dateien ist keine zwingend erforderlich!)

<b>CP.OPT</b>	Datei mit den Vorgaben für das <i>CockPit</i> (=> 0.11), falls man damit arbeitet. Diese Datei kann auch anders heißen; sie wird über CP.BAT aufgerufen. (Muß auf Startverz. liegen)
<b>H DAT</b>	datenbankspezifische Hilfsseite; erscheint bei Shift+F1 vom Funktionsmenü
<b>H CDAT</b>	datenbankspezifische Hilfsseite bei Shift+F1 im Editor (Hinweise für die Erfassung)
<b>H*spr</b>	evtl. abgewandelte H-Dateien (spr = Sprachkennung: GER, ENG, etc.)
<b>UIF?spr</b>	evtl. abgewandelte UIF-Dateien (UIF0, UIF1, UIFA, spr = Sprachkennung)
<b>*.KPR</b>	evtl. datenbankspezifische Export-Parameterdateien für die Listenproduktion etc.
<b>*.KPT</b>	zugehörige Tabellendateien, z.B. Druckertreiber (=> 10.2)
<b>*.KIM</b>	evtl. spezifische Import-Parameterdateien für die Zufuhr von Fremddaten
<b>*.BAT</b>	evtl. Stapeldateien für bestimmte Produktionen, z.B. CP.BAT, PR-LIST.BAT.

## 0.9.1 Schnelle Antworten auf häufige Fragen

*Was ist bei der Installation zu beachten?* Das steht alles im Abschnitt 0.10.

*Wie bearbeite ich einen Datensatz?* Satz aufblättern, dann den Großbuchstaben 'E' drücken. (⇒ 3.1)

*Wie kopiere ich einen Datensatz?* Satz aufblättern, 'C' drücken, dann bearbeiten und mit F10 speichern.

*Wie lösche ich einen Datensatz?* Satz aufblättern, Taste [Entf] drücken, mit 'j' bestätigen.

*Und wie mache ich das rückgängig?* Satz wieder aufblättern (mit [Bild↑]), dann 'E' F10 'j'.

*Wie lösche ich eine Kategorie?* Im Editormodus den Befehl #v benutzen, z.B. #v27, um #27 zu löschen (⇒ 3.3).

*Und wenn das aus Versehen passiert ist?* Mit Befehl #t zurückholen: z.B. #t27 .

*Kann ich den Kurztitel ändern oder löschen?* Nein, der wird automatisch erzeugt und korrigiert. Wenn die Kurztitel eine andere Struktur haben sollen: Änderungen in der Indexparameterdatei im Abschnitt #-0 machen, dann vom **CockPit** `µr o k` (Organisieren / Kurzanzeige erneuern)

*Kann ich die unbelegten Felder eines Datensatzes anzeigen lassen?* Nein. Diejenigen, die zur Abfrageliste gehören, kann man nachträglich abfragen lassen mit Befehl #E.

*Wieviele Sätze enthält meine Datenbank?* Auf dem Anzeigebildschirm Taste [Ende] drücken, dann erscheint der letzte Satz und oben sieht man die letzte Satznummer. Die gelöschten Sätze sind dabei mitgezählt.

*Wieviele Sätze sind gelöscht?* Im Register 1 eingeben: F6 //?. Die Zahl steht danach links neben // .

*Wie wird man die gelöschten Sätze endgültig los?* Sie werden automatisch wieder zum Speichern neuer Sätze benutzt. Allerdings kann man die Datenbank auch "entlüften" (⇒ 0.11.2 "Organisieren / R"), danach sind sie weg.

*Wieviele Einträge stehen in den Registern?* Index kompaktieren, dann wird am Ende ein Protokoll erstellt, angezeigt und in der Datei PROTOQ gespeichert. Darin steht auch eine Statistik der gelöschten Sätze. Achtung beim Mehrplatzbetrieb: Nur kompaktieren, wenn keiner an der Datenbank arbeitet.

*Soll man regelmäßig kompaktieren oder den Index erneuern?* Notwendig ist es nicht. Der Index wird vielleicht etwas schneller, weil die Datei kleiner wird. Erneuern ist nur nötig, wenn man die Indexparameter verändert oder wenn der Index nicht mehr funktioniert (z.B. nach Stromausfall).

*Kann man die Schlüssel sehen, die alle zu einem Datensatz gehören?* Ja, mit F7, wenn der Satz in der Anzeige steht. *Daten einmischen: Ist INDEX oder UPDATE besser?* Wenn Datensätze durch neue ersetzt werden sollen, geht das nur mit UPDATE. INDEX darf man nicht im Mehrplatzbetrieb einsetzen (⇒ Kap.7).

*Kann man die ganze Datenbank als Ergebnismenge nehmen?* Ja, mit dieser Tastenfolge: [Pos1] '( [Ende] ')' .

*Kann ich Fremddaten oder Altdaten übernehmen?* Im Prinzip ja. Studieren Sie Kapitel 5.

*Kann man eigene Routinen in das CockPit einbauen?* Zeilen mit R oder S in die OPT-Datei einbauen (⇒ 0.11.6)

*Welcher Aufruf gehört zu einem CockPit-Menüpunkt?* Bei "eigenen Routinen" sieht man das, wenn man nach dem Anwählen [Tab] statt [Enter] drückt. Ansonsten: **CockPit** mit `acp` statt `cp` starten, dann den Menüpunkt auswählen und starten. Es entsteht eine Batchdatei CCC.BAT. Schauen Sie hinein.

*Wie kann man Menüpunkte aus dem CockPit entfernen?* Die zugehörige Zeile in der Datei UIFCGER entfernen.

*Wie erstellt und testet man Parameterdateien?* Das steht in Kap. 10.1 und 10.3.

*Wie kann man Eingabedaten prüfen lassen?* Es gibt einige eingebaute Prüfmöglichkeiten (⇒ Anh. A.1.2). Mit Parametrierkenntnissen kann man eigene Prüfungen programmieren (⇒ Kap. 10.2.8)

*Kann man Registerauszüge drucken?* Mit dem Programm QRIX (⇒ 0.11.1 Funktion q, auch Kap. 7.9 und 1.4.1).

*Kann man die Registeranzeige beschleunigen?* Taste '%' drücken, oder PRESTO z.B. mit Option `-x10` starten.

*Wie lang darf ein Datenfeld sein?* Bis zu 3000 Zeichen. *Jedes?* Ja.

*Wieviele verschiedene Datenfelder kann es geben?* Über 1000. Mehr über Begrenzungen: Anh. A.1.3)

*Kann jedes Datenfeld mehrfach belegt werden?* Ja. Man kann dies jedoch in der CFG-Datei einschränken (Anh.A).

*Muß man zuerst ausreichend Platz reservieren, wenn eine neue Datenbank entstehen soll?* Nein, die Programme benötigen am Anfang nur das absolute Minimum und erweitern die Dateien ständig automatisch.

*Kann man Änderungen einer Datenbank auf eine andere übertragen?* Siehe Kap. 0.7, letzter Abschnitt.

*Ist es möglich, zwischen zwei oder drei Datenbanken umzuschalten?* Ja, aber man muß richtig starten ⇒ Kap. 1.3.2.

*Welche Satztypen gibt es?* Das ist allein Sache der Parametrierung, vor allem der Indexparameter. Die Software behandelt alle Datensätze gleich.

*Können im Netzwerk zwei oder mehr Updates gleichzeitig laufen?* Ja. Dann aber beim Start von UPDATE nicht die Option `-F` verwenden (⇒ Kap.9.1) und jedem UPDATE mit `-xname` seine eigene Prorokolldatei zuweisen.

*Wie muß man eine Datenbank für UNIX ändern?* Gar nicht, nur an den Parametern sind evtl. Änderungen nötig.

## 0.9.2 Änderungen : Was? Wo? Wie?

Hier finden Sie einen schlagwortartigen, alphabetischen Überblick, was Sie als Anwender alles ändern **können** und wo Sie bei Bedarf eingreifen **müssen**.

Alle Änderungen an Parameterdateien etc. sind mit jedem **Textprogramm** oder **Texteditor** ausführbar. Bei Textsystemen wie WORD oder WordPerfect oder WinWord ist darauf zu achten, die Dateien als "ASCII files" oder im sog. "non-document mode" abzuspeichern. Empfehlenswert sind aber eher solche Editoren, die in die bekannten DOS-Hilfssysteme wie NORTON COMMANDER oder PCTools integriert sind. Der mitgelieferte Texteditor X, der auch vom **CockPit** aus normalerweise aufgerufen wird, ist im Anhang D beschrieben.

**Nehmen wir an**, Sie arbeiten an einer Datenbank namens CAT, die auf der Konfiguration \$A.CFG beruht. (Wird eine andere Konfiguration verwendet, z.B. P.CFG, muß in den Dateitypen jeweils 'A' durch 'P' ersetzt werden, also .PPR statt .APR etc. Für die Bildschirmanzeige verwenden Sie D-1.APR, für die Druckausgabe (Option -q) P-KARTE.APR. Zum Sortieren von Recherche-Ergebnissen gibt es die Parameter S-\*.APR.

- Abfrageliste (Eingabemaske)** : ist enthalten in Konfigurationsdatei **\$A.CFG** (=> Anh.A.1) µo k
- Anzeige der Datensätze am Bildschirm** : Parameterdatei **D-1.APR** (=> Kap.10) µd e
- Benutzeroberfläche** : **CockPit** bietet eine Vielzahl von Funktionen in übersichtlichen Menüs, ermöglicht aber eigene Anpassungen und Erweiterungen (=> 0.11). Möglich sind aber auch eigene Menüprogramme auf der Basis von Batchfiles (=> 12).
- Bildschirmfarben** : einstellbar mit den w-Befehlen in der Konfigurationsdatei **A.CFG** (=> Anh.A.3) µo k
- CockPit** : Anpassungen macht man in der Datei CP.OPT (früher DEFAULT.OPT) (=> 0.11.6) µo v  
Die Texte der Menüs stehen in der Datei UIFCGER µd m
- Codierung** : Zeichencodes für die Druckausgabe lassen sich individuell ändern in der Parameterdatei, die man für die Druckausgabe benutzt (=> 10.2.4), s.a. **Druckertreiber**
- Datenbank-Definition, Datenformat** : siehe **Kategoriesystem** und **Index**
- Druckertreiber** : "Druck" fällt unter "Export", deshalb sind Druckertreiber innerhalb der Exportsprache realisiert (=> 10.). Es sind Dateien des Typs .APT (=> 0.3, 0.11), z.B. P-DSKJET.APT. µo d
- Druckformatierung** : Parameterdatei ändern (z.B. P-NORMAL.APR bei Listendruck, P-KARTE.APR bei Kartendruck), die für die Ausgabe benutzt wird (=> 10.)
- Ergebnismenge** : globale Korrekturen in der Datenbank: F10 (=> 1.5.4, 3.3 Befehl #X)
- Fehlermeldungen** : Die Meldungstexte stehen in den UIF-Dateien (=> 0.3, siehe auch Anh.C.2) µd m
- Funktionstasten-Belegungen** : Grund- und ALT-Belegung sollte man nicht ändern, SHIFT- und CTRL-Belegung außer F1 möglich (=> 1.5, 2.6). (Die UNIX-Version erfordert andersartige Belegungen!)
- Füllzeichen** : Code und Anzahl einstellbar mit Befehl F bzw. f in \$A.CFG (=> Anh. A.1.3)
- Grunddatei** : (Typ .ALG) Bearbeitung mit Programm 3 = ALLEGRO (=> 2.4) µf 2/3
- Hilfstexte** : (H-Dateien) frei bearbeitbar mit jedem ASCII-Texteditor (=> 0.3, 0.6) µd h
- Index** : Art und Gestalt der Eintragungen sind in einer Index-Parameterdatei festgelegt, z.B. CAT.API. µd i  
Bei nachträglichen Änderungen an dieser Datei muß man den Index oder wenigstens diejenigen Teil-Indizes neu generieren (=> Kap.1.2 und Kap.7, Option -fi), deren Einträge betroffen sind. Auch die Kurzanzeige und die Restriktionen werden durch die Index-Parameterdatei bestimmt: es gibt in CAT.API Abschnitte, die mit den Sprungmarken #-0 und #-/ beginnen; dort sind die Strukturen der Kurzanzeige (Datei CAT.STL) und der Restriktionen (CAT.RES) mit Hilfe der Exportsprache definiert.

- Interpunktion** : Diese ist in Form von "Präfix-" und "Postfix-Angaben" in einer Export-Parameterdatei definiert. Z.B. ist das im Normalfall für die Bildschirmanzeige die Datei D-1.APR, für die Druckausgabe z.B. P-KARTE.APR (=> 10.2.0)
- Ja-Nein-Codes** : bei fremdsprachigen Versionen müssen u.U. die Werte für die Antworten "ja" und "nein" bei Ja-Nein-Abfragen geändert werden: Befehl Y in A.CFG (=> Anh.A).
- Kategoriesystem** : In der Datei A.CFG steht das Schema des "konsolidierten Formates"; dieses kann mit Hilfe eines Texteditors geändert werden, oder man erstellt oder verwendet eine andere Konfiguration, z.B. M.CFG für das NW-Format oder \$U.CFG für USMARC (=> Anh. A, B)  
Vorwahl durch Option bzw. Umgebungsvariable -k (=> Kap.12).
- Köpfe der Katalogkarten** : Definitionen stehen in P-KARTE.APR, siehe Befehl **ak** (=> 10.2.1)
- Kurzanzeige** : siehe Index
- Listengestaltung** : Auswahl einer Datei P-\*.APR (evtl. Erstellung einer neuen Datei dieses Typs, => Kap.6) μo 3
- Menüsystem** : siehe **Benutzeroberfläche**
- Nichtsortierzeichen** : Code bzw. Modus: Befehle N bzw. n in A.CFG (=> Anh.A.1.3)
- Ordnung in sortierten Listen** : in den Dateien S-\*.APR werden die Sortierbegriffe durch den Befehl **ak** in Verbindung mit den Sonderkategorien **#u1** und **#u2** gesteuert. So läßt sich die Ordnung präzise festlegen. (=> 6.1, 6.4, 10.2.1, 10.2.6) μo 1
- Pflichtkategorien** : Abfragezeilen mit '?' und '?' in A.CFG (=> Anh.A.2) μo k
- Phrasen** : werden automatisch (in der Datei PHRASE.APH) gespeichert und geladen. .APH-Dateien sind aber auch per Textprogramm bearbeitbar! (=> 0.3, 3.3 Befehl #p) μd p
- Qualität der Daten** : evtl. entscheidend zu verbessern durch Fremddaten-Übernahme, vor allem aus Nationalbibliographien und Literaturdatenbanken (=> 2.0, 5.0)
- Regelwerk** : *allegro* ist nicht auf RAK fixiert. Es duldet jede eigenmächtige Interpretation oder sogar Änderung von Katalogregeln, allerdings nicht mühelos: sofern davon irgendeine Ausgabeform oder Indexeinträge betroffen sind, ändere man die zugehörige Export-Parameterdatei (.APR-Datei) (=> 10.)
- Reports / Sortierte Listen** : per Exportparametrierung (=> 6. und 10.)
- Speicherbedarf** : Mit 640K RAM hat man normalerweise keine Probleme. Wenn doch: Bedarf der geladenen Geräte- und Netzwerktreiber und anderer residenter Programme prüfen. Anpassung des Speicherbedarfs evtl. mit den m-Befehlen in der .CFG-Datei (=> Anh.A.1). Prüfung des aktuellen Zustands mit <Alt+F7> vom Anzeigebildschirm aus. *Tip*: Wenn PRESTO.EXE wegen Speichermangel nicht läuft, statt dessen MENUED.EXE versuchen.
- Sprache der Menüs** : falls entsprechende UIF-Dateien vorhanden (=> 0.6): Vorwahl durch Befehl 'l' in der A.CFG oder Option bzw. Umgebungsvariable -l verwenden (=> 12.). Default ist GER.
- Stoppwörter** : die Tabelle SWL1.APT kann erweitert, gekürzt und beliebig geändert werden (=> 0.3, 10.2.1).
- Tonausgabe abstellen** : bei Programm 5 und 7 (akustische Endemeldung) mit Option -h0
- Übersetzung** : Menü- und Hilfstexte (UIF\*- bzw. H\*-Dateien) können und dürfen übersetzt werden, siehe **Sprache**
- Vorbearbeitung von Fremddaten** : Umfangreiche Möglichkeiten durch eine eigene "Importsprache" (=> 5., 11.)
- Wiederholbarkeit von Kategorien** : grundsätzlich immer gegeben, kein Eingriff nötig (=> 3.1); Alternative: Mehrfacheinträge innerhalb einer Kategorie, mit definiertem Trennzeichen (Empfehlung: "; " oder "¶" ). Begrenzte Wiederholbarkeit einstellbar in der Konfiguration (=> Anh.A.1.2)
- Zeichensatz** : für Bildschirm und Tastatur können bei VGA-Bildschirmen die Zeichensätze softwaremäßig verändert werden, ohne daß dies *allegro* stören würde; für den Drucker: siehe **Codierung**. Der Normzeichensatz DIN 31628/2 wurde für VGA-Bildschirme realisiert. Man schaltet den Zeichensatz ein mit dem Befehl `ostwest.bat`. Mit <Alt>+w erscheint dann jederzeit eine Hilfstabelle mit allen Zeichen.

## 0.10 Installation

Für UNIX gibt es eine eigene Anleitung, die per FTP zugänglich ist, wie alle zu den UNIX-Versionen gehörigen Dateien. Für DOS/Windows: Wenn Sie die CD-ROM der Version 15 erhalten haben, starten Sie das darauf befindliche Programm AUTORUN32 (für Win'95) bzw. START16 (für Windows 3.1). Sie brauchen dann die mitgelieferte, 13stellige Schlüsselzahl, um die zwei Installationsdisketten erstellen zu lassen. Darauf befinden sich dann die eigentlichen Programme und die Installationsroutinen INSTALL.BAT (für Neu-Installation) und INST.BAT (für Update-Installation). Wenn Sie kein CD-Laufwerk haben, können Sie die zwei 3.5" Disketten aus Braunschweig anfordern.

Größere Programmpakete enthalten in der Regel ein Installationsprogramm namens INSTALL. So ist es auch bei *allegro*. Nur muß ein Unterschied gemacht werden zwischen der Neu-Installation und der Update-Installation:

### 1. Neu-Installation

Sie wollen *allegro* auf einem Rechner installieren, auf dem noch keine ältere Version vorhanden ist, oder Sie wollen eine vorhandene Version vollständig und radikal erneuern - ohne Rücksicht auf schon durchgeführte eigene Anpassungen.

#### 1.1 Voraussetzungen

Sie brauchen: einen PC mit 540K freiem RAM (im 640K-Bereich!) und das Betriebssystem MS-DOS in einer Version ab 3.20. Wenn man weniger Platz hat, muß man in der Konfigurationsdatei (\$A.CFG, => Anh.A.1) die Werte in den **m**-Befehlszeilen heruntersetzen.

>>> **Sehr wichtig:** *allegro* benötigt den Gerätetreiber **ANSI.SYS**, damit die Bildschirmsteuerung richtig funktioniert. Dazu muß in Ihrer MS-DOS-Systemdatei CONFIG.SYS eine Zeile dieser Art vorkommen:

```
DEVICE=ANSI.SYS oder z.B. DEVICEHIGH=C:\DOS\ANSI.SYS (je nach Situation)
```

Für Windows'95 gilt dasselbe, nur liegt die Datei ANSI.SYS in C:\WINDOWS\COMMAND.

Bei Windows NT liegt die CONFIG.NT auf C:\WINNT\System32, und in diese Datei muß folgende Zeile:

```
device=%SystemRoot%\System32\ansi.sys
```

Außerdem müssen in der CONFIG die Angaben FILES=20 (oder mehr) und BUFFERS=20 stehen. Stellen Sie dies sicher, bevor Sie installieren. Es kommt vor, daß ANSI.SYS zwar vorhanden ist, aber nicht zur aktuellen Version von MS-DOS paßt oder fehlerhaft ist. Wenn Ihre übrige Software diesen Treiber nicht benutzt, haben Sie davon noch nichts gemerkt. Unter *allegro* würden Sie ein chaotisches Bild mit typischerweise vielen eckigen Klammern sehen, z.B. [**1;32;47m**. Das liegt daran, daß etwas mit ANSI.SYS nicht stimmt. Konsultieren Sie einen Kenner.

*Tip:* Unter Windows NT läuft alles schneller, wenn Sie den Vollbild-Modus benutzen.

Wenn *allegro* von anderen Verzeichnissen aus aufrufbar sein soll, ergänzen Sie C:\ALLEGRO zu dem Befehl PATH in der Systemdatei AUTOEXEC.BAT. Wenn eine Fehlermeldung mit "Umgebungsspeicher.." kommt, ergänzen Sie diese Zeile (dann haben Sie 1000 Byte dafür)

```
SHELL=C:\DOS\COMMAND.COM /P /E:1000
```

#### 1.2 Vorgehen

Nach Einschalten des Rechners legen Sie die Programmdiskette in das Diskettenlaufwerk, schalten auf dieses um und geben den Befehl (ohne Doppelpunkte hinter den Buchstaben!)

```
install d z z.B. install a c für Installation von Disk. A: auf Platte C:
```

wobei *d* der Buchstabe des Diskettenlaufwerks (normalerweise **a**) ist und *z* der Buchstabe des Ziellaufwerks, also der Festplatte, normalerweise **c**.

Die gesamte Installation läuft dann selbsttätig ab. Sie erhalten auf Ihrer Platte:

- ein Unterverzeichnis \ALLEGRO, auf dem alle Programme, Parameterdateien und sonstiges Zubehör stehen (=> 0.3, 0.4)
- ein Datenbankverzeichnis \ALLEGRO\DEMO mit einer kleinen Beispieldatenbank mit dem Namen CAT.

#### Achtung:

Wenn Sie *allegro* nicht auf dem Verzeichnis C:\ALLEGRO installiert haben, ist es notwendig, in der Datei CP.OPT die Einstellung "Programmverzeichnis" zu ändern. Diese Datei ist kommentiert (=> 0.11.6), und Sie sehen sofort, wo man eingreifen muß. Auch Ihr Datenbankpfad ist dort einzutragen, und zwar wo

```
d C:\ALLEGRO\DEMO steht.
```

In der Datei \$A.CFG (=> Anh.A.4) nehmen Sie Änderungen an den Speichergrößen, Bildschirmfarben etc. vor.

### 1.3 Einarbeitung

Nach abgeschlossener Installation wird automatisch das *CockPit* gestartet. Es ist dann so eingestellt, daß Sie mit einem weiteren Tastendruck und <Enter> die Beispieldatenbank sofort benutzen können.

Nach dem Sammeln erster Eindrücke gehen Sie am besten über zu einer ausführlichen Lektüre der Kapitel 0 und 1, wobei Sie z.B. die in 1.4 und 1.5 beschriebenen Such- und Zugriffsfunktionen alle an der Beispieldatenbank ausprobieren können. Ansonsten gehen Sie am besten nach dem "Leitfaden für die Einarbeitung" vor.

Das *CockPit* wird neu gestartet mit dem Befehl **cp** vom Programmverzeichnis aus.

Zum Anlegen einer eigenen Datenbank nach dem Muster der Beispieldatenbank wählen Sie den Menüpunkt "Datenbank anlegen" im *CockPit*-Hauptmenü.

Um eine ganz **neue Datenbank** mit eigener Struktur zu konfigurieren, benutzen Sie das Hilfsprogramm PRONTO.BAT (Dokumentation in der Datei PRONTO.TXT).

### 1.4 Einzelplatzbetrieb

**Entweder** man ruft die Programme PRESTO und UPDATE immer mit Option **-s** (für "single user") auf. Wenn man mit *CockPit* arbeitet, genügt es, in der Startdatei CP.BAT die Option **-s** an den Aufruf des Programms ACP anzuhängen. Bei Lieferung ist dies so eingestellt; es muß ein großes S sein.

**Oder** man startet vor jeder *allegro*-Benutzung das Programm SHARE. Dann wird so getan, als ob man im Mehrplatzbetrieb arbeitet. Das ist jedoch unnötig und macht die Programme ein wenig langsamer.

Ab V15.0 kann die Option **-s** weggelassen werden. Die Programme merken dann selbst, daß sie nicht im Mehrplatzmodus arbeiten können. Der Start dauert dann nur eine Sekunde länger.

### 1.5 Mehrplatzbetrieb (Lokales PC-Netz)

In der Regel genügen diese Vorkehrungen:

- alle Dateien der Datenbank für gleichzeitige Benutzung freigeben (bei Novell: mit Befehl FLAG \*.\* s)
- Einzelplatz-Option **-s** aus CP.BAT herausnehmen (in der Zeile, wo ACP aufgerufen wird)
- Aufrufe von PRESTO und UPDATE ohne Option **-s** durchführen (Batchdateien daraufhin prüfen)

Absicherung des Mehrplatzbetriebs: Auf der Platte, wo die User ihre Heimverzeichnisse haben, z.B. F:, richte man ein Verzeichnis \ALLEGRO\OPT ein, wo nur der Systemverwalter Schreibrecht hat. Dort speichere man für jeden User xyz eine Datei XYZ.OPT. Dann wird *CockPit* automatisch diese und keine andere statt CP.OPT nehmen, und der User kann sie nicht ändern.

## 2. Update-Installation

Wer bereits eine funktionierende *allegro*-Installation hat, braucht nur die Diskette 2. Man erstellt sie wie oben beschrieben von der CD, falls sie nicht als Diskette vorliegt.

Grundsätzlich brauchen Sie keine Veränderungen an Ihren Datenbanken vorzunehmen, wenn Sie eine neue Programmversion installieren, denn die Programme sind immer **abwärtskompatibel**. Es kann nur in seltenen Fällen vorkommen, daß Kleinigkeiten der Bildschirmanzeige oder der Registereinträge sich verändern.

Man schaltet um auf das Diskettenlaufwerk, meistens A:, und gibt

```
inst d z      (d = Diskettenlaufwerk, z Plattenlaufwerk, ohne Doppelpunkte)
```

also z.B.

```
inst a c      für Installation von A: nach C:
```

Oder **viel bequemer:** vom *CockPit* aus den Menüpunkt **mu n u** (Update-Installation) aktivieren.

Die alten Programme bleiben erhalten, werden aber umkopiert auf ein Verzeichnis V15C. Desgleichen die Konfigurations-, Parameter- und Batchdateien. Diese Dateien, z.B. D-1.APR und \$.CFG, sind oftmals vom Anwender schon verändert worden und sollen daher nicht einfach "übergebügelt" werden. Wenn nach der Installation also die Titelanzeige nicht mehr so aussieht wie vorher, holen Sie sich die alte D-1.APR zurück. Das geht am einfachsten über den Menüpunkt "X : EXTERN". Dieser startet die Stapeldatei EXTERN.BAT, die Ihnen alle Dateien und Programme zur Auswahl anzeigt, die zurückgeholt werden können. Die Originale bleiben natürlich auf der Diskette erhalten, d.h. zur Not können Sie jede Originaldatei auch von dort erneut kopieren. (Z.B. mit dem Befehl `lharc x a:par15 d-1.apr`)

**Vorsicht:** Wer Veränderungen an den **UIF-** und **H-Dateien** vorgenommen hat, sollte diese vorher **in Sicherheit bringen** und anschließend Vergleiche mit den mitgelieferten Dateien vornehmen, insbesondere bei UIF0... und UIF1... In den deutschen Versionen dieser Dateien sind Änderungen kommentiert (jeweils mit \$\$ und Datum).

## 0.11 CockPit

Der Name *CockPit* soll andeuten, daß es sich um so etwas wie eine Pilotenkanzel handelt, in der man mit Hilfe geeigneter Instrumente alles überwachen und steuern kann. Allerdings wird dadurch dieses Systemhandbuch keineswegs überflüssig. *CockPit* erleichtert die Routinearbeit, indem es eine Reihe von Vorgängen, die man immer wieder braucht, auf wenige Knopfdrücke reduziert. Es ist ferner vom Anwender erweiterbar: es ermöglicht, eigene Prozeduren, die man als "Batchfiles" geschrieben hat, mit einzubinden. Es bietet aber wenig Unterstützung für Gestaltungsaufgaben: Datenbankdefinition, Export- und Importprogrammierung vereinfacht es kaum. Auf diesen Gebieten ist eine solide Systemkenntnis auf absehbare Zeit unerlässlich, und dafür gibt es dieses Systemhandbuch.

### Startbildschirm

Nach dem Start mit dem Befehl **cp** erhalten Sie diesen Bildschirm:

```

Funktionen      Routinen      Makros      Dateien      Optionen      HELP      EXIT
CockPit
Sie möchten eine Datenbank
benutzen
anlegen
sichern
wiederherstellen
löschen
organisieren
neue Version installieren
Volltextsuche/Listen
Zurücksetzen der Optionen
X : EXTERN
eigene Routinen:
1 : Einfache Anwendungen
2 : OPAC
3 : MENUED
4 : ALFA
5 : Download
6 : Import-Aktionen

Datenbank mit den unten angegebenen Optionen starten; Änderungen: ALT+o
C: \ALLEGRO\DEMO
Datenbank = CAT           Konfiguration = A   Progr.Verz. =      C: \ALLEGRO
Einzelplatzbetrieb       Sprache = GER       Laufwerk = C   V15

```

Die Angaben auf den zwei untersten Zeilen stammen aus der Vorgabendatei CP.OPT - dazu unten noch mehr. Der Leuchtbalken steht zu Anfang auf der wichtigsten Funktion: **benutzen**. Somit reicht ein weiterer Druck auf <Enter>, um in die vorgegebene Datenbank per Schnellzugriff sofort einzusteigen.

Dieses Menü funktioniert so, wie es von vielen Standardprogrammen bekannt ist:

- Mit den Cursorstasten bewegt man den Leuchtbalken auf- und abwärts, mit <Enter> aktiviert man den gewählten Menüpunkt. Oder: Schnellwahl durch Betätigen des Anfangsbuchstabens.
- Zum Umschalten auf ein anderes Pulldown-Menü fährt man mit <Cursor links> und <Cursor rechts> oben an der Menüzeile entlang. Oder: <Alt> + Anfangsbuchstabe, also z.B. <Alt>+d für das Menü "Dateien". Als symbolische Kurzbezeichnung verwendet dieses Handbuch das Zeichen 'μ' für "Menü". Z.B. bedeutet die Angabe μ**r** o **i**, daß man die Tasten <Alt>+r o i hintereinander betätigen soll.
- Aus Untermenüs entkommt man stets ohne Folgen mit <Esc>
- Über "EXIT" oder <Alt>+x verläßt man das *CockPit*.
- *Besonderheit:* Die Funktion **z** auf dem Routinenmenü stellt die Grundoptionen auf den Fußzeilen auf die Werte, die in der Vorgabendatei stehen.

## 0.11.0 Die Hauptmenüs

Zunächst ein schlagwortartiger Überblick, die nachfolgenden Seiten erklären jedes Menü im Detail.

<b>Funktionen</b>	Aufruf der einzelnen <i>allegro</i> -Programme mit der Möglichkeit, über programmspezifische Menüs alle Aufruf-Optionen (=> Kap.12) einzustellen
<b>Routinen</b>	oberer Teil: Routine-Verwaltungsaktionen an der ausgewählten Datenbank unterer Teil: eigene Prozeduren, definiert mit den Befehlen R und S in CP.OPT
<b>Makros</b>	vorhandene Batchfiles können gestartet oder bearbeitet werden.
<b>Dateien</b>	alle Parameterdatei-Typen und andere Hilfsdateien des Systems können aufgelistet, betrachtet und bearbeitet werden. Dazu wird der Editor X benutzt (oder Ihr eigener, siehe oben). Es erscheint jeweils die Liste der vorhandenen Parameterdateien des betreffenden Typs, und zwar zuerst die auf dem Datenverzeichnis; mit dem Menüpunkt "umschalten" bekommt man die entsprechenden Dateien des Programmverzeichnisses. (Dazu muß aber in der CP.BAT der Aufruf von ACP mit der Option -a2 erfolgen.)
<b>Optionen</b>	Hier sind die auf den zwei Fußzeilen (Statuszeilen) stehenden Angaben veränderbar: <b>Datenverzeichnis</b> (bei Anwahl eines anderen Verzeichnisses stellen sich die Statuszeilen automatisch auf die dort liegende Datenbank ein) <b>Konfiguration</b> (hier auch Bearbeitung der .CFG-Dateien) <b>Sprache</b> (GER und ENG werden mitgeliefert) <b>Laufwerke</b> (nach Umschaltung: "Datenverzeichnis" anwählen) <b>Vorgaben</b> (Bearbeitung der CP.OPT-Datei - mit sofortiger Wirkung) <b>Ressourcen</b> (Anzeige wesentlicher Speichergrößen) <b>Druck-Optionen</b> (Einstellmöglichkeiten für Listenproduktionen (Auswahl bestimmter Parameterdateien für Sortierfolge, Ausführlichkeit, Listengestaltung und Druckertreiber.

## Diese Dateien gehören zum *CockPit*

<b>CP.BAT</b>	hiermit startet man das <i>CockPit</i> . Es ruft dann selbsttätig das eigentliche Programm auf: (und zwar im Einzelplatzmodus; für Mehrplatzbetrieb nehme man die Option <b>-S</b> aus CP.BAT heraus)
<b>ACP.EXE</b>	wird folglich nur indirekt, eben durch CP.BAT, aufgerufen. Dieser Umweg ist notwendig, damit das <i>CockPit</i> nach jedem Vorgang erneut die Kontrolle übernehmen kann.
<b>CP.OPT</b>	diese Datei enthält die Voreinstellungen, mit denen <i>CockPit</i> dann arbeitet. Alle Einstellungen lassen sich ändern: CP.OPT ist eine Textdatei und kann, wie alle Parameterdateien, mit jedem Texteditor geändert werden. Dafür ist sogar ein eigener Menüpunkt vorgesehen: der Punkt <b>v</b> im Menü "Optionen". Die Datei ist ausführlich kommentiert und in 0.11.6 abgedruckt.
<b>UIFCGER</b> <b>UIFCENG</b>	Das deutsche und das englische "User Interface File" (=> 0.3). Wie bei den anderen Programmen stehen darin die Menütexte, Fragen und Meldungen, die am Bildschirm erscheinen. Auch bei diesem Programm kann man also Änderungen und Übersetzungen durchführen. <b>Unerwünschte Menüpunkte</b> löscht man heraus, dann erscheinen sie gar nicht! Sehr zu achten ist auf die Längen der Texte, die in vielen Fällen keine großen Änderungen vertragen - die Resultate sollte man immer prüfen. Soll <i>CockPit</i> auf Englisch laufen: in der CP.BAT ergänzt man die Zeile <code>acp ...</code> durch <code>-l eng</code> .

### Mitgeliefert werden ferner

(jedoch gehören sie nicht eigentlich zum *CockPit*, sondern sind durch andere Programme ersetzbar)

<b>X.EXE</b>	Ein Texteditor, der für die Bearbeitung der Parameter- und Konfigurationsdateien benutzt wird; kann durch einen anderen Editor ersetzt werden (dann Befehl E in CP.OPT ändern) Anleitung: siehe Anhang D
<b>V.COM</b>	Ein "View"-Programm zum Betrachten von Textdateien; kann gleichfalls durch etwas entsprechendes anderes ersetzt werden (dann Befehl V in CP.OPT ändern)

### 0.11.1 µf : Das Menü "Funktionen"

Es erlaubt Aufrufe der zum Kernsystem gehörenden Programme (=> Kap.0.4), wobei alle Optionen per Menü einstellbar sind. Dieses Menü kann man nur mit einiger Kenntnis der Grundlagen des *allegro*-Systems benutzen.

Nehmen wir als Beispiel den Aufruf des IMPORT-Programms. Ein typischer Bildschirm sähe so aus:

```

Funktionen  Routinen  Makros  Dateien  Optionen  HELP  EXIT

HELP
1 : PRESTO
2 : OFFLINE
4 : SRCH+EX
5 : IMPORT
7 : INDEX
q : QRIX
8 : SORT
9 : UPDATE

k : Konfiguration:      A
c : Laufwerke:         A
d : Datenverzeichnis:  A:\FREMD
s : Selektierbegriff:  0
manuelle Unterbrechung  1
visuelle Überwachung   1
i : Import Parameter:  DBDISK
e : Export Parameter:  I-1
  Export Datei :       katalog\ergeb.n.alg
e : Export Parameter:
  Export Datei :
e : Export Parameter:
  Export Datei :

GO (START)

in welche Datei soll exportiert werden?
C:\ALLEGRO\DEMO
Datenbank = CAT      Konfiguration = A    Progr.Verz. = C:\ALLEGRO
Einzelplatzbetrieb   Sprache = GER      Laufwerk = C  V15
  
```

Die einzelnen Einstellungen nimmt man so vor:

- man bewegt den Leuchtbalken im linken Teil des Untermenüs "Optionen",
- man drückt <Enter>, wenn man den betreffenden Punkt bearbeiten will,
- es erscheint eine Frage oder eine Eingabeaufforderung oder ein weiteres Untermenü, aus dem man etwas auszuwählen hat - so z.B. bei dem Punkt "Import-Parameter".
- Ist alles richtig eingestellt, bewegt man den Leuchtbalken auf **GO (START)** und drückt nochmals <Enter>.

Eine Besonderheit bei Funktion 9 = UPDATE: man stellt vorher über **µo d** als Datenverzeichnis dasjenige ein, wo die einzuspeisende .ALG- oder .LOG-Datei liegt, und dann im Update-Menü dasjenige, auf dem die zu aktualisierende Datenbank liegt.

*CockPit* produziert nun eine Batchdatei namens CCC.BAT. Darin steht der folgende Aufruf:

```
import -f5 -dA:\FREMD -iDBDISK -kA -lGER -e I-1/katalog\ergeb.n.alg -s0 -ml -v1
```

Anschließend wird CCC.BAT ausgeführt, also IMPORT mit allen erforderlichen Angaben gestartet. Nur die umzuwandelnde Datei (die auf A:\FREMD stehen muß) ist anschließend noch auszuwählen (=> Kap.0.5). Nach Ablauf des Importvorgangs sorgt CCC.BAT dafür, daß *CockPit* wieder aktiviert wird.

Man könnte den IMPORT-Aufruf genauso mit der Hand geben oder IMPORT ohne Optionen starten (dann fragt es selbst die nötigen Angaben ab - man muß aber dann wissen, was gemeint ist). Der Vorteil vom *CockPit* liegt darin, daß die Optionen für den Programmabruf überschaubar und kommentiert zur Auswahl und kontrollierten Eingabe und Bearbeitung angeboten werden. (Das hier gezeigte Beispiel wird in Kap.5.1 noch näher erläutert.)

## 0.11.2 µr : Das Menü "Routinen"

Dieses erscheint als erstes, weil es das wichtigste Menü ist, und der Leuchtbalken steht auf "benutzen". Das Menü enthält die Routineprozeduren (daher heißt es so), die an einer Datenbank oft oder gelegentlich auszuführen sind.

Der **obere Teil** bietet fest vorgegebene Prozeduren, die man unmittelbar auf jede Datenbank anwenden kann.

Der **untere Teil** ("eigene Routinen") ist vom Benutzer frei variierbar. Die Menüpunkte (bis zu 6 Hauptpunkte mit je bis zu 10 Unterpunkten) sind in die Datei CP.OPT einzutragen (=> 0.11.6). Darunter können auch Programmaufrufe sein, die mit *allegro* nichts zu tun haben. Wer konsequent ist, kann seine gesamte Programmumgebung unter dem *CockPit* zusammenfassen und alles vom Menü "Routinen" aus starten, denn dieses erscheint immer wieder automatisch nach Abschluß eines Programmlaufs, nimmt aber dem aufgerufenen Programm keinen Platz weg.

Die Standardpunkte des Routinen-Menüs sind weitgehend selbsterklärend, deshalb hier nur eine Kurzdarstellung: (bis auf "anlegen" und "neue Version installieren" bezieht sich alles auf die angewählte Datenbank, die aus den Fußzeilen ersichtlich ist)

<b>benutzen</b>	PRESTO aufrufen, um an der Datenbank zu arbeiten (=>1.3)
<b>anlegen</b>	völlig neue Datenbank anlegen (=> 1.1)
<b>sichern</b>	BACKUP-Kopie machen / .LOG-Datei kopieren (=> 0.7)
<b>wiederherstellen</b>	Playback (RESTORE, .LOG-Datei einspielen, => 0.7)
<b>löschen</b>	zugehörige Dateien (.ALD, .TBL, .STL, .ADX, .LOG) beseitigen
<b>organisieren</b>	... siehe unten ...
<b>neue Version installieren</b>	wenn eine neue <i>allegro</i> -Diskette geliefert wird (=> 0.10)
<b>Volltextsuche / Listen</b>	Produktion von sortierten Listen (siehe auch 0.11.5)
<b>Zurücksetzen der Optionen</b>	Statuszeilen werden wieder auf die Werte der Datei CP.OPT gesetzt.
<b>X : EXTERN</b>	startet sofort und ohne Nachfrage die Datei EXTERN.BAT

Für das Sichern werden die MS-DOS-Befehle BACKUP und RESTORE verwendet. Wenn man andere Techniken anwenden will oder DOS 6.x verwendet, muß man eigene Batchfiles namens **PBACKUP.BAT** und **PRESTORE.BAT** schreiben; diese können dann vom *CockPit* aus den Routinen "Sichern" und "Wiederherstellen" alternativ aufgerufen werden.

Von größter Bedeutung für den Umgang mit einer *allegro*-Datenbank ist das Untermenü "**Organisieren**". Hier sind Prozeduren zusammengefaßt, die zwar fast alle durch Programmaufrufe mit den richtigen Optionen auch bisher schon ausführbar waren. Jetzt aber sind sie auf einen Knopfdruck reduziert. So sieht es aus, wenn man µr o wählt:

```

Funktionen      Routinen      Makros      Dateien      Optionen      HELP      EXIT
-----
CockPit
Sie möchten eine Datenbank

benutzen
----- Organisieren -----
i = Index wiederherstellen
c = Index kompaktieren
t = Satztabelle erneuern
k = Kurzanzeige erneuern
x = Restriktionsdatei erneuern
7 = Datenbank völlig neu aufbauen
n = nummerntreue Erneuerung der Datenbank
r = einzelne Datenbankdatei entlüften
R = ganze Datenbank entlüften
u = ungenutzten Platz prüfen
s = Datenbank sperren
e =                freigegeben
a = Altdateien (.c1D) löschen

Register erneuern (Datei .CDX)
C:\ALLEGRO\DEMO
Datenbank = CAT      Konfiguration = A      Progr.Verz. =      C:\ALLEGRO
Einzelplatzbetrieb      Sprache = GER      Laufwerk = C V15
    
```

Im Kapitel 7 erfahren Sie, welche Bewandtnis es ganz genau mit den einzelnen Punkten dieses Menüs hat. Für den "Normalgebrauch" reicht eine Kurzbeschreibung, die wir auf der folgenden Seite geben:

## Datenbank organisieren

Die meisten dieser Vorgänge benötigen einigen Platz für Zwischenspeicherungen. Wenn es auf Ihrer Platte eng geworden ist, beachten Sie die angegebenen Werte. Im Mehrplatzbetrieb: Funktionen außer **u**, **s** und **e** nur aktivieren, wenn die Datenbank nicht benutzt wird.

- i** **Index erneuern** (schließt **k** ein; wenn .TBL nicht existiert, auch **t**) : Dieser Punkt wird benötigt, wenn
    - a) der Index nicht mehr funktioniert (er erscheint einfach nicht, meist mit Fehlermeldung), oder
    - b) eine Änderung an der Index-Parameterdatei ausgeführt wurde (**µd i**). Eine geänderte Indexdefinition bewirkt nicht automatisch eine Neugenerierung des Index, sondern dafür gibt es diesen Menüpunkt.
 Vor dem Start sollte etwa doppelt so viel Platz frei sein, wie der aktuelle Index beansprucht. In der Regel wird weniger gebraucht, außer wenn man starke Erweiterungen an der Index-Parameterdatei vornimmt.
  - c** **Index kompaktieren** : in größeren Abständen kann es sein, daß der Index sich vergrößert hat, da er durch viele Löschungen oder andere Aktionen intern Leerräume enthält und dadurch unnötig viel Platz beansprucht und evtl. auch etwas langsamer wird. Wenn man sich die Größe unmittelbar nach dem Aufbau notiert und dann gelegentlich vergleicht, wird man einen Eindruck gewinnen, wann diese Funktion sich lohnen könnte. Die Ausführung geht wesentlich schneller als die Funktion **i**, bei der die Schlüssel alle neu berechnet werden müssen! Beim Start sollte in etwa so viel Platz frei sein, wie der aktuelle Index selbst einnimmt. Meistens reicht zwar weniger, doch ist das nicht vorhersehbar. Funktion **c** bringt nichts, wenn man gerade vorher **i**, **7** oder **n** gemacht hat.
  - t** **.TBL-Datei** : Die .TBL-Datei wird neu produziert. Das ist höchstens dann notwendig, wenn sie unbeabsichtigt gelöscht wurde, oder wenn die Fehlermeldung "kein Zugriff unter dieser Satznummer" kommt, denn diese deutet darauf hin, daß die .TBL-Datei möglicherweise nicht in Ordnung ist. Geht sehr schnell.
  - k** **Kurztitelregister erneuern** : Nur die .STL-Datei (das Register 0) wird erneuert oder erstmals angelegt. Dies geht sehr schnell, ca. 10 bis 30 Datensätze je Sekunde. Es braucht 72 Bytes je Datensatz.
  - x** **Restriktionsdatei erneuern** : Nur die .RES-Datei wird erneuert oder erstmals angelegt (falls eine solche in den Indexparametern definiert ist, => 10.2.9). Geht genauso schnell wie .STL.
  - 7** **Datenbank völlig neu aufbauen** : Index, .TBL- und .STL-Dateien werden gelöscht, die .ALD-Dateien hintereinander eingelesen und auf neue Dateien geschrieben, die Schlüssel berechnet, die gelöschten Dateien werden neu erstellt. Gelöschte Aufnahmen verschwinden völlig, alle Aufnahmen werden neu durchnummeriert, die Dateinummern bleiben aber erhalten, wenn man mehrere .ALD-Dateien hat. Anschließend sind die alten .ALD-Dateien noch unter Dateityp .c1D vorhanden. Man sollte die alten Dateien dann, wenn alles korrekt gelaufen ist, unbedingt löschen, also nachdem man gesehen hat, daß der Index wieder funktioniert. Dazu gibt es Funktion 'a'. Speicherplatz: es muß vorher doppelt soviel Platz vorhanden sein, wie die Datenbank momentan einnimmt.
  - n** **Nummerntreue Erneuerung** : wie Funktion 7, aber die Satznummern bleiben erhalten. Zwangsläufig gibt es dann unbesetzte Nummern (unter denen vorher gelöschte Sätze standen). Unter dem Sonderschlüssel / [ 0 ] im Index 1 sind diese Nummern registriert und werden im weiteren Verlauf wiederverwendet. Die Funktionen **7** und **n** brauchen relativ viel Zeit. Sie schließen **i** und **R** mit ein.
  - r** **R Entlüftung** einer einzelnen .ALD-Datei bzw. der ganzen Datenbank. Bei **r** kann man anschließend auswählen, welche der .ALD-Dateien entlüftet werden soll, wenn man mehrere hat, ansonsten wird die gesamte Datenbank entlüftet. Dabei werden die .ALD-Dateien gelesen und neu geschrieben, jede Aufnahme erhält die Anzahl Füllzeichen, die in der .CFG-Datei festgelegt ist (=> Anh.A.1, evtl. vorher ändern!). Satznummern bleiben erhalten. Für gelöschte Aufnahmen und die Ausgangsdateien gilt dasselbe wie bei Funktion **n**. Das Entlüften braucht nur einen Bruchteil der Zeit des Neu-Indexierens. Vorher muß etwa genauso viel Platz frei sein, wie die zu entlüftenden Dateien beanspruchen. In der Regel werden die Dateien kleiner - das ist ja auch der Sinn. Wie groß die Platzersparnis etwa sein wird, erfährt man mit Funktion **u**:
- Empfehlung:* vor **7**, **i**, **n**, **r**, **R** prüfen, ob alle .ALD-Dateien zur Datenbank gehören und keine .ALG vorhanden ist.
- u** **ungenutzten Platz prüfen** : diese Funktion stellt (mit Hilfe des Programms QRIX, => Kap.7.5) die Gesamtlänge und -anzahl der in der Datenbank vorhandenen gelöschten Sätze fest, aufgeschlüsselt nach Dateien. Das geht sehr schnell und bietet eine Entscheidungshilfe, ob man eine Entlüftung durchführen sollte.
  - s** **Datenbank sperren** : diese zwei Funktionen sind im Mehrplatzbetrieb wichtig, denn
  - e** **Datenbank entsperren** : eine gesperrte Datenbank erlaubt nur den Lesezugriff, d.h. OPAC-Benutzer merken nichts. Während der Sperre kann man gefahrlos z.B. eine Sicherungskopie, Entlüftung oder Index-Kompaktierung machen. Wenn während der Sperrung jemand einen Schreibvorgang versucht, wird durch die Meldung *bitte warten* der Sperrzustand angezeigt. Ansonsten erscheint diese Meldung höchstens einmal für wenige Sekunden, wenn unmittelbar vorher jemand anders eine Speicherung veranlaßt hat. Wenn ausnahmsweise beim Einzelplatz endlos **bitte warten** kommt und kein Speichern möglich ist, läßt sich das mit der Funktion **e** beheben. Der Fall ist allerdings selten und auf vorangegangene Fehler zurückzuführen.
  - a** **Alte Dateien löschen** : Wenn man eine der Aktionen **7**, **n**, **r**, **R** gemacht hat, bleiben am Ende die alten Datendateien unter dem Dateityp .c1D zurück. Diese müssen gelöscht werden, bevor man erneut eine der genannten Aktionen machen kann; *CockPit* macht selber darauf aufmerksam. Dafür gibt es diesen Menüpunkt.

### 0.11.3 µm : Das Menü "Makros"

Mit "Makros" sind hier Batchdateien gemeint.

Das Menü besteht normalerweise aus nur zwei Punkten:

**Produktion starten** : Die Liste der vorhandenen Stapeldateien (= Batchdateien, Dateityp .BAT) wird aufgeblättert. Das Listenmenü ermöglicht ein Umschalten zwischen Daten- und Programmverzeichnis. Man setzt den Leuchtbalken auf den gewünschten Namen und startet den Vorgang mit <Enter><Enter>. Ohne Probleme kann man auch solche Batchdateien starten, die mit *allegro* nichts zu tun haben. Vor dem Start wird *CockPit* völlig aus dem Arbeitsspeicher entfernt, es nimmt also dem aufgerufenen Programm keinen Platz weg.

**Makros bearbeiten** : Dieselbe Liste wird aufgeblättert, aber jetzt kann man sich die einzelnen Stapeldateien zwecks Durchsicht oder Bearbeitung vornehmen, auch neue produzieren. Die Möglichkeiten sind dieselben wie sie im Menü "Dateien" für die Parameterdateien angeboten werden.

Darunter können noch weitere "eigene Routinen" erscheinen, wenn der Platz im Hauptmenü "Routinen" dafür nicht reicht (siehe dort).

In der Praxis braucht man nur die wichtigsten Vorgänge an das Menü "Routinen" anzubinden (=> 0.11.6). Alle anderen Stapeldateien, die man irgendwann erstellt hat, kann man ohne sonstige Vorkehrungen über das Menü "Makros" aufrufen. Ein Problem ist höchstens die

**Parameterübergabe** : *CockPit* ermöglicht auch eine Übergabe eigener, beliebiger Parameter an aufgerufene Stapeldateien. Es legt aber außerdem vor deren Start die wichtigsten Angaben im DOS-Environment ab, und zwar unter den Namen

-B	Name der Datenbank	z.B. -B=CAT
-D	Name des Datenverzeichnisses	z.B. -D=C:\ALLEGRO\KATALOG
-K	Konfiguration	z.B. -K=PICA
-K1	Konfiguration, 1. Buchstabe	z.B. -K1=P
-L	Sprache	z.B. -L=GER oder -L=ENG
-P	Programmpfad	z.B. -P=C:\ALLEGRO

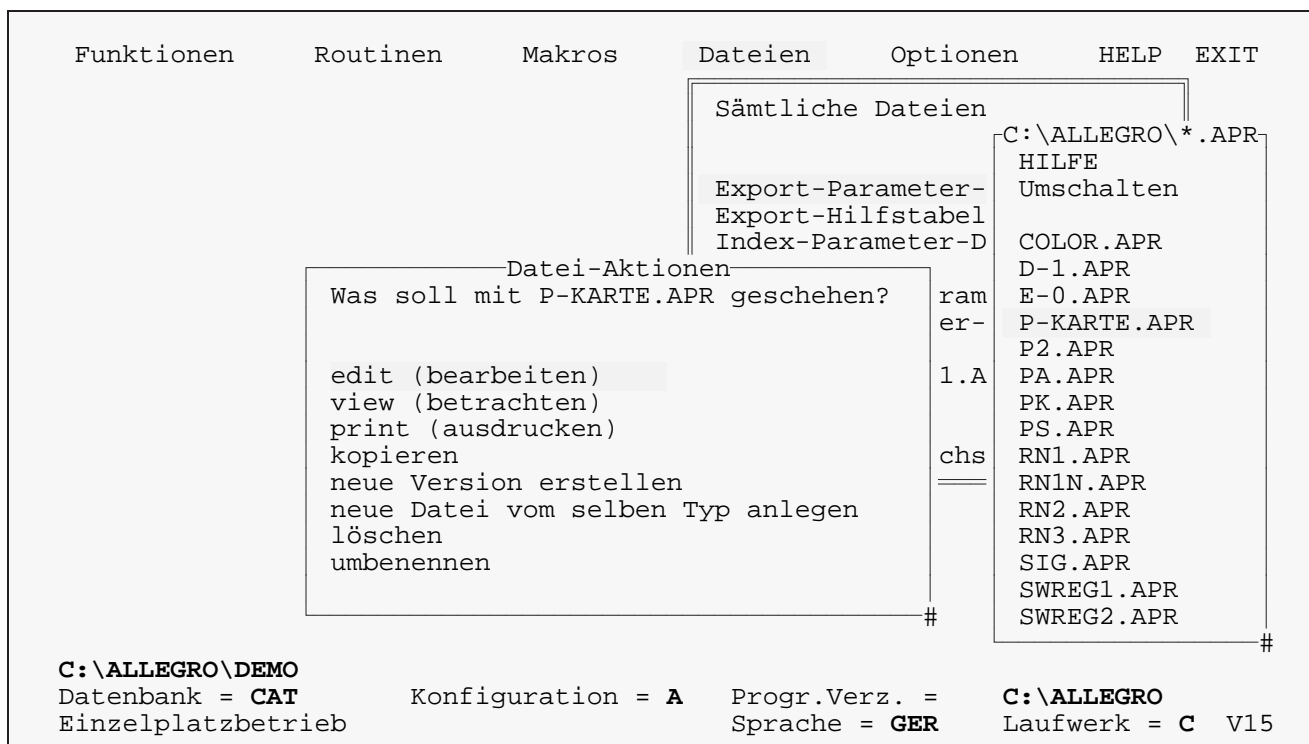
Um eigene Parameter zu übergeben, braucht man nur in der **R** oder **S**-Zeile, die für einen Batchjob angelegt wurde (=> 0.11.6), an den Namen der .BAT-Datei hinten ein Fragezeichen und einen Fragetext anzuhängen. Vor dem Start dieser Batchdatei wird dann die Frage gestellt und die Antwort wird dem Batch als Parameter %1 übergeben.

### 0.11.4 µd : Das Menü "Dateien"

Um die Übersicht in den vielen Dateitypen zu gewinnen, die zur *allegro*-Umgebung gehören, ist dieses Menü besonders hilfreich. Alle Typen lassen sich getrennt und alphabetisch geordnet sichten und dann bearbeiten, umbenennen, kopieren oder löschen. Das alles wäre sonst nur über DOS mit den entsprechenden Kenntnissen und gutem Gedächtnis zu machen, oder aber mit Fremdhilfsmitteln wie Norton-Commander, der jedoch keine systemspezifischen Hilfen anbieten kann. Jeder Unterpunkt dieses Menüs bringt eine alphabetische Liste auf den Schirm, die eben genau die entsprechenden Dateien enthält. Das Blättern in diesen Listen bietet einigen Komfort:

- Wenn die Liste zu lang für den Bildschirm ist, blättert man mit den Tasten <Bild↓> und <Bild↑> sowie <Pos1> und <Ende>, wie man es erwartet, auf- und abwärts.
- Sucht man einen ganz speziellen Namen: nur die ersten paar Buchstaben eintippen. Der Balkencursor springt bei jedem Buchstaben sofort auf den ersten dazu passenden Namen. Bei Verschreiben korrigiert man mit der Rücksetztaste. Jederzeit kann man aus diesem Blättermodus wieder mit den Cursortasten anderswo hinspringen.
- Der Menüpunkt "Umschalten", oberhalb der Liste stehend, ermöglicht ein Hin- und Zurückschalten zwischen den auf den unteren Zeilen angegebenen Verzeichnissen: Datenbank- und Programmverzeichnis sowie dem Startverzeichnis. Dazu muß man wissen: die *allegro*-Programme suchen eine angeforderte Parameterdatei immer zuerst auf dem **Datenbank**verzeichnis. Wenn sie dort nicht ist, wird automatisch auf dem **Start**verzeichnis gesucht, dann auf dem **Programm**verzeichnis. Die hier gebotene Umschaltmöglichkeit ist deshalb eine besonders wichtige, dem Konzept entsprechende Hilfe. **Zu empfehlen** ist, die datenbankspezifischen Parameterdateien und die Konfiguration stets auf dem zugehörigen Datenverzeichnis anzuordnen, also der Datenbank direkt anzugliedern. Parameterdateien von allgemeiner Bedeutung, die man für mehrere Datenbanken nutzen kann (z.B. I-1.APR und E-1.APR), sind dagegen auf dem Programmverzeichnis besser aufgehoben, nutzerspezifische auf dem Startverzeichnis des Nutzers.

Wenn wir nun auf dem Menü "Dateien" als Beispiel den Punkt "Export-Parameter" anwählen, auf der dann erscheinenden Liste die Datei P-KARTE.APR anfahren und dann <Enter> drücken, sieht das Bild so aus:



Ein weiterer Druck auf <Enter> würde den (evtl. vom Benutzer gewählten, => 0.11.6) Texteditor aufrufen und diesem die Bearbeitung von P-KARTE.APR übergeben.

Bei Wahl von "sämtliche Dateien" erscheinen nur dann wirklich alle, wenn mit **-a3** gestartet wurde.

Ansonsten hat das Untermenü "Datei-Aktionen" die folgenden Funktionen:

- view (betrachten)** hiermit kann man sich die angewählte Datei am Bildschirm anzeigen lassen. Benutzt wird dazu ein "View-Programm" namens V.COM. Der Anwender kann es durch ein anderes ersetzen, evtl. auch durch einen alternativen Editor (=> 0.11.6)
- print (ausdrucken)** *CockPit* gibt den Befehl `print P-KARTE.APR`. Statt PRINT kann ebenfalls in CP.OPT ein anderes Hardcopy-Programm (oder auch irgendetwas völlig anderes) eingesetzt werden.
- kopieren** ein neuer Name wird vom Benutzer abgefragt; dieser Name kann eine Pfadangabe enthalten. *CockPit* kopiert P-KARTE.APR dann auf diesen Namen. Wenn es den aber schon gibt, muß man bestätigen, ob die betreffende Datei überschrieben werden soll.
- neue Version erstellen:** Auch hier muß zunächst ein Name eingegeben werden, dann wird der Texteditor aufgerufen und die angewählte Datei zur Bearbeitung vorgelegt. Abgespeichert wird sie aber unter dem vorher angegebenen neuen Namen.
- Neue Datei vom selben Typ:** Fast dasselbe, aber man bekommt eine Kopie der Prototyp-Datei P-0.APR. (=> 10.0)
- löschen** Die Bedeutung dieser Funktionen ist unmittelbar verständlich. Sicherungen sind eingebaut:
- umbenennen** Das Löschen muß bestätigt werden, bzw. wenn der neue Name schon existiert, also die Umbenennung nicht stattfinden kann, wird man informiert.

Wenn man den obersten Punkt des Menüs "Dateien", den Punkt "sämtliche Dateien", anwählt, werden alle auf dem betreffenden Verzeichnis stehenden Dateien angezeigt. Im Untermenü "Datei-Aktionen" erscheinen dann zusätzlich Datum und Uhrzeit der Datei-Erstellung sowie deren Größe. Außerdem gibt es einen weiteren Menüpunkt:

- diagnost** Anzeige der Datei in Form eines Dezimal-Dump, d.h. alle Zeichen werden mit ihrem dezimalen Codewert angezeigt. Sehr nützlich ist dies für denjenigen, der einen Import programmieren will und die genauen Werte der Satz- und Feldbegrenzungen nicht kennt, denn diese müssen in den Importparametern als Dezimalwerte angegeben werden (=> 11.2).

## 0.11.5 µo : Das Menü "Optionen"

Man verändert hier die grundlegenden Einstellungen, die für die aktuelle Sitzung gelten, man kann aber auch die Standardeinstellungen für künftige Sitzungen umdefinieren.

**CockPit** liest beim Start über CP.BAT die Datei CP.OPT (=> 0.11.6). Darin stehen die **Vorgaben**, die in den Statuszeilen am unteren Rand erscheinen und mit denen dann die Hauptroutinen arbeiten. Änderungen der Vorgaben wirken sich somit auf spätere Sitzungen aus. Wenn man aber etwa mehrere Datenbanken mit unterschiedlichen Charakteristika betreibt, wird man nicht jedesmal zuerst die Vorgaben ändern und dann **CockPit** verlassen und neu starten wollen. Deshalb lassen sich auf dem Menü "Optionen" alle Grundeinstellungen während einer Sitzung verändern. Und so sieht das Menü aus:

```

Funktionen      Routinen      Makros      Dateien      Optionen      HELP      EXIT

               Datenverzeichnis
               Konfiguration
               Sprache
               Laufwerke
               Vorgaben
               Ressourcen
                 Druck-Optionen
                 1 Sortierfolge
                 2 Vollständigkeit
                 3 Listengestaltung
                 4 Drucker

Umschalten auf ein anderes Verzeichnis
C: \ALLEGRO\DEMO
Datenbank = CAT      Konfiguration = A      Progr.Verz. =      C: \ALLEGRO
Einzelplatzbetrieb      Sprache = GER      Laufwerk = C      V15

```

Zu den Punkten im einzelnen:

- Datenverzeichnis :** Eine Auswahlliste der Verzeichnisse erscheint. Man wählt daraus genauso wie bei den Datei-Auswahllisten. Die Angaben "Datenverzeichnis", "Konfiguration" und "Datenbank" auf den unteren Zeilen ändern sich entsprechend dieser Wahl, falls auf dem Verzeichnis eine Datenbank liegt, auch erfährt man dann deren Größe. Anschließend beziehen sich die "Routinen" auf diese Datenbank. Mit **µr b** springt man sofort in die Benutzung derselben. So geht der Umstieg auf eine andere Datenbank sehr schnell.
- Konfiguration :** Die vorhandenen .CFG-Dateien werden zur Auswahl oder Bearbeitung angezeigt wie beim Menü "Dateien". In der untersten Zeile links ändert sich dann die Eintragung, wenn man eine andere Auswahl trifft. Anschließend werden im Menü "Dateien" nur die zu dieser Konfiguration gehörigen Parameterdateien gezeigt.
- Sprache :** Nur GER und ENG sind tatsächlich wählbar. Ansonsten haben Anwender UIF-Dateien für Spanisch und Niederländisch erstellt. Eigene Übersetzungen sind möglich. Angepaßte Versionen der UIF-Dateien können unter eigenen "Sprachbezeichnungen" angelegt und über den Unterpunkt "USR" gewählt werden.
- Laufwerk :** Umstellen des aktuellen Laufwerks; anschließend Datenverzeichnis wählen.
- Vorgaben :** Bearbeitung der Vorgabendatei, in der Regel CP.OPT (=> 0.11.6)
- Ressourcen :** Zeigt die Größe des für *allegro* verfügbaren Arbeitsspeichers und des freien Plattenraums.
- Druck-Optionen :** Diese 4 Punkte sind wichtig für die Produktion sortierter Listen. Sie finden eine genaue Beschreibung dieser Möglichkeiten im Zusammenhang mit dem Export in den Kapiteln 4 und 6. Dieselben Punkte sind unter "Routinen , Volltextsuche/Listen" (**µr v**) zu finden.

## 0.11.6 Die Vorgabedatei CP.OPT

Auf den folgenden zwei Seiten ist die mitgelieferte Datei CP.OPT abgedruckt. Sie enthält genügend viele Kommentare, um weitere Beschreibungen fast überflüssig zu machen. Über `µo v` kommt man in die Bearbeitung dieser Datei und kann die nötigen Anpassungen vornehmen. Die Kommentare werden in **anderer Schriftart** dargestellt, der relevante und wirksame Text dagegen in **Courier**. Sie können ansonsten Kommentare auch daran erkennen, wie in den Parameterdateien, daß die Zeile mit einem Leerzeichen beginnt bzw. innerhalb einer Zeile 2 Leerzeichen vorangehen.

**Unerwünschte Menüpunkte** können Sie entfernen: nehmen Sie die zugehörigen Zeilen aus der Datei UIFCGER heraus!

CP.OPT = Datei mit Variablen für den Start von CockPit

! Ergänzen Sie in der Datei CP.BAT

-M für Monochrom-Bildschirme,  
(und zwar anhängen an die Zeile `acp ...`)

Alle Zeilen, die mit Leerzeichen beginnen, sind Kommentare!

**c C** wenn Sie z.B. auf D: arbeiten, D statt C einsetzen  
**l GER** Sprache der Menütexe (diese Zeile darf nicht weiter unten stehen)  
**C cp** Name der Batchdatei, die nach einer Routine wieder aufgerufen wird

In die folgende Zeile tragen Sie **Ihr eigenes Datenverzeichnis** ein:

**d C: \ALLEGRO \DEMO** wiederholbar (2fach oder 3fach)  
 -----  
 aber dann muß der Datenbankname, z.B. \cat noch angehängt werden  
 und in die nächste das Programmverzeichnis:

**P C: \ALLEGRO**  
 -----

Kommentar für MS-DOS-Kenner:

(CP.BAT ist ein Batch-File, in dem ACP.EXE aufgerufen wird.

Wenn man ACP direkt aufruft, werden die angewählten Programme nicht gestartet, sondern es entsteht ein neues Batch-File namens CCC.BAT, in welchem der Programmaufruf mit den gewählten Optionen steht; am Ende wird wieder CP.BAT aufgerufen. )

ALLGEMEINE OPTIONEN:

(diese erscheinen in den 2 Fusszeilen)

Für die Beispiel-Datenbank werden folgende Einstellungen mitgeliefert:

ACHTUNG:

Die Funktionen zum Löschen, Regenerieren etc. sind nur verfügbar, wenn man ACP mit Option -a2 bzw. -a3 startet. Bauen Sie das auf Wunsch in CP.BAT ein!

**a 2** Zugriffsberechtigung (0: Menü "Funktionen" entfällt, 3: alle Rechte)  
**a 320** wenn 3 Datenbanken gleichzeitig, dann z.B. so  
**A 3** Berechtigungsstufe für **CockPit**, falls von **a** abweichend  
**b CAT** Datenbankname (Name **Ihrer Datenbank**)  
**k A** Konfiguration (z.B. U für USMARC)  
 es wird \$A.CFG genommen, wenn dies nicht existiert, dann A.CFG

**E x**

Editor-Aufruf für die Parameterdateien etc.

(schreiben Sie z.B. `E WP`, falls Sie WordPerfect benutzen wollen

`E EDIT` für den neuen Editor von MS-DOS 5)

(Default ist sonst das mitgelieferte X.EXE: die Anleitung dazu steht in der Datei X.DOC)

Achtung: bei Hercules-Monitoren gibt es meist Abstürze, wenn man Dateien bearbeiten will

Abhilfe: ersetzen Sie die Zeile `E x` durch:

**E XXX**

und legen Sie eine Datei XXX.BAT an, die diesen Befehl enthält:

COMMAND /C x %1

**V v**

Viewer-Aufruf

(falls Sie ein besonderes Programm zum Betrachten der Dateien benutzen wollen, z.B. DEBUG - Default ist sonst das mitgelieferte V.COM)

**B B:**

für das Backup kann hiermit ein Laufwerk vorgegeben werden (default ist A:)

**H PRINT** (PRINT ist default)

zum Ausdrucken von Parameterdateien etc. kann man eine HARDCOPY-Funktion vorgeben.

Achtung: wenn man PRINT benutzt, muß man vor Start von ACP wenigstens einmal den Befehl PRINT geben, damit der residente Teil geladen ist. Unter Novell wird man evtl. NPRINT nehmen.

Man könnte auch H TYPE geben, dann würde die angewählte Datei am Bildschirm erscheinen!

**L 120000**

maximale Logfile-Größe in Byte. Wenn die Datei .LOG größer wird, informiert CockPit Sie bei jedem Start darüber, daß eine Sicherung sinnvoll wäre.

\*\*\*\*\*

\* Außerdem können Sie eine Anzahl **eigene Menüpunkte** ins CockPit einbauen: \*

Die eigenen Menüpunkte werden an das Menü ROUTINEN angehängt.

Die dafür nötigen Zeilen beginnen mit R oder S

R	<i>MenuItem, Batch?Frage, InfoLine</i>	Hauptpunkt, bis zu 16 Stück wenn mehr als 6, dann kommen die weiteren auf Menü <b>Makros</b>
S	<i>MenuItem, Batch?Frage, InfoLine</i>	Submenü-Punkt, bis zu 10 je R-Hauptpunkt

mit:

<i>MenuItem</i>	=	Bezeichnung des Menüpunkts
<i>Batch</i>	=	Name des aufzurufenden Batch-Files. Hier kann auch ein MS-DOS-Befehl stehen (Batch-Files müssen auf dem aktuellen Verzeichnis liegen, sonst Pfadangabe nötig). Bei R KANN hier '&' stehen, dann MÜSSEN S-Zeilen folgen
<i>InfoLine</i>	=	dieser Text erscheint auf der Hilfszeile, sobald der Menüpunkt aktiviert wird.
<i>Frage</i>	=	Frage an den Benutzer, bevor <i>Batch</i> gestartet wird Die Antwort auf die <i>Frage</i> wird als Parameter %1 an <i>Batch</i> weitergegeben.

Die folgenden Zeilen sind Beispiele, die zeigen, wie die Struktur aussehen muß. Vergleichen Sie dazu, wie die Menüpunkte im Menü "Routinen" herauskommen, insbes. wie die Untermenüs funktionieren:

```
R 1 Einfache Anwendungen, &, Pakete QUEX, QUANT, PRONTO, EXPEX (ab V14c)
S 1 Sortierte Listen, QUEX.BAT, Menuegesteuerte Produktion von Titellisten
S 2 Statistik-Auswertungen, QUANT.BAT, Quantitative Auswertungen von Feld- und Teilfeldinhalten
S 3 Neues Projekt anlegen, PRONTO.BAT, Menuegefuhrtes Definieren und Anlegen einer neuen Datenbank
S 4 Parameter-Experimente, EXPEX\EXPEX.BAT, Lernen Sie die Exportsprache mit USMARC!

R 2 OPAC, apac -e I-1/DOWNLOAD.-%K1%LG, Online-Katalog
R 3 MENUED, menued -a3 -n1 -S , Editor mit Komfort
R 4 ALFA, alfa -a3 -n1 -S -s"|3m", Einfach-Ausleihe

R 5 Download, &, Erzeugen von Downloads und ihre Verarbeitung
S Download mit PRESTO, P-DOWN.BAT, Daten aus der Datenbank herunterladen in eine Grunddatei
S Download mit SRCH, S-DOWN.BAT, Mit SRCH eine Grunddatei aus der Datenbank herausziehen
S Sortierte Liste, S-LIST.BAT, aus der Download-Datei eine sortierte Liste produzieren
S Unsortierte Liste, U-LIST.BAT, letzte Download-Datei unsortiert ausgeben
S Download bearbeiten, allegro -f3 -ddownload, Download-Datei offline bearbeiten
S Liste anzeigen, v liste, zuletzt produzierte Liste anzeigen
S Liste löschen, del liste, zuletzt produzierte Liste löschen
S Download löschen, del liste, zuletzt produzierte Download-Datei löschen
S Registerauschnitte, QR.BAT?Welches Register, Teile des Index ausgeben

R 6 Import-Aktionen, &, Fremddaten-Import und Einarbeitung in eigene Datenbanken
S 1 BIBLIOFILE, bf.bat, Import von Bibliofile-Daten in die eigene Datenbank
S 2 DEUTSCHE BIBLIOTHEK DISK, dbdisk.bat, DB-Diskettendaten importieren
S 3 DEUTSCHE BIBLIOTHEK CD, dbcd.bat, DB-Daten von CD-ROM importieren
S 4 VLB-Daten, vlbkonv.bat, Daten aus dem Verzeichnis Lieferbarer Bücher
S 5 OCLC, oclc.bat, MARC-Daten vom OCLC oder von der LC
```

S 6 PICA, pica.bat, Pica3-Daten vom GBV

R 7 Aktionen, &, besondere Routinen

S MS-DOS Exkursion, COMM.BAT, Ausstieg in's MS-DOS - Rückkehr mit EXIT

S Indexparameter, edit-ixp.bat, Indexparameter-Datei bearbeiten

S Anzeigeparameter, edit-d1.bat, Datei D-1.APR bearbeiten

S Konfiguration, edit-cfg.bat, CFG-Datei bearbeiten

S Sniffer, sniffer -a3 -x, Untersuchung der Datenbank auf formale Fehler

S Erw. Zeichensatz, ostwest.bat, Umstellen auf den OSTWEST-Zeichensatz

S Norm-Zeichensatz, VGANORM.bat, Rücksetzen auf den VGA-Zeichensatz

Wenn an der Position *Batch* kein Batchfile, sondern ein EXE- oder COM-Programm eingesetzt wird, kann der Name ohne .EXE bzw. .COM angegeben werden, .BAT darf dagegen nicht fehlen. (Vgl. oben die Zeilen unter R 7)

\*\*\*\*\*

\* Voreinstellungen für die einzelnen "allegro"-Programme \*

Hier können für die einzelnen Programme eigene Optionen stehen, die dann im Menüpunkt 1 "Funktionen" benutzt werden. Hinter einer Zeile mit der Kennziffer des Programms folgen Zeilen mit dem Optionsbuchstaben und dem zuzuordnenden Wert:

```
1 presto (Optionen unterhalb dieser Zeile gelten nur im Menü "Funktionen")
s a Index wird zuerst bei 'a' aufgeklappt
p ANZEIGE statt D-1.APR soll ANZEIGE.APR benutzt werden
q P-KARTE für den Druck (mit F2) die Parameter P-KARTE.APR
```

```
4 srch
f 4
d c:\allegro\beispiel
m 1
```

```
5 import
v 0
e i-1
...
```

## Nachwort zu Kapitel 0

*allegro* ist noch immer nicht in jeder Hinsicht vergleichbar "anwenderfreundlich" wie einige andere Systeme. Der Einstieg wird deshalb in der Regel als schwierig empfunden. Eingearbeitete Anwender verstehen jedoch, daß dies der Preis der Flexibilität ist. Die Einübung erfordert einige Zeit und genaues Formulieren der eigenen Wünsche und Vorstellungen. Dieses jedoch sollte am Anfang **jeder** DV-Anwendung stehen.

Wahrscheinlich wünschen sich die meisten Interessenten eine strengere oder mehr ins Detail gehende Menüführung, und sicher wird man dieser Vorstellung allmählich näher kommen - das *CockPit* geht schon ein gutes Stück in diese Richtung, und die Programme für Erwerbung und Ausleihe setzen dieselbe Menü- und Fenstertechnik ein. Das Programm MENUED hat alle Funktionen von PRESTO, aber zusätzlich ein Menü (mit TAB zu aktivieren), das viele Bearbeitungsfunktionen auf Knopfdrücke reduziert. Wer länger mit *allegro* arbeitet, weiß aber die Freiheitsgrade und das schnelle Arbeiten zu schätzen, und daran soll es auch in Zukunft keine Abstriche geben. Es handelt sich bei *allegro*, wie man auch sagen könnte, um ein Katalogisierungssystem für "Erwachsene", die ihre Spielregeln kennen und in eigenem Interesse danach handeln. Auch ein noch so ausgefeiltes Menü- und Maskensystem ersetzt ja nicht die Regelwerkskenntnisse. Ob und wie z.B. ein Urheber einzugeben ist, und ob der, den man eingegeben hat, tatsächlich einer ist, das kann unmöglich alles über Hilfstexte abrufbar gemacht und schon gar nicht automatisch überprüft werden.

Da alle *allegro*-Programme als Module aufrufbar und von anderen Prozessen (vor allem MS-DOS-Batchprogrammen oder UNIX-Shellprozeduren) aktivierbar sind (=> Kap.12), können Anwender, die das *CockPit* nicht einsetzen wollen, sich komplette Menüsysteme schaffen, in denen alle Abläufe, die im Normalbetrieb vorkommen, mit wenigen Tasten aktiviert werden können. Menügesteuerte Abläufe für das Erstellen neuer Datenbanken sowie für statistische Auswertungen und die Produktion sortierter Listen werden bei Version 14c erstmals mitgeliefert.

Benutzeroberflächen sind im übrigen besonders problematisch hinsichtlich der Portabilität. Für *allegro* wurde angestrebt, daß es weitgehend identisch unter MS-DOS und UNIX lauffähig ist und Datenbanken unmittelbar zwischen diesen Plattformen ausgetauscht werden können. Dazu ist es notwendig, Oberfläche und Kern in den Programmen so zu trennen, daß nur relativ kleine Programmteile betriebssystemspezifisch zu schreiben sind. Längerfristig wird sich diese Bemühung auszahlen, zunächst ging es dadurch aber langsamer voran, als wenn man sich ausschließlich auf ein bestimmtes Betriebssystem konzentriert hätte. Die Weichen sind jedoch gestellt: alle Züge fahren jetzt auf den Schienen "Client/Server-Technik" und "Internet/Intranet". Erste Anwendungen für Windows und WWW sind in Betrieb und breiten sich zur Zeit stark aus. Das Handbuch kann diesen Bereich noch nicht behandeln, doch liegen eigene Dokumentationen dafür vor.