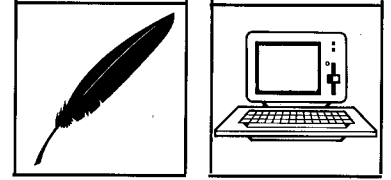


# MS-DOS → SINIX Teil 2

Bernd Sommer



Wichtig sind die in der zweiten bis fünften Spalte mit „7c“ dargestellten Bytes. In der letzten Spalte (zwischen den senkrechten Strichen, | und |, die hier aber nur der Optik dienen) werden druckbare Zeichen von „xd“ als „Buchstaben“ wiedergegeben, also „7c“ (Hex) als „|“. Wichtig ist außerdem die Zeile (erste Spalte links), die mit „1c0“ beginnt. Sie sehen dort, daß der Datensatz mit „0a“ (= Dezimal „10“ = LF = Line-Feed = neue Zeile) abgeschlossen wird. Ein aufmerksamer Leser könnte entdecken, daß ein – im Verhältnis zur Ausgangs-dBase-Datei – zusätzliches, zwanzigstes, Feld vorhanden ist. Daran muß er sich hier aber nicht stören. Auf Grund und Zweck komme ich noch zu sprechen.

Die beschriebenen Begrenzungen sollten nun auch beim Export aus der MS-DOS-Datenbank erscheinen. Also habe ich zwei Maschinen (MS-DOS und SINIX) sowie meinen Kopf heißlaufen lassen, um – wie ich glaube – alle Export-Variationen, welche in den Handbüchern zu dBase IV verraten werden, dazu noch einige, die mir gefallen hätten, aber nicht zum Ziel geführt haben, durchzuprobieren.

Es gibt eine Alternative. Sie können auch die SINIX-Begrenzer an die MS-DOS-Möglichkeiten anpassen. Der Feldbegrenzer kann mit der Shell-Variablen DBDELIMITER geändert werden. Wenn Sie das wollen, z.B. um dieselben Begrenzer wie in der MS-DOS-Datenbank einzustellen, legen Sie sich zweckmäßiger Weise eine „profile“-Datei an, oder ergänzen die schon vorhandene Datei. Da vielleicht nicht jeder Benutzer auf die Änderungen gefaßt ist, empfiehlt es sich, nicht global zu ändern, sondern lediglich die .profile-Datei in Ihrem eigenen Home-Directory zu ändern oder anzulegen. Im eigenen Home-Directory landen Sie, wenn Sie aus dem Menu-System mit „!“ aussteigen, oder auf der Shell-Ebene „cd“ eingeben. In die .profile-Datei könnten Sie dann eintragen:

```
DBDELIMITER=„neuer Delimiter“ export DBDELIMITER usw.
dBase IV: bietet an:
```

## 1. copy to ... delimited

(Felder durch Komma getrennt, alphanumerische Felder zusätzlich durch doppelte Anführungszeichen eingeschlossen)

Ergebnis, z.B.:

```
1,377,„HGB“,477,„BGB“,459,„BGB“,„Rügelast Nebenpflicht“,„BGH NJW 89/2532“,
„Tiedtke“,„NJW“,90,14,„Die Rügelast des Käufers bei Verletzung von
Nebenpflichten dch d Verkäufer.“,„Besprg. v. NJW 89/2532“,„“,„“,„“,„“
```

```
2,535,„BGB“,0,„“,0,„“,„Wohnraummietrecht-88“,„“,„Sonnensche“,„NJW“,90,17,
„Die Entwicklung des privaten Wohnraummietrechts 1986 bis 1988“,„“,„“,„“,„“,„“
```

## 2. copy to ... delimited with

(Felder durch Komma getrennt, alphanumerische Felder zusätzlich (!) durch Begrenzer Ihrer Wahl, also z.B. | eingeschlossen: Kommata stören dann immer noch!)

## 3. copy to ... delimited with blank

(schreibt ASCII-Datei, in welcher Felder aller Art lediglich durch ein Leerzeichen getrennt werden: Gibt spätestens bei den Feldern inh1 bis inh5 Ärger, weil dort die Wörter durch Leerzeichen getrennt sind!)

## 4. copy to ... type sdf (System Data Format)

(Alphanumerische Felder ohne Begrenzungszeichen, aber dafür weisen die Datensätze eine feste und dieselbe Länge auf: Äußerst schwierig, d.h. es ist mir nicht gelungen, das Zeichen | ausschließlich an die richtigen Stellen zu setzen)

Weitere im Handbuch „Befehle“ auf S. 2.83 ff. (insbes. Version 1.1) beschriebene Optionen gelten speziellen Zielformaten, die für Informix nicht in Betracht kommen. Einige nicht beschriebene Ideen habe ich erfolglos probiert.

Also habe ich mich an der erstgenannten Option (copy to ... delimited) einige Tage vertüfelt.

Es stellt sich dann – wie gesagt – die Aufgabe, die Kommata zwischen sämtlichen Feldern und die doppelten Anführungszeichen, welche die alphanumerischen Felder (= Zeichen-Felder) einschließen, durch |-Zeichen zu ersetzen.

Informix:

Feldbegrenzer = 7c Hex = 124

Dezimal = |

Datensatzende = 0a Hex = 10

Dezimal = Line-Feed.

SINIX-Begrenzer an MS-DOS-Format anpassen: .profile-Datei

dBase IV: Exportformate

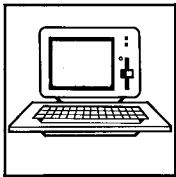
copy to ... delimited.

copy to ... delimited.with

copy to ... delimited.with blank

copy to ... type sdf

Bernd Sommer ist Richter am LG Coburg und Autor des Programmes TEN. Er ist auch in der jur-pc Mailbox erreichbar.



Suchen und ersetzen mit TRENNEX

Wie sucht und ersetzt man?

Die gängigen Textverarbeitungsprogramme und Editoren machen Schwierigkeiten, indem sie meist Zeilen bzw. Absätze (mehr als 500 Zeichen je Datensatz!) als „zu lang“ beklagen, die problematischen Zeichen nicht oder nicht günstig darstellen usw. Hier hilft TRENNEX ((C) Copyright 1988-1990, Hans-Jochen Pasenow, Vertrieb durch: MAKROLOG, Wiesbaden, V 2.9), ein Programm, das OPTOPUS-Anwendern geläufig sein dürfte. Man kann damit in einem Durchgang jedes beliebige Zeichen eines Textes suchen und ersetzen lassen, also auch Kommata und doppelte Anführungszeichen durch den senkrechten Strich (|). Nach dem Export von numerischen Feldern und Zeichenfeldern mit „copy to ... delimited“ besteht eine gewisse Schwierigkeit darin, daß die numerischen Felder lediglich mit Kommata getrennt sind. Da die Zeichenfelder Kommata als Satzzeichen enthalten, kann nicht einfach im gesamten Text jedes allein-stehende Komma durch | ersetzt werden. Man würde dann ja die Felder und ihr Zahl verändern. Man könnte durchaus einzeln ersetzen. Bei einigen Hundert Datensätzen ist das aber keine gute Idee.

Schon aus diesem Grund könnte es sich empfehlen, eine Kopie der dBase-Datenbank anzulegen, in der sämtliche Felder in Zeichenfelder umgewandelt werden. Der Export mit „copy to ... delimited“ zeigt dann etwa folgendes Ergebnis:

„ 1“, „ 377“, „HGB“, „ 477“, „BGB“, „ 459“, „BGB“,  
„Rügelast Nebenpflicht“, „...“, „...“, „...“

Mit TRENNEX müssen Sie dann Kommata und Anführungszeichen suchen und ersetzen. Ich habe es in verschiedenen Durchläufen gemacht, um mir jeweils das Ergebnis anzusehen, und dann eine weitere Zeichenkombination zusammenzubasteln. Beispielsweise können Sie

Die „Tauschpartner“

suchen und ersetzen:

), „, < durch | <  
)> < (Anf.zeichen durch > < (1 Leerzeichen!)  
und 1 (ein!)  
Leerzeichen,

wenn das allererste Anführungszeichen des Datensatzes gefunden und gelöscht werden soll. Am Leerzeichen erkennt TRENNEX das erste Feld, weil dieses fünf Zeichen lang ist und bei weniger als 10.000 Datensätzen, also maximal 9999, nur vier Stellen rechtsbündig besetzt sind.

)>< durch | <  
(es geht hier nur noch um jeweils das letzte Anführungszeichen, das nicht mit einem Komma kombiniert ist)  
falls z.B. übrig bleibt  
)|>< durch | <  
und so weiter ...

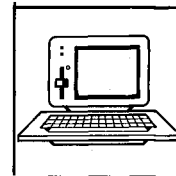
Ein Problem: Die Umlaute

Sodann könnten Sie noch die Umlaute austauschen, denn diese können bei Informix stören. Informix interpretiert „Ä“ als „eckige Klammer auf“, „Ü“ als „eckige Klammer zu“, ä und ü als „geschweifte Klammer auf“ und „zu“, „Ö“ als Backslash „/“ und „ö“ als „|“ (vgl. das Handbuch zu Informix V2.1, Nachschlagen, A-64). Dazu finde ich allerdings kein befriedigendes Ergebnis. Wenn ich z.B. „ö“ durch „oe“ oder „ß“ durch „ss“ ersetze, wird der Feldinhalt u. U. länger als das Feld und dann wird (als geringstes Malheur) der letzte Buchstabe irgendwann abgeschnitten. Ich habe dann einfach „ö“ durch „o“ ersetzt usw., und „ß“ durch „S“. Dann ist eine Gefahr beseitigt, aber die Optik miserabel. Bei der noch vorzuschlagenden Alternative ohne jede Behandlung der Umlaute sieht es auch nicht schlechter aus, wenn im SINIX-Text statt des Umlauts ein Fragezeichen erscheint. Man könnte dann im SINIX-Text global das Fragezeichen suchen und ersetzen.

Die TRENNEX-Grenze von 2<sup>8</sup> Zeichen

Wenn Sie sich nach so einem Austausch-Durchlauf von TRENNEX Ihr neues File anschauen, fällt Ihnen sicher früher als mir auf, daß alle Datensätze etwas kürzer geworden sind. Als ich das gesehen habe, habe ich überschlägig 250 vorhandene Zeichen gezählt. Dann liegt es nahe, daß es 256 (2er-System, 2 hoch 8 = 256) sind. So ist es mir auch auf Nachfrage bestätigt worden. TRENNEX beschränkt sich auf 256 Zeichen und trennt den Rest ab. Somit mußte sich der Perfektionist eine Lösung ohne TRENNEX einfallen lassen. Dazu später. Wenn man bei TRENNEX bleiben will, bietet sich an, die Datenbank nicht in kompletten Datensätzen (etwas mehr als 500 Bytes), sondern beispiels-

TRENNEX und die „Aufteilungslösung“: Nur für Spezialisten



weise bis zum Feld „inh1“, dann noch einmal, nur „inh2“ bis „inh4“, und schließlich „inh5“ und „buchstabe“ zu entladen. Das geht ganz einfach, indem Sie den Befehl „copy to ... delimited“ ergänzen um „ fields dnr, ... (usw. bis) inh1“, usw., wobei Sie allerdings die Anführungszeichen nicht mit eingeben dürfen. Diese Aufteilungslösung kann ich aber nur demjenigen empfehlen, der auch herausbekommt, wie man den zweiten und dritten Teil in die vorhandenen Datensätze einfügt, statt sie jeweils anzuhängen. Mit dem Kommando „load ... insert into ..“ habe ich nur die Zahl der Datensätze verdreifacht, wobei der erste Teil bis „inh1“ angefüllt war, der zweite in „inh2“ bis „inh4“ usw. Ich hätte, als dieses Problem erkannt war, daran gehen können, mehrere Dateien anzulegen und sie anhand der Datensatznummer zu verknüpfen. Vorher wollte ich noch anderes probieren. Das Ergebnis ist, wenn auch von zusätzlicher oder anderer Software abhängig; nicht gerade schlecht.

Mir fiel mein gutes altes Superbase (Relationales Datenbanksystem, eingetragenes Warenzeichen von Precision Software Ltd., England; vertrieben von Markt & Technik) ein.

Superbase hat in seinem Einstellungen-Menü u.a. den Auswahlpunkt „Import/Export“. Mit Import/Export können Sie die Trenner von Feldern und Datensätzen durch Eingabe des jeweils zugehörigen ASCII-Codes setzen (Superbase-Handbuch, S. 124), sowie bestimmen, ob Anführungszeichen benutzt werden. Voreingestellt ist „44“ (= Komma = Dezimal dargestellt) als Feldtrennzeichen und „13“ (Wagenrücklauf = CR) als Datensatztrennzeichen. Geben Sie nun als Feldtrennzeichen „124“ (Dezimalwert von ASCII | ) und als Datensatztrennzeichen „10“ (= LF = Line-Feed = Zeilenvorschub) ein, dann erhalten Sie beim Export aus Superbase genau die richtigen Begrenzer, die Sie für Informix benötigen. Außerdem muß beim Export die Option „Anführungszeichen benutzen“ ausgeschaltet sein. Soweit sind wir aber noch nicht. Jetzt muß erst die Datenbank aus dBase IV exportiert werden, so daß sie von Superbase wieder eingelesen werden kann. Also:

Noch nicht nachahmen! Es gibt Probleme!

MS-DOS-Seite:

(dBase IV)

1. use jdb90  
(Original-dBase-IV-Datei mit numerischen und Zeichenfeldern)
2. copy to j9.db4 delimited  
oder, gleiche Wirkung,  
copy to j9.db4 delimited with "
3. quit
4. copy j9.db4 /sb  
(File in Superbase-Verzeichnis kopieren, auf Betriebssystemebene)

(Superbase)

5. GEM (grafische Benutzeroberfläche, eingetragenes Warenzeichen von Digital Research Inc., USA, vertrieben von Markt&Technik)  
starten, bei mir mit  
cd /  
gem  
(Batch-Datei)
6. sbge.app-Icon doppelt anklicken und damit Superbase starten
7. Trenner (s.o.) einstellen für dBase IV,  
und zwar „44“ (Komma als Feldtrenner)  
„13“ (CR als Satztrenner)  
sowie „Anführungszeichen benutzen“ EIN!
8. Datei/Datenbank gemäß der Struktur der dBase IV-Datei anlegen (Numerische und Textfelder)
9. „Prozeß“-Menü anklicken, und dort
10. „Import“ anklicken
11. Filter-Auswahl mit Return übergehen, da alle Datensätze eingelesen werden sollen.
12. Als Dateinamen angeben (s.o. 8.): j9.db4

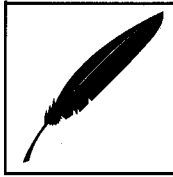
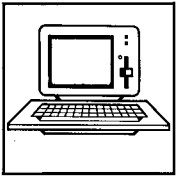
*Der Umweg über Superbase*

*MS-DOS-Seite:*

*Export von dBase nach Superbase*

*dBase-„copy“ für den Export,  
DOS-„copy“ zum Dateikopieren*

*Superbase: Im- und Export*



13. Trenner (s.o.) einstellen für Informix,  
und zwar „124“ (| als Feldtrenner)  
„10“ (LF als Satztrenner)  
sowie „Anführungszeichen benutzen“ AUS!
14. „Prozeß“-Menü anklicken, und dort
15. „Export“ anklicken
16. Filter-Auswahl mit Return übergehen, da alle  
Datensätze exportiert werden sollen.
17. Als Dateinamen angeben (z.B.): j9db4.sb  
(Das wird der Name der Ziel-Datei)
18. Superbase beenden

**(Betriebssystem)**

19. j9db4.sb an MS-DOS - SINIX Transfer-Software übergeben.  
(Die Einzelheiten erfordern ein eigenes Kapitel,  
das in einem weiteren Erfahrungsbericht folgen soll!)

Es gibt verschiedene Softwarelösungen. Wesentlich ist, daß die von MS-DOS aus zu übertragende Datei auf eine für SINIX/UNIX mit den Befehlen tar, far oder cpio etc. lesbare Diskette in dem von SINIX/UNIX geforderten Format übertragen wird.

20. Diese Diskette nehmen Sie aus ihrem MS-DOS-Rechner und  
schieben sie in den Diskettenschacht des  
SINIX-Rechners.

Mit dieser Trivialität möchte ich den Leser nicht beleidigen. Ich schreibe ja einen Erfahrungsbericht. Mit dem vergessenen Austausch der Disketten habe ich nämlich schon einige Zeit vertan.

*MS-DOS-Seite:  
Übergabe an Transfer-Software*

*SINIX-Seite*

**(SINIX-Seite)**

Sodann

21. cd  
um ins Home-Directory zu gelangen.
22. cd DB  
falls Ordner DB schon angelegt, sonst erst  
mkdir DB  
und dann  
cd DB  
(Die Großschreibung ist signifikant! Der MS-DOS-  
Kenner übersieht das gern.)
23. Einlesen in Ordner DB mit dem Shell-Kommando  
(z.B.)  
far xv j9db4.sb

24. isql

Im Informix-Menusystem wählen Sie dann:

Ablauf,  
Dialog,  
Auswahl Datenbank: leitdb  
vorausgesetzt, es existiert schon eine  
Datenbank des Namens leitdb und eine Datei/Tabelle  
mit dem Namen ls90  
entsprechend der Struktur der dBase IV-Datei.

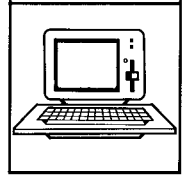
*SINIX unterscheidet Groß-/  
Kleinschreibung*

*Der Informix-Import*

*Der aktuelle Zustand:  
Info- und Status-Abfrage*

Sie kommen dann in das RDSQL-Menü. Dort wählen Sie „Info“ und „Status“, um sich zu vergewissern, daß die Tabelle noch leer ist. Allerdings wird der richtige Status nach Veränderungen erst dann angezeigt, wenn er mit „update statistics“ auf den aktuellen Stand gebracht worden ist. Dazu wählen Sie „Neu“ an, tippen

update statistics  
oder  
update statistics for table ls90



und drücken die „START“-Taste. Es kommt dann irgendwann die Original-Meldung „Statistik auf neuem Stand gebracht“. Mit Info erfahren Sie den aktuellen Zustand. Falls die Bank nicht leer ist (Anzahl Sätze 0), leeren Sie sie mit der Anwahl von „Neu“ und tippen

```
delete from ls90
```

Dann drücken Sie „START“ und bestätigen den Hinweis, daß die where-Klausel fehlt, weshalb sämtliche Datensätze gelöscht werden.

Jetzt wird es – und wurde es jedenfalls für mich – spannend!

Sie wählen an:

„Neu“ oder „Ruf-Editor“

(letzteres führt zum ced-Editor)

Sie tippen: (Sie! Ich wähle „Lesen“, denn ich hätte meine Experimente nicht durchgestanden, wenn ich nicht diese Dialoganweisung mit „Ablegen“ hätte konservieren können)

```
load from „j9db4.sb“
insert into ls90(dnr,par_a,gesetz_a,par_b,gesetz_b,
par_c,gesetz_c,stichw, dverw,autor,ort,jahr,seite,
inh1,inh2,inh3,inh4,inh5,buchst)
```

dann:

```
falls „Neu“ gewählt, START-Taste,
falls im ced,          MENU, v(erlassen), j(a), START-TASTE
```

Hurra, es funktioniert, Informix lädt. Aber, zu früh gefreut, es kommt doch noch ein Protest:

```
„Satz 117: Die Anzahl der Felder der Ladefielder
entspricht nicht der Anzahl der Felder“
```

Immerhin ist einschließlich Satz 116 die Datenbank ordentlich eingelesen worden. Weshalb aber nicht auch Satz 117? Ich schaue mir Satz 117 in der Superbase-Version an und sehe meine Ahnung bestätigt. In Satz 117 soll erstmals das letzte Feld des Satzes, nämlich „buchst“ gefüllt werden. D.h., das ist der erste Datensatz, in welchem eine Norm durch einen zusätzlichen Buchstaben beschrieben wird, z. B. § 89 b HGB. Weshalb macht das Schwierigkeiten, obwohl doch Export und Import die gleichen Felder verwenden?

Weiter hilft die Betrachtung des Ergebnisses eines „unload“ aus einer intakten Informix-Tabelle mit dem SINIX Shell-Kommando xd (s.o.). Die letzten ausgegebenen Zeilen waren:

```
1b0 6c656963 68207761 72656e2e 7c207c20 |leich waren. | | |
1c0 7c0a3131 337c3639 7c5a504f 7c313130 | | 113|69|ZPO|110|
|
| Datensatz-
| Ende = 0a
```

Sie sehen, daß das letzte Feld (das 19., welches den Buchstaben aufnehmen soll) hier leer ist, d.h. kein druckbares Zeichen enthält, sondern ASCII 20 = NUL. Außerdem wird es – wie sonst – durch „7c“ begrenzt. Schauen Sie sich dagegen einen Export aus Superbase z.B. mit den Norton-Utilities in Hexadezimaler Ausgabe an, dann sehen Sie, daß nur das 18. Feld, das „inh5“ enthält, mit „7c“ begrenzt wird. Das 19. Feld, das leer ist, erscheint überhaupt nicht. Es folgt nach dem Begrenzer des 18. Feldes nur noch das Zeichen „0a“ für Datensatz-Ende. Somit kann man den Schluß ziehen, daß Informix auch für das letzte Feld den Begrenzer „7c“ verlangt, Superbase diesen aber nicht liefert. Nun dürfen Sie tricksen: Die Superbase-Datei bekommt ein weiteres Feld als dummy, das nie gefüllt, trotzdem aber mit exportiert wird. Das 19. Feld (buchstabe) wird auf diese Weise wunschgemäß mit 7c begrenzt und das 20. Feld (dummy) interessiert auf der MS-DOS-Seite nicht mehr.

Nun folgt der m.E. nachahmbare Ablauf am Beispiel der Datei für das Jahr 1989.:

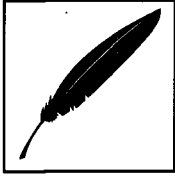
*Importieren mit:  
load from und  
insert into*

*Das Rätsel um Datensatz 117*

*Die Lösung: Superbase begrenzt  
das letzte Feld nicht.*

*Superbase überlistet: Der dummy-  
Trick*

*Endlich: Zum nachahmen*

MS-DOS-Seite:  
(dBase IV)

1. use jdb89  
(Original-dBase IV-Datei mit (numerischen ? und)  
Zeichenfeldern)

\*\*\*\* Vorsicht! \*\*\*\*

*Tücken des „Empire-Stils“: Neue  
Datensätze nicht akzeptiert*

Der Empire-Stil (empirische Methode) zeigt nun seine häßlichere Seite. Als ich auch die 90er Leitsätze übertrug, funktionierte alles wunschgemäß, wie für jdb89 beschrieben. 391 Datensätze gingen rüber. Dann tippte ich weitere Leitsätze (=Datensätze) in die dBase-Datei ein. Diese sollten schließlich an die Informix-Datei angehängt werden. Nichts wie Ärger. Die Programme meldeten Prüfsummenfehler und Informix machte Schwierigkeiten bei der Umwandlung von ASCII-Zeichen geltend. Also: Informix-Tabelle „ls90“ entleeren mit „delete from ls90“ (ohne „where ...“, um sämtliche (391) vorhandenen Datensätze zu entfernen) und versuchen, alle – inzwischen 478 – Datensätze neu einzulesen. Das „Hurra“ erstirbt beim 391. Datensatz. Mehr wird nicht eingelesen.

Also ist etwas zwischen 391 und 392 passiert. Stundenlanges Tüfteln mit „type“ auf der MS-DOS-Seite und „xd“ bei SINIX. Ich sehe keinen Unterschied.

Schließlich: Ich hab's! Bei Datensatz Nr. 391 enthalten sämtliche Paragraphenfelder einen Eintrag, z.B. „242“ oder „631“ usw. oder einfach „0“. Bei 392 finden sich Einträge zu par\_a und par\_b. Aber bei par\_e steht nicht einmal eine Null. Das ist der signifikante Unterschied zwischen den Datensätzen bis 391 und nach 391. Wo kommt er her? Die „Rückschritte“ führen zu dBase. „use jdb90“, „go 391“ und „disp next 3“ zeigt, daß hier schon die Unterschiede (mit 0 und ohne 0) vorkommen. Sie wissen bereits:

*Ursache und Konsequenz*

Ich hatte ursprünglich statt der numerischen Felder Zeichenfelder. Bei der Umwandlung in numerische Felder blieben die Nullen. Bei Neueintragungen gab es keine Nullen. Folglich scheint es – was die weiteren Experimente bestätigen – doch besser, von dBase aus reine Zeichenfelder zu exportieren.

Wenn Sie Ihre funktionierende dBase-Datei nicht verändern wollen, können Sie z.B. so vorgehen (dBase IV 1.1: „with production“ bedeutet: Arbeitsindex (.mdx-Datei) wird mitkopiert):

*Die Struktur der dBase-  
Datenbank ändern:  
modify structure*

```
use jdb89.sdf
copy to j9neu with production
use j9neu
```

Sodann ändern Sie mit

```
modify structure
```

die numerischen Felder von j9neu in Zeichenfelder um. Bei mir genügten die Felder par\_b und par\_c, weil die anderen stets „echte“ Eingaben enthielten. Falls Sie sich auch bei dnr oder par\_a usw. ausdrücklicher Eingaben enthalten haben, müssen Sie auch die betroffenen numerischen Felder in Zeichenfelder ändern! Wenn Sie, eventuell nach neuer Indizierung, dann am dBase-Punkt eingeben:

*go = Positionierung des  
Datensatzzeigers  
disp = display = Datensatzanzeige*

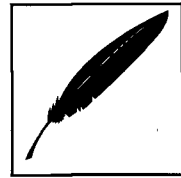
```
go 391
disp next 3
```

sehen Sie, daß wir jetzt in allen früher numerischen Feldern, die keine ausdrückliche Eingabe bekommen haben, Nullen finden. Vor und nach Datensatz 391 sieht es nun gleich aus. Der nächste Schritt aus dBase mit

```
copy to j89.db4 delimited
```

sollte somit – wie beschrieben – von einer dBase-Datei mit Zeichenfeldern („use j9neu“) ausgehen.

2. copy to j89.db4 delimited
3. quit
4. copy j89.db4 /sb  
(File in Superbase-Verzeichnis kopieren, auf  
Betriebssystemebene)

**(Superbase)**

5. cd /  
gem  
(Batch-Datei)
6. sbge.app-Icon doppelt anklicken und damit Superbase starten
7. Trenner (s.o.) einstellen für dBase IV, und zwar „44“ (Komma als Feldtrenner)  
„13“ (CR als Satztrenner)  
sowie „Anführungszeichen benutzen“ EIN!
8. Datei/Datenbank gemäß der Struktur der dBase IV-Datei anlegen (Numerische und Textfelder)
9. „Prozeß“-Menü anklicken, und dort
10. „Import“ anklicken
11. Filter-Auswahl mit Return übergehen, da alle Datensätze eingelesen werden sollen.
12. Als Dateinamen angeben (s.o. 2.,4.): j89.db4
13. Trenner (s.o.) einstellen für Informix, und zwar „124“ (| als Feldtrenner)  
„10“ (LF als Satztrenner)  
sowie „Anführungszeichen benutzen“ AUS!
14. Datei editieren  
und Feld „dummy“, Text, Länge 1 Zeichen, zufügen und  
Dateiaufbau speichern.
15. „Prozeß“-Menü anklicken, und dort
16. „Export“ anklicken
17. Filter-Auswahl mit Return übergehen
18. Als Dateinamen angeben (hier z.B.): j89.sb
19. Superbase beenden

**(Betriebssystem)**

20. j89.sb an MS-DOS - SINIX Transfer-Software übergeben.
21. Diskette aus MS-DOS-Rechner in SINIX-Rechner.

**(SINIX-Seite)**

22. cd
23. cd DB
24. far xv j89.sb
25. isql

**Im Informix-Menusystem wählen Sie dann**

Ablauf,

Dialog,

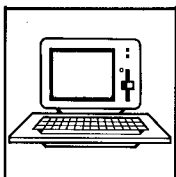
Auswahl Datenbank: leitdb

vorausgesetzt, es existiert schon eine Datenbank des Namens leitdb und eine Datei/Tabelle mit dem Namen ls89 entsprechend der Struktur der dBase IV-Datei für 1989, jedoch ohne !!!! „dummy“, also endend mit „buchst“. Die Tabelle darf schon indiziert sein, z.B. auf „dnr“. Das stört nicht, beschleunigt aber bei der Suche in der Testphase ungemein.

**RDSQL:**

„Info“, „Status“, u.U. „Neu“ für „update statistics“, START, u.U.

„Neu“ für „delete from ls89“, „START“ usw.



„Neu“ oder „Ruf+Editor“

Sie tippen:

```
load from „j89.sb“
insert into ls89(dnr,par_a,gesetz_a,par_b,gesetz_b,
par_c,gesetz_c,stichw,dverw,autor,ort,jahr,seite,
inh1,inh2,inh3,inh4,inh5,buchst)
```

dann:

```
falls „Neu“ gewählt, START-Taste,
falls im ced, MENU, v(erlassen), j(a), START-TASTE
```

*Beharrlichkeit belohnt: 577 Sätze  
geladen*

Nach einiger Zeit, Geduld, Geduld, kommt das erste Erfolgserlebnis:

„Saetze geladen: 577“

```
END-Taste, bis Sie zu
INFORMIX-SQL:
```

kommen,

Ablauf,

Format,

wobei Sie hoffentlich schon das Format für ls89  
erstellt oder kopiert, compiliert und gesichert haben,

und schon meldet sich, nach wenigen Sekunden, die Maske zur Bearbeitung der Tabelle ls89. Mit „s“ für „Suchen“ können Sie sich dann vom Erfolg überzeugen.

*Kommentar des Nordoberfranken:  
„Dessiss aanix“,  
die Umlaute als „?“*

Der Nordoberfranke wird sagen: „Dessiss aanix“, der Gebildete: „Diese Ausgabe fällt nicht zu meiner vollen Zufriedenheit aus!“ Die Umlaute und „ß“ sind ja nicht umgewandelt worden und werden nun als „?“ ausgegeben.

Falls Sie das stört, rege ich folgende Alternativen an:

1. Sie verwenden TRENEX, sind aber auf 256 Zeichen beschränkt (Für viele Dateien wird das genügen).
2. Sie verraten die optimale Lösung.
3. Sie nehmen Ihre MS-DOS-Editoren und probieren durch, welcher ohne Schaden für die Datei Umlaute etc. sucht und ersetzt.
4. Sie schauen mal, ob nicht eines der hervorragenden SINIX Shell-Kommandos die Fragezeichen bekehren kann (grep, awk, tr?)
5. Sie warten ab, ob es eine nächste Folge gibt und ob bis dahin der Autoren-groschen gefallen ist.

#### Elementare SINIX-Handbücher

1. Betriebssystem SINIX, Buch 1: Beschreibung
2. Betriebssystem SINIX, Buch 2: Menüs
3. SINIX-Schnittstellen
4. Betriebssystem SINIX: Systemverwaltung
5. Betriebssystem SINIX: Kommandos  
Teil 1 – Teil 2 – Teil 3 (alphabetisch geordnet)
6. Betriebssystem SINIX: Systemsicherheit
7. SINIX SPOOL: Anwenden –Verwalten –Programmieren

Bestellnummer und Preise dieser Bücher sind von der SINIX-Version abhängig, können aber bei den regionalen Siemens Nixdorf-Geschäftsstellen unter Angabe der Betriebssystemversionsnummer angefordert werden.