

Die in diesen Unterlagen enthaltenen Angaben und Daten können ohne vorherige Ankündigung geändert werden. Die in den Beispielen verwendeten Firmen, sonstige Namen und Daten sind frei erfunden, soweit nichts anderes angegeben ist. Ohne ausdrückliche schriftliche Erlaubnis der Microsoft Corporation darf kein Teil dieser Unterlagen für irgendwelche Zwecke vervielfältigt oder übertragen werden, unabhängig davon, auf welche Art und Weise oder mit welchen Mitteln, elektronisch oder mechanisch, dies geschieht.

Copyright © Microsoft Corporation, 1993. Alle Rechte vorbehalten.

Das FoxPro-Logo, FoxPro, Microsoft, und MS-DOS sind eingetragene Warenzeichen und Windows und Rushmore sind Warenzeichen der Microsoft Corporation in den USA und anderen Ländern.

dBASE, dBASE III Plus und dBASE IV sind eingetragene Warenzeichen von Borland International, Inc.
NetWare ist ein eingetragenes Warenzeichen von Novell, Inc.

Clipper ist ein eingetragenes Warenzeichen von Computer Associates International, Inc.

Alle anderen in diesem Handbuch vorkommenden Produktnamen werden lediglich zur Identifikation der Produkte erwähnt, es handelt sich dabei im Zweifelsfall um Warenzeichen der betreffenden Herstellerfirmen.

Inhaltsverzeichnis

µ	
Der Programm-Konverter	i
FoxPro	1
Der Programm-Konverter	2
Die Konvertierung von dBASE-Programmen	3
Überblick	5
Die Konvertierung	5
Tabellen-, Memo- und Indexdateien aus dBASE verwenden	6
Tabellen	6
Memodateien	6
Indexdateien	7
dBASE-Abfragen	7
Den Programm-Konverter starten	9
Masken, Berichte und Etiketten konvertieren	10
Der Datei-Konverter	10
Maskendateien	12
FMT- und PRG-Dateien konvertieren	13
dBASE-Berichte	14
Etiketten	18
Inkompatibilitäten zur dBASE-Sprache	19
Überblick	19
Hinweise zu Programmgenerator-Anwendungen aus dBASE IV	20
Hauptaspekte in Sachen Kompatibilität	20
Problemlösungsverfahren	21
Der Befehl SET COMPATIBLE	22
Benutzerdefinierte Funktionen (UDFs) aufrufen	22
Syntax-unabhängige Kompatibilitätsfragen	23
Parameterübergaben	23
Tastatureingaben	23
Hartcodierte Dateierweiterungen	24
Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV	24
Die Konvertierung von Clipper-Programmen	80
Überblick	81
Arbeitsschritte	81

Programm-Konverter

Hinweise zu Clipper 5.x Anwendungen	81
Tabellen-, Memo- und Indexdateien aus Clipper verwenden	82
Tabellen	82
Memodateien	82
Indizes	82
.NDX-Indexdateien	82
.NTX-Indexdateien	83
Eine Tabelle auswählen	84
Einen Indextyp auswählen	85
.IDX-Indexdateien erstellen	85
.CDX-Indexdateien	85
Funktionen in Indexausdrücken	86
Aus .FMT- und .PRG-Dateien FoxPro-Maskendateien erstellen	86
Berichte und Etiketten	86
Zugriff auf Funktionen und Prozeduren	87
Funktionen und Prozeduren "extrahieren"	87
Eine Prozedurendatei oder mehrere .PRG-Dateien erstellen	89
Eine Prozedurendatei erstellen	90
Mehrere .PRG-Dateien erstellen	90
Von Funktionen und Prozeduren bereinigte Versionen der Programmdateien erstellen	90
Inkompatibilitäten zur Clipper-Sprache	90
Übersicht	90
Die wichtigsten Kompatibilitätsprobleme	91
Weitere Aspekte in Sachen Kompatibilität	91
Funktionsaufrufe	92
ACHOICE(), DBEDIT() und MEMOEDIT()	92
Fehlerbehandlung	92
Fehlerbehandlung in FoxPro	93
Das Konstrukt BEGIN SEQUENCE...[BREAK]...END in FoxPro simulieren	95
Bibliotheken anderer Hersteller	97
Arrays	97

Farbdarstellung	98
Weitere Kompatibilitätsprobleme	99
Windows-spezifische Probleme	99
Drucken	99
ASCII-Zeichensatz	99
SET Befehle	99
Tastatureingaben	100
Verzeichnis inkompatibler Befehle und Funktionen aus Clipper	100
Das Analyse-Programm	145
Wie das Analyse-Programm arbeitet	146
Eine neue Analyse erstellen (Menüoption: New Analysis)	146
Die Dateien bearbeiten	147
Erforderlicher Festplattenspeicher	148
Einstellungen (Menüoption: Settings)	148
Analysegrad	149
Eine Analyse öffnen (Menüoption: Open Analysis)	149
Das Fenster "Program Analyzer"	149
Filtern und Sortieren	151
Vom Analyse-Programm aus Programmdateien bearbeiten	151
Einen anderen Texteditor einsetzen	152
Das Hilfesystem	153
Analyseberichte	153
Die Befehlsreferenztafel durchsehen	154
Nach der Bearbeitung der Problemfälle	154
FoxPro-Projekte	155
Anhänge	156
Anhang A: Der Befehl SET COMPATIBLE	157
Anhang B: dBASE-Dateien und ihre Verwendung in FoxPro	165
Anhang C: dBASE-Fehlernummern, die in FoxPro für andere Fehler stehen	169
Anhang D: Gleiche Fehlermeldungen mit unterschiedlichen Nummern	174

Programm-Konverter

Appendix E: Bibliotheken mit Netzwerk- und Sicherheitsfunktionen	176
Anhang F: Inkompatible Befehle und Funktionen in Clipper 5.x	189
Anhang G: Tastenbelegung	192
Tastenbezeichnungen für ON KEY LABEL KEY in FoxPro	192
INKEY() Codes in FoxPro, dBASE und Clipper Summer '87	194
Anhang I: Reservierte Wörter in FoxPro	197
Anhang J: Bibliotheken mit BIN-Funktionen	201
INDEX	202

Der Programm- Konverter

FoxPro

Es sprechen viele Gründe dafür, sich für FoxPro von Microsoft zu entscheiden: FoxPro ist schneller als alle Konkurrenzprodukte, es steht für bessere Entwicklungswerkzeuge und Zuverlässigkeit, und da von FoxPro 2.5 sowohl eine Version für MS-DOS® als auch eine für Microsoft Windows™ erhältlich ist, können mit FoxPro Daten und Anwendungen zwischen diesen Plattformen leicht ausgetauscht werden. Mit FoxPro können Sie zu guter Letzt ihre Investitionen in Xbase-Wissen, -Daten und -Anwendungen bewahren und sich dabei die Vorteile modernster Technologie zu nutze machen.

Dieser Programm-Konverter soll den Umstieg zu FoxPro so einfach wie möglich machen. In FoxPro 2.5 für MS-DOS und FoxPro 2.5 für Windows können mit dBASE III Plus® erstellte Programme *unverändert* ausgeführt werden. Die Befehle und Funktionen von dBASE III Plus stellen eine Teilmenge der Befehle und Funktionen von FoxPro dar. Auch viele in dBASE IV geschriebene Programme können in FoxPro unverändert ausgeführt werden. Allerdings weicht dBASE IV vom dBASE III Plus Standard ab. Daher müssen Programme, die dBASE IV-spezifische Funktionen verwenden, unter Umständen modifiziert werden. Dasselbe gilt für in Clipper Summer '87 geschriebene Programme. Deshalb wurde dieser Programm-Konverter entwickelt.

Mit Hilfe des Programm-Konverters und den in FoxPro enthaltenen Funktionen können Sie Ihre Dateien in FoxPro übertragen und dann die Geschwindigkeit und Zuverlässigkeit von FoxPro ausnutzen. Sie werden bald erkennen, wie viele Vorzüge FoxPro bietet, dazu gehören zum Beispiel die hervorragenden Entwicklungswerkzeuge und Dienstprogramme, die uneingeschränkte Mausunterstützung und die Möglichkeit, mit einer fensterorientierten Umgebung zwischen unterschiedlichen Entwicklungsplattformen zu wechseln.

Der Programm-Konverter

Mit dem Programm-Konverter können Sie Programme aus dBASE IV (Version 2.0 und früher) und Clipper Summer '87 konvertieren. Clipper 5.x unterscheidet sich sehr stark von FoxPro und wird nicht unterstützt. Dieses Programmpaket enthält eine in FoxPro 2.5 geschriebene Anwendung namens Programm-Konverter, mit der Sie viele Dateitypen aus dBASE IV und Clipper Summer '87 konvertieren können. Diese Anwendung analysiert darüber hinaus Programmdateien und sucht nach Programmteilen, die unter Umständen in FoxPro nicht korrekt ausgeführt werden können.

Der Programm-Konverter ist sowohl unter FoxPro 2.5 für MS-DOS als auch unter FoxPro 2.5 für Windows lauffähig. Die Anwendung wird dagegen nicht von FoxPro 2.0 oder von Versionen von FoxPro 2.5 für andere Betriebssysteme als MS-DOS und Microsoft Windows unterstützt.

Dieses Handbuch gehört zum Lieferumfang des Programm-Konverters. In diesem Handbuch wird der Konvertierungsvorgang erklärt, es beschreibt, wie Sie die Programm-Konverter-Software einsetzen und in welcher Weise Sie Ihre Programme verändern müssen, damit sie in FoxPro korrekt ausgeführt werden. Dieses Handbuch ist in drei Teile gegliedert:

- I. Die Konvertierung von dBASE-Programmen
- II. Die Konvertierung von Clipper-Programmen
- III. Das Analyse-Programm

Unabhängig davon, ob Sie ein dBASE- oder ein Clipper-Programm konvertieren sollten Sie, den Abschnitt "Das Analyse-Programm" auf jeden Fall lesen. Zuerst sollten Sie jedoch den betreffenden Abschnitt (dBASE oder Clipper) über die Konvertierung von Programmen und anderen Dateien studieren.

Die Konvertierung von dBASE-Programmen



Überblick

Ihr dBASE-Datenbestand setzt sich aus verschiedenen Arten von Dateien zusammen: aus Tabellen (in dBase-Terminologie Datenbestandsdateien), Abfragen, Masken, Berichten, Etiketten und Programmen. All diese Dateien können konvertiert und in FoxPro übernommen werden.

FoxPro verwendet dasselbe Dateiformat für Tabellen wie dBASE, Sie können daher ihre Tabellen (.DBF-Dateien) sofort in FoxPro weiter bearbeiten. Nach nur geringfügigen Veränderungen können Sie dBASE-Abfragen (.QBE-Dateien) in FoxPro ausführen.

Mit Hilfe des Programm-Konverters können Sie schnell und einfach Masken, Berichte und Etiketten von dBASE in die entsprechenden FoxPro-Dateien verwandeln. In FoxPro können auch Formatdateien (.FMT) aus dBASE verwendet werden, allerdings sind in manchen Fällen darin Änderungen vorzunehmen.

dBASE-Programmdateien (.PRG) müssen nicht verändert werden, um in FoxPro ausgeführt werden zu können, sofern sie nur Funktionen und Befehle enthalten, die dem Standard dBASE III Plus entsprechen. Das Analyse-Programm des Programm-Konverters hilft Ihnen dabei, Schlüsselwörter zu orten, die vom dBASE III Plus Standard abweichen, und es gibt Ihnen Hinweise zu Änderungen, die für eine korrekte Ausführung des Programms in FoxPro erforderlich sind.

Die Konvertierung von Programmen kann unterschiedlich lange dauern. Manchmal ist die Konvertierung um Handumdrehen beendet, manchmal dauert es etwas länger. Ganz gleich wie lange es dauert, Ihre Geduld wird in jedem Falls reichlich belohnt durch schnellere Programmausführung, größere Zuverlässigkeit und, sofern Sie FoxPro 2.5 für Windows besitzen, durch die Möglichkeit, Ihre DOS-Anwendungen in Windows-Anwendungen zu verwandeln.

Die Konvertierung

Bei der Konvertierung von Programmen gehen Sie im Prinzip immer wie folgt vor :

1. Erstellen Sie zuerst Sicherungskopien aller Dateien. Arbeiten Sie nie mit den Originaldateien!
2. Kopieren und laden Sie die Tabellen (in dBASE-Terminologie: Datenbestandsdateien) in FoxPro.
3. Verändern Sie die Abfrage-Dateien.
4. Konvertieren Sie mit Hilfe des Datei-Konverters alle Masken, Berichte und Etiketten.
5. Verwenden Sie das Analyse-Programm, um potentielle Inkompatibilitäten zu ermitteln und zu beheben.
6. Falls Ihr Programm mit Formatdateien (.FMT) arbeitet, prüfen Sie, ob diese verändert werden müssen.
7. Genießen Sie die Geschwindigkeit und den Funktionsumfang von Microsoft FoxPro!

Sie brauchen nicht alle dBASE-Dateien zu konvertieren, um ein Anwendung in FoxPro ausführen zu können. Lediglich Tabellen, Masken, Berichte, Etiketten und Programme werden benötigt. In Anhang B finden Sie eine Zusammenstellung aller dBASE-Dateitypen und Angaben dazu, wie diese bei der Konvertierung zu behandeln sind.

Tabellen-, Memo- und Indexdateien aus dBASE verwenden

TabellenXE "Tabellen"§

FoxPro verwendet dasselbe Standarddateiformat (.DBFXE ".DBF"§) für Tabellen wie dBASE. Daher brauchen Sie Ihre dBASE-Daten nicht zu konvertieren, Sie können sie sofort in FoxPro laden und bearbeiten. Dies gilt auch für mit dBASE III PLUS erstellte Datenbanken. Verschlüsselte Dateien müssen jedoch zuerst entschlüsselt werden, bevor FoxPro sie lesen kann.

Sie brauchen nur im Befehlsfenster den Befehl USE <Tabellenname> einzugeben oder im Menü **Datei** die Option **Öffnen** aufzurufen und können dann Ihre dBASE-Tabelle in

FoxPro anschauen und analysieren.

Memodateien

Das XE "Memodateien"§FoxPro-Dateiformat für Memofelder erlaubt Ihnen, in einem Memofeld eine unbegrenzte Menge von Daten zu speichern. (Natürlich sind Ihnen durch die Festplattengröße bestimmte Grenzen gesetzt.) Wenn Sie eine dBASE IV-Tabelle, die mit einer Memodatei verbunden ist, in FoxPro öffnen, müssen Sie angeben, ob die Memodatei konvertiert werden soll. Falls Sie dies bejahen, konvertiert FoxPro die Memodatei zum FoxPro-Format (.FPT) und löscht die ursprüngliche dBASE-Datei (.DBTXE ".DBT"§). (Beachten Sie, daß die Datei für exklusive Nutzung geöffnet werden muß, damit sie konvertieren werden kann. In FoxPro ist die Option EXKLUSIV voreingestellt.)

Sie müssen die Rückfrage sogar bejahen, da FoxPro eine Tabelle, die mit einem dBASE-Memofeld verknüpft ist, erst dann liest, wenn Sie das Memofeld zum .FPT-Format konvertieren lassen.

Sobald ein Memofeld zum FoxPro-Format konvertiert wurde, kann dBASE die .DBF-Datei, mit der das Memofeld verknüpft war, nicht mehr lesen. Allerdings kann FoxPro eine .DBF-Datei mit einem verknüpften FoxPro-Memofeld problemlos in eine .DBF-Datei und eine .DBT-Memodatei, die für dBASE IV lesbar sind, verwandeln. Verwenden Sie dazu den Befehl COPY TO <Tabellenname> TYPE FOXPLUS.

IndexdateienXE "Index"§

FoxPro verwendet ein sehr effizientes Indizierungsschema, das sich sowohl durch kürzere Zugriffszeiten als auch durch Indexfelder, die nur halb oder ein Drittel so groß wie dBASE-Indizes sind, auszeichnet. FoxPro erstellt automatisch anhand der .MDXXE ".MDX"§-Indexeinträge einen .CDX-Index mit gleicher Struktur. Die dBASE-Indizes werden bei der Konvertierung nicht verändert. Wenn Sie mit der Tabelle

arbeiten, wird der im FoxPro-Format strukturierte .CDX-Index ebenso automatisch geöffnet wie eine .MDX-Index-Datei. FoxPro konvertiert .NDXXE ".NDX"§-Indexdateien nicht automatisch. Mit dem Befehl USE und der Angabe des Namens der .NDX-Datei oder dem Befehl SET INDEX TO können Sie jedoch einen .NDX-Index öffnen und damit FoxPro veranlassen, die genannten Indexdateien zu konvertieren. .NDX-Indexdateien werden in FoxPro zu .IDX-Indexdateien konvertiert.

dBASE-Abfragen "Abfragen"§

dBASE-Abfragen sind Dateien mit der Erweiterung .QBEXE ".QBE"§, die die dBASE IV-Befehle enthalten, die für die Durchführung der Abfrage erforderlich sind. Nach geringfügigen Änderungen können Sie diese Dateien in FoxPro als Programm ausführen lassen und erhalten dann dieselben Ergebnisse wie in dBASE. Sie können eine dBASE-Abfrage allerdings nicht mit Hilfe der Abfragefunktion von FoxPro (RQBE) modifizieren. Sie müssen die Befehle, wenn nötig, direkt in der .QBE-Datei verändern. Abfragen, die mit der RQBE-Funktion in FoxPro erstellt wurden, werden in Form von SQL-Befehlen gespeichert, sie sind dadurch kürzer und schneller auszuführen.

Bevor Sie eine .QBE-Datei modifizieren, sollten Sie die Datei kopieren, damit Sie sie weiterhin in dBASE IV ausführen können.

Folgende Arbeitsschritte sind auszuführen, um eine .QBE-Datei in ein in FoxPro ausführbares Programm umzuwandeln:

1. Wählen Sie im FoxPro-Menü Datei die Option Öffnen.
2. Geben Sie im Feld Dateiformat den Dateityp Programm an.
3. Markieren Sie das Kontrollkästchen Alle Dateien.
4. Öffnen Sie eine dBASE-Abfrage (.QBE-Datei).
5. Blättern Sie im Textfenster solange, bis das erste Zeichen erscheint, das kein Textzeichen darstellt.
6. Löschen Sie alle Sonderzeichen am Ende der Datei.
7. Schließen Sie die Datei, und speichern Sie die Änderungen.

Sie können die Abfrage nun mit Hilfe des Befehls DO ausführen lassen. Vergessen Sie nicht, die Erweiterung .QBE anzugeben. dBASE IV-Abfragen geben am Ende den Befehl SET EXACT ON und nicht den Befehl SET EXACT OFF aus. Wenn Sie die von der Abfrage erzeugte Datenbankumgebung verlassen, müssen Sie daher den Status von SET EXACT wiederherstellen.

Abfragen, in denen Tabellen geöffnet und verbunden, in denen Auswahlkriterien angegeben werden oder in denen eine Sortierfolge definiert wird, werden in FoxPro einwandfrei und reibungslos ausgeführt. In Abfragen mit Statistikfunktionen sind unter Umständen Bedingungsklauseln und Berechnungsschleifen enthalten, die in FoxPro Probleme verursachen können. Sie können diese Art von Abfragen sehr schnell mit der RQBE-Funktion in FoxPro neu erstellen und gewinnen damit überdies den Vorteil, sie später verändern zu können.

µ §

Die hervorgehobenen Zeichen sollten gelöscht werden.

Folgende Inkompatibilitäten können in .QBE-Dateien auftreten:

- SET FIELDSXE "SET FIELDS"§ Befehl
- TAGNOXE "TAGNO()"§ Funktion
- NOSAVEXE "NOSAVE"§ Option im Befehl USE (nur bei Version 1.5)
- SET CATALOGXE "SET CATALOG"§ TO Befehl
- SET("CATALOG")XE "SET(\"CATALOG\")"§ Funktion

Der Befehl SET FIELDS wird in FoxPro anders als in dBASE interpretiert. Nähere Angaben zu SET FIELDS finden Sie im Abschnitt "Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV".

FoxPro unterstützt die Funktion TAGNO an sich nicht, allerdings enthält dieser Programm-Konverter eine benutzerdefinierte Funktion, die dieselbe Funktion erfüllt. Sofern das Abfrageprogramm auf diese benutzerdefinierte Funktion zugreifen kann, wird die Funktion TAGNO problemlos ausgeführt. Nähere Informationen hierzu finden Sie im Abschnitt "Benutzerdefinierte Funktionen (UDFs) aufrufen."

In FoxPro ist die Option NOSAVE für den Befehl USE nicht verfügbar. Sie können die Dateien, die während der Ausführung der Abfrage erstellt werden, am Ende der Abfrage löschen, indem Sie die Namen der temporären Dateien speichern und sie durch Einfügen eines Befehls am Ende des Abfrageprogramms löschen lassen. Das folgende Beispiel erläutert, wie Sie dabei vorgehen:

```
temp_datei=SYS(3)+".dbf"  
&&Erstelle eindeutigen Dateinamen mit der  
&&Erweiterung .DBF  
<<Programmcode>>  
delete file (temp_datei)
```

Den Programm-Konverter starten

Dieser Programm-Konverter setzt sich aus zwei in FoxPro geschriebenen Programmen zusammen, dem Datei-Konverter und dem Analyse-Programm. Zur Installation dieser Programme, erstellen Sie zuerst in Ihrem FoxPro-Verzeichnis ein Unterverzeichnis namens MIGRATE. Kopieren Sie dann alle Dateien von den Programm-Konverter-Disketten in das Unterverzeichnis MIGRATE.

Sie starten den Programm-Konverters, indem Sie im Menü **Programm** den Befehl **Ausführen** aufrufen. Nach Aufruf dieses Befehls wird das Dialogfenster Programm ausführen angezeigt. Stellen Sie hier im Listenfeld **Verzeichnis** das Unterverzeichnis MIGRATE ein und im Feld **Programm ausführen** die Datei MIGRATE.APP. Sie können stattdessen auch einfach den Befehl DO MIGRATE.APP im Befehlsfenster eingeben. Stellen Sie sicher, daß das Verzeichnis, das die Datei MIGRATE.APP enthält, als Standardverzeichnis eingestellt ist. Nachdem der Befehl DO MIGRATE.APP aufgeführt wurde, wird ein neues Menü namens **Migration Tools** in die Hauptmenüzeile von FoxPro eingefügt.

Über das Menü **Migration Tools** können alle Funktionen des Programm-Konverters, sowie der Datei-Konverter und das

Analyse-Programm, aufgerufen werden. Solange Sie mit dem Programm-Konverter arbeiten, sind einige Menüs nicht verfügbar. Wenn Sie das Menü **Migration Tools** entfernen und die anderen Menüs wieder reaktivieren wollen, rufen Sie im Menü **Migration Tools** den Befehl **Close Tools** auf.

μ §Mit dem Befehl DO MIGRATE.APP wird der Programm-Konverter gestartet und das Menü Migration Tools in die Hauptmenüzeile von FoxPro eingefügt.

Masken, Berichte und Etiketten konvertieren

Der Datei-KonverterXE "Datei Konverter"§

Zur Konvertierung von Masken (.SCRXE ".SCR"§), Formatdateien (wie auch PRG-Dateien mit @SAY und GET Befehlen), Berichten (.FRMXE ".FRM"§) und Etiketten (.LBLXE ".LBL"§) gehen Sie, unabhängig vom Dateityp, immer auf die gleiche Weise vor. Wählen Sie im Menü **Migration Tools** den Befehl **Convert Files**. Daraufhin wird ein Dialogfenster angezeigt, das eine Liste aller im aktuellen Verzeichnis gespeicherten Dateien mit den Erweiterungen .SCR, .FMT, .PRG, .FRM und .LBL enthält. Es werden auch Dateien mit der Erweiterung .NTX aufgeführt. Dabei handelt es sich um Clipper-Indexdateien, auf diese Dateien werden wir an späterer Stelle eingehen.

Sie markieren die zu konvertierenden Dateien, indem Sie entweder auf dem Dateinamen doppelklicken oder den Dateinamen hervorheben und die Eingabetaste drücken. Die zur Konvertierung markierten Dateien werden durch einen Stern links neben dem Dateinamen (bzw. ein Häkchen in FoxPro 2.5 für MS-DOS) gekennzeichnet. Falls alle Dateien aus einem Verzeichnis konvertiert werden sollen, klicken Sie auf der Schaltfläche **Select All**, Sie markieren damit alle Dateien. Durch Klicken auf der Schaltfläche **Clear All** können Sie die Markierung sämtlicher Dateien aufheben. Wenn Sie die Markierung nur von einer Datei entfernen wollen, doppelklicken

Sie auf den Dateinamen, oder heben Sie den Dateinamen hervor. und drücken Sie die Eingabetaste.

µ §

Im DialogfensterFile Converter geben Sie an, welche Masken, Berichte und Etiketten konvertiert werden sollen.

Der Datei-Konverter kann die Dateitypen .SCR, .FRM und .LBL parallel in einem Konvertierlauf bearbeiten. Wenn Sie eine in einem anderen Verzeichnis gespeicherte Dateien konvertieren wollen, klicken Sie auf der Schaltfläche **Directory**, und wechseln Sie zu einem anderen Verzeichnis. Zur Konvertierung von Dateien aus verschiedenen Verzeichnissen, müssen Sie die Verzeichnisse getrennt bearbeiten, d.h. Sie markieren und konvertieren erst die Dateien in einem Verzeichnis und danach markieren und konvertieren Sie die Dateien des anderen Verzeichnisses.

Nachdem Sie alle Dateien markiert haben, klicken Sie auf der Schaltfläche **Process**. Der Datei-Konverter von FoxPro beginnt dann mit der Arbeit. Masken und Dateien, die Anweisungen mit @SAY und GET enthalten (Dateien mit der Erweiterung .SCR, .FMT und .PRG), werden zu Dateien mit dem FoxPro-Format .SCX konvertiert. Berichtsdateien (.FRM) werden zum FoxPro-Format .FRX und Etiketten (.LBL) zum FoxPro-Format .LBX konvertiert. Diese neu erstellten Dateien werden im selben Verzeichnis wie die ursprünglichen Dateien (die nicht verändert werden) gespeichert.

Wenn Sie diese Masken, Berichte und Etiketten in FoxPro verändern wollen, wählen Sie einfach den Befehl **Öffnen** aus dem FoxPro-Menü **Datei**. Geben Sie den Dateityp und den Namen der Datei, die Sie bearbeiten wollen, an. FoxPro lädt daraufhin automatisch das entsprechende Dienstprogramm. Nähere Information zu den Funktionen und Power Tools von FoxPro finden Sie im *Benutzerhandbuch*, das Teil der Dokumentation zu FoxPro 2.5 ist.

MaskendateienXE "Masken"§

SCR-Dateien konvertieren

Der Datei-KonverterXE "Datei-Konverter"§ kann folgende Elemente von Maskendateien ins FoxPro-Format übertragen: @SAY und GET Befehle, PICTURE-Klauseln, WHEN- und VALID-Klauseln, die angeben, ob ein Feld editiert werden kann oder nicht, ERROR-Meldungen, PROMPT -Meldungen, RANGE-Werte, Vorgabewerte, Auswahllisten, Rahmen und Linien. Die Farbattribute einzelner Felder und andere Formatangaben gehen bei der Konvertierung verloren, alle übrigen Farbattribute werden jedoch in das FoxPro-Format übertragen.

Die konvertierten Maskendateien werden im Format von FoxPro 2.5 für MS-DOS gespeichert. Falls Sie solch eine Datei in FoxPro für Windows laden, fragt FoxPro, ob Sie diese Maske nach Windows portieren wollen. In den meisten Fällen ist es ratsam, diese Frage zu bejahen, da die Maske der Windows-Oberfläche angepaßt sein und nicht wie eine MS-DOS-Maske aussehen soll. Im *Entwicklerhandbuch* der Dokumentation zu FoxPro 2.5 wird näher erläutert, wie Sie mit FoxPro Daten und Programme von MS-DOS nach WINDOWS und umgekehrt portieren können.

SCR-Dateien mit mehrseitigen Masken konvertieren

Der Datei-Konverter konvertiert auch mehrseitige dBASE-Masken, diese müssen jedoch in einzelne einseitige Masken umgewandelt werden. Die einzelnen Masken sollten zwar in einer Maskengruppe zusammengefaßt, aber getrennt gespeichert werden. Sie können innerhalb einer Maskengruppe mit der Taste BILD-AB zur nächsten Maske der Gruppe blättern. Zum Aufspalten der mehrseitige Maske laden Sie die konvertierte Datei. Speichern Sie eine Seite der Maske unter einem neuen Dateinamen ab - vor dem Abspeichern einer Seite müssen Sie natürlich die restlichen Seiten löschen. Laden Sie die konvertierte SCR-Datei erneut und wiederholen Sie diese Arbeitsschritte solange bis alle Seiten in einzelnen Dateien gespeichert sind.

FMT- und PRG-Dateien konvertieren

Damit Sie alle Funktionen von FoxPro ausnutzen können, sollten Sie auch Formatdateien und PRG-Dateien, die @SAY und GET Befehle enthalten, zu FoxPro-Maskendateien konvertieren. Der Datei-Konverter kann Programm- und Formatdateien bearbeiten und daraus SCX-Dateien erstellen. Beim Konvertieren von @SAY Befehlen werden Klauseln zu den Optionen PICTURE, FUNCTION, RANGE, VALID, WHEN, COLOR und MESSAGE übertragen.

Während der Konvertierung einer FMT- oder PRG-Datei wird folgendes Dialogfenster eingeblendet:

µ §

Im Dialogfenster wird angezeigt, welche Datei gerade bearbeitet wird (hier INVOICE.SCX). Die konvertierte Datei erhält nur eine andere Erweiterung (.SCX), ansonsten wird der alte Dateiname beibehalten. Das Dialogfenster ermöglicht Ihnen, die SCX-Datei mit einer Tabelle zu verknüpfen. Wenn Sie eine Tabelle mit der SCX-Datei verknüpfen, öffnet und schließt FoxPro diese Tabelle automatisch, sobald Sie die Maske laden oder modifizieren -- eine sehr bequeme Sache.

Um eine Maske mit einer .SCX-Datei zu verknüpfen, klicken Sie auf der Schaltfläche **Choose**. Daraufhin wird das Dialogfenster Öffnen angezeigt, in dem Sie aus einem beliebigen Verzeichnis Ihrer Festplatte oder einem Netzverzeichnis eine Tabelle auswählen können. Wenn Sie danach auf die Schaltfläche **OK** klicken, wird die neue Datei erstellt. Beachten Sie, daß die Schaltfläche **OK** solange deaktiviert ist, bis Sie den Namen einer Tabelle gewählt haben. Sie können allerdings mit der Schaltfläche **Cancel** oder der Taste ESC den Vorgang abbrechen und auf diese Weise verhindern, daß eine bestimmte Datei konvertiert wird.

Falls Sie die Maskendatei nicht mit einer Tabelle verknüpfen wollen, wählen Sie die Schaltfläche **None**. Die neue Datei wird daraufhin ohne Verknüpfungen erstellt.

Wie SCR-Dateien werden auch FMT- und PRG-Dateien zum Dateiformat von FoxPro 2.5 für DOS konvertiert. Wenn Sie

diese Dateien in FoxPro für Windows laden, wird ein Dialogfenster eingeblendet, in dem Sie angeben müssen, ob die konvertierte Datei ins Windows-Format portiert werden soll. Normalerweise ist dies sinnvoll, wählen Sie in diesem Fall die Schaltfläche "Portieren und Öffnen".

Bevor Sie Formatdateien mit mehrseitigen Masken konvertieren können, müssen Sie die Datei in mehrere Dateien mit jeweils einer einseitigen Maske aufteilen, da der Datei-Konverter den Konvertiervorgang abbricht, sobald er auf eine READ-Anweisung trifft.

FormatdateienXE "Formatdateien"§ modifizieren, die nicht konvertiert wurden

FoxPro unterstützt dBASE-Formatdateien (.FMTXE ".FMT"§) und einschlägige Befehle, wie zum Beispiel SET FORMAT. In manchen mit dBASE IV erstellten Formatdateien werden jedoch auch Befehle zur Einstellung und Rücksetzung von Systemvariablen verwendet. Formatdateien, die nicht nur @SAY- und @GET-Ausdrücke enthalten, können in FoxPro nicht bearbeitet werden. Falls Sie Ihre Formatdateien nicht zum FoxPro-Format konvertieren wollen, müssen Sie die Befehle zur Einstellung und Rücksetzung von Systemvariablen aus der Formatdatei entfernen und an anderer Stelle einfügen. Beispielsweise können Befehle zur Einstellung von Systemvariablen vor und Befehle zur Rücksetzung der Systemvariablen hinter dem Befehl SET FORMAT plaziert werden.

dBASE-BerichteXE "Berichte"§

In dBASE erstellte Berichte können auf zweierlei Weise in FoxPro verwendet werden. Sie können die .FRG-Datei ausführen lassen oder die .FRMXE ".FRM"§-Datei konvertieren. Falls Sie ausschließlich mit FoxPro 2.5 für MS-DOS arbeiten und den Bericht auch nicht verändern wollen, ist die Ausführung der .FRG-Datei wohl das einfachste und beste Verfahren.

Wenn Sie jedoch auch FoxPro 2.5 für Windows einsetzen oder den Bericht verändern wollen, sollten Sie die .FRM-Datei

konvertieren.

Hinweise zur Konvertierung von Berichten

Alle Bereichsdefinitionen, Felder, berechnete Felder, verborgene Felder, Schablonen und Formelfelder werden konvertiert. Dasselbe gilt für Formatattribute wie unterstrichen, fett, kursiv, und Farbeinstellungen. Schriften werden nicht konvertiert. FoxPro für Windows unterstützt jedoch alle Windows-Schriften, auch die True Type Schriften, und damit stehen Ihnen mehr Schrifttypen zur Verfügung als in dBASE. In FoxPro für MS-DOS können Schriften über die Definition von Berichtsvariablen verändert werden.

Bereiche, die automatisch umbrochene Textzeilen enthalten, werden in FoxPro-Berichten zu der entsprechenden Anzahl von einzeiligen Textfeldern konvertiert. Die Bereichseinstellungen für Zeilenabstand und Schriftdicke werden nicht konvertiert. In FoxPro werden weder die Funktion Zeilenlineal noch die Funktion Automatischer Absatzeinzug unterstützt, diese Formatdaten gehen daher bei der Konvertierung verloren. Bei konvertierten Berichten entspricht die Einstellung für den rechten Rand der Breite des Berichts. Die Einstellung für den linken Rand wird nicht konvertiert, kann aber mit dem Berichtsgenerator von FoxPro gesetzt werden.

Bei Berichten, in denen Felder mit Tabulatoren ausgerichtet wurden, müssen die Felder in der Regel neu angeordnet werden. In dBASE kann der Seitenkopf (in dBASE-Terminologie: Kopfzeile) innerhalb des Berichtskopfs (in dBASE-Terminologie: Vorspannbereich) ausgegeben werden. In FoxPro wird bei konvertierten Berichten, die über Daten im Berichtskopf und Berichtsfuß verfügen, der Berichtskopf nach dem Seitenkopf und der Zusammenfassungsbereich vor dem Seitenfuß (in dBASE-Terminologie: Fußzeile) angezeigt. Im Ausdruck wird jedoch der Berichtskopf vor den Seitenkopf und der Seitenfuß vor den Zusammenfassungsbereich gestellt.

In dBASE werden auch leere Berichtsbereiche ausgedruckt. Der Datei-Konverter fügt Leerzeichen in diese Bereiche ein, damit sie auch von FoxPro ausgedruckt werden.

In dBASE-Berichten werden nicht sämtliche Umgebungsinformationen gespeichert, sondern nur die Aliasnamen für die Verknüpfungen zur Tabelle. Infolgedessen erhalten Sie unter Umständen beim Öffnen eines konvertierten Berichts eine Fehlermeldung wie "Kunden.dbf nicht gefunden." In FoxPro-Berichten können Tabellennamen und Informationen über Tabellenverknüpfungen, Feldauswahl und Indizes gespeichert werden. Mit Hilfe des Reportgenerators können Sie die gewünschte Umgebung einstellen und diese Daten zusammen mit dem Bericht abspeichern. Danach werden Sie keine Fehlermeldungen dieser Art mehr erhalten.

Beachten Sie, daß FoxPro nur im Detailbereich dehnbare Felder zur Aufnahme von großen Datenmengen zuläßt. Falls Sie in Ihrem Bericht auch außerhalb des Detailbereichs dehnbare Felder verwenden, sollten Sie die .FRG-Datei einfach ausführen lassen.

Berichtsprogramme (.FRGXE ".FRG"§) ausführen

Falls Sie nicht vorhaben, den Bereich zu verändern, ist es am einfachsten und oft am sinnvollsten, die .FRG-Datei in FoxPro ausführen zu lassen. In FoxPro 2.5 für MS-DOS führen Sie mit dem Befehl REPORT FORM eine .FRG-Datei aus, Sie geben dazu beispielsweise den Befehl REPORT FORM meinrep.frg ein. FoxPro kompiliert die .FRG-Datei in eine .F4X-Datei und führt den Bericht aus. In FoxPro für Windows verwenden Sie den DO Befehl in der Form: DO meinrep.frg. Der Bericht wird in eine .FXP-Datei kompiliert und dann ausgeführt.

Wenn Sie Berichte mit dem Befehl DO ausführen, sollten Sie beachten, daß die Zusatzinformationen, die mit dem Befehl REPORT FORM gegeben werden, dem .FRG-Programm als Parameter übergeben werden können. Etikettenprogramme erfordern keine dieser Parameter (und akzeptieren sie auch nicht). Die ersten drei Parameter sind logische Variablen, die den Wahrheitswert .T. haben, falls der entsprechende REPORT FORM Befehl die Schlüsselwörter NOEJECT, PLAIN oder SUMMARY enthält. Der vierte Parameter ist eine Zeichenfolge, die den Text einer zusätzlichen Titelzeile enthält, und der fünfte

Parameter ist eine Zeichenfolge, die in dBASE IV nicht verwendet wird (Sie können diesen Parameter nach Bedarf in Ihrem Bericht oder in benutzerdefinierten Funktionen, die vom Berichtsprogramm aufgerufen werden, verwenden).

Die beiden Befehle im folgenden Beispiel sind gleichbedeutend:

```
REPORT FORM Namelist TO PRINT NOEJECT SUMMARY HEADING  
"Rechnungen"
```

```
DO Namelist.frg WITH .T., .F., .T., "Rechnungen"
```

Beachten Sie auch, daß die Befehle REPORT FORM und LABEL FORM einen Suchausdruck an einen LOCATE Befehl weiterleiten, um die Datensätze, die bestimmte FOR-Klauseln oder Filterbedingungen erfüllen, zu ermitteln. Da die von dBASE generierte DO WHILE Schleife einen CONTINUE Befehl enthält, *müssen* Sie sogar einen LOCATE Befehl einsetzen, bevor Sie ein Berichts- oder Etikettenprogramm mit einem DO Befehl ausführen.

Druckertreiber

XE "Druckertreiber"§Die meisten Probleme, die bei der Ausführung von dBASE-Berichtsdateien (.FRGs) auftreten, haben etwas mit Druckertreibern zu tun. In dBASE IV sind in Druckertreibern alle Informationen enthalten, die zum Drucken der Standardformate auf einem bestimmten Drucker erforderlich sind (wie die Definitionen für fette, kursive, unterstrichene, hochgestellte und tiefgestellte Schriftzeichen) und darüber hinaus sind fünf dBASE-eigene Schriften in der Datei CONFIG.DB definiert.

Wenn eine Anwendung in dBASE IV mehr als fünf verschiedene Schriften benötigt, kann ein zweiter Druckertreiber für den gewählten Drucker definiert werden. Die Ausführung eines Berichts und die Darstellung von Berichtsteilen stützt sich vor allem darauf, daß Systemvariablen die richtigen Werte zugeordnet werden. Der Name des aktuellen Druckertreibers wird in der Systemvariablen _PDRIVER gespeichert. Sie können in einem Druckformat die Werte der verschiedenen Systemvariablen festlegen (_PADVANCE, _PAGENO, _PBPAGE, _PCOPIES, _PDRIVER, _PECODE, _PEJECT,

`_PEPAGE`, `_PLENGTH`, `_PLOFFSET`, `_PPITCH`,
`_PQUALITY`, `_PSCODE`, `_PSPACING` und `_PWAIT`).

In FoxPro für MS-DOS verwenden Sie den Befehl `Printer driver setup`, um Daten zum gewählten Drucker und einige Einstellungen, die den in dBASE IV-Druckformaten enthaltenen Angaben zum Druckauftrag ähneln, festzulegen. Die Systemvariable `_PDRIVER` speichert den Namen des aktuellen Druckersteuerungsprogramms, und die Variable `_PDSETUP` enthält den Namen der aktuellen Druckertreibereinstellung. Bevor Sie ein dBASE-Programm in FoxPro ausführen können, müssen Sie alle Verweise auf die Systemvariable `_PDRIVER` daraus löschen, da FoxPro sonst den Name des dBASE IV-Druckertreibers für den Namen eines neuen Druckersteuerungsprogramms hält. In den meisten Fällen können Sie die Verweise durch einen Befehl ersetzen, der einen Wert, mit dem der aktuelle Drucker und die allgemeinen Druckparameter eingestellt werden, an `_PDSETUP` übergibt.

Wenn Sie die richtigen Druckertreibereinstellungen ermittelt haben, druckt FoxPro die fünf Standardschriftattribute. Sie müssen möglicherweise weitere Änderungen vornehmen, damit der Bericht genauso wie in dBASE IV ausgegeben wird.

In FoxPro für Windows wird die Druckersteuerung von Windows gehandhabt, und Berichte werden an den Drucker, der mit der Option **Druckereinrichtung** im Menü **Datei** eingestellt wurde, weitergeleitet. FoxPro für Windows ignoriert alle Verweise auf die Systemvariablen `_PDRIVER` und `_PDSETUP`, daher erzeugen Befehle, die `_PDRIVER` einen neuen Wert zuordnen, keinen Fehler. Allerdings werden auch Formatänderungen, die in dBASE IV durch den Wechsel von Druckertreibern erreicht werden, wie beispielsweise ein Wechsel zwischen Hoch- und Querformat, ignoriert.

Wie die Standardschriftattribute werden auch die dBASE-Schriften durch `STYLE`-Klauseln in den `?` und `??` Befehlen, die den Bericht bilden, implementiert. Da FoxPro für MS-DOS alle ihm unbekanntenen Angaben in den `STYLE`-Klauseln ignoriert, erzeugen die in dBASE IV definierten Schriften zwar keine

Fehler, sie werden allerdings auch nicht gedruckt. Sie können diese Schriften, die in der Konfigurationsdatei dBASE IV definiert sind und auf die in Berichts- und Etikettenvorlagen verwiesen wird, auf verschiedene Weise in FoxPro generieren. Eine Möglichkeit ist, das Druckertreiberprogramm zu modifizieren. Dies ist jedoch sehr viel komplizierter als entsprechende Befehle in das Berichts- oder Etikettenprogramm einzufügen. Am einfachsten implementieren Sie diese Schriften, indem Sie die Escape-Sequenzen zur Initialisierung und zum Zurücksetzen der Schriften (Sie finden diese in der Datei dBASE IV CONFIG.DB) in Speichervariablen speichern und ??? Befehle in Berichts- oder Etikettenprogramme einfügen, mit denen diese Steuerzeichen vor und nach dem Ausdruck der betreffenden Daten an den Drucker geschickt werden. Sie machen diese Druckformatsteuerzeichen am einfachsten ausfindig, indem Sie die .FRG- oder .LBG-Programm editieren und nach den STYLE-Klauseln suchen.

In FoxPro für Windows werden Schriftart und Schriftattribute von Ausdrucken durch eine FONT-Klausel in einem ? oder ?? Befehl definiert. FoxPro für Windows ignoriert STYLE-Klauseln; nur wenn Sie zusätzlich in einer FONT-Klausel eine bestimmte Schrift festlegen, werden diese gedruckt. Falls Sie nicht allzu viele verschiedene Schriftattribute verwenden, können Sie einfach die Datei nach STYLE-Klauseln durchsuchen lassen und die erforderlichen FONT-Klauseln einfügen. Sie können mit FONT-Klauseln auch die Schriften emulieren, die in dBASE IV in der Datei CONFIG.DB definiert sind. Da FoxPro für Windows die mit der Systemvariablen _PLENGTH festgelegte Seitenlänge *in Zeilen* berechnet, müssen Sie unter Umständen verschiedene Schriftgrößen und Schriftarten ausprobieren, um einen korrekten Seitenumbruch zu erhalten. Im Gegensatz zu FoxPro für MS-DOS werden jedoch nicht immer sechs Zeilen pro Zoll ausgedruckt, in Windows wirkt sich eine Veränderung der Schriftgröße sowohl auf die Höhe als auch die Breite der Zeichen aus. Für gewöhnlich passen bei einem Bericht, in dem gemäß Standardeinstellung 66

Zeilen auf eine 11 Zoll lange Seite gedruckt werden sollen, mehr als 66 Zeilen auf eine Papierseite.

Für einige Einstellungen, die in dBASE IV mit Systemvariablen vorgenommen werden, insbesondere für Zeichendichte (_PPITCH) und Druckqualität (_PQUALITY), sind in FoxPro andere Druckereinstellungen nötig. Sofern das dBASE IV-Programm diese Einstellungen für den gesamten Bericht definiert, können Sie in FoxPro mit der Option Druckereinrichtung die entsprechenden Einstellungen treffen. Falls diese Einstellungen innerhalb eines Berichtsprogramms verändert werden, müssen Sie das Programm editieren und die Druckertreibereinstellungen verändern oder ??? Befehle zum Initialisieren und Rücksetzen der Druckmodi an den betreffenden Stellen einfügen. FoxPro ignoriert überdies die Systemvariable _PWAIT, die festlegt, ob der Drucker nach dem Drucken jeder Seite anhält. Wenn diese Druckpause in FoxPro ausgeführt werden soll, müssen Sie eine benutzerdefinierte Funktion schreiben, die vom Berichtskopf aus aufgerufen wird.

Etiketten

XE "Etiketten"§Wie bei Berichten können Sie zwischen zwei Vorgehensweisen wählen: Sie können die Etikettenprogramme (.LBG) in FoxPro ausführen oder die Etikettendatei (.LBLXE ".LBL"§) mit dem Datei-Konverter zum FoxPro-Format (.LBX) konvertieren.

Hinweise zur Konvertierung

Alle Felder und Textteile werden konvertiert. Falls sich in einem Etikett mehrere Felder in einer Zeile befinden, werden die Felddefinitionen durch einen Ausdruck, der die einzelnen Feldnamen zusammenfaßt, ersetzt. Beachten Sie auch, daß dBASE ein Feld zentriert, indem es links vom Feldwert Leerzeichen einfügt. Der Datei-Konverter behandelt diese Leerzeichen wie Text und setzt sie in Anführungszeichen. Desweiteren ist zu beachten, daß die Anweisungen zu Schriftattributen (Unterstreichung, fett, kursiv, hochgestellt und tiefgestellt) bei der Dateikonvertierung verloren gehen und in

FoxPro neu erstellt werden müssen.

Etikettenprogramme ausführen (.LBGXE ".LBG"§)

Für die Ausführung von .LBG-Dateien gelten dieselben Hinweise und Regeln wie für .FRG-Dateien. Im allgemeinen sollten diese Programm problemlos ausgeführt werden. Ein wichtiger Unterschied ist jedoch zu beachten. In mit dBASE IV Version 1.5 erstellten Programmen wird die Funktion ISBLANK sehr häufig verwendet, um die Ausgabe von Leerzeilen, die sich durch leere Felder ergeben, zu unterdrücken. Sie lösen dieses Problem, indem Sie alle Vorkommen dieser Funktion durch die fast äquivalente Funktion EMPTY ersetzen.

Inkompabilitäten zur dBASE-Sprache

Überblick

Von den mehreren Hundert Befehlen und Funktionen von dBASE IV verhält sich die überwiegende Mehrzahl in dBASE genauso wie in FoxPro. Es wurde ein extrem hoher Grad an Kompatibilität erreicht, insbesondere gegenüber dBASE IV 1.0 und 1.1. Trotz allem gibt es Bereiche, in denen sich dBASE und FoxPro unterscheiden.

Wenn Sie dBASE-Programme nach FoxPro portieren wollen, ermitteln Sie zuerst mit Hilfe des Analyse-Programms alle inkompatiblen Befehle und Funktionen. Lesen Sie dazu bitte den Abschnitt "Das Analyse-Programm." Nachdem das Analyse-Programm die Problemfälle in einer Tabelle erfaßt hat, können Sie mit der Bearbeitung der einzelnen Probleme beginnen. Sobald Sie alle Probleme bereinigt haben, können Sie versuchen, die Anwendung in FoxPro ausführen zu lassen.

Alle Problemfälle, die vom Analyse-Programm angezeigt werden, sind im Abschnitt "Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV" dokumentiert. Dort werden die Unterschiede in der Interpretation der einzelnen Befehle und Funktionen beschrieben, und es wird dargestellt, wie Sie

Programme verändert können, damit sie in FoxPro das erwartete Verhalten zeigen.

Hinweise zu Programmgenerator-XE "Programmgenerator"§Anwendungen aus dBASE IV

Anwendungen, die vom dBASE IV-Programmgenerator erstellt werden, setzen sich aus vielen verschiedenen Dateien zusammen, die die einzelnen einzelnen Anwendungskomponenten enthalten. Es gibt Dateien für jedes Zeilenmenü, für jedes Popup-Menü, für Dateilisten, für Stapelverarbeitungs-Routinen und so weiter, jedem dieser Dateitypen ist eine eindeutige Erweiterung (.BAR, .POP, .FIL, .VAL, usw.) zugeordnet. Diese Dateien werden nur gebraucht, wenn die Anwendung in dBASE IV verändert wird. Sie müssen nicht konvertiert werden.

Sie sollten die Hauptdatei der Anwendung, die Datei mit der Erweiterung .APP, löschen oder zumindest in ein anderes Verzeichnis verlagern, da FoxPro sonst versucht, diese Datei anstelle der gleichnamigen .PRG-Datei auszuführen (sofern Sie nicht die Erweiterung .PRG angeben). Die Erweiterung .APP wird in FoxPro für Dateien verwendet, die den .DBO-Dateien in dBASE ähneln.

Mit dem Programmgenerator erstellte Anwendungen bestehen aus zwei .PRG-Dateien: einem Initialisierungsprogramm, das die Anwendung startet und die Arbeitsumgebung einrichtet, und einem umfangreichen Programm, das den gleichen Namen trägt wie das Hauptmenü und das die Prozeduren, mit denen die einzelnen Menüoptionen ausgeführt werden, enthält. Das Initialisierungsprogramm hat denselben Namen wie Hauptobjekt der Anwendung. Sie führen solche Anwendungen in FoxPro aus, indem Sie mit einem DO Befehl das Initialisierungsprogramm aufrufen.

Hauptaspekte in Sachen Kompatibilität

Auch wenn das Analyse-Programm in einer Anwendungen viel Problemfälle anzeigt, können die allermeisten Probleme oft durch die Veränderung einer einzigen Programmzeile gelöst werden, so daß Ihre Anwendungen ohne Schwierigkeiten in FoxPro ausgeführt werden. Sie werden bald herausfinden, daß einige wenige Probleme sehr häufig auftreten und daß Sie bei der Fehlerkorrektur meist dieselbe Art von Änderungen durchführen.

Die einzigen Bereiche, in denen signifikante Kompatibilitätsprobleme auftreten, sind Datenschutzroutinen, Netzwerkfunktionen und die weniger häufig verwendeten SQL- und Transaktionsbefehle.

Die meisten SQL-XE " SQL"§Programme (.PRSXE ".PRS"§-Dateien) müssen in FoxPro neu geschrieben werden. FoxPro unterstützt selbst auch weder Datenschutzroutinen noch Transaktionen. Falls Sie FoxPro 2.5 für MS-DOS und Novell Netware® 2.x oder neuer besitzen, können Sie mit den Funktionen von Novell Transaction Tracking Services® arbeiten. Darüber hinaus bieten viele Software-Hersteller umfangreiche Bibliotheken mit Datenschutz- und Netzwerkfunktionen an, solche Bibliotheken bieten meist einen viel größeren Funktionsumfang als die Routinen in dBASE. In Anhang E finden Sie eine Zusammenstellung von Produkten und von Funktionen, die von diesen Produkten unterstützt werden.

Problemlösungsverfahren

Nachdem das Analyse-Programm die potentiellen Problembereiche in Ihrem Programm ermittelt hat, können Sie eines der folgenden sechs Problemlösungsverfahren einsetzen. Sie können

- die Befehle bzw. Funktionen durch äquivalente FoxPro Befehle oder Funktionen ersetzen,
- die Syntax verändern,
- den Befehl SET COMPATIBLE DB4 ON einsetzen,

- benutzerdefinierte Funktionen aufrufen,
- eine neue Prozedur oder Funktion schreiben,
- Software eines anderen Herstellers verwenden.

Das Analyse-Programm und dieses Handbuch enthalten ein Verzeichnis aller Befehle und Funktionen, bei denen bekanntermaßen Kompatibilitätsprobleme auftreten können. (Siehe nachfolgende Abschnitte "Verzeichnis inkompatibler Befehle und Funktionen aus ...") Sie finden hier eine Erläuterung des Problems und Vorschläge zur Problemlösung.

Viele, wenn nicht sogar die meisten Probleme können gelöst werden, indem der Befehl bzw. die Funktion durch einen entsprechenden Befehl bzw. eine Funktion aus FoxPro ersetzt wird. In vielen Fällen können Sie dafür benutzerdefinierte Funktionen schreiben, die diesen Austausch für Sie durchführen. Viele solcher UDFs gehören sogar zum Lieferumfang des Programm-Konverters. In manchen Fällen muß der Programmcode nur ganz geringfügig verändert werden, z.B. wenn nur ein Parameter eingefügt werden muß, um die gewünschten Ergebnisse zu erzielen. Eine Reihe von Problemen kann durch die Aktivierung des dBASE IV-Kompatibilitätsmodus gelöst werden.

Wo solche schnellen Lösungen nicht zur Hand sind, muß man für gewöhnlich nur eine Prozedur oder einige Zeilen zusätzlichen Programmcodes einfügen. Sie werden in den seltensten Fällen, abgesehen von Datenschutzroutinen und Transaktionen, auf Produkte anderer Hersteller zurückgreifen müssen.

Der Befehl SET COMPATIBLE

Sie können die Zahl von Kompatibilitätsproblemen deutlich verringern, indem Sie den Befehl SET COMPATIBLE in Ihr Initialisierungsprogramm einfügen. Dazu geben Sie folgende Befehlszeile ein:

```
SET COMPATIBLE DB4 ON
```

Dieser Befehl bewirkt, daß FoxPro bestimmte Befehle und Funktionen so wie dBASE IV ausführt. Das Analyse-Programm ermittelt alle Schlüsselwörter, auf die sich der Befehl SET COMPATIBLE anwenden läßt. Im Abschnitt "Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV" wird bei den betreffenden Befehlen und Funktionen auch die Wirkungsweise von SET COMPATIBLE beschrieben.

Der Kompatibilitätsmodus ermöglicht Ihnen, Anwendungen im Handumdrehen an FoxPro anzupassen. Mit der Zeit, nachdem Sie die Funktionen von FoxPro kennengelernt haben, werden Sie den Kompatibilitätsmodus wahrscheinlich immer weniger benutzen und doch lieber mit den Funktionen in FoxPro arbeiten wollen.

In Anhang A finden Sie eine Übersicht, die zeigt, wie sich der Befehl SET COMPATIBLE auf die einzelnen Schlüsselworte auswirkt.

Benutzerdefinierte Funktionen (UDFs) aufrufen

Sie können viele Probleme, die dadurch entstehen, daß FoxPro und dBASE unterschiedliche Funktionsbezeichnungen verwenden oder daß FoxPro bestimmte Funktionen nicht unterstützt, durch den Einsatz von UDFs, die das Verhalten von dBASE IV Funktionen nachahmen, lösen.

Beispielsweise ist die dBASE IV Funktion PCOUNT identisch mit der Funktion PARAMETERS in FoxPro. Wenn Sie den nachfolgenden Programmcode in ein Programm einfügen, können Sie damit verhindern, daß Prozeduren oder UDFs aus dBASE IV, die die Funktion PCOUNT verwenden, Fehler verursachen:

FUNCTION Pcount
RETURN PARAMETERS()

Diese UDF und weitere benutzerdefinierte Funktionen finden Sie in der Datei FOXPROC.PRG auf der Programm-Konverter-Diskette. In der folgenden Tabelle sind alle dBASE-Funktionen, für die der Programm-Konverter eine UDF enthält, zusammengestellt:

FDATE()	ISBLANK()
FLDCOUNT()	PCOUNT()
FOR()	TAGCOUNT()
FTIME()	TAGNO()
	WINDOW()

Wenn Sie den Befehl SET PROCEDURE TO FOXPROC.PRG am Beginn Ihrer Anwendung einfügen, werden diese Funktionen genauso wie in dBASE ausgeführt. Falls die Anwendung bereits eine Prozedurdatei verwendet, fügen Sie diese UDFs einfach in diese Datei ein.

In den Fällen, in denen eine integrierte FoxPro-Funktion zwar wie eine dBASE-Funktion heißt, sich aber anders verhält, muß eine andere Lösung gefunden werden. Die integrierte Funktion wird in jedem Fall ausgeführt, auch dann, wenn eine gleichnamige UDF vorhanden ist.

Syntax-unabhängige Kompatibilitätsfragen

Parameterübergaben XE "Parameterübergaben"§

In dBASE IV können Parameter sowohl Befehlen als auch Funktionen als Variablenreferenz übergeben werden. In FoxPro werden Parameter nur Prozeduren als Variablenreferenz übergeben, Funktionen jedoch als *Wert*. Um Parameter auch Funktionen als Variablenreferenz übergeben zu können, fügen Sie den Befehl SET UDFPARMS TO REFERENCE in Ihr Programm ein.

Tastatureingaben

Wenn ein dBASE IV Programm die Funktionen INKEY(), READKEY() XE "READKEY "§oder LASTKEY() XE "LASTKEY "§verwendet, um Benutzereingaben zu ermitteln und anhand des Funktionsergebnisses entscheidet, welche Aktion daraufhin auszuführen ist, können in FoxPro Fehler auftreten. Bei einer Reihe von Tastenkombinationen unterscheidet sich in FoxPro und dBASE die Tastaturbelegung. Aus diesem Grund werden die Funktionen INKEY(), READKEY() und LASTKEY() vom Analyse-Programm als Problemfälle markiert.

FoxPro liest die Tastaturbelegung aus eine Makrodatei mit der Erweiterung .FKY ein. Verwenden Sie die Makrodatei dBASE.FKY, die mit dem Programm-Konverter ausgeliefert wird, damit die Tastatureingaben in FoxPro wie dBASE-Tastatureingaben interpretiert werden. Fügen Sie danach den Befehl RESTORE MACROS FROM dBASE.FKY in Ihr Programm ein. Sie können natürlich auch Ihren Programmcode verändern, sodaß anstelle der dBASE- die FoxPro-Tastenbelegung verwendet wird. In Anhang G finden Sie Liste der verschiedenen Tastaturbelegungen und Tastaturcodes.

Die in FoxPro verwendeten Tastenbezeichnungen unterscheiden sich etwas von dBASE. Lesen Sie dazu die Erklärung zum Befehl ON KEY in Abschnitt "Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV".

Hartcodierte DateierweiterungenXE "Dateierweiterungen"§

Das Analyse-Programm erkennt die gebräuchlichsten Dateierweiterungen, die in FoxPro nicht verwendet werden. In der folgenden Tabelle sind die verschiedenen von Programmen verwendeten dBASE-Dateitypen und die äquivalenten FoxPro-Dateitypen aufgeführt.

dBASE-Datei	FoxPro-Datei
XE ".DBF "§.DBF	Keine Änderung nötig
XE ".DBO"§.DB O	.FXP
XE ".DBT"§.DBT	.FPT
XE ".FMO"§.FM O	.PRX
XE ".FMT "§.FMT	.FMT
XE ".FRG "§.FRG	Keine Änderung nötig
XE ".FRM "§.FRM	.FRX

XE ".FRO "§.FRO	.F4X
XE ".LBG "§.LBG	Keine Änderung nötig
XE ".LBL "§.LBL	.LBX
XE ".LBO "§.LBO	.L4X
.MDX	.CDX
XE ".NDX "§.NDX	.IDX
XE ".PRG"§.PRG	.PRG
XE ".PRS "§.PRS	.PRG
XE ".QBE "§.QBE	.PRG

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Nachfolgend werden in alphabetischer Reihenfolge alle Befehle und Funktionen verzeichnet, die bei der Ausführung von dBASE-Pprogrammen in FoxPro Probleme verursachen können. Jeder Eintrag beschreibt, wie sich der Befehl oder die Funktion in dBASE and in FoxPro verhält. Unter dem Stichwort Vorgehen finden Sie Hinweise dazu, in welcher Weise Sie Ihr Programm verändern sollten, und in den meisten Fällen wird dies anhand eines Beispiels veranschaulicht. Im Abschnitt Anmerkungen werden relevante Zusatzinformationen gegeben. Der Inhalt dieses Verzeichnisses wird auch im Fenster Program Analyzer angezeigt, allerdings ohne Beispiele.

Dieses Verzeichnis stellt in keiner Weise den Versuch dar, die Originaldokumentation dieser Befehle und Funktionen zu reproduzieren, und es wird auch kein Anspruch auf Vollständigkeit erhoben. Bitte schlagen Sie in der Dokumentation zu dBASE IV bzw. FoxPro nach, wenn Sie detailliertere Informationen zu diesen Befehle und Funktionen brauchen.

Jeder Befehl bzw. jede Funktion wird einer der folgenden vier Problemstufen zugeordnet:

- **Stufe XE "Problemstufe" §1** umfaßt Befehle und Funktionen, die von FoxPro nicht unterstützt werden. Diese Befehle und Funktionen müssen gelöscht oder durch die äquivalenten Befehle und Funktionen der FoxPro-Sprache ersetzt werden.
- **Stufe 2** umfaßt Befehle und Funktionen, die in FoxPro Fehler erzeugen, für die es aber häufig sehr ähnliche Befehle und Funktionen in FoxPro gibt.
- **Stufe 3** umfaßt Befehle und Funktionen, die in FoxPro zwar keine Fehler erzeugen, sich jedoch so unterschiedlich verhalten, daß sie Beachtung verdienen.
- **Stufe 4** umfaßt Befehle und Funktionen, die bei der Ausführung des Programms in FoxPro nur selten Probleme verursachen. Wenn Sie nach Problemen der Stufe 4 suchen

lassen, werden viele Programmzeilen markiert, die für
gewöhnlich problemlos ausgeführt werden.

Diese Einstufung mag vielleicht etwas zufällig erscheinen, sie soll Ihnen
lediglich erleichtern einzuschätzen, wie schwerwiegend ein Problem ist und
wieviel Aufwand seine Lösung erfordert.

@...GET MESSAGE

Siehe MESSAGE.

**@...GET XE "@...GET"\$REQUIREDXE "@...GET REQUIRED"\$XE
"REQUIRED"\$**

Stufe: 1

In dBASE IV: Mit dem Schlüsselwort REQUIRED wird festgelegt, daß eine Überprüfung gemäß der RANGE-Parameter oder der VALID-Klausel stattfindet, wenn der Cursor auf einen Datensatz gesetzt wird, unabhängig davon, ob der Datensatz verändert wurde oder nicht. Die Überprüfung findet auch statt, wenn Sie den Mauszeiger über einen Datensatz oder ein Feld bewegen.

In FoxPro: Erzeugt einen Fehler

Vorgehen: Damit eine Überprüfung stattfindet, wenn der Benutzer den Cursor auf einen Datensatz setzt, unabhängig davon, ob Daten verändert wurden, löschen Sie das Schlüsselwort REQUIRED und setzen stattdessen SET COMPATIBLE DB4 ein.

Damit eine Überprüfung stattfindet, wenn der Benutzer den Mauszeiger über einen Datensatz bewegt, löschen Sie das Schlüsselwort REQUIRED. Fügen Sie danach eine VALID-Klausel in den READ Befehl ein, der eine benutzerdefinierte Funktion (kurz UDF) aufruft. Definieren Sie die Überprüfungsrouninen für jeden GET Befehl in der benutzerdefinierten Funktion, die von READ VALID aufgerufen wird.

Anmerkung: Die Maus kann mit dem Befehl SET MOUSE auch deaktiviert werden.

Beispiel: **dBASE**
@ 2,2 SAY "Name:" GET kname VALID

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

REQUIRED UDF()

```
@ 6,2 SAY "Menge:" GET auft_mnge VALID  
REQUIRED ;
```

```
  auftr_menge <> 0  
  READ
```

```
  FUNCTION udf  
  IF LEN(kname) = 0  
    <Fehlermeldung ausgeben>  
    RETURN .F.  
  ELSE  
    RETURN .T.  
  ENDIF
```

FoxPro

```
@ 2,2 SAY "Kundenname:" GET kname  
@ 6,2 SAY "Menge: " GET auft_mnge  
READ VALID udf()
```

```
FUNCTION udf  
ret_val=.T.  
IF LEN(kname) = 0  
  <Fehlermeldung ausgeben>  
ret_val = .F.  
ENDIF  
IF auft_mnge = 0  
  <Benutzer andere Fehlermeldung ausgeben>  
ret_val = .F.  
ENDIF  
RETURN ret_val
```

@...GETXE "@...GET"§ XE "@...GET"§OPEN WINDOWXE
"OPEN WINDOW"§

XE "@...GET OPEN WINDOW"§Stufe: 3

In dBASE IV: dBASE öffnet das Bearbeitungsfenster für ein Memofeld und zeigt den Memotext an, sobald Sie das Memofeld aktivieren.

Verzeichnis inkompatibler Befehle und Funktionen aus

dBASE IV

Sie müssen Strg+Pos1 drücken, um den Cursor in das Fenster zu setzen und den Text bearbeiten zu können.

In FoxPro: Wie in dBASE, allerdings zeigt FoxPro das Wort "Memo" hinter dem vordefinierten Fenster an und setzt den Cursor auf das Wort "Memo".

Anmerkung: Bei Bedarf können Sie die Koordinaten des Fensters, das geöffnet wird dahingehend verändern, daß das Wort "Memo" nicht überdeckt wird. Alternativ dazu können Sie die Befehle EDIT und MODIFY MEMO in FoxPro verwenden, die Ihnen nicht nur größere Flexibilität, sondern auch umfangreichere Steuerungsmöglichkeiten geben.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

@...SAY

XE "@...SAY"§Stufe: Wird vom Analyse-Programm nicht markiert.

In dBASE IV: a) Der Bildschirminhalt wird nach oben geblättert, damit Ausgaben, die über die rechte untere Bildschirmcke hinausgehen, angezeigt werden können.

b) Wenn SET STATUS auf ON gesetzt ist, können Ausgaben die Statuszeile überschreiben. Die Anzeige von Text, der länger als die Statuszeile ist, wird oberhalb der Statuszeile fortgesetzt und von hier nach oben geblättert.

c) Wenn @SAY mit dem Schlüsselwort PICTURE verwendet wird, wird die letzte Dezimalstelle gerundet.

In FoxPro: a) Ausgaben, die über den rechten Bildschirmrand hinausgehen, werden abgeschnitten und nicht angezeigt.

b) Ausgaben können die Statuszeile nicht überschreiben. Über den rechten Bildschirmrand hinausgehender Text wird abgeschnitten.

c) Wenn Daten mit Hilfe einer PICTURE-Klausel angezeigt werden, wird der Wert nicht gerundet sondern abgeschnitten.

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Vorgehen: Setzen Sie SET COMPATIBLE DB4 auf ON.

Anmerkung: Wenn Sie COMPATIBLE auf ON gesetzt haben, wird @SAY in FoxPro genauso interpretiert wie in dBASE. Da mit diesem Befehl keine Daten verändert, sondern nur das Erscheinungsbild von Ausgaben festgelegt wird, ignoriert das Analyse-Programm @SAY Befehle.

ACCESS()

Siehe Datensicherheit.

ACTIVATE MENU "ACTIVATE MENU"§

Stufe: 2

In dBASE IV: dBASE erlaubt es Benutzern nicht, auf deaktivierte Menüoptionen zu klicken.

In FoxPro: FoxPro erlaubt es Benutzern, auf deaktivierte Menüoptionen zu klicken. Das Klicken hat hier dieselbe Wirkung wie das Drücken der ESC-Taste.

Vorgehen: Verlagern Sie den Befehl ACTIVATE MENU in eine DO...WHILE Schleife.

Anmerkung: Die Schleife sollte solange ausgeführt werden, bis der Benutzer durch Klicken auf eine Menüoption die Anwendung verläßt.

Beispiel: **dBASE**
ACTIVATE MENU mein_menü

FoxPro
mein_menü DO WHILE PAD() <>"Beenden"
&&Setzen Sie hier ACTIVATE MENU
&&den String Ihrer
ENDDO
&&Anwendung ein

ACTIVATE POPUPXE "ACTIVATE POPUP"§

Stufe: 2

In dBASE IV: dBASE erlaubt es Benutzern nicht, auf deaktivierte Popup-Menüs zu klicken.

In FoxPro: FoxPro erlaubt es Benutzern, auf deaktivierte Popup-Menüs zu klicken. Das Klicken hat hier dieselbe Wirkung wie das Drücken der ESC-Taste.

Vorgehen: Verlagern Sie den Befehl ACTIVATE POPUP in eine DO...WHILE Schleife.

Beispiel: **dBASE**

```
ACTIVATE POPUP mein_popup
```

FoxPro

```
DO WHILE BAR() <> 5    &&Verwenden Sie  
ACTIVATE MENU mein_menü    &&die
```

Menünummer

```
END DO
```

&&Ihrer

Anwendung

ACTIVATE SCREENXE "ACTIVATE SCREEN"§, ACTIVATE WINDOWXE "ACTIVATE WINDOW"§

Stufe: Wird vom Analyse-Programm nicht markiert.

In dBASE IV: Der Cursor wird gemäß Voreinstellung auf die Position 0,0 gesetzt.

In FoxPro: Die Cursorposition wird nicht verändert, sie bleibt dieselbe wie vor dem Aufruf des Befehls ACTIVATE SCREEN bzw. ACTIVATE WINDOW.

Vorgehen: Setzen Sie SET COMPATIBLE DB4 auf ON.

Anmerkung: Nach Eingabe des Befehls ACTIVATE SCREEN bzw. ACTIVATE WINDOW wird der Cursor auf die Position 0.0 gesetzt. Das Analyse-Programm ignoriert diese Befehle, da sie nur das Erscheinungsbild der Ausgabe beeinflussen..

APPEND MEMOXE "APPEND MEMO"§

Stufe: 3

In dBASE IV: Wenn keine Dateierweiterung angegeben wird, wird die Erweiterung .TXT verwendet.

In FoxPro: Das Fehlen der Angabe einer Dateierweiterung resultiert in einem Fehler.

Vorgehen: Setzen Sie SET COMPATIBLE DB4 auf ON.

BARCOUNT()XE "BARCOUNT()"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Gibt die Anzahl von Menüblöcken im genannten Popup-Menü aus bzw. im aktiven Popup-Menü, falls kein Menüname angegeben wurde.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie die Funktion BARCOUNT() durch die FoxPro-Funktion CNTBAR().

Anmerkung: CNTBAR() verlangt die Angabe eines Menünamens.

Beispiel: **dBASE**

numBars=BARCOUNT(pop_name)

FoxPro

numBars=CNTBAR(pop_name)

BARPROMPT()XE "BARPROMPT()"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Gibt den Text der Eingabeaufforderung des genannten Popup-Menüs aus bzw. falls kein Menüname angegeben wurde, den des aktiven Popup-Menüs.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie BARPROMPT() durch die FoxPro-Funktion

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

PRMBAR().

Anmerkung: PRMBAR() verlangt die Angabe eines Menünamens.

Beispiel: **dBASE**

prmt_txt=BARPROMPT(1)

FoxPro

prmt_txt=PRMBAR(pop_name,1)

BEGIN TRANSACTION

Siehe Transaktionen.

BLANK (nur in Version 1.5)

XE "BLANK (nur in Version 1.5)"§Stufe: 2

In dBASE IV: Füllt ein oder mehrere genannte Felder in einem oder mehreren Datensätzen mit Leerzeichen ("Nullen").

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie den Befehl BLANK. Verwenden Sie die Befehle SCATTER MEMVAR BLANK und GATHER MEMVAR, um Werte aus einem Datensatz zu entfernen. Einzelne Felder bearbeiten Sie mit dem Befehl REPLACE.

Anmerkung: FoxPro unterstützt Nullwerte nur eingeschränkt. Mit Leerzeichen gefüllte Felder stellen Nullwerte dar, FoxPro unterstützt Nullwerte in Datums- und Zeichenfeldern, aber nicht in numerischen und logischen Feldern. Wenn Sie in numerischen Feldern mit Nullwerten arbeiten wollen, müssen Sie ein logisches Feld anlegen, das überprüft, ob das numerische Feld einen Wert enthält oder leer ist.

Beispiel: **FoxPro**

Datumsfelder:

```
REPLACE ung_datum WITH { / / }
```

Zeichenfelder:

```
REPLACE kund_name WITH  
SPACE(LEN(kund_name))
```

Datensätze (füllt den aktuellen Datensatz mit Leerzeichen; in numerischen Feldern wird der Wert Null gesetzt):

```
SCATTER MEMVAR BLANK  
GATHER MEMVAR
```

BROWSE

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

XE "BROWSE"§Stufe: Wird vom Analyse-Programm nicht
angezeigt.

In dBASE IV: Sobald der Benutzer den Cursor aus der Zeile bewegt,
werden die Daten in die Tabelle aufgenommen.

In FoxPro: Sobald der Benutzer den Cursor aus dem Feld bewegt,
werden die Daten in die Tabelle aufgenommen.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

BROWSEXE "BROWSE"§ COMPRESSXE "COMPRESS"§

XE "BROWSE COMPRESS"§Stufe:3

In dBASE IV: Das Schlüsselwort COMPRESS bewirkt, daß der
Tabellenkopf in der Bildschirmanzeige auf eine Zeile
komprimiert wird.

In FoxPro: Wird nicht unterstützt. Das Schlüsselwort COMPRESS wird
ignoriert.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

Anmerkung: Mit COMPRESS können in dBASE bis zu 19 Zeilen anstatt
nur 17 auf einem Bildschirm mit 25 Zeilen angezeigt werden.
FoxPro 2.5 für Windows unterstützt skalierbare Schriften und
Sie können daher eine kleinere Schriftgröße wählen. Dadurch
verringert sich die Zeilenhöhe, und es können mehr
Datensätze gleichzeitig angezeigt werden. Im folgenden
Beispiel wird die Bildschirmausgabe von 25 Datensätzen
(Standardschrift und Schriftgröße) auf 31 Datensätze (Times
New Roman, 6 pt.) erhöht.

Beispiel: **dBASE**
BROWSE COMPRESS

FoxPro
BROWSE FONT 'Times New Roman', 6

BROWSEXE "BROWSE"§ NOFOLLOWXE "NOFOLLOW"§

XE "BROWSE NOFOLLOW "§Stufe: 3

In dBASE IV: Das Schlüsselwort NOFOLLOW bewirkt, daß nach der Bearbeitung eines Kennfeldes der bearbeitete Datensatz zwar im Index neu eingeordnet wird, der zugehörige Datensatzzeiger aber unverändert bleibt. Ohne Angabe dieser Option wird der Datensatz im Index neu angeordnet und bleibt weiterhin der aktuelle Datensatz.

In FoxPro: Das Schlüsselwort NOFOLLOW wird ignoriert. In FoxPro wird BROWSE immer so interpretiert, als wäre das Schlüsselwort NOFOLLOW angegeben. Nach der Bearbeitung eines Kennfeldes wird weder der Zeiger, der geänderte Datensätze anzeigt, noch der Datensatzzeiger verändert.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

Anmerkung: Falls Sie den dBASE-Befehl BROWSE ohne NOFOLLOW simulieren wollen, verwenden Sie den Befehl SHOW WINDOW <Fenstername> REFRESH. Damit wird der aktualisierte Datensatz auch in der Bildschirmanzeige neu angeordnet. Der Datensatzzeiger kann mit der gleichen Routine verändert werden.

BROWSE XE "BROWSE "§NOINITXE "NOINIT"§

XE "BROWSE NOINIT "§Stufe: 3

In dBASE IV: Reaktiviert die bei der letzten BROWSE-Sitzung eingegebenen Parameter der Fensterkonfiguration.

In FoxPro: Wird nicht unterstützt. Das Schlüsselwort NOINIT wird ignoriert.

Vorgehen: Keine besonderen Maßnahmen erforderlich. Das Schlüsselwort LAST erfüllt dieselbe Funktion wie NOINIT.

Beispiel: **dBASE**
BROWSE NOINIT

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

FoxPro

BROWSE LAST

BROWSE "BROWSE" § NOMENUXE "NOMENU" §

XE "BROWSE NOMENU "§Stufe: 3

In dBASE IV: Mit der Option NOMENU wird die Menüleiste ausgeblendet und der Zugriff darauf verhindert.

In FoxPro: Die Option NOMENU entfernt nur den Menüblock Datenblatt aus der Systemmenüleiste.

Vorgehen: Wenn Sie die gesamte Menüleiste ausblenden wollen, geben Sie vor dem BROWSE Befehl den Befehl SET SYSMENU OFF ein. Stellen Sie danach die Menüanzeige wieder her, indem Sie SET SYSMENU auf ON setzen oder SET SYSMENU AUTOMATIC eingeben.

Beispiel: **dBASE**
BROWSE NOMENU

FoxPro

SET SYSMENU OFF
BROWSE
SET SYSMENU ON

**BROWSEXE "BROWSE"§ NOORGANIZEXE "NOORGANIZE"§
XE "BROWSE NOORGANIZE "§**

Stufe: 2

In dBASE IV: Das Schlüsselwort NOORGANIZE (nur in Version 1.5) entfernt den Menüblock Verwaltung aus der Menüleiste.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie dieses Schlüsselwort.

Anmerkung: In FoxPro gibt es kein Menü namens Verwaltung. Ähnliche Befehle wie im Menüblock Verwaltung finden Sie in FoxPro in den Menüblöcken Datenbank und Datenblatt. Das FoxPro-Menüsystem kann leicht verändert werden. Sie können einzelne Menüoptionen wie auch ganze Menüblöcke deaktivieren oder entfernen. Schlagen Sie dazu bitte das Kapitel über Menüs im *Benutzerhandbuch* (Teil der Dokumentation zu FoxPro 2.5) nach.

CALLXE "CALL"§

Stufe: 1

In dBASE IV: CALL kann eine Ausdrucksliste verarbeiten.

In FoxPro: CALL kann nur einen Ausdruck verarbeiten.

Vorgehen: Verändern Sie Ihre Binärdateien so, daß eine Binärdatei jeweils nur eine Routine enthält, und verwenden Sie statt des dBASE-Befehls CALL, mit dem eine Binärdatei mit mehreren Programmmodulen ausgeführt werden kann, in FoxPro entsprechend viele CALL Befehle für die einzelnen Routinen.

CALL()

XE "CALL()"§Stufe: 1

In dBASE IV: Diese Funktion stellt eine Alternative zum Befehl CALL dar,

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

wie dieser führt sie Binärroutinen aus, die zuvor mit dem Befehl LOAD in den Arbeitsspeicher geladen wurden.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen den Befehl CALL ein und modifizieren Sie gegebenenfalls die aufzurufende Routine.

Anmerkung: Der Befehl CALL erfüllt in FoxPro dieselbe Funktion wie in dBASE IV. Sie müssen unter Umständen die Anzahl von Parametern oder die Binärroutine verändern, da in dBASE der Befehl CALL der Funktion CALL einen Rückgabewert übergeben kann. Wenn Sie den Befehl CALL einsetzen, muß die Binärroutine Werte zurückliefern, indem es die Werte der als Parameter übergebenen Speichervariablen verändert.

CATALOG()XE "CATALOG()"

Stufe: 2

In dBASE IV: Liefert den Namen der aktuellen Katalogdatei zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

CERROR()

XE "CALL()"§Stufe: 2

In dBASE IV: Nicht dokumentierte Funktion, die die Nummer des letzten Kompilerfehlers ausgibt.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

CHANGE()

Siehe Netzwerkfunktionen.

Vergleichsoperatoren

XE "Vergleichsoperatoren"§Stufe: 2

In dBASE IV: Für die Operatoren \geq " \geq "§ (größer gleich) und \leq " \leq "§ (kleiner gleich) sind auch die Schreibweisen \Rightarrow und \Leftarrow zulässig.

In FoxPro: Wenn Sie ein Gleichheitszeichen vor das Kleiner- oder vor das Größerzeichen setzen (zum Beispiel \Rightarrow " \Rightarrow "§ oder \Leftarrow " \Leftarrow "§) wird die Fehlermeldung "Fehlender Operand" ausgegeben.

Vorgehen: Ersetzen Sie alle Vorkommen von \Rightarrow und \Leftarrow durch \geq bzw. \leq .

COMPLETED()

Siehe Transaktionen.

CONVERT

Siehe Netzwerkfunktionen.

COPY TO ARRAY

XE "COPY TO ARRAY"§Stufe: 4

In dBASE IV: Ein Feld kann in der FIELDS-Klausel mehrfach genannt werden.

In FoxPro: Wenn ein Feld in der FIELDS-Klausel mehr als einmal vorkommt, wird ein Fehler erzeugt.

Vorgehen: Entfernen Sie Mehrfachverweise auf ein Feld.

CTOD()

XE "CTOD()"§Stufe: 3

In dBASE IV: Wenn der Funktion CTOD eine Zeichenfolge übergeben wird, in der die Monatsangabe größer als 12 oder das

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Tagesdatum größer als die Anzahl von Tagen des betreffenden Monats ist, berechnet dBASE IV die Daten und liefert eine gültige Datumsangabe zurück. Zum Beispiel wird CTOD("13/32/93") ausgewertet zu 02/01/94.

In FoxPro: In FoxPro werden nur gültige Datumsangaben als Parameter akzeptiert. Bei Eingabe eines ungültigen Datums, liefert CTOD einen leeren Datumswert zurück, der in der Form " / / " angezeigt wird.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

Anmerkung: Sie sollten mit Hilfe der Funktion EMPTY den Rückgabewert von CTOD() überprüfen und damit sicherstellen, daß ein gültiges Datum eingegeben wurde.

Datensicherheit

XE "Datensicherheit"§Stufe: 1

In dBASE IV: PROTECTXE "PROTECT"§, ACCESSXE "ACCESS"§, USERXE "USER"§, SET ENCRYPTIONXE "SET ENCRYPTION"§ und SET("ENCRYPTION")XE "SET(\"ENCRYPTION\")"§ gehören in dBASE zu einem System, das die Datensicherheit gewährleistet.

In FoxPro: FoxPro ignoriert den Befehl SET ENCRYPTION. Die Funktion ACCESS liefert immer den Wert 0 zurück. Die Funktion USER und der Befehl PROTECT erzeugen Fehler.

Vorgehen: Löschen Sie die Funktion USER. Kopieren Sie verschlüsselte dBASE-Tabellen in nicht verschlüsselte dBASE-Dateien, bevor Sie sie in FoxPro laden.

Anmerkung: Falls eine Anwendung ein ausgefeiltes Datensicherungssystem oder die Verschlüsselung von Daten erfordert, können Sie in FoxPro Ihr eigenes System entwickeln oder auf Produkte anderer Hersteller zurückgreifen..

DEFINE BAR

Siehe MESSAGE.

DEFINE MENU

XE "DEFINE MENU"§Stufe: 4

In dBASE IV: In dBASE IV Version 1.0 fügt DEFINE MENU ein zusätzliches Leerzeichen an einen Menüblock an (und an die Hervorhebung zur Kennzeichnung eines markierten Menüblocks). Falls für DEFINE PAD keine Koordinaten für den Textanfang angegeben werden, werden die einzelnen Menüblöcke jeweils durch ein Leerzeichen getrennt; infolgedessen können bis zu drei Leerzeichen zwischen den einzelnen Menüblöcken stehen. Die dBASE IV Versionen

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

1.1 und 1.5 stellen Menüblöcken zwar keine Leerzeichen voran oder nach, es wird jedoch ein Leerzeichen zur Trennung der einzelnen Menüblöcke eingefügt.

MESSAGE-Ausdrücke werden im aktuellen Fenster ausgegeben (Siehe MESSAGE).

In FoxPro: Wie in dBASE IV 1.0. An jeden Menüblock wird ein Leerzeichen angefügt.

MESSAGE-Ausdrücke werden auf dem Desktop ausgegeben (Siehe MESSAGE).

Vorgehen: Wenn Sie das Schlüsselwort NOMARGIN zusammen mit dem Befehl DEFINE MENU verwenden, werden keine zusätzlichen Leerzeichen eingefügt und das Menü wird wie in dBASE IV 1.1 oder 1.5 angezeigt.

Anmerkung: Bei in dBASE IV Version 1.1 oder 1.5 geschriebenen Menüzeilen passen aufgrund der zusätzlichen Leerzeichen unter Umständen nicht alle Menüblöcke in eine Zeile.

Beispiel: **dBASE** (1.1 or 1.5 only)
DEFINE MENU Haupt

FoxPro

DEFINE MENU Haupt NOMARGIN

DEFINE PAD

Siehe MESSAGE.

DEFINE POPUP

Siehe MESSAGE.

DESCENDING()XE "DESCENDING()"§ (nur in Version 1.5)

Stufe: 1

In dBASE IV: Die Funktion DESCENDING liefert den Wahrheitswert .T. (True - wahr), wenn ein Index die Datensätze in absteigender

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Reihenfolge einordnet, andernfalls gibt sie .F. (false - falsch) zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie die Funktion DESCENDING.

DISPLAY USERS

Siehe LIST USERS

END TRANSACTION

Siehe Transaktionen.

ERROR()

XE "ERROR()"§Stufe: 3

In dBASE IV: ERROR() liefert die Fehlermeldungsnummer des Fehlers, der vom Befehl ON ERROR registriert wurde.

In FoxPro: Wie in dBASE, gibt jedoch oft andere Fehlermeldungsnummern zurück als die von dBASE.

Vorgehen: Verwenden Sie die entsprechende Fehlermeldungsnummer aus FoxPro. Anhang D enthält eine Übersicht der Fehlermeldungen, die in dBASE und FoxPro zwar gleichlautend sind, jedoch unterschiedliche Nummern haben. Im *FoxPro Entwicklerhandbuch* finden Sie eine Liste aller FoxPro-Fehlermeldungen.

Anmerkung: Sie können unter Umständen einige Teile Ihrer Fehlerbehandlungsroutine löschen, da einige dBASE-Fehler in FoxPro nicht auftreten können. Beispielsweise gibt FoxPro nie die Fehler 76 (" -: Verkettete Zeichenfolge zu lang") oder 77 aus, da FoxPro Zeichenketten mit einer Länge von bis zu 65K unterstützt.

Beachten Sie, daß bei vielen Fehlermeldungsnummern, die in FoxPro und in dBASE für denselben Fehler ausgegeben werden, die Meldungstexte unterschiedlich sind. Falls eine Fehlerbehandlungsroutine die Funktion MESSAGE anstelle der Funktion ERROR verwendet, reagiert sie daher unter Umständen auf Fehler, die von beiden Programmen gleich behandelt werden. Siehe Anhänge C und D.

FCREATE()XE "FCREATE()"§ und FOPEN()XE "FOPEN()"§

Stufe: 2

In dBASE IV: Der zweite Parameter, der den Funktionen FOPEN und FCREATE übergeben wird, gibt die Dateiattribute an und muß einen der folgenden Werte haben "R" (read-only), "W"

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

(write-only), "A" (append-only), "RW" bzw. "WR (read and write) oder "RA" bzw. "AR" (read and append).

In FoxPro: Der zweite Parameter der Funktionen FOPEN and FCREATE gibt in Form von Kennzahlen die Dateiattribute an, sie können hier auch angeben, ob Zugriffe gepuffert oder ungepuffert sind.

Vorgehen: Ersetzen Sie die dBASE-Dateiattributcodes durch die Attributnummern von FoxPro.

Beispiel: dBASE
`datei_handle = FCREATE("Beispiel.txt", "R")`

FoxPro
`datei_handle = FCREATE("Beispiel.txt", "1")`

MS-DOS Attribut(e)	FoxPro Attribut nummer	dBASE Attributcode
Read/write (Standard)	0	RW oder WR
Append only (nur anhängen)	0*	A
Read and append (lesen und anhängen)	0*	RA ode AR
Read only (schreibgeschützt)	1	R
Hidden (verborgen)	2	(nicht definiert)

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

Read Only/Hidden schreibgeschützt/verborgen	3	(nicht definiert)
MS-DOS Attribut(e)	FoxPro Attribut nummer	dBASE Attributcode
System	4	(nicht definiert)
Read Only/System (schreibgeschützt/System)	5	(nicht definiert)
System/Hidden (System/verborgen)	6	(nicht definiert)
Read Only/Hidden/System (schreibgeschützt/verborgen/ System)	7	(nicht definiert)

*In FoxPro ist Append nicht in derselben Weise definiert.
Sowohl der Lese- als auch der Schreibmodus sind verfügbar.

FDATE()XE "FDATE()" § (nur in Version 1.5)

Stufe: 2

In dBASE IV: Die Funktion FDATE gibt das Erstellungsdatum der als

Parameter eingegebenen Datei aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen die Funktion ADIR ein, oder verwenden Sie eine der in FOXPROC.PRG enthaltenen benutzerdefinierten Funktionen (UDF).

Anmerkung: In FoxPro gibt eine Funktion, ADIR(), das Erstellungsdatum, aber auch die Größe, Erstellungszeit und die Attribute einer Datei aus. Als Parameter sind ein Dateiname und ein Dateiname mit Platzhaltern zulässig. Die Daten zu den gesuchten Dateien werden automatisch in ein entsprechend dimensioniertes Array geschrieben. In der dritten Spalte des Arrays werden die Angaben zum Erstellungsdatum der Datei(en) gespeichert.

FOXPROC.PRG enthält eine benutzerdefinierte Funktion namens FDATE(), die es Ihnen erlaubt, Anweisungen mit der dBASE-Funktion FDATE() ohne vorherige Änderung ausführen zu lassen. Lesen Sie dazu bitte im Abschnitt "Benutzerdefinierte Funktionen (UDFs) aufrufen" nach.

Beispiel: **dBASE**

```
datei_datum = FDATE('kunden.dbf')
```

FoxPro

```
temp = ADIR(dir_array,'c:\foxprow\angest.dbf')
datei_datum = dir_array(3)
```

FGETS()

XE "FGETS()"§Stufe: 2

In dBASE IV: Sie können im dritten Parameter der Funktionen FGETS und FPUTS das Zeilenendezeichen festlegen (gemäß Voreinstellung handelt es sich dabei um CR/LF, ein Wagenrücklauf- und Zeilenvorschubzeichen).

In FoxPro: Die Funktionen FGETS und FPUTS können nur Textdateien verarbeiten, in denen als Zeilenende das Zeichen CR/LF

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

(Wagenrücklauf- und Zeilenvorschubzeichen) verwendet wird.

Vorgehen: Falls in einer Datei das Zeilenende durch ein anderes Zeichen als dem Wagenrücklaufzeichen (CR) (ASCII 13) oder dem Zeilenvorschubzeichen (LF) (ASCII 10) markiert wird, ersetzen Sie alle Zeilenendezeichen durch CR/LF.

Sie können die Funktion FGETS auch durch eine FREAD-Schleife ersetzen, mit der jedes Zeichen der Datei gelesen und daraufhin überprüft wird, ob es sich um ein Zeilenendezeichen handelt.

FLDCOUNT()XE "FLDCOUNT()"§ (nur in Version 1.5)

Stufe: 2

In dBASE IV: Diese Funktion liefert die Anzahl von Feldern einer Tabelle zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen die entsprechende FoxPro-Funktion FCOUNT ein, oder verwenden Sie eine benutzerdefinierte Funktion aus FOXPROC.PRG.

Anmerkung: FCOUNT hat dieselbe Funktionalität wie FLDCOUNT.

Wenn Sie die Funktion FLDCOUNT() aus FOXPROC.PRG verwenden, brauchen Sie Routinen mit der dBASE-Funktion FLDCOUNT() nicht zu verändern. Lesen Sie dazu bitte im Abschnitt "Benutzerdefinierte Funktionen (UDFs) aufrufen" nach.

Beispiel: **dBASE**
FLDCOUNT('kunden.dbf')

FoxPro
FCOUNT('kunden.dbf')

FLDLIST()XE "FLDLIST()"§ (nur in Version 2.0)

Stufe: 1

In dBASE IV: Gibt die in der mit SET FIELDS TO definierten Liste enthaltenen Felder zurück oder, falls ein numerischer Parameter übergeben wurde, den Namen eines bestimmten Feldes.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie FLDLIST() durch SET("FIELDS", 1).

Anmerkung: Siehe SET FIELDS, dort finden Sie weitere Erläuterungen zu mit SET FIELDS definierten Feldlisten, die Felder aus

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

unterschiedlichen Arbeitsbereichen enthalten.

FOPEN()

Siehe FCREATE()

FOR()XE "FOR()"§ (nur in Version 1.5)

Stufe: 2

In dBASE IV: Die Funktion FOR liefert die FOR-Klausel, mit der ein an eine Bedingung geknüpfter Indexeintrag erzeugt wird, zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen die entsprechende FoxPro-Funktion SYS(2021) ein, oder verwenden Sie eine benutzerdefinierte Funktion aus FOXPROC.PRG.

Anmerkung: Wenn Sie die Funktion FOR() aus FOXPROC.PRG verwenden, brauchen Sie Routinen mit der dBASE-Funktion FLDCOUNT() nicht zu verändern. Lesen Sie dazu bitte im Abschnitt "Benutzerdefinierte Funktionen (UDFs) aufrufen" nach.

Beispiel: **dBASE**
for_clause = FOR('kund_kein.mdx',1)

FoxPro
for_clause = SYS(2021, 1)

FPUTS()

XE "FPUTS()"§ Stufe: 2

In dBASE IV: Sie können im dritten Parameter der Funktionen FPUTS und FGETS das Zeilenendezeichen festlegen (gemäß Voreinstellung handelt es sich dabei um CR/LF, ein Wagenrücklauf- und Zeilenvorschubzeichen).

In FoxPro: Die Funktionen FPUTS und FGETS können nur Textdateien verarbeiten, in denen als Zeilenende das Zeichen CR/LF (Wagenrücklauf- und Zeilenvorschubzeichen) verwendet wird.

Vorgehen: Falls in einer Datei das Zeilenende durch ein anderes Zeichen

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

als dem Wagenrücklaufzeichen (CR) (ASCII 13) oder dem Zeilenvorschubzeichen (LF) (ASCII 10) markiert wird, ersetzen Sie alle Zeilenendezeichen durch CR/LF.

Sie können mit Hilfe der Funktion FWRITE eine Datei mit anderen Zeilenendezeichen erzeugen, indem Sie der Funktion einen String mit einer Textzeile und dem gewünschten Zeilenendezeichen übergeben.

FSIZE()

XE "FSIZE() "§Stufe:2

In dBASE IV: Die Funktion FSIZE liefert die Größe der angegebenen *Datei* zurück.

In FoxPro: Die Funktion FSIZE liefert die Größe des angegebenen *Feldes* zurück.

Vorgehen: Setzen Sie stattdessen die Funktion ADIR ein.

Anmerkung: In FoxPro gibt eine Funktion, ADIR(), die Dateigröße, aber auch das Erstellungsdatum, die Erstellungszeit und die Attribute einer Datei aus. Als Parameter ist ein Dateiname mit Platzhaltern zulässig. Die Daten zu den gesuchten Dateien werden automatisch in ein entsprechend dimensioniertes Array geschrieben. In der zweiten Spalte des Arrays werden die Angaben zur Größe der Datei(en) gespeichert.

Beispiel: **dBASE**

```
datei_größe = FSIZE(kunden.dbf)
```

FoxPro

```
temp = ADIR(dir_array,'c:\foxprow\angest.dbf')  
datei_größe = dir_array(2)
```

FTIME() XE "FTIME() "§(nur in Version 1.5)

Stufe: 2

In dBASE IV: Die Funktion FTIME gibt das Erstellungsdatum der als Parameter angegebenen Datei aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen die Funktion ADIR ein, oder verwenden Sie eine der in FOXPROC.PRG enthaltenen benutzerdefinierten Funktionen (UDF).

Anmerkung: In FoxPro gibt eine Funktion, ADIR(), die Erstellungszeit,

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

aber auch die Größe, das Erstellungsdatum und die Attribute einer Datei aus. Als Parameter ist ein Dateiname mit Platzhaltern zulässig. Die Daten zu den gesuchten Dateien werden automatisch in ein entsprechend dimensioniertes Array geschrieben. In der vierten Spalte des Arrays werden die Angaben zur Erstellungszeit der Datei(en) gespeichert.

FOXPROC.PRG enthält eine benutzerdefinierte Funktion namens FTIME(), die es Ihnen erlaubt, Anweisungen mit der dBASE-Funktion FTIME() ohne vorherige Änderung ausführen zu lassen. Lesen Sie dazu bitte im Abschnitt "Benutzerdefinierte Funktionen (UDFs) aufrufen" nach.

Beispiel: **dBASE**

```
datei_zeit = FTIME('kunden.dbf')
```

FoxPro

```
temp = ADIR(dir_array,'c:\foxprow\angest.dbf')  
datei_zeit = dir_array(4)
```

HOME()XE "HOME()"§

Stufe: 2

In dBASE: Liefert den Namen des Verzeichnisses, in dem dBASE IV installiert ist, zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen die äquivalente Funktion SYS(2004) ein.

ID()XE "ID()"§

Stufe: 1

In dBASE: Gibt den Namen des aktuellen Benutzers aus, wenn dBASE in einem Mehrplatzsystem betrieben wird.

In FoxPro: Erzeugt einen Fehler.

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Vorgehen: Verwenden Sie stattdessen die Funktion SYS(0).

Anmerkung: SYS(0) gibt die Netzwerkkennung des Rechners (Rechnername und Rechnernummer) aus, sofern FoxPro in einem Netz betrieben wird. Dazu muß jedoch die Netz-Shell geladen und zuvor von der Netzsoftware eine Rechnernummer und ein Rechnername vergeben worden sein. In Novell-Netzen müssen Sie dazu das Login-Script durch folgenden Eintrag ergänzen:

```
MACHINE="%USER_ID,%P_STATION,  
%LOGIN_NAME"
```

Falls FoxPro nicht im Netz installiert ist oder falls im Netz weder eine Rechnernummer noch ein Rechnername vergeben wurden, liefert SYS(0) eine Zeichenkette mit einer Reihe von Leerzeichen (10 in FoxPro für MS-DOS bzw. 15 in FoxPro für Windows), die durch die Zeichenfolge Nummernzeichen (#), Leerzeichen und 0 abgeschlossen wird. Wenn Sie mit einer Einzelplatzversion von FoxPro arbeiten, gibt SYS(0) den Wert 1 aus.

INKEY()XE "INKEY()"§, LASTKEY()XE "LASTKEY()"§, READKEY()XE "READKEY()"§

Stufe: 3

In dBASE: Tasten und Tastenkombinationen haben FoxPro und dBASE nicht immer die gleichen Dezimalcodes.

In FoxPro: Diese Funktionen haben zwar dieselbe Funktionalität wie in dBASE, da FoxPro jedoch oft andere Tastencodes verwendet, müssen dBASE-Programme entsprechend verändert werden. Falls Sie sie nicht modifiziert in FoxPro ausführen, erhalten Sie oft andere Ergebnisse als in dBASE.

Vorgehen: FoxPro liest die Tastaturcodes aus einer Makrodatei mit der Erweiterung .FKY. Verwenden Sie die Makrodatei dBASE.FKY von der Programm-Konverter-Diskette, um dBASE-Tastencodes den entsprechenden FoxPro-

Tastencodes zuzuordnen. Fügen Sie danach den Befehl
RESTORE MACROS FROM dBASE.FKY in Ihr Programm
ein.

Sie können natürlich auch Ihren Programmcode verändern
und an den entsprechenden Stellen die FoxPro-Tastencodes
einsetzen. In Anhang G finden Sie eine Zusammenstellung
der Tastencodes von FoxPro, dBASE und Clipper.

ISBLANK()

XE "ISBLANK()"§Stufe: 2

In dBASE IV: Die Funktion ISBLANK liefert den Wahrheitswert .T., wenn
der Wert einer Variablen ein Nullwert ist.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie ISBLANK durch die FoxPro-Funktion EMPTY
oder verwenden Sie eine UDF aus FOXPROC.PRG

Anmerkung: Bei Feldern vom Datentyp Datum und Zeichen liefern
ISBLANK in dBASE und EMPTY in FoxPro die gleichen
Ergebnisse. Bei numerischen Feldern gibt EMPTY den
Wahrheitswert .T. zurück, wenn das Feld keinen Wert enthält
(Nullwert) oder den Wert 0 hat. Die dBASE-Funktion
ISBLANK liefert .T. immer dann zurück, wenn der Feldwert
ein Nullwert ist..

Wenn Sie in numerischen Feldern mit Nullwerten arbeiten
wollen, müssen Sie ein logisches Feld anlegen, das überprüft,
ob das numerische Feld einen Wert enthält oder leer ist.

FOXPROC.PRG enthält eine benutzerdefinierte Funktion
namens ISBLANK(), die es Ihnen erlaubt, Anweisungen mit
der dBASE-Funktion ISBLANK() ohne vorherige Änderung
ausführen zu lassen. Lesen Sie dazu bitte im Abschnitt
"Benutzerdefinierte Funktionen (UDFs) aufrufen" nach.

Beispiel: **dBASE**

null_status = ISBLANK(ung_datum)

FoxPro

 null_status = EMPTY(ung_datum)

ISMARKED()

Siehe Transaktionen.

ISMOUSE()XE "ISMOUSE()"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Liefert den Wert True (.T.) zurück, wenn ein Maustreiber installiert ist.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

Anmerkung: FoxPro verfügt über keine äquivalente Funktion.

KEYBOARDXE "KEYBOARD"§ CLEARXE "CLEAR"§

XE "KEYBOARD CLEAR "§Stufe: 2

In dBASE IV: Mit dem Schlüsselwort CLEAR werden alle noch gelesenen Zeichen im Tastaturpuffer gelöscht, bevor mit dem Befehl KEYBOARD neue Zeichen gespeichert werden.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie das Schlüsselwort CLEAR, und setzen Sie den Befehl CLEAR TYPEAHEAD vor den Befehl KEYBOARD.

Beispiel: **dBASE**

 KEYBOARD kund_name + adresse + plz + stadt

CLEAR

FoxPro

 CLEAR TYPEAHEAD

 KEYBOARD kund_name + adresse + plz + stadt

dBASE IV

Verzeichnis inkompatibler Befehle und Funktionen aus

KEYMATCH()XE "KEYMATCH()"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Sucht den angegebenen Indexeintrag in der spezifizierten Indexdatei, ohne dabei den aktuellen Index und ohne den Datensatzzeiger zu verändern.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

Anmerkung: FoxPro verfügt über keine äquivalente Funktion.

LIKE()

XE "LIKE()"§Stufe: 3

In dBASE IV: Bei der Ausführung des Vergleichs werden Leerzeichen, die hinter den Zeichenfolgen des Suchmusters und des Vergleichsausdrucks stehen, ignoriert.

In FoxPro: Alle Zeichen in Suchmuster und Vergleichsausdruck sind signifikant, auch nachstehende Leerzeichen.

Vorgehen: Verwenden Sie die Funktion RTRIM oder setzen Sie SET COMPATIBLE DB4 auf ON.

Beispiel: **dBASE**
LIKE(var1,var2)

FoxPro
LIKE(RTRIM(var1),var2)

LIST USERSXE "LIST USERS"§

Stufe: 2

In dBASE: Gibt die Kennung des Rechners, der aktuell im Netz angemeldet ist, aus.

In FoxPro: Wird nicht unterstützt. LIST USERS wird ignoriert.

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

Vorgehen: Keine besonderen Maßnahmen erforderlich.

Anmerkung: Siehe Anhang E.

LKSYS

Siehe Netzwerkfunktionen.

LOCK()XE "LOCK()"§, RLOCK()XE "RLOCK()"§

Stufe: 2

In dBASE IV: Sie können gemäß Standardeinstellung mit dieser Funktion gleichzeitig mehrere Datensätze sperren.

In FoxPro: In FoxPro können Sie mit LOCK() gemäß Standardeinstellung immer nur einen Datensatz sperren.

Vorgehen: Mit dem Befehl SET MULTLOCKS ON können Sie mehrere Datensätze sperren. Fügen Sie diesen Befehl in das Startprogramm für Netzwerkanwendungen ein.

Anmerkung: Wenn SET MULTLOCKS auf ON gesetzt ist, können Sie in FoxPro auf dieselbe Weise wie in dBASE IV mehrere Datensätze sperren.

Beispiel: **FoxPro**
SET MULTLOCKS ON

MEMORY()XE "MEMORY()"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Sie können der Funktion einen Parameter im Wertebereich 0 bis 7 übergeben. Die Werte 0 bis 7 stehen für bestimmte Speicherbereiche, die von der Funktion MEMORY() überprüft werden sollen.

In FoxPro: Parameterangaben resultieren in einem Fehler.

Vorgehen: Ersetzen Sie den Befehl MEMORY() durch den entsprechenden FoxPro-Befehl:

dBASE	FoxPro Äquivalent

Verzeichnis inkompatibler Befehle und Funktionen aus

dBASE IV

MEMORY(0)	SYS(1001)+SYS(1016)
MEMORY(1)	Kein Äquivalent vorhanden
MEMORY(2)	SYS(12)
MEMORY(3)	SYS(1001)
dBASE	FoxPro Äquivalent
MEMORY(4)	SYS(23)
MEMORY(5)	SYS(23)
MEMORY(6)	SYS(1016)
MEMORY(7)	Kein Äquivalent vorhanden

MESSAGEXE "MESSAGE"\$XE "@...GET MESSAGE"\$ (@GET, DEFINE BARXE "DEFINE BAR"\$, DEFINE MENUXE "DEFINE MENU"\$ DEFINE PADXE "DEFINE PAD"\$, DEFINE POPUPXE "DEFINE POPUP"\$)

Stufe: 2

In dBASE IV: MESSAGE-Ausdrücke werden im aktuellen Fenster ausgegeben.

In FoxPro: MESSAGE-Ausdrücke werden auf den Desktop ausgegeben.

Vorgehen: Sie können dieses Problem auf dreierlei Weise angehen:

a) Legen Sie mit dem Befehl SET MESSAGE WINDOW fest, in welches Fenster ein MESSAGE-Ausdruck ausgegeben werden soll.

b) Verkleinern Sie die Fenster um jeweils eine Zeile, damit die auf den Desktop ausgegebene Meldung sichtbar wird.

c) Arbeiten Sie mit dem Desktop statt mit einem Fenster (verwenden Sie die Befehle SAVE SCREEN und RESTORE SCREEN, um Fenster zu simulieren).

Anmerkung: Dieses Problem taucht nur dann auf, wenn ein Fenster definiert wurde, das die unterste Bildschirmzeile, in der Meldungen ausgegeben werden, überdeckt.

NETWORK()

XE "NETWORK()"§Stufe: 2

In dBASE IV: Diese Funktion liefert nur dann .T. zurück, wenn dBASE in einem Netzwerk läuft.

In FoxPro: In FoxPro 2.5 gibt die Funktion NETWORK immer .T. zurück. In FoxPro 2.0 gibt die Funktion NETWORK bei Einsatz der Einzelplatzversion den Wert .F. und bei Einsatz der Mehrplatzversion den Wert .T. zurück.

Vorgehen: Setzen Sie stattdessen die Funktion SYS(0) ein.

Anmerkung: Wenn FoxPro im Netz läuft, liefert SYS(0) die Netzwerkennung des Rechners (Rechnername und -nummer) zurück. Dazu muß jedoch die Netz-Shell geladen und zuvor von der Netzsoftware eine Rechnernummer und ein Rechnername vergeben worden sein. In Novell-Netzen müssen Sie dazu das Login-Script durch folgenden Eintrag ergänzen:

```
MACHINE="%USER_ID,%P_STATION,  
%LOGIN_NAME"
```

Falls FoxPro nicht im Netz installiert ist oder falls im Netz weder eine Rechnernummer noch ein Rechnername vergeben wurden, liefert SYS(0) eine Zeichenkette mit einer Reihe von Leerzeichen (10 in FoxPro für MS-DOS bzw. 15 in FoxPro für Windows), die durch die Zeichenfolge Nummernzeichen (#), Leerzeichen und 0 abgeschlossen wird. Wenn Sie mit einer Einzelplatzversion von FoxPro arbeiten, gibt SYS(0) den Wert 1 aus.

Beispiel: **dBASE**
in_netz = NETWORK()

FoxPro
in_netz = LEFT(SYS(0),10)<>SPACE(10)

Netzwerkfunktionen

XE "Netzwerkfunktionen"§Stufe: 1

In dBASE IV: dBASE IV kann mit Hilfe des Befehls CONVERT und den Funktionen CHANGE, LKSYS und USER feststellen, ob der aktuelle Datensatz im Netzwerkbetrieb verändert wurde.

In FoxPro: FoxPro ignoriert den Befehl CONVERT. Die Funktionen CHANGE, LKSYS und USER erzeugen Fehler.

Vorgehen: Löschen Sie die Funktionen CHANGE, LKSYS und USER. Routinen zur Anzeige von Datensatzsperrern, die auf diesen Funktionen basieren, müssen für FoxPro neu geschrieben

werden.

ON BARXE "ON BAR"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Führt einen Befehl aus, wenn der bezeichnete Popup-Menübalken *hervorgehoben* wird.

In FoxPro: Erlaubt die Ausführung von ACTIVATE POPUP oder ACTIVATE MENU Anweisungen, wenn der bezeichnete Popup-Menübalken *markiert* ist.

Vorgehen: Falls andere Anweisungen als ACTIVATE POPUP oder ACTIVATE MENU ausgeführt werden sollen, ersetzen Sie den Befehl ON BAR in FoxPro durch ON SELECTION BAR.

ON EXIT BARXE "ON EXIT BAR"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Führt einen Befehl aus, wenn der Cursor (Hervorhebung) vom bezeichneten Menübalken *wegbewegt* wird.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Befehl.

Anmerkung: FoxPro verfügt über keinen äquivalenten Befehl.

ON EXIT MENUXE "ON EXIT MENU"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Führt einen Befehl aus, wenn der Cursor (Hervorhebung) vom bezeichneten Menü *weg bewegt* wird.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Befehl.

Anmerkung: FoxPro verfügt über keinen äquivalenten Befehl.

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

ON EXIT PAD

Stufe: 2

In dBASE IV: Führt einen Befehl aus, wenn der Cursor (Hervorhebung)
vom bezeichneten Menübalken weg bewegt wird.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Befehl.

Anmerkung: FoxPro verfügt über keinen äquivalenten Befehl.

ON EXIT POPUPXE "ON EXIT POPUP" § (nur in Version 2.0)

Stufe: 2

In dBASE IV: Führt einen Befehl aus, wenn der Cursor (Hervorhebung) vom bezeichneten Menübalken wegbewegt wird.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Befehl.

Anmerkung: FoxPro verfügt über keinen äquivalenten Befehl.

ON KEY LABELXE "ON KEY LABEL" §

Stufe: 3

In dBASE IV: Verwendet das Zeichen "-" in Bezeichnungen von Tastenkombinationen (zum Beispiel: Ctrl-F1).

In FoxPro: Verwendet das Zeichen "+" in Bezeichnungen von Tastenkombinationen (zum Beispiel: Ctrl+F1).

Vorgehen: Suchen Sie alle Bindestriche ("-") in Tastenbezeichnungen, und ersetzen Sie diese durch Pluszeichen ("+").

Anmerkung: Die Tastenbezeichnungen von dBASE und FoxPro sind identisch, nur das Verbindungszeichen für Tastenkombinationen unterscheidet sich..

Beispiel: **dBASE**
ON KEY LABEL ALT-A DO meinprog.prg

FoxPro
ON KEY LABEL ALT+A DO meinprog.prg
&& change "-" to "+"

ON MENUXE "ON MENU" § (nur in Version 2.0)

Stufe: 2

In dBASE IV: Führt einen Befehl aus, sobald ein Popup-Menübalken, der nicht über ON PAD mit der Option eines Zeilenmenüs

Programm-Konverter

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

verbunden wurde, hervorgehoben wird.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie ON MENU in FoxPro durch ON SELECTION MENU.

Anmerkung: Der FoxPro-Befehl ON SELECTION MENU hat dieselbe Funktionalität wie der Befehl ON MENU in dBASE IV, einziger Unterschied ist, daß ON SELECTION MENU den bezeichneten Befehl ausführt, nachdem ein Balken *ausgewählt* wurde, wogegen ON MENU den bezeichneten Befehl ausführt, sobald ein Balken *hervorgehoben* wurde.

Beispiel: **dBASE**
ON MENU File DO mein_prog

FoxPro
ON SELECTION MENU File DO mein_prog

ON MOUSEXE "ON MOUSE" § (nur in Version 2.0)

Stufe: 2

In dBASE IV: Führt den angegebenen Befehl aus, wenn die linke Maustaste gedrückt wird.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie ON MOUSE durch ON KEY LABEL LEFTMOUSE.

Anmerkung: ON MOUSE wird in dBASE IV normalerweise bei der Definition von benutzerfreundlichen Dialogelementen, wie Kontrollkästchen und Optionsfelder, eingesetzt. Obwohl diese dBASE IV-Routinen in FoxPro problemlos ausgeführt werden, ist es unter Umständen sinnvoll, anstelle dieser Dialogelemente mit der FoxPro-Funktion @..GET Schaltflächen, Optionsfelder, Kontrollkästchen, Listenfelder, etc. neu zu definieren.

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Beispiel: **dBASE**

ON MOUSE DO maus_proc

FoxPro

ON KEY LABEL LEFTMOUSE DO maus_proc

ON PADXE "ON PAD" § (nur in Version 2.0)

Stufe: 2

In dBASE IV: Führt einen beliebigen Befehl aus, wenn der bezeichnete Popup-Menübalken *hervorgehoben* wird.

In FoxPro: Erlaubt die Ausführung von ACTIVATE POPUP oder ACTIVATE MENU Anweisungen, wenn der bezeichnete Popup-Menübalken *ausgewählt* wurde.

Vorgehen: Falls andere Anweisungen als ACTIVATE POPUP oder ACTIVATE MENU ausgeführt werden sollen, ersetzen Sie den Befehl ON POPUP in FoxPro durch ON SELECTION POPUP.

Anmerkung: Der FoxPro-Befehl ON SELECTION PAD hat dieselbe Funktionalität wie der Befehl ON PAD in dBASE IV, einziger Unterschied ist, daß ON SELECTION PAD den bezeichneten Befehl ausführt, nachdem ein Balken *ausgewählt* wurde, wogegen ON PAD den bezeichneten Befehl ausführt, sobald ein Balken *hervorgehoben* wurde.

Beispiel: **dBASE**

ON PAD tuwas OF type DO tuwas_pad

FoxPro

ON SELECTION PAD tuwas OF type DO

tuwas_pad

ON POPUPXE "ON POPUP" § (nur in Version 2.0)

Stufe: 2

In dBASE IV: Führt einen Befehl aus, sobald ein Popup-Menübalken, der nicht über ON PAD mit der Option eines Zeilenmenüs

Programm-Konverter

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

verbunden wurde, hervorgehoben wird.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie ON POPUP in FoxPro durch ON SELECTION POPUP.

Anmerkung: Der FoxPro-Befehl ON SELECTION POPUP hat dieselbe Funktionalität wie der Befehl ON POPUP in dBASE IV, einziger Unterschied ist, daß ON SELECTION POPUP den bezeichneten Befehl ausführt, nachdem ein Balken *ausgewählt* wurde, wogegen ON POPUP den bezeichneten Befehl ausführt, sobald ein Balken *hervorgehoben* wurde.

Beispiel: **dBASE**
ON POPUP just DO it

FoxPro
ON SELECTION POPUP just DO it

ON SELECTION POPUP

XE "ON SELECTION POPUP"§Stufe: 3

In dBASE IV: Datenausgaben oder Fenster, die von der vom Befehl ON SELECTION POPUP aufgerufenen Routine aktiviert wurden, können das Popup-Menü überdecken. Nachdem dieser Befehl oder diese Routine ausgeführt wurde, wird das Popup-Menü wieder angezeigt.

In FoxPro: Während FoxPro die im Befehl ON SELECTION POPUP genannte Prozedur ausführt, bleibt das Popup-Menü über Fenstern oder Datenausgaben sichtbar.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

Anmerkung: Sie können bei Bedarf den Befehl HIDE POPUP in die Routine einfügen, die von ON SELECTION POPUP aufgerufen wird. Wenn der Befehl oder die Routine ausgeführt wurde, wird das Popup-Menü automatisch wieder angezeigt, und die Programmsteuerung wird an das Popup-

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Menü zurückgegeben.

Sie können auch SET COMPATIBLE auf ON setzen, damit der Befehl in derselben Weise wie in dBASE ausgeführt wird.

Beispiel:

dBASE

```
ON SELECTION POPUP edit_pop DO edit_proc
```

```
PROCEDURE edit_proc  
<Definition der Prozedur>
```

FoxPro

```
ON SELECTION POPUP edit_pop DO edit_proc
```

```
PROCEDURE edit_proc  
HIDE POPUP edit_pop  
<Definition der Prozedur>  
SHOW POPUP edit_pop
```

ON SELECTION POPUP BLANK

XE "ON SELECTION POPUP BLANK"§Stufe: 2

In dBASE IV: Mit dem optionalen Schlüsselwort BLANK kann vor der Ausführung eines Befehls das Popup-Menü vom Bildschirm entfernt werden. Nachdem der Befehl ausgeführt wurde, wird das Popup-Menü erneut angezeigt.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie das Schlüsselwort BLANK.

Anmerkung: Siehe ON SELECTION POPUP.

PADPROMPT()XE "PADPROMPT()"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Liefert den Text der Eingabeaufforderung des angegebenen Menübalkens im bezeichneten Menü bzw. des aktuellen Menüs, falls kein Menü spezifiziert wurde.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie PADPROMPT() durch PRMPAD().

Anmerkung: Die FoxPro-Funktion PRMPAD() verlangt die Angabe eines Menüs.

Beispiel: **dBASE**

```
pr_txt=PADPROMPT("Datei")
```

FoxPro

```
pr_txt=PRMAD("menu_1","Datei")
```

PCOUNT()XE "PCOUNT()"§ (nur in Version 1.5)

Stufe: 2

In dBASE IV: Die Funktion PCOUNT ermittelt die Anzahl von Parametern, die einer benutzerdefinierten Funktion übergeben wurden.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie PCOUNT durch die Funktion PARAMETERS oder verwenden Sie eine UDF aus FOXPROC.PRG.

Anmerkung: FOXPROC.PRG enthält eine benutzerdefinierte Funktion namens PCOUNT(), die es Ihnen erlaubt, Anweisungen mit der dBASE-Funktion PCOUNT() ohne vorherige Änderung ausführen zu lassen. Lesen Sie dazu bitte im Abschnitt "Benutzerdefinierte Funktionen (UDFs) aufrufen" nach.

Beispiel: **dBASE**

```
param_zahl = PCOUNT()
```

FoxPro

param_zahl = PARAMETERS()

PROTECT

Siehe Datensicherheit

REPLACE FROM ARRAY

XE "REPLACE FROM ARRAY"§Stufe: 2

In dBASE IV: Ersetzt den Inhalt eines oder mehrerer Felder des aktuellen Datensatzes durch Werte aus einem zweidimensionalen Array.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Verwenden Sie stattdessen die Befehle SCATTER und GATHER.

Anmerkung: Der Befehl SCATTER kopiert die Daten eines Feldes in ein Speichervariablenarray oder eine Speichervariablengruppe. GATHER kopiert die Daten dieser Speichervariablen in die aktuelle Tabelle. Diese Befehle sind schneller auszuführen als COPY TO und REPLACE FROM, zudem werden die von SCATTER erstellten Speichervariablen nach den Feldnamen benannt und sind damit leichter und sicherer zu handhaben, als mit Nummern versehene Einträge eines Arrays.

Falls mit diesem Befehl der Feldinhalt in mehreren Datensätzen ersetzt wird, lassen Sie den Befehl GATHER wiederholt ausführen.

Beispiel: **dBASE**

```
COPY TO ARRAY rec_array  
<<@ SAY- und GET-Anweisungen>>  
READ  
REPLACE FROM ARRAY rec_array
```

FoxPro

```
SCATTER MEMVAR  
<<@ SAY- und GET-Anweisungen>>  
READ  
GATHER MEMVAR
```

RESET

Siehe Transaktionen.

RESTORE FROMXE "RESTORE FROM"§

Stufe: 3

In dBASE IV: Falls keine Dateierweiterung angegeben wird, wird die Erweiterung .MEM angehängt.

In FoxPro: Falls keine Dateierweiterung angegeben wird, sucht FoxPro nach einem Dateinamen ohne Erweiterung.

Vorgehen: Geben Sie eine Dateierweiterung an.

RESTORE SCREENXE "RESTORE SCREEN"§

Stufe: 2

In dBASE IV: Der Befehl SAVE SCREEN speichert den Bildschirminhalt im Hauptspeicher und nicht in einer Speichervariablen. RESTORE SCREEN kann an jeder beliebigen Stelle des Programms aufgerufen werden, der Befehl zeigt dann einen zuvor gespeicherten Bildschirminhalt an.

In FoxPro: Der Befehl SAVE SCREEN speichert den Bildschirminhalt in einer Speichervariablen. Die Prozedur oder Routine, die den Befehl RESTORE SCREEN aufruft, muß auf diese Speichervariable zugreifen können.

Vorgehen: Wenn SAVE SCREEN und RESTORE SCREEN in der gleichen Prozedur oder Funktion aufgerufen werden, sind keine Änderungen erforderlich. Ist dies nicht der Fall, deklarieren Sie die Speichervariable vor dem Aufruf des Befehls SAVE SCREEN als PUBLIC und machen sie damit global verfügbar.

Anmerkung: Falls der Bildschirminhalt nicht in einer globalen Speichervariablen gespeichert wird, erzeugt der Aufruf von RESTORE SCREEN die Fehlermeldung "Variable nicht gefunden", sofern der Aufruf nicht von der Prozedur oder Funktion, die den Befehl SAVE SCREEN aufruft, getätigt

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

wurde.

Beispiel: **dBASE**

SAVE_SCREEN TO screen_var

FoxPro

PUBLIC screen_var

SAVE SCREEN to screen_var

RLOCK()

Siehe LOCK()

ROLLBACK

Siehe Transaktionsbefehle.

ROLLBACK ()

Siehe Transaktionsbefehle.

RUN()XE "RUN()"§

Stufe: 2

In dBASE: Führt ein externes Programm aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen den Befehl RUN ein.

SELECT()

XE "SELECT()"§Stufe: 2

In dBASE IV: In Version 1.1 nimmt diese Funktion keine Parameter an, sie ermittelt lediglich die Nummer des höchsten nicht benutzten Arbeitsbereichs.

In Version 1.5 ermittelt sie die Nummer des niedrigsten nicht benutzten Arbeitsbereichs, sofern sie ohne Parameter aufgerufen wurde. Wenn ein Aliasname als Parameter übergeben wurde, liefert sie die Nummer des Arbeitsbereichs, in dem die Tabelle mit dem angegebenen Aliasnamen geöffnet ist.

Falls der aktuelle Arbeitsbereich durch eine benutzerdefinierte Funktion (UDF) verändert wurde, wird nach Beendigung der UDF der Status des Arbeitsbereichs automatisch wiederhergestellt.

In FoxPro: SELECT() und SELECT(0) geben die Nummer des gewählten Arbeitsbereichs aus.

SELECT(1) gibt die Nummer des höchsten nicht benutzten Arbeitsbereichs aus.

Falls der aktuelle Arbeitsbereich durch eine

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

benutzerdefinierte Funktion (UDF) verändert wurde, wird in FoxPro nach Beendigung der UDF der veränderte Arbeitsbereich zum aktuellen Arbeitsbereich, sofern nicht explizit ein anderer Arbeitsbereich gewählt wird.

Vorgehen: Wenn Sie mit SELECT eine Tabelle in einem bestimmten Arbeitsbereich öffnen wollen, sollten Sie den Ausdruck in SELECT(1) abändern, um zu vermeiden, daß im aktuellen Arbeitsbereich geöffnete Tabellen geschlossen werden.

Verwenden Sie den Befehl SELECT, um den Arbeitsbereich einer bestimmten Datenbank auszuwählen.

Sie können auch SET COMPATIBLE auf ON setzen, damit der Befehl SELECT() dieselbe Wirkung wie in dBASE hat.

Anmerkung: Falls Ihr Programm sich auf die Auswahl numerierter Arbeitsbereiche stützt und die Werte von SELECT abfragt, müssen Sie wahrscheinlich Programmabläufe neu definieren, um dem Umstand Rechnung zu tragen, daß die Funktion SELECT in FoxPro nicht den niedrigsten nicht benutzten Arbeitsbereich, sondern den höchsten ermittelt.

Beispiel: **dBASE**
 SELECT()

FoxPro
 SELECT(1)

SET("ATTRIBUTES")XE "SET(\ "ATTRIBUTES\ ") "§

Stufe: 2

In dBASE IV: SET("ATTRIBUTES") liefert eine Zeichenfolge zurück, die sich aus den mit dem Befehl SET COLOR OF eingestellten Farbcodes der sieben Teilbereiche des Bildschirms zusammensetzt.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Bei Anwendungen, die mit dem dBASE-

Anwendungsgenerator erstellt wurden, brauchen diese Befehlszeilen normalerweise einfach nur als Kommentar gekennzeichnet und damit deaktiviert zu werden.

Falls ein Programm von Farbeinstellungen abhängig ist und diese mit dem Befehl SET COLOR TO ermittelt, sollten Sie eine Routine schreiben, die nach Aufruf des Befehls SET COLOR TO die hier definierten Farben in Speichervariablen speichert. Sie können diese Speichervariablen dann später abfragen und können damit den Einsatz der Funktion SET("ATTRIBUTES") umgehen.

Anmerkung: SET COLOR TO wird in FoxPro genauso interpretiert wie in dBASE, daher zeigen auch Programme, die Farbeinstellungen verändern, für gewöhnlich dieselben Resultate.

Wenn Sie die Funktionen SYS(2001,'COLOR') oder SET("COLOR") verwenden, gibt FoxPro die Farbcodes in dBASE II-Notation aus.

SET("BORDER")XE "SET(\\"BORDER\\")"§

Stufe: 3

In dBASE IV: Liefert die Zeichenfolge zur Definition des Rahmentyps.

In FoxPro: Liefert eine Zeichenfolge mit den Rahmenzeichen.

Vorgehen: Analysieren Sie die zurückgegebene Zeichenfolge, und ermitteln Sie mit der Funktion ASC() den ASCII-Code der Zeichen. Überprüfen Sie, ob es sich um Zeichen des einfachen oder erweiterten ASCII-Zeichensatzes oder um Grafikzeichen handelt.

SET CATALOG

XE "SET CATALOG"§Stufe: 1

In dBASE: Eine Katalogdatei ist eine Tabelle mit der Erweiterung .CAT und einem festgelegten Datensatzformat, in der jede zu einer

Verzeichnis inkompatibler Befehle und Funktionen aus
dBASE IV

Anwendung gehörige Datei in einem Datensatz verzeichnet wird. dBASE unterstützt die folgenden Befehle und Funktionen zur Erstellung und Bearbeitung von Katalogdateien:

SET CATALOG TO <Dateiname> legt eine Katalogdatei an oder öffnet eine Katalogdatei.

SET CATALOG ON | OFF aktiviert bzw. deaktiviert die Aktualisierung der Katalogdatei.

In FoxPro: Wird nicht unterstützt. Katalogbefehle werden ignoriert.

Vorgehen: Suchen Sie mit ADIR() nach Dateien, die einem bestimmten Dateimuster entsprechen sollen.

SET("CATALOG")XE "SET(\\"CATALOG\\")"§

Stufe: 2

In dBASE IV: Gibt ON oder OFF aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion, sie wird in FoxPro nicht unterstützt.

SET COLOR TO

XE "SET COLOR TO"§Stufe: 3

In dBASE IV: SET COLOR TO (ohne Angabe von Farbcodes für Vorder- und Hintergrundfarbe) stellt die Bildschirmdarstellung auf den Monochrommodus um.

In FoxPro: SET COLOR TO läßt den aktuellen Farbsatz unverändert.

Vorgehen: Keine besonderen Maßnahmen erforderlich. Wenn Sie in FoxPro zur Monochromdarstellung umschalten wollen, verwenden Sie die Routine im nachstehenden Beispiel.

Beispiel: **dBASE**
SET COLOR TO

FoxPro
SET COLOR TO W/N, N/W, N
or
SET COLOR SCHEME TO monochrome

SET DBTRAPXE "SET DBTRAP" § (nur Versionen 1.1 und 1.5)

Stufe: 1

In dBASE IV: Wenn DBTRAP auf ON gestellt ist, können benutzerdefinierte Funktionen und Interrupt-Routinen bestimmte Befehle und Funktionen nicht ausführen.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

Anmerkung: Sie sollten in Ihre benutzerdefinierte Funktion Routinen zum Speichern und Wiederherstellen der Umgebung einfügen. FoxPro verläßt sich darauf, daß in benutzerdefinierten Funktionen oder Interrupt-Routinen keine Befehle, die die sie aufrufende Funktion oder Prozedur unterbrechen, wie beispielsweise PACK oder MODIFY STRUCTURE, aufgerufen werden.

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

SET DESIGNXE "SET DESIGN"§

Stufe: 2

In dBASE IV: Mit diesem Befehl können Programmentwickler verhindern, daß Benutzer Tabellen, Berichte, Masken, Abfragen, Etiketten oder Anwendungen erstellen oder verändern können.

In FoxPro: Wird nicht unterstützt. SET DESIGN wird ignoriert.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

Anmerkung: Sie können das FoxPro-Menüsystem leicht verändern und die Menüoptionen Neu und Öffnen aus dem Datei-Menü entfernen oder deaktivieren, um zu verhindern, daß Benutzer Dateien erstellen oder verändern.

SET("DESIGN")XE "SET(\"DESIGN\")"§

Stufe: 2

In dBASE IV: Gibt ON oder OFF aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

Anmerkung: Siehe Befehl SET DESIGN.

SET DIRECTORY

XE "SET DIRECTORY"§Stufe: 2

In dBASE IV: Legt den Suchpfad (Laufwerk und Verzeichnis) des Standardverzeichnisses fest.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Verwenden Sie stattdessen den Befehl SET DEFAULT.

Anmerkung: Die Befehle SET DEFAULT (FoxPro) und SET DIRECTORY (dBASE) sind äquivalent. Dem FoxPro-Befehl SET DEFAULT kann eine Pfadangabe mit

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Unterverzeichnissen übergeben werden. In dBASE IV wird nur eine Laufwerksangabe und ein Verzeichnisname akzeptiert.

Beispiel: **dBASE**
SET DIRECTORY TO c:\data

FoxPro
SET DEFAULT TO c:\data

SET("DIRECTORY")XE "SET(\\"DIRECTORY\\")"§

Stufe: 2

In dBASE IV: Liefert den Suchpfad des Standardverzeichnisses zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie die Funktion SET("DEFAULT") durch die Funktion SYS(2003).

Anmerkung: Die Funktionen SET("DEFAULT") (FoxPro) und SET("DIRECTORY") (dBASE) sind äquivalent. Dem FoxPro-Befehl SET DEFAULT kann eine Pfadangabe mit Unterverzeichnissen übergeben werden. In dBASE IV wird nur eine Laufwerksangabe und ein Verzeichnisname akzeptiert.

Beispiel: **dBASE**
akt_dir = SET("DIRECTORY")

FoxPro
akt_dir = SET("DEFAULT")+ SYS(2003)

SET("DISPLAY")XE "SET(\\"DISPLAY\\")"§

Stufe: 2

In dBASE IV: Liefert den aktuellen Anzeigemodus zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Speichern Sie die aktuelle Einstellung des Anzeigemodus in

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

einer Variablen, bevor Sie den Anzeigemodus verändern.

Beispiel: **FoxPro**

Anzeigemodus speichern

```
vmd=LEFT(SYS(2006),AT("/",SYS(2006))-1)
+STR(SROWS(),2)
SET DISPLAY TO VGA50
```

Anzeigemodus wiederherstellen

```
SET DISPLAY TO &vmd
```

SET ENCRYPTION und SET("ENCRYPTION")

Siehe Datensicherheit.

SET FIELDS

XE "SET FIELDS"§Stufe: 1

In dBASE IV: a) SET FIELDS definiert eine Liste mit Feldern, auf die in einer oder mehreren Tabellen zugegriffen werden kann.

b) Wenn keine Feldliste mit SET FIELDS TO angegeben wird, wird SET FIELDS auf OFF gesetzt.

c) SET FIELDS /r öffnet ein Feld nur zum Lesen, nicht zum Schreiben.

In FoxPro: a) Die mit dem Befehl SET FIELDS definierte Feldliste darf nur Felder aus einer Tabelle enthalten, allerdings kann für jeden Arbeitsbereich eine eigene Feldliste definiert werden.

b) Wenn SET FIELDS TO ohne Parameter aufgerufen wird, wird eine leere Zeichenfolge als Feldliste eingesetzt.

c) Die in einem BROWSE Befehl genannten Felder können mit einer Schreibsperre versehen werden, nicht jedoch die Felder der mit dem Befehl SET FIELDS festgelegten Feldliste.

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Vorgehen: a) Speichern Sie die Feldliste in einer Speichervariablen und lassen Sie mit Hilfe eines Makros daraus eine Befehlsfolge generieren.

oder

Ersetzen Sie SET FIELDS Befehle, deren Feldliste Felder aus verschiedenen Tabellen enthält, durch entsprechend viele SET FIELDS Befehle mit den Feldern nur einer Tabelle.

b) Stellen Sie SET DB4 COMPATIBLE auf ON, wenn der Aufruf von SET FIELDS ohne Feldliste die Einstellung von SET FIELDS auf OFF bewirken soll.

Beispiel: **dBASE**

```
SET FIELDS TO kunde -> kund_id,  
auftrag -> auftr_menge  
LIST
```

FoxPro

```
fld_strng = "kunde.kund_id, auftrag.auftr_menge"  
LIST &fld_strng
```

oder

```
SET FIELDS TO kunde.auftr_menge  
LIST  
SET FIELDS TO auftrag.kund_id  
LIST
```

SET FORMAT

XE "SET FORMAT"§Stufe: 2

In dBASE IV: dBASE-Formatdateien (.FMTs) können neben @SAY- und @GET-Anweisungen Befehle zur Initialisierung zum Freigabe von Speichervariablen enthalten.

In FoxPro: Formatdateien, die nicht nur @SAY- und @GET-Anweisungen enthalten, erzeugen einen Fehler.

Vorgehen: Verlagern Sie Routinen zur Initialisierung und zum Schließen Programm-Konverter

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

der Formatdatei in eine separate Datei.

Anmerkung: Wenn Sie eine .SCR -Datei zum Format .SCX konvertieren, ersetzen Sie alle Befehle, die zum Öffnen und Bearbeiten der Formatdatei dienen (EDIT, CHANGE, APPEND oder READ) und die die Formatdatei nach der Bearbeitung schließen, und rufen Sie stattdessen den Befehl DO mit dem Namen der erstellten Maskendatei auf.

SET IBLOCK TO (nur in Version 2.0)

Stufe: 2

In dBASE IV: Verändert die Blockgröße von Indexdateien, damit Daten schneller bearbeitet werden können.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie den Befehl SET IBLOCK. Er ist in FoxPro nicht erforderlich.

Anmerkung: Da die Blockgröße keinen Einfluß auf die in FoxPro verwendete Rushmore-Optimierung hat, ist dieser Befehl in FoxPro überflüssig.

SET INSTRUCT

XE "SET INSTRUCT"§Stufe: 3

In dBASE IV: Öffnet bei Auswahl einer Datei oder eines Menüs im Regiezentrum ein Dialogfenster mit Bedienungshinweisen.

In FoxPro: Wird nicht unterstützt. SET INSTRUCT wird ignoriert.

Vorgehen: Löschen Sie diesen Befehl.

SET("INSTRUCT")XE "SET("INSTRUCT")"§

Stufe: 2

In dBASE IV: Liefert ON oder OFF zurück.

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

SET KEYXE "SET KEY"§ (nur in Version 1.5)

Stufe: 2

In dBASE IV: Zeigt nur diejenigen Datensätze an, deren Index eine bestimmte Bedingung erfüllt. Bei der Ausführung dieses Befehls wird die Tabelle nicht sequentiell von oben nach unten durchsucht, sondern mit Indexschlüsseln gearbeitet.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Befehl, und verwenden Sie stattdessen den Befehl SET FILTER.

Anmerkung: SET FILTER arbeitet auch mit Indexschlüsseln. Wenn der Filterausdruck optimiert werden kann, wird die Ausführung durch Rushmore™ sogar noch beschleunigt.

Beispiel: **dBASE**

```
SET ORDER TO TAG PLZ  
SET KEY TO "94000","94999"
```

FoxPro

```
SET FILTER TO plz >= "94000" AND plz <=  
"94999"
```

SET LDCHECKXE "SET LDCHECK"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Aktiviert bzw. deaktiviert die Überprüfung der ID-Kennung des Treibers für Ländereinstellungen.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Befehl.

Anmerkung: FoxPro verfügt über keinen äquivalenten Befehl. In Version FoxPro 2.5a werden länderspezifische

Einstellungen, unterschiedliche Zeichensatztabellen und
mehrere Sortierfolgen unterstützt.

SET LIBRARYXE "SET LIBRARY"§ (nur in Version 1.5)

Stufe: 1

In dBASE IV: Öffnet eine besondere Prozedurendatei, die geöffnet bleibt, auch wenn mit dem Befehl SET PROCEDURE TO weitere Prozedurendateien geöffnet werden.

In FoxPro: SET LIBRARY TO öffnet eine FoxPro-API-Bibliothek. Wenn im Befehlsparameter keine Dateierweiterung angegeben wird, wird die Fehlermeldung "Datei ist nicht vorhanden" ausgegeben. Falls der Dateiname vollständig angegeben wurde, die genannte Datei jedoch keine API-Bibliothek ist, wird die Fehlermeldung "Ungültige .LIB-Datei" ausgegeben.

Mit dem FoxPro-Befehl SET PROCEDURE TO können Sie nur eine Prozedurendatei geöffnet halten.

Vorgehen: Setzen Sie anstelle von SET LIBRARY TO den Befehl SET PROCEDURE TO. (Beachten Sie, daß SET PROCEDURE andere offene Prozedurendateien schließt, bevor die bezeichnete Prozedurendatei geöffnet wird.) Wenn die Routinen der dBASE IV-Prozedurenbibliothek während des gesamten Programmablaufs einer Anwendung verfügbar sein sollen, fügen Sie diese Routinen in die Prozedurendatei ein, die Sie mit SET PROCEDURE TO öffnen, oder verlagern Sie sie in den Initialisierungsteil der Anwendung.

SET("LIBRARY")XE "SET(\\"LIBRARY\\")"§ (nur in Version 1.5)

Stufe: 2

In dBASE IV: Liefert den Namen der geöffneten Prozedurendatei zurück.

In FoxPro: Wird nicht unterstützt. SET ("LIBRARY") wird ignoriert.

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Vorgehen: Setzen Sie statt SET ("LIBRARY") die Funktion SET ("PROCEDURE") ein.

Anmerkung: Siehe SET LIBRARY weiter oben.

Beispiel: **dBASE**

```
akt_lib = SET("LIBRARY")
```

FoxPro

```
akt_lib = SET("PROCEDURE")
```

SET MBLOCKXE "SET MBLOCK"§ (nur in Version 2.0)

Stufe: 2

In dBASE IV: Legt die Standardblockgröße für neue Memofelder fest.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie SET MBLOCK durch SET BLOCKSIZE, oder löschen Sie den Befehl SET MBLOCK.

Anmerkung: Der Befehl SET MBLOCK aus dBASE IV hat eine ähnliche Funktion wie der Befehl SET BLOCKSIZE, der sowohl in FoxPro als auch in dBASE IV verfügbar ist. Der einzige Unterschied besteht darin, daß mit SET MBLOCK die Blockgröße von Memofeldern unabhängig von der Blockgröße von Indexdateien (SET IBLOCK) definiert werden kann, wogegen SET BLOCKSIZE aus dBASE IV sowohl die Blockgröße von Memofeldern als auch die von Indexdateien festlegt. Da die Blockgröße keinen Einfluß auf die in FoxPro verwendete Rushmore-Optimierung hat und da der Befehl SET BLOCKSIZE in FoxPro die Blockgröße von Indexdateien nicht verändert, ist der Befehl SET MBLOCK überflüssig.

SET MESSAGEXE "SET MESSAGE"§

Stufe: 3

In dBASE IV: dBASE unterstützt die Option AT, mit der die

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Ausgabeposition von Meldungen festgelegt werden kann.

In FoxPro: FoxPro ignoriert das Schlüsselwort AT.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

Anmerkung: Sie können statt AT eines der FoxPro-Schlüsselwörter LEFT, CENTER oder RIGHT zur Angabe der horizontalen Bildschirmposition verwenden.

SET PAUSEXE "SET PAUSE"§

Stufe: 3

In dBASE IV: Steuert die Anzeige des SQL-Befehls SELECT und unterbricht die Datenausgabe nach jeder vollen Bildschirmseite.

In FoxPro: SET PAUSE wird ignoriert. In FoxPro erfolgt die Datenausgabe gemäß Voreinstellung immer so, als sei PAUSE auf ON gestellt.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

SET("PAUSE")XE "SET(\"PAUSE\")"§

Stufe: 2

In dBASE IV: Liefert ON oder OFF zurück.

In FoxPro: Erzeugt einen Fehler. In FoxPro erfolgt die Datenausgabe gemäß Voreinstellung immer so, als sei PAUSE auf ON gestellt.

Vorgehen: Löschen Sie diese Funktion.

SET PRECISIONXE "SET PRECISION"§

Stufe: 3

In dBASE IV: Legt in Dezimalstellen die Rechengenauigkeit für mathematische Operationen fest, zulässig sind Werte

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

zwischen 10 und 20. Der voreingestellte Wert ist 16.

In FoxPro: Wird nicht unterstützt. SET PRECISION wird ignoriert.

Vorgehen: Löschen Sie diesen Befehl.

Anmerkung: In FoxPro werden Berechnungen immer mit einer Genauigkeit von 16 Dezimalstellen ausgeführt, die Rechengenauigkeit kann hier nicht eingestellt werden.

SET("PRECISION")XE "SET(\\"PRECISION\\")"§

Stufe: 2

In dBASE IV: Gibt die Anzahl von Dezimalstellen aus, mit der mathematische Berechnungen ausgeführt werden.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

Anmerkung: In FoxPro werden Berechnungen immer mit einer Genauigkeit von 16 Dezimalstellen ausgeführt, die Rechengenauigkeit kann hier nicht eingestellt werden.

SET PRINTER TO FILEXE "SET PRINTER TO FILE"§

Stufe: 3

In dBASE IV: Falls keine Dateierweiterung angegeben wird, erstellt dBASE eine Datei mit der Erweiterung .PRT.

In FoxPro: Falls keine Dateierweiterung angegeben wird, erstellt FoxPro eine Datei ohne Erweiterung.

Vorgehen: Keine besonderen Maßnahmen erforderlich. Wenn SET DB4 COMPATIBLE auf ON esetzt ist, werden Druckdateien standardmäßig mit der Erweiterung .PRT ausgestattet.

SET("SQL")XE "SET(\\"SQL\\")"§

Stufe: 2

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

In dBASE IV: Liefert ON oder OFF zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

Anmerkung: SQL ist in FoxPro integriert und muß daher nicht ein- oder ausgeschaltet werden. Siehe SQL.

SET TITLEXE "SET TITLE"§

Stufe: 3

In dBASE IV: Blendet die Eingabeaufforderung für eine Beschreibung der Katalogdatei ein und aus.

In FoxPro: Wird nicht unterstützt. SET TITLE wird ignoriert.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

Anmerkung: Katalogdateien werden in FoxPro nicht unterstützt.

SET("TITLE")XE "SET(\\"TITLE\\")"§

Stufe: 3

In dBASE IV: Liefert ON oder OFF zurück.

In FoxPro: Liefert immer den Wert False (Falsch) zurück.

Vorgehen: Keine besonderen Maßnahmen erforderlich.

Anmerkung: Katalogdateien werden in FoxPro nicht unterstützt.

SET TRAP

XE "SET TRAP"§Stufe: 3

In dBASE IV: Bei der Einstellung ON wird der Debugger aktiviert, sobald Sie die Taste ESC drücken.

In FoxPro: Wird nicht unterstützt. SET TRAP wird ignoriert.

Vorgehen: Keine besonderen Maßnahmen erforderlich. Öffnen Sie das

Trace- und das Debug-Fenster, bevor Sie Programme ausführen lassen, und verfolgen Sie die Programmausführung mit Hilfe dieser Fenster.

Anmerkung: Da FoxPro eine fensterorientierte Anwendung ist, können Sie Ihre Anwendung, das Trace- und das Debug-Fenster nebeneinander anzeigen lassen. Sie brauchen also nicht zwischen Ihrer Anwendung und dem Debug-Fenster hin- und herzuschalten.

Mit dem Befehl SET ECHO können Sie einen ähnlichen Effekt wie mit SET TRAP erzielen. Sobald ein Fehler auftritt, wird in diesem Fall das Trace-Fenster eingeblendet.

Beispiel: **FoxPro**
ON ERROR SET ECHO ON

SET("TRAP")XE "SET("TRAP")"§

Stufe: 2

In dBASE IV: Liefert ON oder OFF zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Öffnen Sie das Trace- und das Debug-Fenster, bevor Sie Programme ausführen lassen, und verfolgen Sie die Programmausführung mit Hilfe dieser Fenster.

SET VIEWXE "SET VIEW"§

XE "SET ("TRAP")"§Stufe:2

In dBASE IV: Führt eine Abfrage durch oder aktualisiert die Sicht aus einer dBASE III Plus Sichtdatei (.VUE).

In FoxPro: Führt bei Angabe einer Abfrage (.QBE- oder .QBO-Datei) zu einem Fehler.

Vorgehen: Falls SET VIEW eine Abfrage durchführt, ersetzen Sie den Befehl durch einen entsprechenden DO Befehl. Falls SET VIEW mit dem Namen einer .VUE-Datei aufgerufen wird,

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

sind keine besonderen Maßnahmen erforderlich.

Anmerkung: Lesen Sie bitte den Abschnitt "dBASE-Abfragen", da manche Abfragen modifiziert werden müssen, bevor sie in FoxPro korrekt ausgeführt werden können.

Beispiel: **dBASE**
SET VIEW TO abfrage1.qbe

FoxPro

DO abfrage1.qbe

SET("VIEW")XE "SET(\"VIEW\")"§

XE "SET (\\"TRAP\\")"§Stufe:2

In dBASE IV: Liefert einen Dateinamen zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Anstatt die Funktion SET("VIEW") aufzurufen, können Sie nach dem Aufruf des Befehls SET VIEW den übergebenen Dateinamen in einer Speichervariablen, deren Wert Sie später abfragen können, speichern.

SET("WINDOW")XE "SET(\"WINDOW\")"§

Stufe: 2

In dBASE IV: Gibt den Namen des Standardbearbeitungsfensters für Memofelder aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Speichern Sie den Namen des Fensters, das mit dem Befehl SET WINDOW OF MEMO erstellt wird, in einer Variablen und fragen Sie den Wert dieser Variablen ab.

SQLXE "SQL"§

Stufe: 1

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

In dBASE IV: Wenn Sie dBASE-Daten in SQL verwenden wollen, müssen Sie eine SQL-Tabelle erstellen und die dBASE-Tabellen zu SQL-Tabellen konvertieren (SQL-Tabellen werden in einer Reihe von .DBF-Dateien gespeichert). Mit SET SQL ON | OFF aktivieren bzw. deaktivieren Sie den SQL-Modus.

In FoxPro: FoxPro unterstützt die SQL-Befehle CREATE CURSOR, CREATE TABLE, INSERT und SELECT. Diese Befehle können in Kombination mit Standardbefehlen aus Xbase verwendet werden, und die Befehlszeile muß nicht durch ein Semikolon abgeschlossen werden. Die Befehle SET SQL ON und SET SQL OFF sind nicht erforderlich und werden in FoxPro ignoriert.

Vorgehen: Komplexe in dBASE IV geschriebene SQL-Programme sind in FoxPro nicht lauffähig und müssen in FoxPro neu geschrieben werden.

STOREXE "STORE"§XE "Arrays"§

Stufe: 4

In dBASE IV: Wenn Sie den Befehl STORE mit einer Array-Bezeichnung ohne Angabe der Array-Elemente aufrufen, gibt dBASE IV den vom Array belegten Speicher frei und legt dafür eine Speichervariable gleichen Namens an.

In FoxPro: Wenn Sie den Befehl STORE mit einer Array-Bezeichnung ohne Angabe der Array-Elemente aufrufen, initialisiert FoxPro jedes Array-Element mit dem angegebenen Wert.

Vorgehen: Sofern sich ein Programm nicht darauf stützt, daß STORE ein Array in eine Speichervariable umwandelt, sind keine besonderen Maßnahmen erforderlich.

Um zu verhindern, daß in FoxPro STORE alle Elemente eines Arrays mit dem gleichen Wert initialisiert, setzen Sie SET COMPATIBLE DB4 auf ON.

SUM()

XE "SUM()"§Stufe: 3

In dBASE IV: Die Funktion SUM gibt das Ergebnis mit der Anzahl von Dezimalstellen aus, die mit dem Befehl SET DECIMALS eingestellt wurde.

In FoxPro: Auf wieviele Dezimalstellen genau das Ergebnis ausgegeben wird, hängt davon ab, mit wievielen Dezimalstellen die Feldwerte der mit SUM berechneten Felder in der Tabelle gespeichert sind.

Vorgehen: Stellen Sie SET COMPATIBLE DB4 auf ON, wenn Sie mit der mit SET DECIMALS festgelegten Anzahl von Dezimalstellen arbeiten wollen.

TAG()XE "TAG()"§

Stufe: 2

In dBASE IV: TAG() liefert den Namen des aktuellen Index oder eine leere Zeichenfolge, falls kein Index aktiv ist.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Verwenden Sie stattdessen SYS(22).

Beispiel: **dBASE**

TAG()

FoxPro

SYS(22)

TAGCOUNT() XE "TAGCOUNT() "§(nur in Version 1.5)

Stufe: 2

In dBASE IV: Die Funktion TAGCOUNT liefert die Anzahl von Indexeinträgen in einer .MDX-Datei.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Verwenden Sie eine benutzerdefinierte Funktion.

Anmerkung: Die in der Bibliothek des Programm-Konverters enthaltene benutzerdefinierte Funktion TAGCOUNT() verhält sich genauso wie die dBASE-Funktion TAGCOUNT. Lesen Sie dazu bitte im Abschnitt "Benutzerdefinierte Funktionen (UDFs) aufrufen" nach.

TAGNO()XE "TAGNO()"§ (nur in Version 1.5)

Stufe: 2

In dBASE IV: Die Funktion TAGNO gibt die Nummer des angegebenen Index aus.

In FoxPro: Erzeugt einen Fehler.

Programm-Konverter

Verzeichnis inkompatibler Befehle und Funktionen aus dBASE IV

Vorgehen: Verwenden Sie eine benutzerdefinierte Funktion.

Anmerkung: Die in der Bibliothek des Programm-Konverters enthaltene benutzerdefinierte Funktion TAGNO verhält sich genauso wie die dBASE-Funktion TAGNO. Lesen Sie dazu bitte im Abschnitt "Benutzerdefinierte Funktionen (UDFs) aufrufen" nach.

Transaktionen XE "Transaktionen" §

Stufe: 1

In dBASE IV: Obwohl sie nur selten eingesetzt werden, verfügt dBASE IV über folgende Befehle und Funktionen zur Bearbeitung von Transaktionen:

BEGIN TRANSACTIONXE "BEGIN TRANSACTION" § ...
END TRANSACTIONXE "END TRANSACTION" § Befehl
ROLLBACKXE "ROLLBACK" § Befehl
RESETXE "RESET" § Befehl
NOLOGXE "NOLOG" § Schlüsselwort im Befehl USE
COMPLETEDXE "COMPLETED" § Funktion
ROLLBACK Funktion
ISMARKEDXE "ISMARKED" § Funktion

In FoxPro: Diese Befehle und Funktionen werden nicht unterstützt und erzeugen Fehler.

Vorgehen: Ersetzen Sie diese Befehle und Funktionen durch entsprechende vom Novell Netware's Transaction Tracking System (TTS) unterstützte Funktionen und Befehle oder durch Software anderer Hersteller. FoxPro 2.5 für MS-DOS enthält eine Bibliothek, mit Hilfe derer Anwendungen das Netware TTS von Novell aufrufen können. Im Anhang E finden Sie ein Verzeichnis von Produkten, die die Bearbeitung von Transaktionen erlauben und die über vielfältige andere Netzwerk- und Datensicherungsfunktionen verfügen.

UNIQUE()XE "UNIQUE()" § (nur in Version 1.5)

Stufe: 2

In dBASE IV: Die Funktion UNIQUE liefert den Wahrheitswert .T., falls ein Index mit der Einstellung UNIQUE ON oder unter Verwendung des Schlüsselworts UNIQUE angelegt wurde, andernfalls wird .F. ausgegeben.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion.

USEXE "USE"& NOSAVEXE "NOSAVE"&, NOLOGXE "NOLOG"&, EXCLUSIVE, AGAINXE "EXCLUSIVE"&

Stufe: 2

In dBASE IV: a) Bei Angabe des Schlüsselworts NOSAVE löscht dBASE IV die Tabelle, wenn sie geschlossen wird.

b) Bei Angabe des Schlüsselworts NOLOG wird die Aufzeichnung von Änderungen im Transaktionsprotokoll, sofern eines geöffnet ist, ausgesetzt.

c) Der Vorgabewert von SET EXCLUSIVE ist OFF.

d) In dBASE IV 1.1 wird mit dem Schlüsselwort AGAIN der Kennbuchstabe des Arbeitsbereichs als Aliasname verwendet. In dBASE IV 1.5 setzen sich Aliasnamen aus einem Unterstrich und der Nummer des Arbeitsbereichs zusammen (z.B. _2).

In FoxPro: a) Das Schlüsselwort NOSAVE wird nicht unterstützt und erzeugt einen Fehler.

b) Das Schlüsselwort NOLOG wird nicht unterstützt und erzeugt einen Fehler.

c) In FoxPro ist der Vorgabewert von SET EXCLUSIVE ON.

d) AGAIN hat dieselbe Wirkung wie in dBASE IV 1.1, der Aliasnamen entspricht dem Kennbuchstaben des Arbeitsbereichs, in dem die Kopie der Tabelle geöffnet wird.

Vorgehen: a) Definieren Sie mit CREATE CURSOR eine temporäre Tabelle, die automatisch beim Schließen aus dem Arbeitsspeicher gelöscht wird. Sie können auch mit dem Befehl DELETE FILE die Tabelle löschen, nachdem sie geschlossen wurde.

b) Das Schlüsselwort NOLOG wird nicht unterstützt, löschen Sie es.

c) Falls in Ihrer Anwendung mehrere Benutzer gleichzeitig

auf eine Datenbank zugreifen müssen, setzen Sie EXCLUSIVE auf OFF.

d) Setzen Sie der dBASE-Version entsprechenden Aliasnamen ein, also für mit dBASE IV 1.5 erstellte Programme, einen Unterstrich plus der Nummer des Arbeitsbereichs, in dem die Tabelle geöffnet wird.

USER

Siehe Netzwerkfunktionen.

WINDOW()

XE "WINDOW()"§Stufe: 2

In dBASE IV: Liefert den Namen des aktuellen Fensters zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Verwenden Sie stattdessen WONTOP (liefert den Namen des obersten Fensters) oder WOUTPUT (liefert den Namen des aktuellen Ausgabefensters).

Anmerkung: FoxPro verfügt über eine hochentwickelte Fenstertechnik, ein Fenster kann hier auf zweierlei Weise "aktiv" sein, es kann eines oder mehrere Fenster überdecken und an oberster Position angezeigt werden oder es kann Ausgaben empfangen oder beides.

FOXPROC.PRG enthält eine benutzerdefinierte Funktion namens WINDOW() die es Ihnen erlaubt, Anweisungen mit der dBASE-Funktion WINDOW() ohne vorherige Änderung ausführen zu lassen. Diese UDF liefert den Wert der FoxPro-Funktion WONTOP() zurück. Lesen Sie dazu bitte im Abschnitt "Benutzerdefinierte Funktionen (UDFs) aufrufen" nach, und verwenden Sie die benutzerdefinierte Funktion.

Beispiel: **dBASE**

```
fenster_name = WINDOW()
```

FoxPro

```
fenster_name = WONTOP()
```

dBASE IV

Verzeichnis inkompatibler Befehle und Funktionen aus

or

fenster_name = WOUTPUT()

Die Konvertierung von Clipper-Programmen



Überblick

In Clipper Summer '87 geschriebene Anwendungen setzen sich aus einer Reihe verschiedener Dateien zusammen. Der Programm-Konverter und die integrierten Konvertierungsfunktionen von FoxPro sorgen dafür, daß Sie Ihre Tabellen, Indizes, Format- und Programmdateien problemlos in FoxPro verwenden können. In dBASE III Plus geschriebene Berichte (.FRM) und Etiketten (.LBL) müssen in FoxPro oft neu erstellt werden.

Arbeitsschritte "Arbeitsschritte"

1. Erstellen Sie zuerst Sicherungskopien aller Dateien.
Arbeiten Sie nie mit den Originaldateien!
2. Kopieren und laden Sie die Clipper-Tabellen in FoxPro.
3. Konvertieren Sie die .NTX-Indexdateien zu FoxPro-Indexdateien.
4. Bei Bedarf konvertieren Sie mit dem Datei-Konverter die .FMT- und .PRG-Dateien zu FoxPro-Masken.
5. Erstellen Sie Berichte und Etiketten im FoxPro-Format neu.
6. "Extrahieren" Sie die Prozeduren und Funktion, auf die FoxPro nicht zugreifen kann.
7. Verwenden Sie das Analyse-Programm, um potentielle Inkompatibilitäten zu ermitteln und zu beheben.
8. Genießen Sie die Geschwindigkeit und den Funktionsumfang von Microsoft FoxPro!

Hinweise zu Clipper 5.x Anwendungen

Der Programm-Konverter unterstützt nur Clipper Summer '87. Sofern in mit Clipper 5.x geschriebenen Programmen 5.x-spezifische Programmelemente nicht überwiegen, kann der Programm-Konverter auch hier die Konvertierung erleichtern. Clipper-Programme, die auf 5.x-spezifische Sprachelementen, wie Codeblöcken, benutzerdefinierten Befehlen, Klassen, neuen Operatoren, austauschbaren Datenbanktreibern, etc., aufbauen, sind sehr schwer zu konvertieren. Microsoft kann Ihnen bei der

Konvertierung dieser Dateien leider nicht behilflich sein.
In Anhang F finden Sie eine Liste der Clipper 5.x-
Sprachelemente, die von FoxPro nicht unterstützt werden.

Tabellen-, Memo- und Indexdateien aus Clipper verwenden

Tabellen

Clipper Summer '87 und FoxPro benutzen dasselbe Standarddateiformat (.DBF), daher brauchen Sie Ihre Clipper-Daten nicht zu konvertieren. Sie können sie sofort in FoxPro laden und bearbeiten. Geben Sie den Befehl USE <Tabellenname> in das Befehlsfenster ein, oder wählen Sie im Hauptmenü von FoxPro aus dem Menü Datei die Option Öffnen.

Memodateien

FoxPro und Clipper verwenden zum Speichern von Memofeldern unterschiedliche Formate. Im XE "Memodateien"§FoxPro-Format ist es möglich, eine unbegrenzte Menge von Daten und beliebige Binärdaten in einem Memofeld zu speichern. (Natürlich sind Ihnen durch die Festplattengröße bestimmte Grenzen gesetzt.) FoxPro kann auch .DBT-Dateien im Clipper-Format aktualisieren und anlegen. Wenn Sie diese Option wählen, können Sie allerdings nicht die Vorteile der FoxPro-Memodateien nutzen.

Sie können .DBT-Dateien auf zweierlei Weise in Memodateien im FoxPro-Format umwandeln. Sie können zum einen die Struktur der Tabelle (Datenbestandsdatei) verändern. FoxPro legt dann beim Abspeichern der neu strukturierten Tabelle eine Memodatei im FoxPro-Format an. Zum anderen können Sie die Tabellenstruktur in eine neue Datei kopieren und danach die Datensätze aus der alten Datei an die neue Datei anfügen. Sobald die Datei zum FoxPro-Format konvertiert wurde, kann Clipper die .DBF-Datei, mit der das Memofeld verknüpft war, nicht mehr lesen. Sie sollten daher immer mit Kopien der Tabellendateien arbeiten.

FoxPro kann eine Memodatei problemlos zum .DBT-Format konvertieren, verwenden Sie dazu den Befehl COPY TO

<Tabellenname> TYPE FOXPLUS.

Indizes

.NDX-Indexdateien

Falls Sie den .NDX-Treiber in Ihrem Clipper Summer '87 Programm verwenden, kann FoxPro (auch ohne den Programm-Konverter) diese Indexdateien im .IDX-Format reproduzieren. FoxPro verwendet ein sehr effizientes Indizierungsschema, das sich sowohl durch kürzere Zugriffszeiten als auch durch Indexfelder, die nur halb oder ein Drittel so groß wie NDX-Indizes sind, auszeichnet.

FoxPro konvertiert .NDX-XE ".NDX"§Indizes, wenn Sie einen USE Befehl mit dem Namen der .NDX-Datei eingeben oder mit einem SET INDEX TO Befehl, eine .NDX-Indexdatei öffnen.

.NTX-Indexdateien

Wenn Sie mit .NTX-Indizes arbeiten, können Sie diese mit dem Programm-Konverter zu FoxPro-Indizes konvertieren. Zur Konvertierung dieser Indizes müssen Sie den Programm-Konverter starten. Lesen Sie dazu bitte den Abschnitt "Den Programm-Konverter starten". Nachdem Sie den Programm-Konverter installiert haben, wählen Sie im Menü **Migration Tools** die Option **Convert files**. Daraufhin wird das Dialogfenster File Converter angezeigt.

µ §

Im Dialogfenster Convert Files können Sie mehrere .NTX-Dateien auswählen, die auf einmal konvertiert werden.

Sie wählen eine .NDX-Datei zum Konvertieren aus, indem Sie entweder auf dem Dateinamen doppelklicken oder den Dateinamen hervorheben und die Eingabetaste drücken. Die zur Konvertierung markierten Dateien werden durch ein Stern links neben dem Dateinamen (bzw. ein Häkchen in FoxPro 2.5 für MS-DOS) gekennzeichnet. Falls alle Dateien aus einem Verzeichnis konvertiert werden sollen, klicken Sie auf der

Schaltfläche **Select All**, Sie markieren damit alle Dateien. Durch Klicken auf der Schaltfläche **Clear All** können Sie die Markierung sämtlicher Dateien aufheben. Wenn Sie die Markierung nur von einer Datei entfernen wollen, doppelklicken Sie auf den Dateinamen, oder heben Sie den Dateinamen hervor, und drücken Sie die Eingabetaste. Der Datei-Konverter kann verschiedene Dateitypen, auch dBASE-Dateien vom Typ .SCR, .FRM und .LBL, parallel in einem Konvertierlauf bearbeiten. Dateien vom Typ .FMT und .PRG können zu Fox-Pro-Maskendateien konvertiert werden. Nähere Informationen hierzu finden Sie im Abschnitt "FMT- und PRG-Dateien konvertieren".

Wenn Sie eine in einem anderen Verzeichnis gespeicherte Dateien konvertieren wollen, klicken Sie auf der Schaltfläche **Directory**, und wechseln Sie zu einem anderen Verzeichnis. Zur Konvertierung von Dateien aus verschiedenen Verzeichnissen, müssen Sie die Verzeichnisse getrennt bearbeiten, d.h. Sie markieren und konvertieren erst die Dateien in einem Verzeichnis, und danach markieren und konvertieren Sie die Dateien des anderen Verzeichnisses.

Nachdem Sie die Dateien ausgewählt haben, klicken Sie auf der Schaltfläche **Process**. Daraufhin wird das nachfolgend abgebildete Dialogfenster angezeigt:

μ §

Wählen Sie den Indextyp und eine Tabelle, die mit dem konvertierten Index verknüpft werden soll.

Eine Tabelle auswählen

Die .NTX-Indexdatei kann erst konvertiert werden, wenn Sie sie mit der Tabelle, auf der der Index beruht, verknüpfen. Klicken Sie auf der Schaltfläche **Choose**. Damit wird ein Dialogfenster aufgerufen, in dem Sie die betreffende Tabelle suchen und auswählen können.

Einen Indextyp auswählen

Das Dialogfenster zeigt an, welche .NTX-Datei konvertiert

wird. Dieses Dialogfenster wird immer dann eingeblendet, wenn Sie eine .NTX-Datei zum Konvertieren auswählen. Sie können hier einstellen, ob eine .CDX- oder .IDX-Indexdatei erstellt werden soll. .IDX-Indizes und .NTX-Indizes sind vom gleichen Indextyp, beides sind Einfachindexdateien, die einen Indexeintrag enthalten. Falls Sie Ihr Programm möglichst wenig verändern wollen, sollten Sie die .NTX-Indexdateien zu .IDX-Indexdateien konvertieren.

Das .CDX-Format hat jedoch eine Reihe von Vorteilen. .CDX-Indizes sind Mehrfachindexdateien, sie können mehrere als Schlüssel bezeichnete Einträge enthalten. Darüber hinaus öffnet FoxPro .CDX-Indexdateien automatisch, wenn Sie die mit dem Index verknüpfte Tabelle laden.

.IDX-Indexdateien erstellen

Wenn eine .IDX-Datei erstellt werden soll, klicken Sie auf dem Optionsfeld **.IDX**. Der Programm-Konverter liest den Indexeintrag automatisch aus der .NTX-Datei in das Feld namens "Index expression" ein. Wenn Sie eine Tabelle ausgewählt haben, können Sie die .IDX-Datei benennen. Gemäß Vorgabe wird nur die Erweiterung von .NTX zu .IDX geändert. Nachdem Sie alle Einstellungen getroffen haben, klicken Sie auf **OK**. Die .NTX-Datei wird zu einer .IDX-Datei konvertiert. Die neue Datei wird in das Verzeichnis, in dem sich die konvertierte .NTX-Datei befindet, geschrieben.

.CDX-Indexdateien

Wenn eine .CDX-Datei erstellt werden soll, klicken Sie auf dem Optionsfeld **.CDX**. Der Programm-Konverter liest den Indexeintrag genau wie bei der Erstellung von .IDX-Indizes aus der .NTX-Datei in das Feld namens **Index expression** ein. Nachdem Sie eine Tabelle ausgewählt haben, können Sie den Indexschlüssel (Feld **Tag name**) beliebig umbenennen, Sie müssen dabei nur die MS-DOS-Konventionen für Dateinamen beachten. Standardmäßig werden die ersten zehn Zeichen des

Indexausdrucks (Feld **Index expression**) verwendet. Beachten Sie jedoch, daß Sie diesen Ausdruck in manchen Fällen sogar ändern müssen, nämlich wenn es sich dabei nicht um einen gültigen Indexschlüssel handelt.

Wenn Sie die nötigen Einstellungen getroffen haben, klicken Sie auf **OK**. Die .NTX-Datei wird daraufhin zu einer .CDX-Datei konvertiert. Die neue Datei wird im Unterschied zu .IDX-Dateien nicht in dem Verzeichnis der .NTX-Datei erstellt, sondern in dem Verzeichnis, in dem sich die zugehörige Tabelle befindet.

Wenn Sie weitere .NTX-Dateien konvertieren und den Namen einer bereits bestehenden .CDX-Datei einsetzen, fügt der Programm-Konverter in die genannte .CDX-Datei neue Indexschlüssel ein.

Funktionen in Indexausdrücken

Falls Ihr .NTX-Indexausdruck eine Funktion enthält, muß FoxPro auf diese Funktion zugreifen können. Benutzerdefinierte Funktionen müssen (zumindest temporär) in .PRG-Dateien verlagert werden. Diese .PRG-Dateien sollten in dem Verzeichnis gespeichert werden, in dem sich MIGRATE.APP und die anderen Programm-Konverter-Dateien befinden. Wenn auf eine Funktion nicht zugegriffen werden kann, wird eine Fehlermeldung ausgegeben, und Sie werden aufgefordert, die Datei dem Programm-Konverter zugänglich zu machen.

Falls es sich bei der Funktion um eine Clipper-Funktion handelt, die in FoxPro nicht unterstützt wird, müssen Sie sie entweder in FoxPro als benutzerdefinierte Funktion neu erstellen, oder Sie müssen den Index in Clipper ohne diese Funktion neu erstellen und danach mit dem Programm-Konverter konvertieren. (Der Indexausdruck kann im Dialogfenster Convert .NTX file nicht editiert werden.) Manchmal ist es einfacher, den Index in FoxPro von grund auf neu anzulegen.

Aus .FMT- und .PRG-Dateien FoxPro-Maskendateien erstellen

Dateien, die @SAY Befehle enthalten, können zu FoxPro-Maskendateien (.SCX) konvertiert werden. Lesen Sie dazu den Abschnitt "FMT- und PRG-Dateien konvertieren".

Berichte und Etiketten

Berichte (.FRM-Dateien), die Sie mit Hilfe des Clipper RL-Dienstprogramms erstellt haben, können Sie in FoxPro 2.5 für MS-DOS ausführen. Sie können diese Dateien in FoxPro für MS-DOS oder Windows allerdings nicht verändern. Diese Berichte sind zudem in FoxPro für Windows nicht lauffähig. Sie müssen diese Dateien mit dem Berichtsgenerator von FoxPro neu erstellen.

Hartcodierte Berichte (oder Etiketten) werden in FoxPro normalerweise wie jede andere Programmdatei ausgeführt, sicherheitshalber sollten Sie mit dem Analyse-Programm jedoch überprüfen, ob darin inkompatible Befehle und Funktionen enthalten sind. Falls Sie Software eines anderen Herstellers verwendet haben, wie z.B. R&R Reportwriter von Concentric, können Sie dieses Produkt, sofern es FoxPro-Indizes unterstützt (bei R&R ist dies der Fall), weiterhin verwenden.

In dBASE III Plus erstellte Etikettendateien (.LBL) können ausgeführt und zum FoxPro-Format .LBX konvertiert werden, Sie müssen nur die Dateierweiterung .LBL angeben.

Hinweis: Der Programm-Konverter kann in dBASE IV geschriebene Berichte (.FRM) und Etiketten (.LBL) konvertieren. Der Datei-Konverter kann jedoch in dBASE III Plus erstellte Dateien nicht bearbeiten und gibt eine Fehlermeldung aus, wenn Sie solch eine Datei konvertieren wollen.

Zugriff auf Funktionen und Prozeduren

In Clipper Summer '87 können .PRG-Dateien Funktionen und Prozeduren aufrufen, die in irgendeiner anderen .PRG-Datei

definiert worden sind. In FoxPro müssen Funktionen und Prozeduren entweder in der .PRG-Datei, aus der sie aufgerufen werden, definiert sein, oder in dem der aufrufenden .PRG-Datei übergeordneten Aufruf-Stack.

Der Programm-Konverter ermöglicht Ihnen, die Programmdateien auf einfache Weise so zu verändern, daß die unterschiedlichen Aufrufkonventionen kein Problem mehr darstellen. Der Programm-Konverter kann die in einer Gruppe von .PRG-Dateien enthaltenen Funktionen und Prozeduren sowohl in eine einzelne Prozedurendatei als auch in separate .PRG-Dateien kopieren. Er kann auch den Inhalt der .PRG-Dateien ohne Funktionen und Prozeduren kopieren.

Funktionen und Prozeduren "extrahieren"

Wählen Sie die Option **Extract procs/funcs** aus dem Menü **Migration Tools**. Daraufhin wird ein Dialogfenster eingeblendet, das so wie die anderen Dialogfenster des Programm-Konverters aussieht. **Hinweis: Die Originaldateien werden NICHT verändert.** Die Prozeduren und Funktionen werden tatsächlich aus den Originaldateien **herauskopiert** ("extrahiert") und in eine neue Datei geschrieben.

µ §

Wählen Sie die .PRG-Dateien, aus denen Prozeduren und Funktionen extrahiert werden sollen.

Denken Sie daran, daß diese Dateien in zwei Ausfertigungen vorliegen, wenn Sie das Analyse-Programm eingesetzt haben. Eine Kopie befindet sich auf der Festplatte, die andere im Memofeld der Analysedatei (.EXP). Achten Sie darauf, daß die Prozeduren und Funktionen aus den richtigen Dateien "extrahiert" werden.

Sie wählen eine Datei aus, indem Sie entweder auf dem Dateinamen doppelklicken oder den Dateinamen hervorheben und die Eingabetaste drücken. Die markierten Dateien werden durch einen Stern links neben dem Dateinamen (bzw. ein Häkchen in FoxPro 2.5 für MS-DOS) gekennzeichnet. Falls alle

Dateien aus einem Verzeichnis konvertiert werden sollen, klicken Sie auf der Schaltfläche **Select All**, Sie markieren damit alle Dateien. Durch Klicken auf der Schaltfläche **Clear All** können Sie die Markierung sämtlicher Dateien aufheben. Wenn Sie die Markierung nur von einer Datei entfernen wollen, doppelklicken Sie auf den Dateinamen, oder heben Sie den Dateinamen hervor, und drücken Sie die Eingabetaste.

Wenn Sie eine in einem anderen Verzeichnis gespeicherte Dateien konvertieren wollen, klicken Sie auf der Schaltfläche **Directory**, und wechseln Sie zu einem anderen Verzeichnis. Zur Konvertierung von Dateien aus verschiedenen Verzeichnissen, müssen Sie die Verzeichnisse getrennt bearbeiten, d.h. Sie markieren und konvertieren erst die Dateien in einem Verzeichnis und danach markieren und konvertieren Sie die Dateien des anderen Verzeichnisses.

Nachdem Sie alle Dateien markiert haben, klicken Sie auf der Schaltfläche **Process**. Daraufhin wird das nachfolgend abgebildete Dialogfenster eingeblendet.

μ §

Hier geben Sie an, ob die Prozeduren und Funktionen in eine Prozedurdatei oder in verschiedene .PRG-Dateien kopiert werden sollen.

Eine Prozedurdatei oder mehrere .PRG-Dateien erstellen

Sie können die Funktionen und Prozeduren in eine einzige Prozedurdatei (eine Programmdatei mit der Erweiterung .PRG) extrahieren lassen. Wenn Sie dann den Befehl SET PROCEDURE TO <Prozedurdatei> in den Initialisierungsteil Ihres Programms einfügen, sind alle in der Prozedurdatei enthaltenen Prozeduren und Funktionen während der gesamten Ausführungszeit Ihrer Anwendung verfügbar.

Die andere Möglichkeit ist, die Funktionen und Prozeduren in verschiedene .PRG-Dateien kopieren zu lassen. Dabei wird für jede Funktion oder Prozedur eine Datei angelegt, die nach der Funktion bzw. der Prozedur benannt wird (falls der Funktions- oder Prozedurname mehr als acht Zeichen lang ist, wird er

gekürzt).

Gemäß Standardeinstellung wird nur eine Prozedurendatei erstellt. Wählen Sie eines der beiden Verfahren, indem Sie auf das entsprechende Optionsfeld klicken.

Eine Prozedurendatei erstellen

Wenn nur eine einzige Prozedurendatei erstellt werden soll, müssen Sie den Namen und das Verzeichnis für diese Datei festlegen. Klicken Sie auf der Schaltfläche **Choose**. Stellen Sie das Verzeichnis ein, in dem die Datei erstellt werden soll, und geben Sie einen Dateinamen ein. Klicken Sie dann auf der Schaltfläche **Create**. Sie sollten die Datei nicht in dem Verzeichnis, das die Quelldateien enthält, erstellen lassen. Der Programm-Konverter gibt vor dem Überschreiben einer Datei zwar immer eine Meldung aus, die Sie bestätigen müssen, es bleibt aber immer noch das Risiko, daß Sie diese Rückfrage versehentlich mit Ja (bzw. "yes") beantworten. Am sichersten ist es, ein anderes Verzeichnis zu wählen.

Mehrere .PRG-Dateien erstellen

Wenn Sie sich für mehrere .PRG-Dateien entscheiden, müssen Sie angeben, in welchem Verzeichnis diese Dateien erstellt werden sollen. Klicken Sie auf der Schaltfläche **Choose** und stellen Sie das gewünschte Verzeichnis ein. Danach klicken Sie auf der Schaltfläche **Select**.

Von Funktionen und Prozeduren bereinigte Versionen der Programmdateien erstellen

Falls Sie eine Kopie der .PRG-Quelldateien (den Programmdateien, die nach Prozeduren und Funktionen durchsucht werden) anfertigen lassen wollen, die keine Prozeduren und Funktionen mehr enthält, markieren Sie das unterste Kontrollkästchen im Dialogfenster. Die Datei wird dann in dasselbe Verzeichnis wie die Prozedurendatei bzw. die verschiedenen .PRG-Dateien, je nachdem welches Verfahren

Sie gewählt haben, geschrieben.

Beachten Sie auch hier eventl. Rückfragen des Programm-Konverters, ob bestehende Dateien überschrieben werden sollen, und überlegen Sie gut, bevor Sie die Frage bejahen.

Inkompatibilitäten zur Clipper-Sprache

Übersicht

Obwohl es einige Inkompatibilitäten zwischen Clipper Summer '87 und FoxPro gibt, können die meisten Befehle und Funktionen aus Clipper ohne Schwierigkeit in FoxPro ausgeführt werden.

Um einen reibungslosen Programmablauf in FoxPro zu erreichen, ermitteln Sie zuerst mit Hilfe des Analyse-Programms alle Befehle und Funktionen, die in FoxPro Probleme verursachen können. Lesen Sie dazu bitte den Abschnitt "Das Analyse-Programm." Nachdem das Analyse-Programm die Problemfälle in einer Tabelle erfaßt hat, können Sie mit der Bearbeitung der einzelnen Problemfälle beginnen. Sobald Sie alle Probleme bereinigt haben, können Sie versuchen, die Anwendung in FoxPro ausführen zu lassen.

Alle Problemfälle, die vom Analyse-Programm angezeigt werden, sind im Abschnitt "Verzeichnis inkompatibler Befehle und Funktionen aus Clipper" dokumentiert. Dort werden die Unterschiede in der Interpretation der einzelnen Befehle und Funktionen beschrieben, und es wird dargestellt, wie Sie Programme verändert können, damit sie in FoxPro das erwartete Verhalten zeigen.

Die wichtigsten Kompatibilitätsprobleme

Auch wenn das Analyse-Programm in einer Anwendungen viele Probleme anzeigt, können die allermeisten Probleme oft durch die Veränderung einer einzigen Programmzeile gelöst werden, so daß Sie Ihre Anwendungen ohne Schwierigkeiten in FoxPro ausführen können. Sie werden bald herausfinden, daß einige

wenige Probleme sehr häufig auftreten und daß Sie bei der Fehlerkorrektur meist dieselbe Art von Änderungen durchführen.

In welchem Umfang Programm verändert werden müssen, hängt davon ab, in welchem Maße Clipper-spezifische Programmier-techniken und Funktionen eingesetzt wurden. Wenn Ihre Programme in reinem Xbase geschrieben sind, müssen Sie sie nur minimal verändern. In einigen Fällen müssen Sie Programmzeilen löschen, da FoxPro über integrierte Funktionen verfügt, die Sie in Clipper durch eigene Routinen erst implementieren oder durch Bibliotheken von anderen Herstellern in Clipper einbinden müssen. Sie müssen Ihr Hauptaugenmerk auf die folgenden sechs Bereiche legen:

1. Syntax von Funktionsaufrufen
2. die Funktionen ACHOICE(), DBEDIT() und MEMOEDIT()
3. Fehlerbehandlung
4. Bibliotheken anderer Hersteller
5. Arrays
6. Farbdarstellung

Weitere Aspekte in Sachen Kompatibilität

Neben den oben aufgeführten Hauptproblembereichen, müssen folgende Aspekte bei der Konvertierung von in Clipper Summer '87 geschriebenen Anwendungen beachtet werden:

1. Windows-spezifische Probleme
2. SET Befehle
3. Tastatureingaben
4. Hartcodierte Dateierweiterungen

FunktionsaufrufeXE "Funktionsaufrufe"§

Der Großteil der Änderungen, die Sie in Programmen durchführen müssen, ist durch die unterschiedliche Syntax von Funktionsaufrufen in Clipper und FoxPro bedingt.

In Clipper können Funktionen aufgerufen werden, ohne daß der Rückgabewert einer Variablen zugeordnet wird. In FoxPro muß der Rückgabewert einer Variablen zugeordnet oder der Funktion muß ein Gleichheitszeichen vorangestellt werden. Das nachfolgende Beispiel zeigt, wie eine Clipper-Funktion dieses Typs verändert werden müßte.

Clipper:

FSEEK(handle,15,0)

FoxPro:

= FSEEK(handle,15,0)

oder

dummy = FSEEK(handle,15,0)

oder

? FSEEK(handle,15,0)

Das Analyse-Programm markiert alle Funktionen, auch benutzerdefinierte Funktionen, die in Clipper-Manier aufgerufen werden. Die Funktion wird unter dem Eintrag "Bad function call" (Unzulässiger Funktionsaufruf) aufgeführt. Sie sollten Ihr Programm wie im Beispiel gezeigt verändern, also entweder ein Gleichheitszeichen vor die Funktion stellen, den Rückgabewert einer Variablen zuordnen oder den Befehl ? einsetzen. Sofern FoxPro die Funktion unterstützt, wird sie danach korrekt ausgeführt.

ACHOICE(), DBEDIT() und MEMOEDIT()

Diese drei häufig verwendeten Funktionen werden in FoxPro nicht unterstützt. In FoxPro gibt es jedoch verschiedene Alternativen zu ACHOICE() und fast äquivalente Funktionen

für DBEDIT() und MEMOEDIT(). Detaillierte Erläuterungen und Beispiele dazu finden Sie im Abschnitt "Verzeichnis inkompatibler Befehle und Funktionen aus Clipper".

FehlerbehandlungXE "Fehlerbehandlung"§

Fehlerbehandlungsroutinen stellen einen weiteren Bereich dar, in dem Änderungen unumgänglich sind. In Clipper werden Fehlerbehandlungsroutinen mit einem BEGIN SEQUENCE...[BREAK]...END Konstrukt implementiert. Der Programmierer kann damit einen Funktionsaufruf in eine BEGIN SEQUENCE...END Anweisung einfügen und dann mit Hilfe des Schlüsselwortes BREAK, von einer Funktion, die einige Ebenen tiefer aufgerufen wird, zur END-Anweisung zurückkehren. Dies ähnelt einem Longjump in C oder einer GoTo-Anweisung in Basic.

Beispiel:

Clipper

```
OldScreen=SAVESCREEN( 0, 0, 24, 79,)  
CLEAR  
@ 2, 12 SAY "Bitte warten, Ausdruck läuft!"  
BEGIN SEQUENCE  
    SET PRINT ON  
    Foo()  
END  
SET PRINT OFF  
CLEAR  
@ 2, 12 SAY "Bericht kann nicht gedruckt werden!"  
@ 3, 12 SAY "Bitte später nochmals versuchen!"  
RESTSCREEN( 0, 0, 24, 79, OldScreen )
```

```
FUNCTION Foo
```

```
    USE Test Exclusive
```

```
    IF NETERR()
```

```
        BREAK      &&Wenn nicht exklusiv
```

geöffnet,

```
        &&zur END-Anweisung gehen
```

Programmen

```

                                ENDIF
                                PrintIt()                                && hartcodierter Bericht, der
auf
                                && Dateien exklusive zugreifen muß
                                && Nicht ausführen, falls Break-
                                RETURN .T. && bedingung erfüllt wird.

```

Solch ein Programmablauf wäre in FoxPro nur durch eine ziemlich aufwendige Programmierung zu erreichen. Anstatt zu versuchen, diese SEQUENCE...[BREAK]...END Konstrukte zu simulieren, sollten Sie die Fehlerbehandlungsroutine besser neu schreiben und dabei den FoxPro-Befehl ON ERROR verwenden.

Fehlerbehandlung in FoxPro

Manche Fehler, die in Clipper zu Programmabbrüchen führen, werden von FoxPro abgefangen. Beide Sprachen verfügen über interne Fehlerbehandlungsroutinen. In Clipper Summer '87 enthält die Bibliothek EXTEND.LIB Fehlerbehandlungsroutinen, der zugehörige Programmcode findet sich in der Datei ERRORSYS.PRG. Viele Programmierer haben zudem eigene Routinen geschrieben, die beim Auftreten von Fehlern ein eleganteres und durchsichtigeres Verhalten zeigen. In FoxPro sind jedoch viel umfangreichere Fehlerbehandlungsroutinen, die zudem mehr Fehlerarten behandeln können, integriert als in Clipper.

Die Fehlerbehandlung in FoxPro stützt sich auf den Befehl ON ERROR, der festlegt, welche Prozedur beim Auftreten eines Fehlers auszuführen ist. Bei der Fehlerbehandlungsprozedur handelt es sich gewöhnlich um eine längere CASE-Anweisung, die bestimmte Fehlerbedingungen überprüft. Falls es sich um einen schwerwiegenden Fehler handelt, wird die Programmausführung abgebrochen (und die Programmsteuerung zurück an FoxPro oder das Betriebssystem gegeben). Bei nicht so schwerwiegenden Fehlern, meldet die Prozedur den Fehler und gibt die Programmsteuerung an das aufrufende Programm

Die Konvertierung von Clipper-Programmen

zurück.

Sie können in FoxPro beispielsweise die folgenden Programmzeilen in den Aufrufabschnitt Ihres Programms einfügen:

```
Beispiel:      FoxPro
                ON ERROR DO Fehl_Rout WITH ERROR(),
MESSAGE(), ;
                MESSAGE(1), SYS(16), LINENO(), SYS(102), ;
                SYS(100), SYS(101), LASTKEY(), ALIAS(), ;
                SYS(18), SYS(5), ; SYS(12), SYS(6),
SYS(2003), ;
                WONTOP(), SYS(2011), SYS(2018),
SET("CURSOR")
```

Die CASE-Anweisung der Fehlerbehandlungsroutine würde dann aus verschiedenen Teilen bestehen, die jeweils einen bestimmten Fehler behandeln. Um den Fehler abzufangen, der im oben angeführten Clipper-Beispiel behandelt wurde, müssten Sie in FoxPro überprüfen, ob eine der Fehlerbedingungen "Datei wird bereits benutzt" (Fehlernummer 3) oder "Datei wird von einem anderen Benutzer benutzt" (Fehlernummer 108) zutrifft. Wenn Sie ein Meldungsfenster am Anfang der Fehlerbehandlungsroutine definieren, können Sie eine Meldung ausgeben lassen, die den Benutzer über das Auftreten des Fehlers unterrichtet.

```
Beispiel:      FoxPro
                PROCEDURE Fehl_Rout
                PARAMETERS m_fehler, m_meldung,
m_meldung2, ;
                m_prgrname, m_zlnr, m_prtset, m_console, m_dev, ;
                m_tasteing, m_alias, m_aktget, m_deflfw, m_mem, ;
                m_ausg, m_aktdir, m_wontop, m_lockstat, ;
                m_fenstername, m_cursor

                DO CASE
```

```

CASE m_fehler= <zu ermittelnder Fehler>
  <Fehler handhaben>
CASE m_fehler = 3 or ERROR() = 108
  Meldungsfenster aktivieren
  JN=" "
  @ 1,1 Say "Datei ist bereits geöffnet! ;
    Wiederholen J/N ? "
  Get JN
  Read
  If Upper(JN) $ "J"
    Retry
  Else
    Set Print Off
    Return
  Endif
CASE m_fehler= <anderer Fehler>
  <Fehlerbehandlung>
...
ENDCASE

```

In FoxPro sind mehr Fehlermeldungen integriert als in Clipper. Mit einer gut geschriebenen Fehlerbehandlungsroutine können Sie diese integrierten Fehlermeldungen leicht verwalten. Fehlerbehandlungsroutinen sind auch bei der Suche nach Programmfehlern sehr hilfreich, da Sie Ihnen Informationen über die Fehlerposition und die Fehlerart geben.

Das Konstrukt BEGIN SEQUENCE...[BREAK]...END in FoxPro simulieren

Sofern Aufrufe nicht zu tief geschachtelt sind, können BEGIN SEQUENCE...[BREAK]...ENDXE "BEGIN SEQUENCE... [BREAK]...END"§ Konstrukte durch DO WHILE Schleifen simuliert werden. Denken Sie jedoch daran, daß es umso ratsamer ist, die Routinen mit ON ERROR Befehlen neu zu implementieren, je mehr Schachtelungsebenen in Clipper verwendet wurden.

Die Konvertierung von Clipper-Programmen

Kurz gesagt, simuliert man dieses Konstrukt, indem man alle BREAK-Anweisungen sucht, die in Routinen unterer Ebenen geschachtelt sind, und dann diese Routinen und alle davon abhängigen Routinen abändert, so daß am Ende nur noch ein BREAK-Aufruf im Block BEGIN SEQUENZ vorkommt.

Beispiel: **FoxPro**

```
DO WHILE .T.  
    <Anweisungen>...  
    IF break_beding  
        EXIT  
    ENDIF  
    <Anweisungen>...  
ENDDO  
    <recovery statements>...
```

Falls es sich bei der Routine um eine Prozedur handelt, müssen Sie sie in einen Funktionsaufruf verwandeln.

Beispiel: **Clipper**

```
DO DieRoutine [ WITH parm1, parm2, ... ]
```

FoxPro

```
IF !DieRoutine ( [ parm1, parm2 ... ] )  
    EXIT                   &&Falls Aufruf in BEGIN
```

SEQ Block

```
    * or RETURN .F.     &&Falls Aufruf auf unterer  
                          &&Ebene
```

```
ENDIF
```

Falls es sich bei der Routine bereits um eine Funktion handelt, müssen Sie andere Änderungen vornehmen.

Beispiel: **Clipper**

```
DieRoutine ( [ parm1, parm2 , ... ] )
```

FoxPro

```
PRIVATE RVAL
```

Programmen

```
RVAL = .F.
IF !DieRoutine( @RVAL [ parm1, parm2, ... ] )
    EXIT                                &&Falls Ref in BEGIN
```

SEQ Block

```
* or RETURN .F.  &&Falls Ref auf unterer Ebene
ENDIF
RETURN RVAL
```

Oder

```
PRIVATE RVAL
RVAL = .F.
x = DieRoutine( parm1, parm2,@RVAL)
IF !rval
    EXIT
    * or RETURN .F.
ENDIF
```

Verändern Sie die Funktion dann wie folgt:

Beispiel:

Clipper

```
FUNCTION DieRoutine
PARAMETERS parm1, parm2,...
DO Irgendwas
BREAK
DO Irgendwas
RETURN Irgendwas
```

FoxPro

```
FUNCTION DieRoutine
PARAMETERS RVAL, parm1, parm2,...
DO Irgendwas
RETURN .F.
DO Irgendwas
RVAL=Irgendwas
RETURN .T.
```

Die Konvertierung von Clipper-Programmen

Bibliotheken anderer Hersteller

Für Clipper waren zahlreiche Bibliotheken und Dienstprogramme anderer Hersteller verfügbar. Falls Sie eines dieser Produkte in Ihren Programmen verwendet haben, müssen Sie die betreffenden Funktionen löschen oder ersetzen. Manche dieser Bibliotheken, wie beispielweise NetLib, CommTools und dGE sind auch für FoxPro verfügbar. (dGE kann sowohl mit Clipper als auch mit FoxPro verwendet werden.) Eine sehr häufig verwendete Bibliothek, OVERLAY(), kann leicht durch das FoxPro-Speicherverwaltungsprogramm Foxswap ersetzt werden.

Zu anderen Bibliotheken, wie z.B. Funky, Grumpfish und Artful, sind keine entsprechenden Produkte oder Funktionen in FoxPro verfügbar. Für einige der darin enthaltenen Funktionen können Sie selbst benutzerdefinierte Funktionen schreiben. Getit, eine häufig in Clipper Summer '87 verwendete Bibliothek, ermöglichte verschachtelte READ- und GET-Anweisungen, in FoxPro können ist dies ohnehin zulässig. Sie können bis zu fünf READ-Anweisungen schachteln. Darüber hinaus finden Sie verschiedene Shareware im Foxforum von CompuServe. Falls Sie mit Funky II arbeiten, können Sie die "C" Funktionen verwenden und mit dem Befehl RUN in FoxPro ausführen.

Hinweis: Das Analyse-Programm sucht nur nach Ausdrücken mit Clipper Summer '87 Syntax, die zu FoxPro inkompatibel sind. Aufrufe an Bibliotheken von Drittherstellern werden nicht erkannt. Sie müssen diese Aufrufe selbst finden und bearbeiten, da sie ansonsten Compiler- oder Laufzeitfehler verursachen.

Arrays

In Clipper Summer '87 sind Arrays eindimensional, in FoxPro können Arrays dagegen auch mehrdimensional sein. FoxPro verfügt über sehr umfangreiche Array-Funktionen. (Eine Liste dieser Funktionen finden Sie im Handbuch *Befehle und*

Funktionen zu FoxPro 2.5.) Da FoxPro mehrdimensionale Arrays unterstützt, müssen Sie Funktionen wie ADIR() und AFIELDS(), denen in Clipper mehrere Arrays als Parameter übergeben werden, in FoxPro nur ein Array als Parameter übergeben.

Beachten Sie, daß FoxPro die Funktion AFILL() nicht unterstützt und daß die Funktion TYPE() in FoxPro Arrays nicht in der gleichen Weise Arrays bearbeitet wie in Clipper.

Im Abschnitt "Verzeichnis inkompatibler Befehle und Funktionen aus Clipper" wird genau erklärt, wie Sie diese vier Funktionen verändern müssen.

Farbdarstellung

In FoxPro wird die Farbdarstellung ganz anders gehandhabt als in Clipper. Es ist einfacher, die Farbeinstellungen in FoxPro neu zu setzen, als zu versuchen, die Clipper-Farbeinstellungen in FoxPro zu verwenden.

FoxPro verwendet Farbschemata, die aus jeweils zehn Farbpaaren bestehen. Die in einem Schema definierten Farbpaare legen Vorder- und Hintergrundfarbe fest, und allen Elemente der FoxPro-Benutzeroberfläche und der Oberfläche von in FoxPro erstellten Anwendungen wird jeweils ein Farbpaar zugeordnet. (Beachten Sie, daß in Windows bestimmte Farbeinstellungen im Dialogfenster Farben der Windows-Systemsteuerung für alle Windows-Anwendungen gesetzt werden, also auch für FoxPro für Windows gelten.)

Beispielsweise verwendet FoxPro ein bestimmtes Farbpaar eines Farbschemas, um anzuzeigen, ob eine Schaltfläche verfügbar oder nicht verfügbar ist. Sie müssen für diese vom Status abhängige Farbdarstellung keine zusätzlichen Routinen schreiben. Analog dazu ändert sich die Farbe des Fensterrahmens, sobald ein Fenster aktiv bzw. nicht aktiv ist. FoxPro führt diese Farbänderung automatisch durch, Sie brauchen sich nicht darum zu kümmern.

FoxPro enthält eine Reihe vordefinierter Farbschemata. Sie können diese Schemata verändern, und sie können neue

Schemata definieren. Mit dem Befehl SET COLOR SET können Sie Farbschemata auswählen. Zum Verändern oder Erstellen eines Farbschemas verwenden Sie die Befehle SET COLOR OF SCHEME bzw. CREATE COLOR SCHEME.

Beispiel: **FoxPro:**

* Farbschema ändern und danach laden

SET COLOR OF SCHEME 24 TO B/BG,W+/N, ;

BG+/BG,N/BG,N/BG,W+/GR,GR+/RB,N+/N,GR+/B,R+/B

SET COLOR SET TO SCHEME 24

Um die Kompatibilität zu Vorgängerversionen zu wahren, unterstützt FoxPro auch die Funktion SET('COLOR'), die die aktuellen Farbeinstellungen ausgibt, und den Befehl SET COLOR TO, mit dem eine neue Farbeinstellung definiert werden kann. Wenn Sie diesen Befehl zusammen mit der Funktion SET('COLOR') verwenden, können Sie dieselben Ergebnisse wie mit der Clipper-Funktion SETCOLOR() erzielen.

Hinweis: Zum Lieferumfang von FoxPro 2.5 für MS-DOS gehört eine sehr nützliche Anwendung namens PROCOLOR, die erklärt, wie Sie die FoxPro-Schemata am besten einsetzen.

Weitere Kompatibilitätsprobleme

Windows-spezifische Probleme

Drucken

Verschiedentlich wurden Probleme mit den Windows-Druckertreibern beim Drucken hartcodierter Berichte gemeldet. Falls Sie die Befehle `???` zum Ausdrucken verwenden, können Sie das Problem lösen, indem Sie die FoxPro-Befehle `PRINTJOB` und `ENDPRINTJOB` vor bzw. hinter den Druckdaten einfügen. Darüber hinaus müssen Sie die MS-DOS-Druckertreiber in Windows verwenden. Falls Sie `@ SAY-`Anweisungen eingesetzt haben, werden die Berichte mit dem Befehl `SET DEVICE TO PRINTER` korrekt ausgedruckt, in diesem Fall werden sogar die Schriften richtig gedruckt. Beim Einsatz unterschiedlicher Schriften ist jedoch Vorsicht geboten, da Probleme mit Zeichendichte und Zeilenabständen auftreten können.

ASCII-Zeichensatz

Falls Sie ASCII-Zeichencodes für Bildschirmausgaben verwenden, werden die betreffenden Aufrufe in FoxPro für Windows nicht das gewünschte Ergebnis zeigen. Die Windows-Version von FoxPro arbeitet mit dem ANSI-Zeichensatz, nicht mit dem ASCII-Zeichensatz. FoxPro für Windows verfügt über eine Funktion, `OEMTOANSI`, mit der ASCII-Zeichen in ANSI-Zeichen umgewandelt werden können, sofern es sich nicht um Grafikzeichen, die zum Zeichnen von Linien verwendet werden, handelt. Diese Grafikzeichen sind im ANSI-Zeichensatz nicht definiert.

SET Befehle

Vielen SET Befehlen kann in Clipper ein logischer Parameter übergeben werden. In FoxPro wird dadurch ein Fehler erzeugt. Da SET Befehle auch Speichervariablen als Parameter

akzeptieren, kann das Analyse-Programm nicht feststellen, ob einem SET Befehl ein logischer Parameter übergeben wurde. Sie müssen den Programmcode daher selbst überprüfen, um diese Fälle ausfindig zu machen. Sie können das Programm aber auch in FoxPro ausführen und anhand der auftretenden Laufzeitfehler diese SET Befehle aufspüren. Folgende SET Befehle können mit logischen Werten arbeiten:

Die Konvertierung von Clipper-Programmen

SET CENTURYXE "SET CENTURY"§	SET EXCLUSIVEXE "SET EXCLUSIVE"§
SET CONFIRMXE "SET CONFIRM"§	SET FIXEDXE "SET FIXED"§
SET CONSOLEXE "SET CONSOLE"§	SET INTENSITYXE "SET INTENSITY"§
SET CURSORXE "SET CURSOR"§	SET PRINTXE "SET PRINT"§
SET DATEXE "SET DATE"§	SET SCOREBOARDXE "SET SCOREBOARD"§
SET DELETEDXE "SET DELETED"§	SET SOFTSEEKXE "SET SOFTSEEK"§
SET DELIMITERSXE "SET	SET UNIQUEXE "SET UNIQUE"§

DELIMITERS"§	
SET ESCAPEXE "SET ESCAPE"§	SET WRAPXE "SET WRAP"§
SET EXACTXE "SET EXACT"§	

Tastatureingaben

Die in FoxPro und Clipper Summer '87 verwendeten Tastencodes stimmen nicht immer überein. Die Befehle SET FUNCTION und SET KEY (nicht unterstützter Befehl) sowie die Funktionen INKEY()XE "INKEY()"§, LASTKEY()XE "LASTKEY()"§ und NEXTKEY()XE "NEXTKEY()"§ (nicht unterstützte Funktion) liefern daher in FoxPro unter Umständen nicht die erwarteten Ergebnisse. Schlagen Sie in Anhang G in der Tastencode-Tabelle nach, welche FoxPro-Tastencodes für die in Clipper verwendeten Tastencodes einzusetzen sind, und verändern Sie das Programm entsprechend.

Das Analyse-Programm führt diese Funktionen und Befehle als Problemfälle an, damit Sie feststellen können, welche Teile Ihres Programms gegebenenfalls verändert werden müssen.

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

Nachfolgend werden in alphabetischer Reihenfolge alle Befehle und Funktionen verzeichnet, die bei der Ausführung von Clipper-Programmen in FoxPro Probleme verursachen können. Jeder Eintrag beschreibt, wie sich der Befehl oder die Funktion in Clipper and in FoxPro verhält. Unter dem Stichwort Vorgehen finden Sie Hinweise dazu, in welcher Weise Sie Ihr Programm verändern sollten, und in den meisten Fällen wird

dies anhand eines Beispiels veranschaulicht. Im Abschnitt Anmerkungen werden relevante Zusatzinformationen gegeben. Der Inhalt dieses Verzeichnisses wird auch im Fenster Program Analyzer angezeigt, allerdings ohne Beispiele.

Dieses Verzeichnis stellt in keiner Weise den Versuch dar, die Originaldokumentation dieser Befehle und Funktionen zu reproduzieren, und es wird auch kein Anspruch auf Vollständigkeit erhoben. Bitte schlagen Sie in der Dokumentation zu Clipper bzw. FoxPro nach, wenn Sie detailliertere Informationen zu diesen Befehle und Funktionen brauchen.

- **Stufe XE "Problemstufe" §1** umfaßt Befehle und Funktionen, die von FoxPro nicht unterstützt werden. Diese Befehle und Funktionen müssen gelöscht oder durch die äquivalenten Befehle und Funktionen der FoxPro-Sprache ersetzt werden.
- **Stufe 2** umfaßt Befehle und Funktionen, die in FoxPro Fehler erzeugen, für die es aber häufig sehr ähnliche Befehle und Funktionen in FoxPro gibt.
- **Stufe 3** umfaßt Befehle und Funktionen, die in FoxPro zwar keine Fehler erzeugen, sich jedoch so unterschiedlich verhalten, daß sie Beachtung verdienen.
- **Stufe 4** umfaßt Befehle und Funktionen, die bei der Ausführung des Programms in FoxPro nur selten Probleme verursachen. Wenn Sie nach Problemen der Stufe 4 suchen lassen, werden viele Programmzeilen markiert, die für gewöhnlich problemlos ausgeführt werden.

Diese Einstufung mag vielleicht etwas zufällig erscheinen, sie soll Ihnen lediglich erleichtern einzuschätzen, wie schwerwiegend ein Problem ist und wieviel Aufwand seine Lösung erfordert.

== (Vergleichsoperator)XE "==" (Vergleichsoperator)"§

Stufe: 3

In Clipper: Beim Vergleich von Zeichenfolgen wird TRUE zurückgeliefert, wenn die Zeichenfolgen die gleiche Länge haben. Nachstehende Leerzeichen werden ignoriert.

In FoxPro: Der Wert TRUE wird nur dann ausgegeben, wenn sämtliche Zeichen der beiden Zeichenfolgen genau übereinstimmen, Leerzeichen werden hier nicht ignoriert.

Vorgehen: Entfernen Sie mit den Funktionen RTRIM(), LTRIM() oder ALLTRIM() Leerzeichen, bevor Sie Zeichenfolgen vergleichen.

Beispiel: **Clipper**

```
IF string1==string2
    <Code ausführen>
ENDIF
```

FoxPro

```
IF ALLTRIM(string1)==ALLTRIM(string2)
    <Code ausführen>
ENDIF
```

ACHOICE()XE "ACHOICE()"§

Stufe: 1

In Clipper: ACHOICE() erstellt ein Menü gemäß der Werte zweier als Parameter übergebener Arrays. In einem Arrays werden die Eingabeaufforderungen definiert, das andere Array enthält logische Werte, die festlegen, ob eine Meldung hervorgehoben wird oder nicht.

In FoxPro: Erzeugt einen Fehler.

aus Clipper

Vorgehen: Löschen Sie ACHOICE(). Falls ACHOICE() in einem Menü verwendet wurde, erstellen Sie das Menü mit dem FoxPro-Menügenerator erneut. Ersetzen Sie Auswahllisten durch Popup-Menüs. Für lange Listen sollten Sie Listenarrays verwenden. Bei kurzen Listen sollte ein Popup-Menü mit Eingabefeldern funktionieren.

Beispiel: **Beispiel 1: Ein mit erstelltes ACHOICE() Menü**

Clipper:

```
DO WHILE .T.
    menü_grösse=6

    DECLARE cues[ menü_grösse ], msgs[ menü_grösse ]

    cues[1] = " Aufträge "
    cues[2] = " Formate "
    cues[3] = " Tabellen "
    cues[4] = " Gesamt "
    cues[5] = " Berichte "
    cues[6] = " Summen "

    msgs[1] = "  Aufträge eingeben, bearbeiten, löschen,
sichten."
    msgs[2] = "  Formattabelle erstellen, bearbeiten, ;
        löschen oder sichten.;
        Einträge hinzufügen oder ändern."
    msgs[3] = "  Tabellen verwalten."
    msgs[4] = "  Gesamtsummen sichten."
    msgs[5] = "  Bericht ausführen."
    msgs[6] = "  Summen berechnen."

    mchoice=ACHOICE( 1, 1, 2, 80, cues, msgs)
```

```
DO CASE
  CASE mchoice = 1
    THE_USUAL('ORD', .f., .f., .t.)
  CASE mchoice = 2
    THE_USUAL('STY')
  CASE mchoice = 3
    TABL_MENU()
  CASE mchoice = 4
    TOTL_MENU()
  CASE mchoice = 5
    RPT_MENU()
  CASE mchoice = 6
    GET_MENU()
  OTHERWISE
    Exit
ENDCASE
ENDDO
QUIT
```

FoxPro: Ersetzen Sie das mit ACHOICE() erstellte Menü durch ein FoxPro-Menü

Das oben beschriebene Clipper-Beispiel würde wie folgt aussehen, wenn es mit dem Menügenerator in FoxPro erstellt würde. FoxPro würde dann auch den Programmcode zur Definition dieses Menüs für Sie schreiben.

µ §

Wenn Sie den Programmcode selbst schreiben wollen, sollte er folgendermaßen aussehen:

```
SET SYSMENU TO
SET SYSMENU AUTOMATIC
```


Verzeichnis inkompatibler Befehle und Funktionen

aus Clipper

"Aufträge"	DEFINE PAD Aufträge OF _MSYSMENU PROMPT
"Formate"	DEFINE PAD Formate OF _MSYSMENU PROMPT
"Tabellen"	DEFINE PAD Tabellen OF _MSYSMENU PROMPT
"Berichte"	DEFINE PAD Berichte OF _MSYSMENU PROMPT
"Summen"	DEFINE PAD GesSumme OF _MSYSMENU PROMPT
"Beenden"	DEFINE PAD mQuit OF _MSYSMENU PROMPT
auftrmenü	ON SELECTION PAD Aufträge OF _MSYSMENU do
formmenü	ON SELECTION PAD Formate OF _MSYSMENU do
tablmenü	ON SELECTION PAD Tabellen OF _MSYSMENU do
bermenu	ON SELECTION PAD Berichte OF _MSYSMENU do
do gesmenu	ON SELECTION PAD GesSumme OF _MSYSMENU
	ON SELECTION PAD mQuit OF _MSYSMENU QUIT

Beispiel 2:

Clipper: Eine mit ACHOICE() erstellte lange Auswahlliste

```
select 0
use Hersteller
DECLARE raTemp[RECCOUNT()]
FOR i = 1 TO RECCOUNT()
    raTemp[ i ] = Konto_nam
    SKIP
NEXT
nChoice= ACHOICE(nTrow+1, nTcol+1, ;
```

nBrow-1, nBcol-1, raTemp)

FoxPro -Popup-Menü mit Listenarray

```
DIMENSION Meinarray[RECCOUNT()]
Select Hersteller.Konto_nam, recno() ;
from Profile into array Meinarray
=Asort(Meinarray)
```

```
@ 0,0 GET Hersteller.Konto_nam ;
  PICTURE "@&T" ;
  FROM Meinarray ;
  SIZE 16,33 ;
  DEFAULT 1 ;
  WHEN check_prompt() ;
  COLOR SCHEME 2
```

Beispiel 3: Clipper: Eine mit ACHOICE() erstellte kurze Auswahlliste

Siehe Beispiel oben

FoxPro-Popup-Menü mit Eingabefeldern

```
Select Selcodes
Set Filter to Popname="KontMed"
Go Top
m.medium="Telefon"
```

```
DEFINE WINDOW KontaktMed FROM 8, 2 TO 16,21 ;
  TITLE " Kontaktmedium " NOFLOAT ;
  NOCLOSE NOMINIMIZE COLOR SCHEME 1
```

```
DEFINE POPUP Kontakt PROMPT FIELD pcue ;
SCROLL MARGIN MARK ""
```

```
ACTIVATE WINDOW KontaktMed SAME
@ 0,0 GET m.typ PICTURE "@&T" POPUP Kontakt ;
```

```
SIZE 7,18 DEFAULT " " COLOR SCHEME 2  
READ CYCLE MODAL ;  
VALID Finishit()
```

```
RELEASE WINDOW KontaktMed  
RELEASE POPUPS Kontakt
```

```
m.kont_medium=pcue  
SELECT Calls  
SHOW GET m.kont_medium ENABLE
```

```
FUNCTION FinishIt  && Lesen zulässig
```

ADIR()XE "ADIR()"§

Stufe: 2

In Clipper: ADIR() können bis zu fünf Arrays als Parameter übergeben werden.

In FoxPro: ADIR() erfordert und akzeptiert nur ein Array als Parameter.

Vorgehen: Verändern Sie die Syntax des Funktionsaufrufs ADIR().

Anmerkung: FoxPro unterstützt mehrdimensionale Arrays, daher ist es nicht notwendig, mehrere eindimensionale Arrays zu verwenden.

Beispiel: **Clipper**

```
DECLARE aName[10],aGrösse[10],  
aDatum[10],aZeit[10]  
ADIR(aName,aGrösse,aDatum,aZeit)
```

FoxPro

```
=ADIR(Arin,"*.dbf")
```

AFIELDS()XE "AFIELDS()"§

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

Stufe: 2

In Clipper: AFIELDS() können bis zu vier Arrays als Parameter übergeben werden.

In FoxPro: AFIELDS() erfordert und akzeptiert nur ein Array als Parameter.

Vorgehen: Verändern Sie die Syntax des Funktionsaufrufs AFIELDS().

Anmerkung: FoxPro unterstützt mehrdimensionale Arrays, daher ist es nicht notwendig, mehrere eindimensionale Arrays zu verwenden.

Beispiel: **Clipper**

```
DECLARE aNamen[FCOUNT()],aTypen[FCOUNT()] ;  
aBreite[FCOUNT()], aDez[FCOUNT()]  
=AFIELDS(aNamen, aTypen, aBreite, aDez)  
fld_name=aNamen  
fld_typ=aTypen  
fld_breit=aBreite  
fld_dez=aDez
```

aus Clipper

FoxPro

```
=AFIELDS(Meinarray)
fld_name=Meinarray(1,1)
fld_typ=Meinarray(2,1)
fld_breit=Meinarray(3,1)
fld_dez=Meinarray(4,1)
```

AFILL()XE "AFILL()"§

Stufe: 2

In Clipper: AFILL() initialisiert die Elemente eines Arrays mit dem angegebenen Wert.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie AFILL(). Setzen Sie einfach den Arraynamen (ohne eckige Klammern und ohne Verweise auf Arrayelemente) gleich dem gewünschten Wert. Jedes Arrayelement wird daraufhin mit dem angegebenen Wert initialisiert.

Beispiel: **Clipper**
AFILL(Meinarray,"")

FoxPro

```
Meinarray = ""
```

ALTD()XE "ALTD()"§

Stufe: 2

In Clipper: ALTD() aktiviert den Clipper-Debugger.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie ALTD(). Fügen Sie stattdessen eine Routine ein, die die Programmausführung unterbricht und das Trace-

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

und/oder das Debug-Fenster aktiviert.

Beispiel: **Clipper**
 Do WHILE !EOF()
 ALTD()
 ENDDO

aus Clipper

FoxPro

```
Do WHILE !EOF()
  ACTIVATE WINDOW Trace
  ACTIVATE WINDOW Debug
  SUSPEND
ENDDO
```

APPEND FROM FIELDSXE "APPEND FROM FIELDS"§

Stufe: 2

In Clipper: Die FIELDS-Klausel wird vor die FROM-Klausel gestellt.
Zudem sind WHILE-Klauseln zulässig.

In FoxPro: FIELDS-Klauseln, die vor einer FROM-Klausel stehen,
erzeugen einen Fehler.

Vorgehen: Stellen Sie die FROM-Klausel vor die FIELDS-Klausel

Beispiel: **Clipper**

```
APPEND FIELDS Id, Vorname, Nachname FROM
DieDatei ;
FOR Id > 10 WHILE Nachname = "Schmidt"
```

FoxPro

```
APPEND FROM DieDatei Fields Id, Vorname,
Nachname ;
FOR Id > 10 AND Nachname = "Schmidt"
```

APPEND FROM WHILEXE "APPEND FROM WHILE"§

Stufe: 2

In Clipper: WHILE-Klauseln sind zulässig.

In FoxPro: Das Vorkommen einer WHILE-Klausel erzeugt einen Fehler.

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

Vorgehen: Verwenden Sie anstelle der WHILE-Klausel eine FOR-Klausel.

Beispiel: **Clipper**

```
APPEND FIELDS Id, Vorname, Nachname FROM  
DieDatei ; FOR Id > 10 WHILE Nachname = "Schmidt"
```

FoxPro

```
APPEND From DieDatei Fields Id, Vorname, Nachname  
;  
FOR Id > 10 and NACHNAME = "Schmidt"
```

**BEGIN SEQUENCE...[BREAK]...ENDXE "BEGIN SEQUENCE..
[BREAK]...END"§**

Level 1

In Clipper: Definiert eine Sequenz von Anweisungen zur Fehlerbehandlung.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Sie können an Clipper-Fehlerbehandlungsroutinen auf zweierlei Weise herangehen. Die beste Methode ist, die betreffenden Routinen in FoxPro neu zu schreiben und dazu den FoxPro-Befehl ON ERROR zu verwenden. Die zweite, in der Regel sehr viel kompliziertere Methode ist, mit DO WHILE Schleifen die Wirkungsweise der BEGIN SEQUENCE...END Konstrukte zu imitieren.

Beispiel: Siehe Abschnitt "Fehlerbehandlung"

BIN2I()XE "BIN2I()"§

Stufe: 1

In Clipper: Konvertiert eine Zeichenfolge, die das Format eines in 16 Bit mit Vorzeichen gespeicherten Integerwerts hat, in einen

Programm-Konverter

153

aus Clipper

numerischen Wert.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Funktionsaufruf. FoxPro bietet keine Alternativlösung zu dieser Funktion. In Anhang J finden Sie jedoch eine Liste mit Produkten von Drittherstellern, die diese Funktion erfüllen.

BIN2L()XE "BIN2L()"§

Stufe: 1

In Clipper: Konvertiert eine Zeichenfolge, die das Format eines in 32 Bit mit Vorzeichen gespeicherten Integerwerts hat, in einen numerischen Wert.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Funktionsaufruf. FoxPro bietet keine Alternativlösung zu dieser Funktion. In Anhang J finden Sie jedoch eine Liste mit Produkten von Drittherstellern, die diese Funktion erfüllen.

BIN2W()XE "BIN2W()"§

Stufe: 1

In Clipper: Konvertiert eine Zeichenfolge, die das Format eines in 16 Bit gespeicherten vorzeichenlosen Integerwerts hat, in einen numerischen Wert.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Funktionsaufruf. FoxPro bietet keine Alternativlösung zu dieser Funktion. In Anhang J finden Sie jedoch eine Liste mit Produkten von Drittherstellern, die diese

Funktion erfüllen.

CALLXE "CALL"§

Stufe: 1

In Clipper: Führt unabhängig voneinander kompilierte Routinen und Programme aus. Der Funktion kann eine Ausdrucksliste als Parameter übergeben werden.

In FoxPro: Führt Binärdateien aus, die mit dem Befehl LOAD in den Hauptspeicher geladen wurden. Dem Befehl CALL kann nur eine einzige Zeichenfolge als Parameter übergeben werden.

In Clipper Summer '87 geschriebene Programme, die den Befehl CALL einsetzen, generieren in FoxPro Fehler.

Vorgehen: Löschen Sie diese Funktion, oder kompilieren Sie die Routine erneut in FoxPro.

Anmerkung: Routinen, die in Clipper getrennt kompiliert und assembliert wurden, können in FoxPro nicht in dieser Form eingesetzt werden. Sie müssen neu kompiliert und in Binärdateien (.BIN) verwandelt oder mit Hilfe des FoxPro Library Construction Kits (LCK) in eine .PLB- oder .FLL-Datei verwandelt werden. Nähere Informationen zum LCK erhalten Sie auf Nachfrage von Microsoft.

CANCELXE "CANCEL"§

- Stufe: 3
- In Clipper: Beendet die Programmausführung und kehrt zum Betriebssystem zurück.
- In FoxPro: Beendet die Programmausführung und kehrt zum letzten aufrufenden Programm zurück.
- Vorgehen: Wenn Sie zum Betriebssystem zurückkehren wollen, ersetzen Sie CANCEL durch QUIT.

CLEARXE "CLEAR"§

- Stufe: 2
- In Clipper: Der Befehl CLEAR kann mit einer SCREEN-Klausel verwendet werden, damit wird das automatische Löschen von GET-Anweisungen unterdrückt.
- In FoxPro: Die SCREEN-Klausel erzeugt einen Fehler.
- Vorgehen: Löschen Sie die SCREEN-Klausel. In FoxPro werden GET-Anweisungen standardmäßig mit dem Befehl CLEAR nicht gelöscht bzw. aufgehoben. Verwenden Sie den Befehl CLEAR GETS, um GET-Anweisungen zu löschen.
- Anmerkung: In Clipper wird der Bildschirminhalt mit CLEAR SCREEN gelöscht, setzen Sie den Cursor zur Position 0,0 und löschen Sie die seit dem letzten CLEAR ALL bzw. CLEAR GETS gegebenen GET-Anweisungen. In FoxPro müssen Sie den Befehl CLEAR GETS oder CLEAR ALL eingeben, um GET-Anweisungen zu löschen. In FoxPro wird mit dem Befehl CLEAR nur der Bildschirminhalt gelöscht. Falls Sie in Clipper die Option SCREEN zum Löschen von GET-Anweisungen

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

eingesetzt haben, müssen Sie in FoxPro den Befehl `CLEAR GETS` in Ihr Programm einfügen, um dasselbe Ergebnis zu erzielen.

Beispiel: **Clipper**
 `CLEAR`

aus Clipper

FoxPro

CLEAR

CLEAR GETS

&& Falls in Ihrem Programm mit

dem

&& Befehl CLEAR GET-Anweisungen

&& gelöscht werden sollen

COMMITXE "COMMIT"§

Stufe: 2

In Clipper: COMMIT gibt den Inhalt aller Clipper-Puffer an DOS aus und schreibt die Daten auf Festplatte.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen den äquivalenten FoxPro-Befehl FLUSH ein.

Beispiel: **Clipper**

COMMIT

FoxPro

FLUSH

DBEDIT()XE "DBEDIT()§

Stufe: 1

In Clipper: DBEDIT() ermöglicht die Anzeige und Bearbeitung von Datensätzen aus einem oder mehreren Arbeitsbereichen in Form einer BROWSE-Tabelle, die innerhalb eines definierten Fensterbereichs angezeigt wird. DBEDIT() kann mit benutzerdefinierten Routinen zur Bearbeitung von Tastatureingaben verwendet werden und zeigt dann ein anderes Verhalten.

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie die Funktion DBEDIT, und setzen Sie stattdessen den Befehl BROWSE ein.

Anmerkung: Passen Sie das BROWSE-Fenster der Funktionsweise von DBEDIT() an. Sie müssen Ihre Routinen zur Behandlung von Tastatureingaben nicht verändern, da FoxPro die Tastatureingaben für Sie verarbeitet. Sie sollten das BROWSE-Fenster mit Hilfe von Kommandozeilenbefehlen verändern, anstatt mit einem Funktionsaufruf.

BROWSE verfügt über eine ganze Reihe Funktionen, die Sie in Clipper erst selbst erstellen mußten. Wenn Sie beispielsweise mit DBEDIT schreibgeschützte Daten anzeigen lassen wollten, müßten Sie zuerst eine Routine schreiben, die Tastatureingaben nicht zuläßt. In FoxPro geben Sie einfach nur das Schlüsselwort NOMODIFY nach dem Befehl BROWSE ein.

Beispiel: **Clipper**

```
USE Kunden
DECLARE feld_list[3]
feld_list[1] = "Filiale"
feld_list[2] = "Verkäufer"
feld_list[3] = "Menge"
DBEDIT(4, 0, 22, 79, FELD_LIST, "UserFunc")
```

FoxPro

```
DEFINE WINDOW Tbrowse FROM 4, 0 to 22, 79
USE Kunden
BROWSE FIELDS Filiale, Verkäufer, Menge ;
    LAST NORMAL WINDOW TBROWSE;
    TITLE "Kunden"
```

DBFILTER()XE "DBFILTER()"§

aus Clipper

Stufe: 2

In Clipper: DBFILTER() gibt die im aktuellen Arbeitsbereich definierte Filterbedingung als Zeichenfolge aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie DBFILTER() durch die äquivalente FoxPro-Funktion FILTER().

Beispiel: **Clipper**
filt_string=DBFILTER()

FoxPro
filt_string=FILTER()

DBRELATION()XE "DBRELATION()"§

Stufe: 2

In Clipper: DBRELATION() gibt den relationalen Ausdruck als Zeichenfolge aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie DBRELATION() durch die äquivalente FoxPro-Funktion RELATION().

Beispiel: **Clipper**
rel_string=DBRELATION()

FoxPro
rel_string=RELATION()

DBRSELECT()XE "DBRSELECT()"§

Stufe: 2

In Clipper: DBRSELECT() liefert die Nummer des Arbeitsbereichs, der die

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

genannte Beziehung enthält.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Erstellen Sie eine benutzerdefinierte Funktion namens DBRSELECT(), die das Verhalten der gleichnamigen Clipper-Funktion imitiert. Fügen Sie diese Funktion in Ihr Programm oder in eine Prozedurenbibliothek ein.

Die UDF verwendet die FoxPro-Funktion TARGET() zur Ermittlung des richtigen Wertes. Beachten Sie, daß der Arbeitsbereich, der nach der Beziehung durchsucht wird, geöffnet sein muß.

Beispiel: **Clipper**

```
ArbBereich=DBRSELECT(1)
```

FoxPro

```
ArbBereich=DBRSELECT(1)
```

```
FUNCTION DBRSELECT
PARAMETERS relnum
PRIVATE i, malias, relnum
mAlias=TARGET(relnum)
i=0
If !EMPTY(malias)
FOR i= 1 TO 255
    IF ALIAS(i) = malias
        RETURN i
    ENDIF
NEXT
IF i = 226
    i =0
ENDIF
ENDIF
```


RETURN i

DESCEND()XE "DESCEND()"§

Stufe: 2

In Clipper: DESCEND() ermöglicht die Erstellung und das Durchsuchen von Indizes mit absteigender Sortierfolge.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen die äquivalenten Befehle ASCENDING | DESCENDING im Ausdruck INDEX ON ein. ASCENDING und DESCENDING können mit dem Befehl SET ORDER TO in FoxPro aktiviert und deaktiviert werden.

Beispiel: **Clipper**

```
INDEX ON DESCEND(Nachname) TO Nachname  
SET INDEX TO Nachname  
SEEK "Jonas"
```

FoxPro

```
INDEX ON Nachname TO Nachname DESCENDING  
SET INDEX TO Nachname  
SET ORDER TO Tag Nachname Descend  
SEEK "Jonas"
```

DOSERROR()XE "DOSERROR()"§

Stufe: 2

In Clipper: DOSERROR() liefert die Fehlernummer des zuletzt aufgetretenen MS-DOS-Fehlers.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Verändern Sie die Fehlerbehandlungsroutinen, setzen Sie den Befehl ON ERROR ein.

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

Anmerkung: Siehe Abschnitt "Fehlerbehandlung."

Da Clipper-Anwendungen kompiliert werden, sind die meisten zur Laufzeit auftretenden Fehler so kritisch, daß man den Benutzer zum Betriebssystem zurückkehren lassen muß. Mit dem Befehl DOSERROR() wird in Clipper ermittelt, woran die Ausführung des Befehls RUN gescheitert ist.

Beispiel:

Clipper

```
IF NETERR() .AND. model == "USE"  
    RETURN .F.  
ENDIF  
BREAK_SEQ()
```

```
RETURN ERR_EXIT( ERR_LINE() + M->info + ", " ;  
+ M->_1 + IF( DOSERROR() > 0, ", DOS-Fehler " ;  
+ STR( DOSERROR() ) , "" ))
```

FoxPro

```
ON ERROR DO FP25EROR WITH ERROR(),  
MESSAGE(), ;  
MESSAGE(1), SYS(16), LINENO(), SYS(102), ;  
SYS(100), SYS(101), LASTKEY(), ;  
ALIAS(), SYS(18), SYS(5), SYS(12), SYS(6), ;  
SYS(2003), WONTOP(), ;  
SYS(2011), SYS(2018), SET("CURSOR")
```

ERRORLEVEL()XE "ERRORLEVEL()"§

Stufe: 2

In Clipper: ERRORLEVEL() gibt den aktuellen MS-DOS Fehlerstatus aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: ERRORLEVEL() wurde hauptsächlich mit dem Clipper-

Dienstprogramm SWITCH.EXE verwendet und zwar immer dann, wenn mehrere ausführbare Programme aufgerufen wurden. Wenn Sie mit FOXSWAP arbeiten, brauchen Sie den MS-DOS-Fehlerstatus nicht zu überprüfen. Sobald Sie SWITCH.EXE durch FOXSWAP ersetzen, können Sie diese Funktion löschen.

EXTERNALXE "EXTERNAL"§

- Stufe: 2
- In Clipper: EXTERNAL deklariert ein Symbol für den Clipper-Kompiler.
- In FoxPro: Erzeugt einen Fehler.
- Anmerkung: In FoxPro wird EXTERNAL verwendet, um externe Dateien und Verweise auf externe Routinen in ein FoxPro-Projekt einzubinden. EXTERNAL wird nur vom Projektmanager bearbeitet, während der Programmausführung wird der Befehl ignoriert. In Clipper ist EXTERNAL eine an den Kompiler gerichtete Anweisung, damit der Kompiler die nach dem Schlüsselwort EXTERNAL aufgeführten Funktionen während des Bindens - nicht erst während des Kompilierens - des Programms findet.
- Vorgehen: Löschen Sie alle mit EXTERNAL definierten Verweise. In FoxPro müssen Programmierer keine Kompilerdirektiven schreiben.
- Anmerkung: In FoxPro bearbeitet der Projektmanager Verweise auf externe Routinen. Wenn Sie mit dem Projektmanager eine Anwendung schreiben, sucht der Projektmanager nach den Programmen, Funktionen und Prozeduren, die von der Anwendung aufgerufen werden. Wenn er sie findet, fügt er sie in die Anwendung ein. Falls er sie nicht findet, wird der Programmierer aufgefordert, die Datei zu nennen, in der sie sich befinden.

FREAD()XE "FREAD()"§

Stufe: 2

In Clipper: FREAD() gibt die Anzahl gelesener Bytes aus und liest diese Bytes in die Speichervariable ein, die als Parameter angegeben wurde.

In FoxPro: FREAD() gibt die tatsächlich gelesenen Daten aus, als Parameter kann die Anzahl der zu lesenden Bytes angegeben werden.

Vorgehen: Verändern Sie die Syntax des Aufrufs von FREAD().

Beispiel: **Clipper**

```
block = 128
buffer = SPACE(512)
handle = FOPEN("Test.txt")
IF FERROR() <> 0
    bytes = FREAD(handle, @buffer, block)
    IF bytes <> block
        ? "Fehler beim Lesen von Test.txt"
    ENDIF
ENDIF
ENDIF
```

FoxPro

```
*Hinweis: Dieses Beispiel geht davon aus,
*daß TEST.TXT existiert
STORE FOPEN('test.txt') TO file_handle
STORE FSEEK(file_handle, 0, 2) TO ifp_size
IF ifp_size <= 0                                && Ist Datei leer?
    WAIT WINDOW 'Die Datei ist leer!' NOWAIT
ELSE
    l_string = FREAD(file_handle, ifp_size)
ENDIF
```

FREADSTR()XE "FREADSTR()"§

aus Clipper

Stufe: 2

In Clipper: FREADSTR() liest Zeichen aus einer DOS-Datei.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie FREADSTR() durch die äquivalente FoxPro-Funktion FREAD().

Beispiel: **Clipper**

```
buffer = FREADSTR(handle,16)
```

FoxPro

```
buffer = FREAD(handle,16)
```

HARDCR()XE "HARDCR()"§

Stufe: 2

In Clipper: HARDCR() ersetzt alle harten Zeilenumbrüche (Wagenrücklaufzeichen) durch bedingte Zeilenumbrüche.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion HARDCR(). FoxPro führt in Memofeldern automatisch Zeilenumbrüche durch, diese Funktion wird daher nicht benötigt.

Anmerkung: Gemäß Vorgabe ist die Zeilenumbruchfunktion in FoxPro aktiviert. Falls Sie die Funktion HARDCR() jedoch simulieren wollen, können Sie die Funktion STRTAN() verwenden.

Beispiel: **Clipper**

```
memo_var=HARDCR(memo_var)
```

FoxPro

*Diese Zeile ist eigentlich überflüssig!

memo_var=STRTRAN(memo_var,CHR(141),CHR(13)+CHR(10))

I2BIN()XE "I2BIN()"§

Stufe: 2

In Clipper: I2BIN() konvertiert einen numerischen Wert vom Typ Integer zu einem zwei Bytes umfassenden Zeichen, das das Format eines binären in 16 Bit gespeicherten Integerwert hat.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Funktionsaufruf. FoxPro bietet keine Alternativlösung zu dieser Funktion. In Anhang J finden Sie jedoch eine Liste mit Produkten von Drittherstellern, die diese Funktion erfüllen.

IFXE "IF"§

Stufe: 2

In Clipper: ELSEIF-Klauseln werden unterstützt.

In FoxPro: ELSEIF-Klauseln erzeugen einen Fehler.

Vorgehen: Verwenden Sie anstelle von IF..ENDIF Anweisungen geschachtelte IF- oder CASE Anweisungen.

Anmerkung: Bei langen ELSEIF Ausdrücken ist der Einsatz von CASE empfehlenswert.

Beispiel: **Clipper**

```
IF A
    <tue irgendwas >
ELSEIF B
    <tue irgendwas anderes>
ELSE
    <tue nichts>
```

aus Clipper

Verzeichnis inkompatibler Befehle und Funktionen

ENDIF

FoxPro: geschachtelte If-Anweisung

```
IF A
  <tue irgendwas>
ELSE
  IF B
    <tue irgendwas anderes>
  ELSE
    <tue nichts>
  ENDIF
ENDIF
```

FoxPro: Case-Anweisung

```
DO CASE
  CASE A
    <tue irgendwas>
  CASE B
    <tue irgendwas anderes>
  CASE <N>
    <tue etwas ganz anderes>
  OTHERWISE
    <tue nichts>
ENDCASE
```

IF()XE "IF()"§

Stufe: 2

In Clipper: Gibt einen von zwei Ausdrücken entsprechend dem Wert eines logischen Ausdrucks zurück (Zwischen-If-Funktion).

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie IF() durch die äquivalente FoxPro-Funktion IIF().

Beispiel: **Clipper**

Verzeichnis inkompatibler Befehle und Funktionen

aus Clipper

IF(Bezahlt, SPACE(0), "Jetzt aber hinter her!")

FoxPro

IIF(Bezahlt, SPACE(0), "Jetzt aber hinter her!")

INDEXEXT()XE "INDEXEXT()"§

Stufe: 2

In Clipper: INDEXEXT() gibt den Indextyp aus, der in einer Anwendung verwendet wird.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion. Da FoxPro nur seine eigenen Indexdateiformate verarbeitet, muß der Indextyp nicht ermittelt werden.

Anmerkung: **Hinweis:** FoxPro unterstützt sowohl .IDX-Indexdateien (Indizes mit einfachen Einträgen, .NTX- und .NDX-Indizes ähnlich) als auch .CDX-Indizes, die mehrere Indexschlüssel enthalten können. Darüber hinaus öffnet FoxPro .CDX-Indexdateien automatisch, wenn Sie die mit dem Index verknüpfte Tabelle laden.

INDEXKEY()XE "INDEXKEY()"§

Stufe: 2

In Clipper: INDEXKEY() gibt den Ausdruck für Indexschlüssel der geöffneten Indexdatei zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie INDEXKEY() durch die äquivalente FoxPro-Funktion SYS(14), oder verwenden Sie die Funktion KEY().

Beispiel: **Clipper**
Schlüssel=INDEXKEY(1)

FoxPro
Schlüssel=KEY(1)

aus Clipper

Verzeichnis inkompatibler Befehle und Funktionen

oder
Schlüssel=SYS(14,1)

INDEXORD()XE "INDEXORD()"§

Stufe: 2

In Clipper: INDEXORD() gibt die Positionsnummer des kontrollierenden Hauptindex einer Liste geöffneter Indexdateien aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie INDEXORD() durch die FoxPro-Funktion SYS(21).

Anmerkung: Beachten Sie, daß SYS(21) eine Zeichenfolge zurückliefert.

Beispiel: **Clipper**

pos_num=INDEXORD(1)

FoxPro

pos_num=VAL(SYS(21))

**INKEY()XE "INKEY()"§, LASTKEY()XE "LASTKEY()"§, SET
FUNCTIONXE "SET FUNCTION"§**

Stufe: 3

In Clipper: In FoxPro und Clipper Summer '87 werden Tasten oft unterschiedliche Tastencodes zugeordnet.

In FoxPro: Diese Funktionen und Befehle verhalten sich in FoxPro genauso wie in Clipper, da die FoxPro-Tastencodes jedoch nicht immer mit den Clipper-Tastencodes übereinstimmen, zeigen Clipper-Programme in FoxPro oft nicht das erwartete Verhalten.

Vorgehen: Schlagen Sie in Anhang G in der Tastencode-Tabelle nach, welche FoxPro Tastencodes für verwendeten Clipper-Tastencodes einzusetzen sind, und verändern Sie das Programm entsprechend.

ISPRINTER()XE "ISPRINTER()"§

Stufe: 2

In Clipper: ISPRINTER() überprüft, ob LPT1 bereit ist.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen die FoxPro-Funktion PRINTSTATUS() ein.

Anmerkung: Die Überprüfung ist nicht auf den Druckeranschluß LPT1 beschränkt, die Funktion liefert den Status des aktuellen Druckers zurück.

Beispiel: **Clipper**

```
    If ISPRINTER()
        REPORT FORM Calls TO PRINT
    ENDIF
```

FoxPro

```
If Printstatus()  
    REPORT FORM Calls TO PRINT  
ENDIF
```

L2BIN()XE "L2BIN()"§

Stufe: 1

In Clipper: L2BIN() wandelt numerische Clipper-Datentypen in eine vier Bytes lange Zeichenfolge um, die das Format eines in 32 Bit mit Vorzeichen gespeicherten Integerwerts hat.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Funktionsaufruf. FoxPro bietet keine Alternativlösung zu dieser Funktion. In Anhang J finden Sie jedoch eine Liste mit Produkten von Drittherstellern, die diese Funktion erfüllen.

LASTKEY()

Siehe INKEY().

LASTREC()XE "LASTREC()"§

Stufe: 2

In Clipper: LASTREC() gibt die Anzahl der in der aktiven Tabelle enthaltenen physikalischen Datensätze aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen die Funktion RECCOUNT() ein.

Beispiel: **Clipper**

num_recs=LASTREC()

FoxPro

num_recs=RECCOUNT()

MEMOEDIT()XE "MEMOEDIT()"§

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

Stufe: 2

In Clipper: MEMOEDIT() ermöglicht die Bearbeitung von Memofeldern.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie MEMOEDIT() durch MODIFY MEMO.

Anmerkung: Zur Bearbeitung von Memofeldern steht in FoxPro der integrierte Memoeditor zur Verfügung, Sie brauchen daher nicht wie in Clipper erst aufwendige Routinen programmieren. Sie müssen nur einfach alle Vorkommen von MEMOEDIT durch den Befehl MODIFY MEMO ersetzen. Sie können natürlich auch eine Routine schreiben, die genau dasselbe tut wie Ihr Clipper-Programm, dies wird im folgenden Beispiel gezeigt.

Sie können mit Memofeldern und BROWSE-Fenstern Operationen ausführen, die in Clipper nur sehr schwer, wenn überhaupt, zu realisieren wären. Siehe Beispiel 2.

Memofelder können mit dem FoxPro-Befehl SCATTER MEMVAR MEMO auch in ein Speichervariablenarray oder eine Speichervariablengruppe kopiert werden, damit sind indirekte Leseoperationen leichter zu implementieren.

Beispiel: **Clipper**

```
mNotiz = MEMOEDIT(Notiz,12,6,18,73,.T.)
```

FoxPro

```
DEFINE WINDOW TNotiz FROM 12, 6 TO 18, 73  
SET WINDOW OF MEMO TO Tnotiz  
MODIFY MEMO Notiz
```

Beispiel 2: **FoxPro**

Memofeld-
*Damit werden ein BROWSE-Fenster und ein
*Fenster, in dem Text geblättert werden kann,

Verzeichnis inkompatibler Befehle und Funktionen

aus Clipper

*nebeneinander angezeigt.

```
DEFINE WINDOW Tbrowser FROM 3,3 TO 15,15 ;  
TITLE "Aufrufe"  
DEFINE WINDOW Tkommentar FROM 3,17 TO 15,78
```

;

```
TITLE "Kommentar"  
SELECT CALLS  
SET FILTER TO Analyst_id=Anal_id AND ;  
Contact_id=Rolodex->Contact_id  
SET WINDOW OF MEMO TO TCOMMENTS  
SET ORDER TO TAG CALLDATE  
MODI MEMO COMMENTS NOWAIT
```

```
*Fenster Tbrowser aktivieren  
GO TOP  
BROWSE FIELDS CALLDate LAST NORMAL ;  
NOMENU WINDOW Tbrowser  
DEACTIVATE WINDOW Tbrowser  
DEACTIVATE WINDOW Tkommentar  
RELEASE WINDOWS Tbrowser, Tkommentar  
SET WINDOW OF MEMO TO Close Memo
```

Kommentar

```
SET FILTER TO ANALYST_ID=Anal_id  
SET ORDER TO TAG ACCOUN_NAM  
SELECT (cAlias)
```

MEMOLINE()XE "MEMOLINE()"\$

Stufe: 2

In Clipper: MEMOLINE() extrahiert eine formatierte Textzeile aus einem Zeichenausdruck oder einem Memofeld.

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

In FoxPro: Erzeugt einen Fehler.

Vorgehen: In FoxPro tun Sie dies in zwei Arbeitsschritten. Mit ATLINE() oder ATCLINE() ermitteln Sie die Zeilennummer, dann lassen Sie von MLINE() die Textzeile ausgeben.

Beispiel: **Clipper**

```
mNotiz=MEMOLINE(Notiz,79,3)
*Notiz ist das Memofeld
*79 ist die Zeilenlänge
*3 ist die Zeilennummer
```

FoxPro

```
SET MEMOWIDTH TO 79
mNotiz=MLINE(Notiz,3)
```

aus Clipper

MEMOREAD()XE "MEMOREAD()"§

Stufe: 2

In Clipper: MEMOREAD() liefert den Inhalt einer Textdatei als Zeichenfolge zurück.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie die Funktion, und setzen Sie stattdessen MODIFY FILE oder MODIFY MEMO ein.

Beispiel: **Clipper**

```
MEMOWRIT("Datei.txt,Memoedit(Memoread("Datei.txt")))
```

FoxPro

```
MODIFY FILE DieDatei.txt
```

Beispiel 2: **FoxPro**

```
*Mit MODIFY FILE wird eine temporäre Datei
```

```
*bearbeitet
```

```
COPY FILE Datei.txt TO DieTemp.txt
```

```
MODIFY FILE DieTemp.txt
```

```
IF NOT LASTKEY() == 27
```

```
SET SAFETY OFF
```

```
COPY FILE DieTemp.txt TO Datei.txt
```

```
SET SAFETY ON
```

```
ENDIF
```

```
ERASE DieTemp.txt
```

MEMORY()XE "MEMORY()"§

Stufe: 2

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

In Clipper: MEMORY(0) gibt die Größe des verfügbaren Speicherplatzes aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen MEMORY(), SYS(12) oder SYS(1001) ein.

Anmerkung: In FoxPro für MS-DOS geben MEMORY() und SYS(12) aus, wieviel Speicherplatz unter der 640KB-Grenze für die Ausführung externer Programme verfügbar ist. SYS(1001) gibt aus, wieviel Speicherplatz der FoxPro-Speicherverwaltung zur Verfügung steht und berücksichtigt dabei auch den HMA-Speicher im Bereich zwischen 640KB und 1 MB, sofern DOS auf diesen Bereich zugreifen kann .

In FoxPro für Windows gibt MEMORY() immer den Wert 640 und SYS(12) den Wert 655,360 aus.

Beispiel: **Clipper**

```
mem_avail=MEMORY(0)
```

FoxPro

```
mem_avail=MEMORY()
```

```
or
```

```
mem_avail=SYS(12)
```

```
or
```

```
mem_avail=SYS(1001)
```

MEMOTRAN()XE "MEMOTRAN()"§

Stufe: 2

In Clipper: MEMOTRAN() ersetzt die Zeichenkombination Wagenrücklauf/Zeilende (CR/LF).

In FoxPro: Erzeugt einen Fehler.

aus Clipper

Vorgehen: Löschen Sie diese Funktion. Falls nötig, können Sie stattdessen STRTRAN() einsetzen.

Anmerkung: Da in FoxPro Zeilen standardmäßig umbrochen werden, wird die Funktion MEMOTRAN() nicht benötigt. Falls sich Ihr Programm jedoch genau so wie bei Einsatz der Funktion MEMOTRAN() verhalten soll, verwenden Sie die Funktion STRTRAN().

Beispiel: **Clipper**

*Gemäß Vorgabe ersetzt MEMOTRAN harte
Zeilenumbrüche

*durch Semikolons und bedingte Zeilenumbrüche durch
*Leerzeichen, Zeilenvorschübe werden gelöscht.
memo_var=MEMOTRAN(memo_var, "", "")

FoxPro

*Harte Zeilenumbrüche durch Semikolons ersetzen
memo_var=STRTRAN(memo_var,CHR(13),";")
*Bedingte Zeilenumbrüche durch Leerzeichen ersetzen
memo_var=STRTRAN(memo_var,CHR(141)," ")
*Zeilenvorschübe löschen
memo_var=STRTRAN(memo_var,CHR(10))

MEMOWRIT()XE "MEMOWRIT()"§

Stufe: 2

In Clipper: Schreibt eine Zeichenfolge in die genannte Datei.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen COPY FILE oder COPY MEMO ein.

Beispiel: **Clipper**

Memowrit(MyVar)

FoxPro

```
REPLACE Test.notiz WITH MeineVar  
COPY MEMO Test.notiz TO DieDatei.txt
```

MLCOUNT()XE "MLCOUNT()"§

Stufe: 2

In Clipper: MLCOUNT() gibt die Anzahl umbrochener Zeilen, die in einer Zeichenfolge oder einem Memofeld enthalten sind, aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion, und setzen Sie stattdessen die äquivalente FoxPro-Funktion MEMMLINES() ein.

Beispiel: **Clipper**
mNO=MLCOUNT(NOTIZ)

FoxPro

```
mNO=MEMMLINES(NOTIZ)
```

MLPOS()XE "MLPOS()"§

Stufe: 2

In Clipper: MLPOS() ermittelt in einer Zeichenfolge oder einem Memofeld die Position der durch die Zeilennummer bezeichneten Textzeile.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie diese Funktion durch die FoxPro-Funktion MLINE().

Anmerkung: Die Systemvariable _MLINE speichert die Position des Memofeld-Offsets für die Funktion MLINE().

Beispiel **Clipper**
string = MEMOREAD("Temp.txt")

aus Clipper

? MLPOS(string, 40, 5)

Foxpro

SET MEMOWIDTH to 40
string = Einedbf.Notiz
= MLINE(string,5,_mline)
? _MLINE

NETERR()XE "NETERR()"§

Stufe: 1

In Clipper: NETERR() ermittelt, ob die Befehle USE, USE...EXCLUSIVE oder APPEND BLANK bei Arbeitssitzungen in einem Netzwerk erfolgreich ausgeführt werden konnten.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Funktionsaufruf. Sie können mit Hilfe einer ON ERROR Routine feststellen, ob USE und USE...EXCLUSIVE oder APPEND BLANK Befehle fehlerfrei ausgeführt werden. Lesen Sie dazu bitte den Abschnitt "Fehlerbehandlung."

NETNAME()XE "NETNAME()"§

Stufe: 1

In Clipper: NETNAME() liefert die Netzwerkkennung des Arbeitsplatzrechners.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie NETNAME() durch SYS(0).

Anmerkung: Wenn FoxPro im Netz läuft, liefert SYS(0) die Netzwerkkennung des Rechners (Rechnername und -nummer) zurück. Dazu muß jedoch die Netz-Shell geladen und zuvor von

der Netzsoftware eine Rechnernummer und ein Rechnername vergeben worden sein. In Novell-Netzen müssen Sie dazu das Login-Script durch folgenden Eintrag ergänzen:

```
MACHINE="%USER_ID,%P_STATION,%LOGIN_NAME"
```

Falls FoxPro nicht im Netz installiert ist oder falls im Netz weder eine Rechnernummer noch ein Rechnername vergeben wurden, liefert SYS(0) eine Zeichenkette mit einer Reihe von Leerzeichen (10 in FoxPro für MS-DOS bzw. 15 in FoxPro für Windows), die durch die Zeichenfolge Nummernzeichen (#), Leerzeichen und 0 abgeschlossen wird.

NEXTKEY()XE "NEXTKEY()"§

Stufe: 2

In Clipper: NEXTKEY() liest die nächste Tastatureingabe aus dem Tastaturpuffer, ohne sie aus dem Puffer zu löschen, und liefert den entsprechenden INKEY() Wert (oder 0, falls der Puffer leer ist).

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion, und setzen Sie stattdessen LASTKEY() ein.

Anmerkung: Beachten Sie, daß Clipper und FoxPro teilweise unterschiedliche Tastencodes und Tastaturbelegungen verwenden. In Anhang G finden Sie eine Zusammenstellung der Tastencodes, verändern Sie Ihr Programm gegebenenfalls entsprechend.

Eine andere FoxPro-Funktion, die in diesem Zusammenhang unter Umständen hilfreich ist, ist CHRSAW(), die den Wahrheitswert TRUE liefert, wenn ein Zeichen im Tastaturpuffer vorhanden ist, ohne jedoch den Pufferinhalt zu verändern.

aus Clipper

Beispiel: **Clipper**

```
IF NEXTKEY()=27
  RETURN .F.
ENDIF
```

FoxPro

```
IF LASTKEY()=27
  RETURN .F.
ENDIF
```

PCOUNT()XE "PCOUNT()"§

Stufe: 2

In Clipper: PCOUNT() liefert die Anzahl der Parameter, die einer Prozedur oder benutzerdefinierten Funktion tatsächlich übergeben wurden.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie PCOUNT durch die äquivalente Funktion PARAMETERS().

Beispiel: **Clipper**

```
prm_zahl=PCOUNT()
```

FoxPro

```
prm_zahl=PARAMETERS()
```

PROCLINE()XE "PROCLINE()"§

Stufe: 2

In Clipper: PROCLINE() gibt die Zeilennummer des aktuell ausgeführten Programmcodes aus, wobei die Zeilen ab dem Beginn des Hauptprogramms gezählt werden.

In FoxPro: Erzeugt einen Fehler.

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

Vorgehen: Löschen Sie diese Funktion, und setzen Sie stattdessen die äquivalente FoxPro-Funktion LINENO() ein.

Beispiel: **Clipper**
zeile_num=PROCLINE()

FoxPro
zeile_num=LINENO()

PROCNAME()XE "PROCNAME()"§

Stufe: 2

In Clipper: PROCNAME() liefert den Namen des Programms oder der Prozedur zurück, das bzw. die aktuell ausgeführt wird.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen die äquivalente FoxPro-Funktion PROGRAM() ein.

Beispiel: **Clipper**
prog_name=PROCNAME()

FoxPro
prog_name=PROGRAM()

READEXIT()XE "READEXIT()"§

Stufe: 2

In Clipper: Verändert die Tastenbelegung so, daß die Tasten NACH-OBEN und NACH-UNTEN zum Verlassen einer READ-Operation verwendet werden können.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Dieser Befehl wird in FoxPro nicht benötigt. Falls Sie die Pfeiltasten zum Verlassen einer READ-Operation einsetzen

aus Clipper

wollen, erstellen Sie die entsprechenden Anweisungen mit dem Befehl ON KEY LABEL.

Beispiel: **Clipper**

READEXIT(.T.)

FoxPro

ON KEY LABEL UPARROW CLEAR READ
ON KEY LABEL DNARROW CLEAR READ

READINSERT()XE "READINSERT()"§

Stufe: 2

In Clipper: Liefert die aktuelle Einstellung des Einfügemodus für READ und MEMOEDIT().

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen INSMODE() ein.

Beispiel: **Clipper**

READINSERT(.T.) &&Einfügemodus eingeschaltet

FoxPro

= INSMODE(.T.) &&Einfügemodus eingeschaltet

READVAR()XE "READVAR()"§

Stufe: 3

In Clipper: READVAR() gibt den Namen der aktuellen GET- oder MENU-Variablen aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Für GET-Variablen ersetzen Sie READVAR() durch VARREAD(). Für MENU-Variablen ersetzen Sie READVAR() entweder durch PROMPT() oder BAR().

Anmerkung: In Clipper wird READVAR() hauptsächlich zum Debuggen verwendet, in FoxPro kann es nicht für diese Zwecke eingesetzt werden. Verwenden Sie stattdessen das Trace- und das Debug-Fenster.

Beispiel: **Clipper**

```
FUNCTION MGültig
xvar=READVAR()
xwert=&READVAR()
SELECT Pcode
SET ORDER TO &xvar
SEEK xwert
IF !FOUND()
    RETURN .F.
ELSE
    RETURN .T.
ENDIF
```

FoxPro

```
FUNCTION MGültig
```

Verzeichnis inkompatibler Befehle und Funktionen

aus Clipper

```
xvar=VARREAD()  
xwert=&READVAR()  
SELECT Pcode  
SET ORDER TO (xvar)
```

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

```
SEEK xwert
IF !FOUND()
    RETURN .F.
ELSE
    RETURN .T.
ENDIF
```

RESTSCREEN()XE "RESTSCREEN()"§

Stufe: 2

In Clipper: RESTSCREEN() zeigt den Inhalt eines zuvor abgespeicherten Bildschirmbereichs an.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie RESTSCREEN() durch den äquivalenten FoxPro-Befehl RESTORE SCREEN.

Anmerkung: FoxPro verfügt über sehr elegante Mechanismen zur Verwaltung von Fenstern, die mehr können, als den Fensterinhalt speichern und wiederanzeigen. Befehle wie DEFINE WINDOW, ACTIVATE WINDOW, SHOW WINDOW und HIDE WINDOW stellen eine ausgezeichnete Alternative dar zum fortdauerndem Speichern und Wiederherstellen von Bildschirminhalten.

Beispiel: **Clipper**
RESTBILD(0, 0, 24, 29, AltBild)

FoxPro
RESTORE SCREEN FROM MeinBild

SAVESCREEN()XE "SAVESCREEN()"§

Stufe: 2

In Clipper: SAVESCREEN() speichert den Inhalt des genannten Programm-Konverter

Verzeichnis inkompatibler Befehle und Funktionen

aus Clipper

Bildschirmbereichs, damit dieser später erneut angezeigt werden kann.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie SAVESCREEN() durch den äquivalenten FoxPro-Befehl SAVE SCREEN.

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

Beispiel: **Clipper**

AltBild = SAVESCREEN(0,0,24,79)

FoxPro

SAVE SCREEN TO MeinBild

SCROLL()XE "SCROLL()"§

Stufe: 2

In Clipper: SCROLL() ermöglicht, daß der Inhalt eines bestimmten Bildschirmbereichs nach oben oder unten geblättert oder ausgeblendet werden kann.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen den FoxPro-Befehl SCROLL ein.

Beispiel: **Clipper**

SCROLL(4, 5, 21, 74, 0)

FoxPro

SCROLL 4, 5, 21, 74, 0

SET CURSORXE "SET CURSOR"§

Stufe: 3

In Clipper: SET CURSOR OFF deaktiviert die Cursoranzeige während die Bildschirmanzeige erneuert wird.

In FoxPro: SET CURSOR OFF verhindert, daß der Cursor während der Ausführung einer @ ... GET, @ ... EDIT, WAIT oder INKEY() Anweisung dargestellt wird.

Vorgehen: Keine Änderungen erforderlich.

Anmerkung: Sie sollten die SET CURSOR OFF Befehle löschen. FoxPro zeigt standardmäßig den Cursor nicht an, während die

aus Clipper

Bildschirmanzeige neu aufgebaut wird. Wenn SET CURSOR auf OFF eingestellt ist, wird die Cursoranzeige auch während der Ausführung von GET-, EDIT- und WAIT-Anweisungen unterdrückt.

SET FUNCTION

Siehe INKEY().

SET KEYXE "SET KEY"§

Stufe: 2

In Clipper: SET KEY führt eine Prozedur aus, sobald die bezeichnete Taste gedrückt wird. Als Parameter wird SET KEY der INKEY() Wert einer Tastatureingabe übergeben.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Ersetzen Sie SET KEY durch den Befehl ON KEY LABEL. Setzen Sie für den in SET KEY übergebenen Parameter die entsprechende FoxPro-Tastenbezeichnung ein.

Anmerkung: Falls in FoxPro keine zum SET KEY Wert äquivalente Tastenbezeichnung existiert, verwenden Sie den Befehl ON KEY, der wie SET KEY eine numerische Parameterangabe erfordert (der betreffende numerische Tastencode in FoxPro kann sich allerdings vom Clipper-Wert unterscheiden). Der Befehl SET KEY wird nur noch aus Kompatibilitätsgründen unterstützt.

In Anhang G finden Sie eine Zusammenstellung der Tastencodes von FoxPro und Clipper.

Beispiel: **Clipper**

SET KEY -1 TO mein_prog && -1 steht für die Taste
F2

FoxPro

ON KEY LABEL F2 DO mein_prog && Empfohlene
Methode
oder

aus Clipper

ON KEY = 316 DO mein_prog && 316 steht für
die
 && Taste F2

SET RELATIONXE "SET RELATION"§

Stufe: 2

In Clipper: Das Schlüsselwort TO wird bei aufeinanderfolgenden SET
RELATION Befehlen, die einzelne Beziehungen definieren, in
jedem einzelnen Befehl verwendet.

In FoxPro: Das Schlüsselwort TO darf nur einmal verwendet werden,
ansonsten wird ein Fehler erzeugt.

Vorgehen: Löschen Sie alle TO Schlüsselwörter mit Ausnahme des ersten.

Anmerkung: Es handelt sich hier um einen kleinen Syntaxunterschied. Die
Befehle erfüllen ansonsten dieselbe Funktion.

Beispiel: **Clipper**

 SET RELATION TO kund INTO rechng, TO stat INTO
stat

FoxPro

 SET RELATION TO kund INTO rechng, stat INTO stat

SET SOFTSEEKXE "SET SOFTSEEK"§

Stufe: 2

In Clipper: SET SOFTSEEK schaltet das Suchverfahren SOFTSEEK ein
bzw.aus. Mit SOFTSEEK wird der Datensatzzeiger auf den
nächst höheren Schlüsselwert gesetzt, wenn der Befehl SEEK
scheitert, d.h. kein Datensatz den Suchbedingungen genau
entspricht.

In FoxPro: Erzeugt einen Fehler.

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

Vorgehen: Löschen Sie den Befehl. Setzen Sie stattdessen den äquivalenten FoxPro-Befehl SET NEAR ein.

Beispiel: **Clipper**
SET SOFTSEEK ON

FoxPro
SET NEAR ON

SET WRAPXE "SET WRAP"§

Stufe: 2

In Clipper: SET WRAP schaltet die Zeilenumbruchfunktion ein bzw. aus.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Befehl. Er wird in FoxPro nicht gebraucht.

Anmerkung: FoxPro arbeitet so, als sei SET WRAP immer auf ON gesetzt.

SETCANCEL()XE "SETCANCEL()"§

Stufe: 2

In Clipper: SETCANCEL() legt fest, ob das Programm mit der Tastenkombination ALT-C beendet werden kann oder nicht.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Befehl. Er wird in FoxPro nicht gebraucht. Die meisten FoxPro Programmierer verwenden entweder den Befehl ON ESCAPE oder legen eine Tastenkombination fest, mit der die Ausführung eines Programms abgebrochen werden kann.

Beispiel: **Clipper**

Programm-Konverter

aus Clipper

SETCANCEL(.T.) &&ALT-C beendet Programm

FoxPro

ON KEY LABEL F12 SUSPEND

Or

SET ESCAPE ON

Or

ON ESCAPE SUSPEND

SETCOLOR()XE "SETCOLOR()"§

Stufe: 2

In Clipper: SETCOLOR() gibt die aktuelle Farbeinstellung aus und erlaubt Ihnen, eine neue Farbeinstellung zu definieren.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diesen Befehl. Sie sollten die Routine neu schreiben und dabei mit den Farbschemata von FoxPro arbeiten. Allerdings können Sie auch mit der Funktion SET('COLOR') die aktuelle Farbeinstellung ermitteln und mit dem Befehl SET COLOR TO eine neue Farbeinstellung festlegen.

Anmerkung: Nähere Informationen zu den Farbschemata in FoxPro finden Sie im Abschnitt "Farbdarstellung".

Beispiel: **Clipper**

```
akt_farbe=SETCOLOR()  
neu_farbe=SETCOLOR("W+/B,W+/BG,N")
```

FoxPro

```
akt_farbe=SET('COLOR')  
SET COLOR TO "W+/B,W+/BG,N"
```

SETPRC()XE "SETPRC()"§

Stufe: 2

In Clipper: SETPRC() legt die Werte der internen Funktionen PROW() und PCOL() fest.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion. Definieren Sie die Werte von PROW() und PCOL() in FoxPro direkt mit Speichervariablen.

aus Clipper

Beispiel:

Clipper

SETPRC(0,0) && Setzt Zeile & Spalte auf
&& Position 0,0 zurück

FoxPro

```
_LMARGIN=0  
Oder  
EJECT
```

TEXTXE "TEXT"§

Stufe: 2

In Clipper: TEXT kann mit den Optionen TO PRINT und TO FILE aufgerufen werden.

In FoxPro: Die Optionen TO PRINT und TO FILE erzeugen Fehler.

Vorgehen: Löschen Sie TO PRINT und/oder TO FILE. In FoxPro können Sie mit SET CONSOLE OFF und SET PRINTER TO denselben Effekt erzielen.

Beispiel: **Clipper**

```
TEXT TO PRINT  
<An Bildschirm oder Drucker auszugebenden Text>  
ENDTEXT
```

FoxPro

```
SET CONSOLE OFF  
SET PRINTER TO LPT1    && oder Datei Datei.txt  
TEXT  
<Text, der an den Drucker ausgegeben wird>  
ENDTEXT
```

TONE()XE "TONE()"§

Stufe: 2

In Clipper: TONE() erzeugt ein Tonsignal mit der genannten Frequenz und Dauer.

aus Clipper

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Setzen Sie stattdessen SET BELL TO und ?? Befehle ein.

Beispiel: **Clipper**

TONE(150,18) && 1 Sekunde langes

Tonsignal

FoxPro

SET BELL TO 150, 1&& 1 Sekunde langes Tonsignal
?? CHR(7)

TYPE()XE "TYPE()"§

Stufe: 3

In Clipper: TYPE() kann unter anderem Arrays ermitteln und die Werte Array, syntaktischer Fehler und unbestimmter Fehler zurückliefern.

In FoxPro: TYPE() hat dieselbe Funktionalität wie in Clipper Summer '87, kann jedoch nicht die Werte ArrayXE "Array"§, syntaktischer Fehler und unbestimmter Fehler zurückliefern.

Vorgehen: Falls die Überprüfung den Datentyp Arrays einschließen soll, ersetzen Sie die Funktion TYPE() durch die unten abgedruckte Funktion ISARRAY(). die von Malcolm Rubel geschrieben wurde.

Beispiel: **Clipper**

TYPE('array_1')

FoxPro

=ISARRAY('array_1')

* Beispielcode aus FoxPro 2.0 Power Tools

* von Malcolm Rubel, Performance Dynamics

Verzeichnis inkompatibler Befehle und Funktionen aus Clipper

Associates

- * Bantam Computer Books. Alle Rechte vorbehalten.
- * Copyright (c) 1989, 90, 91
- * Funktion ergibt TRUE, falls genannte Variable ein
- * Array ist.

```
FUNCTION isarray
PARAMETERS var_name
PRIVATE var_name
IF type(var_name + '[1]') = 'U' .and. ;
   type(var_name + '[1,1]') = 'U'
   RETURN(.F.)
ELSE
   RETURN(.T.)
ENDIF
```

aus Clipper

WORD()XE "WORD()"§

Stufe: 2

In Clipper: WORD() wandelt die numerischen Parameter, die dem Befehl CALL übergeben wurden und die vom Typ DOUBLE sind, zum Typ INT um.

In FoxPro: Erzeugt einen Fehler.

Vorgehen: Löschen Sie diese Funktion. Diese Typumwandlung ist in FoxPro nicht erforderlich.

Das Analyse- Programm



Wie das Analyse-Programm arbeitet

Das Analyse-Programm lädt ein Programm in ein FoxPro-Memofeld und sucht nach Befehlen und Funktionen aus dBASE IV oder Clipper Summer '87, die nicht unterstützt werden oder die sich in FoxPro anders verhalten. (Die Originaldateien werden dabei nicht verändert.) Das Analyse-Programm legt eine Tabelle an, in der die potentiellen Problemfälle und deren Position in der Programmdatei (deren Inhalt in Memofelder kopiert wurde). Nachdem das Analyse-Programm diese Tabelle erstellt hat, können Sie darin nach einem Problembereich suchen. Wenn Sie einen Eintrag auswählen, öffnet das Analyse-Programm das Memofeld, lädt die zugehörige .PRG-Datei in das Textbearbeitungsfenster und setzt den Cursor in die Zeile, die das potentielle Problem enthält.

Denken Sie daran, daß das Analyse-Programm unter Umständen auch Programmzeilen markiert, die in FoxPro korrekt ausgeführt werden. Das Analyse-Programm sucht nur nach bestimmten Zeichenfolgen, es führt weder eine Syntaxprüfung durch noch führt es den Programmcode aus. Manche Befehle und Funktionen hängen von anderen Programmteilen ab, das Analyse-Programm ist nicht mit Funktionen zur Überprüfung dieser Abhängigkeiten ausgestattet.

Das Analyse-Programm verändert Dateien in keiner Weise.
Es kopiert die Dateien lediglich in Memofelder.

Eine neue Analyse erstellen (Menüoption: New Analysis)

Damit Sie ein Programm analysieren können, müssen Sie den Programm-Konverter installieren. (Die Installation wird weiter oben im Abschnitt "Der Programm-Konverter" beschrieben.) Führen Sie dann den Befehl DO MIGRATE.APP aus, indem Sie im Menü Programm die Option Ausführen wählen oder indem Sie den Befehl DO MIGRATE.APP in das Befehlsfenster eingeben.

Es wird ein neues Menü namens Migration Tools in die Hauptmenüzeile von FoxPro eingefügt. Wählen Sie im Menü

Migration Tools die Option **New Analysis**. Daraufhin wird ein Dialogfenster eingeblendet, das so ähnlich aussieht wie das Dialogfenster des Datei-Konverters. Hier werden jedoch nur .PRG-Dateien angezeigt.

Sie wählen eine Datei zur Analyse aus, indem Sie auf dem Dateinamen doppelklicken oder heben Sie den Dateinamen hervor und drücken Sie die Eingabetaste. Ausgewählte Dateien werden durch einen Stern links neben dem Dateinamen markiert. Falls alle Dateien aus einem Verzeichnis analysiert werden sollen, klicken Sie auf der Schaltfläche **Select All**, Sie markieren damit alle Dateien. Durch Klicken auf der Schaltfläche **Clear All** können Sie die Markierung sämtlicher Dateien aufheben. Wenn Sie die Markierung nur von einer Datei entfernen wollen, doppelklicken Sie auf dem Dateinamen, oder heben Sie den Dateinamen hervor und drücken Sie die Eingabetaste.

Wenn Sie in einem anderen Verzeichnis gespeicherte Dateien analysieren wollen, klicken Sie auf der Schaltfläche **Directory**, und stellen Sie das neue Verzeichnis ein. Zur Analyse von Dateien aus verschiedenen Verzeichnissen, müssen Sie die Verzeichnisse getrennt bearbeiten, d.h. Sie markieren und analysieren erst die Dateien in einem Verzeichnis und danach markieren und analysieren Sie die Dateien des anderen Verzeichnisses. Sie können auch die Dateien in ein gemeinsames Verzeichnis verlagern.

μ §

*Wählen Sie .PRG-Dateien aus, und klicken Sie auf der Schaltfläche **Process**, um eine neue Analyse zu erstellen.*

Die Dateien bearbeiten

Nachdem Sie alle Dateien markiert haben, klicken Sie auf der Schaltfläche **Process**. Das Analyse-Programm beginnt daraufhin, die Datei(en) zu analysieren. Zuerst werden die markierten .PRG-Dateien in FoxPro-Memofelder kopiert. Dann

durchsucht FoxPro die .PRG-Dateien nach Befehlen und Funktionen, die bei der Ausführung des Programms in FoxPro Probleme verursachen können. Die Bearbeitungsdauer hängt von der Anzahl der zu analysierenden Dateien und der Anzahl der gefundenen Problemfälle ab.

Beachten Sie, daß das Analyse-Programm zwar nach ganz ausgeschriebenen Befehlen wie auch nach Befehlsabkürzungen sucht (z.B. activate menu und acti menu), nicht korrekt abgekürzte Befehle (z.B. activ menu) aber nicht erkennt.

Erforderlicher Festplattenspeicher

Das Analyse-Programm erstellt praktisch Kopien Ihrer Programmdateien, wenn es die .PRGs in Memofelder einliest. Sie können den für die Analyse erforderlichen Speicherplatz wie folgt berechnen: Gesamtgröße aller zu analysierenden Dateien mal 1,2.

Einstellungen (Menüoption: Settings)

Der Programm-Konverter kann sowohl in dBASE IV als auch in Clipper Summer '87 geschriebene Programme bearbeiten. Falls Sie im Dialogfenster Settings den Programmtyp, der analysiert werden soll, noch nicht eingestellt haben, werden Sie aufgefordert, eine der Optionen **dBASE IV 1.5 or earlier**, **dBASE IV 2.0 or earlier** oder **Clipper Summer '87** auszuwählen.

μ §

Falls Sie noch nicht angegeben haben, welche Art von Programmen konvertiert und analysiert werden soll, fordert Sie der Prgoramm-Konverter zu einer entsprechenden Eingabe auf.

Sie können dieses Dialogfenster auch durch Auswahl der Option **Settings** aus dem Menü **Migration Tools** aufrufen. Stellen Sie den Programmtyp, der konvertiert werden soll, und den Analysegrad (Optionen **Sensitiy level**) ein, klicken Sie anschließend auf der Schaltfläche **OK**.

Analysegrad

Das Analyse-Programm unterscheidet vier Problemstufen:

- **Stufe XE "Problemstufen" §1** umfaßt Befehle und Funktionen, die von FoxPro nicht unterstützt werden. Diese Befehle und Funktionen müssen gelöscht oder durch die äquivalenten Befehle und Funktionen der FoxPro-Sprache.
- **Stufe 2** umfaßt Befehle und Funktionen, die in FoxPro Fehler erzeugen, für die es aber häufig sehr ähnliche Befehle und Funktion in FoxPro gibt.
- **Stufe 3** umfaßt Befehle und Funktionen, die in FoxPro zwar keine Fehler erzeugen, sich jedoch so unterschiedlich verhalten, daß sie Beachtung verdienen.
- **Stufe 4** umfaßt Befehle und Funktionen, die bei der Ausführung des Programms in FoxPro nur selten Probleme verursachen. Wenn Sie nach Problemen der Stufe 4 suchen lassen, werden viele Programmzeilen markiert, die für gewöhnlich problemsol ausgeführt werden.

Vorgabegemäß berücksichtigt das Analyse-Programm Probleme der Stufen 1, 2 und 3. Beachten Sie, daß die in diesem Dialogfenster vorzunehmende Einstellung der zu analysierenden Problemstufen etwas anderes ist als die Definition eines Filters im Analyse-Programm-Fenster. Die Definition von Filtern basiert auf dem eingestellten Analysegrad und wird weiter unten besprochen.

Alle im Dialogfenster Settings vorgenommenen Einstellungen bleiben auch nach Beendigung einer Arbeitssitzung in Kraft, sie werden gespeichert und in nachfolgenden Arbeitssitzungen verwendet.

Eine Analyse öffnen (Menüoption: Open Analysis)

Mit dieser Menüoption können Sie eine früher erstellte Analyse laden und weiterbearbeiten. Die Analyseergebnisse werden in Tabellen mit der Erweiterung .EXP gespeichert. Beachten Sie, daß Sie keine neuen Dateien in eine bereits erstellte Analyse

aufnehmen können.

Das Fenster "Program Analyzer"

Das Analyse-Programm erstellt eine Tabelle und legt für jeden potentiellen Problemfall einen Datensatz an. Darin wird der Name des Befehls bzw. der Funktion und die Position (Zeilennummer) in der Programmdatei gespeichert. Diese Daten werden im Fenster "Program Analyzer" angezeigt.

μ §Das Analyse-Programm zeigt eine Liste der in Programmen enthaltenen Problemfälle an.

Im oberen linken Fensterbereich werden alle Befehle und Funktionen, die bei der Ausführung in FoxPro möglicherweise Probleme verursachen, angezeigt. In der äußersten linken Spalte wird unter der Überschrift "Level" angezeigt, welcher Problemstufe (1, 2, 3 oder 4) der Befehl oder die Funktion zugeordnet wird. (Diese Problemstufen werden weiter unten im Abschnitt "Filtern und Sortieren" näher erläutert.)

In der nächsten Spalte wird unter der Überschrift "Program" der Name des Programms und die Zeilennummer, in der der Problemfall gefunden wurde, angezeigt. Der Name des Befehls bzw. der Funktion wird in der Spalte mit der Überschrift "Issue" ausgegeben.

Wenn Sie im linken Fensterbereich einen Problemfall markieren, wird im rechten Fensterbereich unter der Überschrift "Overview" eine Beschreibung des betreffenden Problems eingeblendet. Diese Problembeschreibung erläutert, wie der Befehl bzw. die Funktion in dBASE oder Clipper Summer '87 interpretiert wird (abhängig von der Einstellung im oben beschriebenen Dialogfenster Settings), wie in FoxPro, und es werden Hinweise gegeben, wie Sie sich daraus ergebende Probleme angehen und lösen können.

Die Texte, die im Fenster Overview angezeigt werden, sind in diesem Handbuch in den Abschnitten mit Übersichten zu dBASE- bzw. Clipper-Problemen abgedruckt und zusätzlich mit Beispielen versehen.

Filtern und Sortieren

Damit Sie potentielle Probleme systematisch angehen können, gibt Ihnen das Analyse-Programm die Möglichkeit, Problemfälle nach unterschiedlichen Kriterien zu sortieren: nach Programmname, nach den Namen der Befehle und Funktionen (issue) und nach Problemstufen (level). Mit den Optionsfeldern unter der Überschrift "Order" legen Sie das Sortierkriterium fest.

Sie können auch einen Filter definieren, wenn Sie sich nur die Problemfälle einer bestimmten Problemstufe anzeigen lassen möchten: 1, 2, 3 oder 4. Sie setzen einen Filter, indem Sie das entsprechende Optionsfeld unter der Überschrift "Sensitivity" einstellen. Wenn nur Problemfälle aufgeführt werden sollen, die schon bearbeitet wurden oder nur solche, die noch nicht bearbeitet wurden, markieren Sie das betreffende Optionsfeld unter der Überschrift "View". Es stehen die Optionen **All items** (Alle Fälle), **Edited Only** (Nur bearbeitete Fälle) und **Unedited Only** (Nur unbearbeitete Fälle) zur Auswahl. (Im nächsten Abschnitt wird die Bearbeitung von Programmdateien erläutert.)

Vom Analyse-Programm aus Programmdateien bearbeiten

Sie können vom Analyse-Programm aus zu einer Programmdatei wechseln, indem Sie im linken Bereich des Fensters Program Analyzer auf einem Eintrag doppelklicken oder indem Sie einen Eintrag hervorheben und die Eingabetaste drücken. Das Analyse-Programm lädt automatisch die entsprechende Datei in den Programmmeditor und setzt den Cursor in die Zeile, in der sich das potentielle Problem befindet.

Falls Sie Änderungen vorgenommen haben, können Sie mit der Tastenkombination STRG+W oder durch Klicken auf das Systemmenüfeld in der linken oberen Fensterecke diese Änderungen in die Programmdatei schreiben lassen. Danach wird wieder das Fenster Program Analyzer eingeblendet.

Problemfälle, die bearbeitet wurden, werden durch einen Stern neben der Zahl, die die Problemstufe anzeigt, markiert.

μ §Wenn Sie auf einem Eintrag doppelklicken, wird der Texteditor geladen und der Cursor in die Zeile mit der betreffenden Zeilennummer gesetzt.

Wenn Sie zum Fenster Program Analyzer zurückkehren wollen, ohne daß Änderungen gespeichert und Problemfälle als gelöst markiert werden, verlassen Sie den Programmeditor durch Drücken der Taste ESC.

Falls Sie Programmzeilen eingefügt oder gelöscht haben, wird bei späteren Aufrufen der Programmdatei vom Fenster Program Analyzer aus der Cursor unter Umständen nicht genau in die Zeile gesetzt, die das Problem enthält, da die Zeilennummern nach der Erstellung der Analysedatei verändert wurden. Für gewöhnlich befindet sich die gesuchte Zeile nur wenige Zeilen ober- oder unterhalb des Cursors. Falls Sie jedoch größere Codesegmente eingefügt oder gelöscht haben, sollten Sie die Option **Suchen** aus dem Menü **Bearbeiten** aufrufen, den Namen des Befehls oder der Funktion eingeben und FoxPro nach dem Eintrag suchen lassen.

Einen anderen XE "Texteditor, extern"§Texteditor einsetzen

Wenn Sie mit Ihrem gewohnten Texteditor arbeiten wollen, können Sie nicht wie im vorhergehenden Abschnitt beschrieben aus dem Analyse-Programm heraus einen Problemfall in der Programmdatei bearbeiten. Sie müssen stattdessen einen Analysebericht ausdrucken lassen, und durch Auswahl der Option **Close Tools** im Menü **Migration Tools** zu FoxPro zurückkehren. Aus FoxPro heraus können Sie nun Ihren Editor laden und anhand des Analyseberichts die Problemfälle bearbeiten. Für dBASE-Benutzer sollte der FoxPro-Editor eine Verbesserung darstellen, er hat einen größeren Funktionsumfang und ist schneller als der dBASE-Editor. Außerdem sind meist nur geringfügige Änderungen vorzunehmen, und der Einsatz eines externen Texteditors, mit dem Sie vertraut sind, ist daher nicht so lohnend.

Falls Sie einen externen Editor einsetzen, bearbeiten Sie die ursprünglichen Programmdateien und nicht die in die FoxPro-Memofelder kopierten Programme. **Exportieren Sie keine Dateien aus dem Analyse-Programm, die dadurch von Ihnen bearbeitete Dateien unter Umständen überschrieben werden!**

Das HilfesystemXE "Hilfe"§

Klicken Sie auf der Schaltfläche **Help**, oder drücken Sie die Funktionstaste F1, wenn Sie Hilfeinformationen zum Dialogfenster Program Analyzer benötigen. Sie können das reguläre Hilfesystem von FoxPro aufrufen und sich Informationen zu den in FoxPro enthaltenen Befehlen und Funktionen anzeigen lassen, indem Sie auf der Schaltfläche **Fox Help** klicken. Das Analyse-Programm versucht in diesem Fall, Informationen zu dem Eintrag zu finden, der im linken Fensterbereich gerade hervorgehoben ist.

AnalyseberichteXE "Berichte, Analyse-Programm"§

Das Analyse-Programm erstellt Berichte, die Sie sich auf dem Bildschirm anschauen oder ausdrucken können. Sie können die Tabelle potentieller Problemfälle sich auf dreierlei Weise ausgeben lassen:

- Sortiert nach Programmnamen und Problem
- Sortiert nach Programmnamen und Zeilennummer
- Sortiert nach Problem

Diese Berichte sollen es Ihnen erleichtern, die Teile Ihrer Anwendung, die möglicherweise geändert werden müssen, zu orten. Zudem ermöglichen diese Berichte, daß Sie Ihre eigene Textverarbeitung einsetzen oder andere Programme für globale Suchen- und Ersetzen-Operationen verwenden. Sie können mit dem Berichtsgenerator von FoxPro auch selbst Berichte aus einer Analysedatei (.EXP) erstellen. Beachten Sie, daß alle im Analyse-Programm gesetzten Filter (beispielsweise Filter zur Anzeige nur der bearbeiteten Fälle) auch für die Ausgabe von Berichtsdaten gelten.

Die Befehlsreferenztafel durchsehen "Befehlsreferenz"

Wenn Sie schnell nachschlagen wollen, wie sich bestimmte Befehle oder Funktionen in FoxPro, dBASE oder Clipper, können Sie sich die Befehlsreferenztafel ansehen. Sie rufen diese Tabelle über das Menü **Migration Tools** mit der Option **View Issues** auf. Auf diesen Befehl hin wird ein Fenster eingeblendet, in dem alle Befehle und Funktionen aus dBASE IV oder Clipper Summer '87, die bei der Programmausführung in FoxPro Probleme verursachen können, angezeigt werden. Das Analyse-Programm durchsucht die Programmdateien nach diesen Schlüsselwörtern. Sie können auch einen Filter einsetzen und sich nur Befehle oder nur Funktionen ausgeben lassen. Sobald Sie einen bestimmten Befehl oder eine Funktion auswählen, wird die zugehörige Beschreibung angezeigt. Dieser Text erklärt, wie der Befehl bzw. die Funktion in dBASE oder Clipper interpretiert wird und wie FoxPro auf diesen Befehl oder die Funktion reagiert. Es handelt sich hierbei um dieselbe Beschreibung, die im Fenster Program Analyzer eingeblendet wird und die Sie hier in den Abschnitten mit den Verzeichnissen der dBASE- und Clipper-Probleme abgedruckt finden.

Nach der Bearbeitung der Problemfälle

Nachdem Sie alle vom Analyse-Programm angezeigten Problemfälle überprüft und die nötigen Änderungen durchgeführt haben, können Sie das Programm in eine reguläre .PRG-Datei exportieren.

Das Analyse-Programm liest die markierten .PRG-Dateien in einzelne Memofelder ein. Mit der Schaltfläche **Export .PRGs** " Export .PRG" im Fenster Program Analyzer können Sie Programme in die .PRG-Dateien zurück schreiben.

Stellen Sie zuerst sicher, daß eine Analysedatei geöffnet ist. Klicken Sie dann auf der Schaltfläche **Export .PRGs**. Daraufhin wird das Dialogfenster Export angezeigt.

Wählen Sie die .PRG-Dateien aus, in die der Programmcode

exportiert werden soll. Doppelklicken Sie dazu auf dem Dateinamen oder heben Sie den Dateinamen hervor, und drücken Sie die Eingabetaste. Mit der Schaltfläche **Directory** können Sie festlegen, in welchen Verzeichnis die .PRG-Dateien geschrieben werden sollen. Es ist normalerweise sinnvoll, die .PRG-Dateien nicht in das Verzeichnis, aus dem sie gelesen wurden, zu exportieren. Damit bewahren Sie Ihren ursprünglichen Quelltext. Wenn Sie die Dateien und das Zielverzeichnis gewählt haben, klicken Sie auf der Schaltfläche **Export**. Falls eine Datei gleichen Namens bereits existiert, macht Sie eine Warnung darauf aufmerksam, daß diese Datei beim Export überschrieben wird.

Versichern Sie sich noch einmal, daß sie diese Datei nicht mehr brauchen, bevor Sie mit dem Export fortfahren! Überschriebene Dateien können nicht wiederhergestellt werden.

Falls Sie die Programmdateien mit einem externen Texteditor bearbeitet haben, dürfen Sie die .PRG-Dateien nicht exportieren. Die in den Memodateien gespeicherten .PRG-Dateien wurden in diesem Fall nicht korrigiert und beim Export würden wahrscheinlich die korrigierten Programmdateien überschrieben werden.

Hinweis: .PRG-Dateien, die exportiert wurden, werden nicht aus der Analyse-Tabelle gelöscht und können später nochmals exportiert werden. Wenn Sie mit dem Programm-Konverter alle gewünschten Dateien analysiert und konvertiert haben, können Sie alle vom Analyse-Programm erstellten Dateien löschen. Es handelt sich dabei um die Dateien mit den Extensionen .CDX, .FPT, .EXP und .PXP. (Die beiden letzten Extensionen werden in FoxPro normalerweise nicht verwendet.) Passen Sie beim Löschen auf, daß Sie nicht unbeabsichtigt Dateien löschen, die Sie später noch brauchen!

FoxPro-Projekte

Wie in dBASE können Sie .PRG-Dateien mit dem Befehl DO direkt aufrufen und ausführen lassen. Sie sollten den FoxPro-

Projektmanager nutzen, der Projektmanager verwaltet alle zu einer Anwendung gehörenden Dateien und erstellt darauf aufbauend eine Anwendungsdatei. Sie aktivieren den Projektmanager, indem Sie ein Projekt erstellen und den Namen des aufrufenden Programm in das Projekt einfügen. Erstellen Sie Ihre Anwendung und überlassen Sie es dem Projektmanager, alle zugehörigen Programmdateien einzubinden.

Falls FoxPro beim Kompilieren einer Datei, die zu einer neu erstellten Anwendung gehört, auf einen Fehler stößt, wird eine Datei mit dem Namen des Projekts und der Extension .ERR angelegt. Anhand dieser Datei können Sie herausfinden, ob und welche Fehler vom Analyse-Programm übersehen wurden.

Wenn Sie ausführbare Dateien (.EXE) erstellen wollen, brauchen Sie das FoxPro Distribution Kit für MS-DOS bzw. Windows. Mit dem Distribution Kit wird der Projektmanager dazu befähigt, selbständig ausführbare Programmdateien zu erstellen.

Anhänge

Anhang A: Der Befehl SET COMPATIBLE

Hinweis: Der Befehl SET COMPATIBLE ist in FoxPro vorgabegemäß auf OFF gestellt.

Befehl oder Funktion	SET COMPATIBLE	Erklärung
Befehle zur Dateibearbeitung mit Angabe eines Laufwerks	DB4 ON	Wenn der Befehl ein Laufwerk angibt, dann wird nur im Befehl genannten Laufwerk nach der Datei gesucht, auch wenn es eine Pfadangabe für FoxPro gibt.
	OFF	Wenn der Befehl ein Laufwerk angibt, wird zuerst das genannte Laufwerk durchsucht, danach der FoxPro-Suchpfad, sofern eine Pfadangabe zu FoxPro existiert.
<p>Beispiel:</p> <pre>SET PATH TO D:\TESTDIR DO C:TEST</pre> <p>In diesem Beispiel ist TEST eine Programmdatei, die sich im Verzeichnis D:\TESTDIR befindet. Wenn SET COMPATIBLE auf DB4 gesetzt ist, wird die Datei nicht gefunden und nicht ausgeführt, da nur das genannte Laufwerk durchsucht wird. Wenn SET COMPATIBLE allerdings auf OFF gesetzt ist, wird auch der FoxPro-Pfad durchsucht und die Datei wird gefunden und ausgeführt.</p>		
	DB4 ON	Parameter, die durch Variablen-

PARAMETERS Übergabe durch Variablen-Referenz		Referenz übergeben werden (SET UDFPARMS TO REFERENCE), können von der aufgerufenen Prozedur verändert werden, und Änderungen werden an das aufrufende Programm zurückgegeben.
	OFF	Bei Parametern, die durch Variablen-Referenz übergeben werden (SET UDFPARMS TO REFERENCE), werden Änderungen nicht an das aufrufende Programm zurückgeleitet.
@...GET...RANGE	DB4 ON	Die RANGE-Klausel wird immer überprüft.
	OFF	Die RANGE-Klausel wird nur überprüft, falls Daten verändert wurden.
@...SAY (Sonderzeichen)	DB4 ON	Alle Sonderzeichen mit Ausnahme von CHR(7), das den Warnton aktiviert, werden ausgegeben.
	OFF	Alle Sonderzeichen einschließlich CHR(7), das hier den Warnton nicht aktiviert, werden ausgegeben.

COMPATIBLE

Befehl oder Funktion	SET COMPATIBLE	Erklärung
@...SAY (SCREEN SCROLL)	DB4 ON	Ausgaben, die über die rechte untere Bildschirmecke hinausgehen, werden angezeigt, indem der Bildschirminhalt nach oben geblättert wird.
	OFF	Ausgaben, die über die rechte untere Bildschirmecke hinausgehen, werden nicht angezeigt sondern abgeschnitten.
@...SAY (SET STATUS ON)	DB4 ON	Ausgaben können die Statuszeile überschreiben. Text, der über das Zeilenende hinausgeht, wird oberhalb der Statuszeile angezeigt und von hier nach oben geblättert.
	OFF	Ausgaben können die Statuszeile nicht überschreiben. Über den rechten Bildschirmrand hinausgehender Text wird abgeschnitten.
@...SAY (m/PICTURE)	DB4 ON	Wenn mit einer PICTURE-Klausel numerische Daten ausgegeben werden, wird die letzte Dezimalstelle gerundet.
	OFF	Wenn Daten mit Hilfe einer PICTURE-Klausel angezeigt werden, werden numerische Werte nicht gerundet sondern abgeschnitten.

ACTIVATE SCREEN und ACTIVATE WINDOW	ON	Bei der Aktivierung des Bildschirms oder eines Fensters wird der Cursor auf die Position 00,00 gesetzt.
	OFF	Die Cursorposition wird nicht verändert.
APPEND MEMO	ON	Falls keine Dateierweiterung angegeben wird, wird gemäß Vorgabe die Erweiterung .TXT verwendet.
	OFF	In FoxPro ist kein Vorgabewert definiert. Wenn keine Dateierweiterung angegeben wird, wird daher auch keine Erweiterung verwendet.
GO GOTO (mit TALK ON)	ON	FoxPro gibt eine Meldung aus, in der der Aliasname des aktuellen Arbeitsbereichs und die Position des Datensatzzeigers genannt werden.
	OFF	FoxPro gibt keine Meldung zur Position des Datensatzzeigers aus.

COMPATIBLE

Befehl oder Funktion	SET COMPATIBLE	Erklärung
MENU- und POPUP-Befehle	ON	<p>Popup-Menüs werden im aktiven Fenster erstellt, bei Aktivierung des Menüs wird der Cursor auf eine Option des Popup-Menüs gestellt. Falls das Popup-Menü in Zeile 0 plaziert wird, wird die gesamte Zeile Bestandteil der Menüleiste.</p>
	OFF	<p>Popup-Fenster werden in einem eigenen Fenster erstellt, und der Cursor bleibt bei Aktivierung des Menüs unverändert auf seiner Position im aktiven Fenster. Bei Platzierung in Zeile 0 werden nur die Menüblöcke zu Bestandteilen der Menüleiste.</p>
PLAY MACRO	ON	<p>Wenn ein Makro, dem eine Taste im Bereich A-Z zugeordnet ist, abgespielt wird (z.B. PLAY MACRO A), wird impliziert, daß die Tastenkombination ALT+F10 vor dem Buchstaben gedrückt wurde. FoxPro führt das Makro aus, dem die Tastenkombination ALT+F10+Buchstabe, hier also ALT+F10+A, zugeordnet ist. Bei Makros mit der Tastenbelegung F1 bis F9, wird impliziert, daß die Taste ALT vor der Funktionstaste betätigt wurde. FoxPro führt das</p>

		Makro aus, dem die Tastenkombination ALT+F1, z. B. ALT+F1, zugeordnet ist. .
PLAY MACRO	OFF	FoxPro führt das bezeichnete Makro aus, ohne implizite Tastatureingaben einzufügen. Sie finden in der FoxPro Dokumentation eine Liste der zulässigen Makro-Tastenkombinationen.
READ (mit einer VALID-Klausel nach einem GET-Befehl)	ON	Wenn die Taste ESC gedrückt wird, während sich der Cursor in einem Feld mit einer VALID-Klausel befindet, wird damit eine Überprüfung initiiert. Falls die Eingabe ungültig ist, wird die Warnung "Ungültige Eingabe" eingeblendet.
Befehl oder Funktion	SET COMPATIBLE	Erklärung
	OFF	Die Taste ESC initiiert keine Überprüfung der Eingabe. Auch wenn bei einer Überprüfung (ESC nicht gedrückt) unzulässige Eingaben gefunden werden, wird keine Warnung ausgegeben.
READ and TRANSFORM (mit einer numerischen PICTURE-Klausel)	ON	Der Wert der Variablen oder des Ausdrucks wird auf die in der PICTURE-Klausel angegebene Zahl von Dezimalstellen gerundet.

Der Befehl SET

COMPATIBLE

	OFF	Der Wert der Variablen oder des Ausdrucks wird so weit verkürzt, bis er die in der PICTURE-Klausel genannte Anzahl von Stellen hat.
READ-Befehle (geschachtelt)	ON	FoxPro löscht alle in der untergeordneten Ebene definierten GET-Variablen (führt implizit einen CLEAR GETS Befehl aus), bevor es zu einem READ-Befehl höherer Ebene zurückkehrt.
	OFF	Bei der Rückkehr zu einem READ-Befehl höherer Ebene werden die in der niederen Ebene definierten GET-Variablen nicht gelöscht.
RUN/!	ON	Der Cursor wird in Zeile 24, Spalte 0 gesetzt. Alle nachfolgenden Ausgaben werden ab dieser Position angezeigt.
	OFF	Die Cursorposition wird nicht verändert. Alle nachfolgenden Ausgaben werden ab dieser Position angezeigt.
RUN/! (mit STATUS ON)	ON	Ausgaben des Programms, das mit einem RUN Befehl ausgeführt wurde, werden um zwei Zeilen nach oben geblättert, bevor die Programmsteuerung an FoxPro zurückgegeben wird.
	OFF	Ausgaben des Programms, das mit

		einem RUN Befehl ausgeführt wurde, werden um drei Zeilen nach oben geblättert, bevor die Programmsteuerung an FoxPro zurückgegeben wird.
SET COLOR TO	ON	Wenn SET STATUS auf ON gesetzt ist, werden die letzte (n) und die vorletzte Zeile (n-1) mit den im Befehl SET COLOR TO angegebenen Farben neu gezeichnet.
	OFF	Wenn SET STATUS auf ON gesetzt ist, werden die letzten (n, n-1 und n-2) Zeilen mit den im Befehl SET COLOR TO angegebenen Farben neu gezeichnet.

COMPATIBLE

Befehl oder Funktion	SET COMPATIBLE	Erklärung
SET FIELDS	ON	Wenn SET FIELDS TO ohne Angabe einer Feldliste oder mit der Option ALL aufgerufen wird, wird SET FIELDS OFF gesetzt. Damit werden alle Felder aus der Feldliste gelöscht.
	OFF	Wenn SET FIELDS TO ohne Angabe einer Feldliste oder mit der Option ALL aufgerufen wird, wird SET FIELDS TO gleich einem Nullwert gesetzt. (Damit werden alle Felder aus der Feldliste gelöscht.)
SET MESSAGE	ON	Die in diesem Befehl angegebene Zeichenfolge wird sofort in der untersten Bildschirmzeile angezeigt, wobei der Text in einer anderen Farbe ausgegeben wird..
	OFF	Die Zeichenfolge wird nur angezeigt, wenn Sie SET STATUS auf ON setzen, dann werden sowohl Zeichenfolge als auch die Zeile, in der sie angezeigt wird, in einer anderen Farbe angezeigt.
SET PRINT TO <Datei>	ON	Die Ausgabedatei erhält die Erweiterung .PRT, falls keine andere Erweiterung angegeben wird.

	OFF	Die Ausgabedatei erhält nur eine Erweiterung, wenn eine Dateierweiterung explizit angegeben wurde.
STORE	ON	STORE kann nicht eingesetzt werden, um alle Elemente eines Arrays zu initialisieren.
	OFF	STORE kann eingesetzt werden, um alle Elemente eines Arrays mit dem angegebenen Wert zu initialisieren.
SUM	ON	Der von SUM gelieferte Ergebniswert wird mit der in SET DECIMALS angegebenen Zahl von Dezimalstellen ausgegeben.
	OFF	Der von SUM gelieferte Wert wird mit der Anzahl von Dezimalstellen ausgegeben, die in der Datenbankstruktur für das mit SUM berechnete Feld definiert ist.
INKEY() und LASTKEY()	ON	Die Taste POS1 und die Tastenkombination UMSCHALT+POS1 liefern den Wert 26, die Tastenkombination STRG+LINKS liefern den Wert 1.
	OFF	Schlagen Sie die Tastencodes bitte in der unter INKEY() aufgeführten Tabelle im Handbuch <i>Befehle & Funktionen</i> nach.

Der Befehl SET

COMPATIBLE

Befehl oder Funktion	SET COMPATIBLE	Erklärung
LIKE()	ON	Vor dem Vergleich werden nachstehende Leerzeichen aus den zu vergleichenden Zeichenfolgen entfernt.
	OFF	Alle Zeichen sind signifikant, beim Vergleich werden daher nachstehende Leerzeichen nicht ignoriert.
SELECT()	ON	SELECT() liefert die Nummer des höchsten nicht benutzten Arbeitsbereichs.
	OFF	SELECT() liefert die Nummer des aktuell gewählten Arbeitsbereichs.
SYS(2001,'COLOR')	ON	Gibt die aktuelle Einstellung von SET COLOR ON OFF aus.
	OFF	Gibt die mit SET COLOR TO definierten Farbpaare aus.

Anhang B

dBASE-Dateien und ihre Verwendung in FoxPro

FoxPro benötigt nicht alle von dBASE erstellten Dateien zur Ausführung Ihrer Anwendung. Nachfolgend werden alle dBASE-Dateitypen aufgeführt, ihre Funktion beschrieben, Hinweise dazu gegeben, wie diese Dateien zu behandeln sind und in welchen Abschnitten dieses Handbuches Sie dazu nähere Informationen finden.

Datei- typ	Beschreibung	Was tun	Siehe
Datenbankdateien			Tabellen-, Memo- und Indexdateien aus dBASE verwenden
.DBF	Tabellendatei	Wird von FoxPro verwendet. Keine besonderen Maßnahmen erforderlich.	
.DBT	Memodatei	Wird von FoxPro konvertiert (und gelöscht), wenn die zugehörige Tabelle geöffnet wird.	
.MDX	Mehrfach- indexdatei	Wird von FoxPro ignoriert. FoxPro legt automatisch.CDX- Indexdateien an, wenn die zugehörige Tabelle geöffnet wird.	
.NDX	Einfachindex-	Wird von FoxPro ignoriert.	

	datei	FoxPro erstellt einen äquivalenten .IDX-Index, wenn ein .NDX-Index aufgerufen wird.
--	-------	---

Formatdateien			
.SCR	Bildschirmmaske	Mit dem Datei-Konverter zum FoxPro-Maskenformat (.SCX) konvertieren.	Maskendateien; .SCR-Dateien konvertieren
.FMT	Aus .SCR-Datei erzeugte Formatdatei	Wird von FoxPro verwendet. (Muß unter Umständen modifiziert werden..)	
.FMO	Kompilierte Formatdatei	Wird von FoxPro nicht verwendet.	Formatdateien modifizieren

Berichte			
.FRM	Berichts- maskendatei	Mit dem Datei-Konverter zum FoxPro-Berichtsformat (.FRX) konvertieren.	dBASE- Berichte
.FRG	Aus .FRM- Datei erzeugte Berichtsdatei	Wenn die zugrundeliegende .FRM-Datei konvertiert wird, wird die .FRG- Datei nicht benötigt, andernfalls führen Sie die .FRG-Datei in FoxPro aus.	dBASE- Berichte
.FRO	Kompilierte Berichtsformat- datei	Wird von FoxPro nicht verwendet.	

Abfragen			
.QBE	Abfragedatei	Können nach leichter Modifikation in FoxPro ausgeführt werden..	dBASE- Abfragen
.QBO	Kompilierte Abfragedatei	Wird von FoxPro nicht verwendet.	

Etiketten			
.LBL	Etiketten-	Mit dem Datei-Konverter zum	Etiketten

	formatdatei	FoxPro-Etikettenformat (.LBX) konvertieren.
.LBG	Aus .LBL-Datei erzeugte Etikettenformatdatei	Wenn die zugrundeliegende .LBL-Datei konvertiert wird, wird die .LBG-Datei nicht benötigt, andernfalls führen Sie die .LBG-Datei in FoxPro aus
.LBO	Kompilierte Etikettendatei	Wird von FoxPro nicht verwendet.

Programme		
.PRG	Programm-quellcode	Mit Hilfe des Analyse-Programms an FoxPro anpassen.
.DBO	Kompiliertes Programm	Wird von FoxPro nicht verwendet.
.PRS	SQL-Befehls-oder Prozedurendatei	In FoxPro neu erstellen.

.FMT-
und .PRG-
Dateien
konvertieren

**Vom
Programmgenerator
erzeugte Dateien**

Hinweise zu Programmgenerator-Anwendungen

.APP	Haupt- programmdatei	Wird von FoxPro nicht verwendet.
.BAR	Zeilenmenü- objekt	Wird von FoxPro nicht verwendet.
.BCH	Stapelverarbeitungso- bjekt	Wird von FoxPro nicht verwendet.
.FIL	Dateilisten- objekt	Wird von FoxPro nicht verwendet.
.POP	Popup- Menüobjekt	Wird von FoxPro nicht verwendet.
.STR	Felderlisten- objekt	Wird von FoxPro nicht verwendet.
.VAL	Wertelisten- objekt	Wird von FoxPro nicht verwendet.

Anhang C:
dBASE-Fehlernummern, die in FoxPro für andere Fehler stehen

Fehler	dBASE IV	FoxPro
67	Interner Fehler: EVAL: Arbeitsspeicherüberlauf.	Fehler bei der Auswertung eines Ausdrucks.
95	Quellcode stimmt nicht mit Objektcode überein.	Anweisung im interaktiven Modus nicht zulässig.
103	Befehl kann nicht durch sich selbst aufgerufen werden.	Verschachtelung von Do zu tief.
130	Befehl in Formatdatei nicht zulässig.	Datensatz ist nicht gesperrt.
178	MENÜ wurde nicht aktiviert.	MENU wurde nicht aktiviert.
179	POPUP wurde nicht aktiviert.	POPUP wurde nicht aktiviert.
225	Die Einstellung für den rechten Rand muß 255 sein.	Kein gültige RapidFile-Datenbank.
226	Tabulatorstopps müssen in aufsteigender Reihenfolge eingegeben werden.	"<Name>" ist keine Dateivariablen.

232	Aliasausdruck nicht im gültigen Bereich (1-10)	"<Name>" ist kein Array
279	Eingabebereich für Pop-up-Menü bereits festgelegt.	Menü/Popup wurde nicht auf den Stack gelegt.
411	Maximale Größe von Memofeld überschritten (64K)	RUN/! Befehlszeichenfolge zu lang
412	Kein gültiges Laufwerk: <Laufwerk>	Umgebungsvariable COMSPEC nicht gefunden
1001	Name ist länger als 10 Zeichen.	Option nicht verfügbar
1002	Zeichen ungültig	Eingabe-/Ausgabefehler.
1003	Missing end quotes for string	Keine freie Kennung gefunden.
1004	Symbol ist nicht festgelegt.	Ungültige Kennung.
1010	Name, Konstante oder Ausdruck fehlt/fehlerhaft.	Bereichsgröße bei der Komprimierung überschritten.
1011	Konstante ungültig.	Bereich kann keine Kennung enthalten.
1012	Name fehlt/fehlerhaft.	Betriebssystem-Speicherfehler.

1098	Verschachtelte Funktion nicht zulässig.	API Funktion UserError() wurde aufgerufen.
1101	In Unterabfragen kann nur eine Spalte gewählt werden.	Datei "<Datei>" kann nicht geöffnet werden.
1102	Indexname ist bereits vorhanden.	Datei kann nicht erstellt werden.
1103	Mit WHERE sind Formelfunktionen nicht zulässig.	Unzulässige Positionierung in Tabelle.
1104	Spaltenanzahl der Sicht und SELECT müssen übereinstimmen	Dateilesefehler.
1105	Nach CREATE VIEW muß eine Liste der Spaltenname folgen.	Dateischreibfehler.
1106	INTO:Klausel für Sichtdefinition nicht zulässig.	Transaktion in Arbeit.

Appendix C: dBASE error numbers that represent different errors
in FoxPro

1108	Sicht nicht aktualisierbar: <Sichtname>.	Bild ist zu groß, beschädigt oder im falschen Format.
1111	Mit GROUP BY definierte Sicht ist in Abfrage mit GROUP BY nicht zulässig.	Ungültiger Dateideskriptor.
1112	Argumente der dBASE-Funktion mit falschem Datentyp.	Fehler beim Schließen der Datei.
1113	Anzahl der Argumente in der dBASE-Funktion nicht zulässig.	Datei nicht offen.
1115	FROM bezieht sich auf unzulässige Tabelle: <Tabellenname>.	Ungültige CURSOR-Operation.
1117	Sichen können nicht indexiert werden: <Sichtname>.	Falsche Schüssellänge.
1124	Cursor nicht aktiviert :<Cursorname>.	Schlüssel zu groß.
1126	Anderen Namen in der Cursorfestlegung angeben: <Cursorname>.	Satz zu lang.
1127	INTO Klausel ist in der Cursordeklaration nicht zulässig.	FOR/WHILE verlangt logischen Ausdruck.
1130	AVG,MAX,MIN,SUM und COUNT benötigen Spaltenname	'Feld' -Zusatz nicht gefunden.

Appendix C: dBASE error numbers that represent different errors
in FoxPro

	als Argument.	
1134	Vergleichsbegriff/-operator nötig.	Variable muß in der ausgewählten Tabelle sein.
1140	Katalogtabelle(n) wurde(n) durch anderen Benutzer gesperrt: <Tabellenname(n)>	FILTER-Ausdruck zu lang.
1141	Geöffnete Datenbank kann nicht entfernt werden : <Datenbankname>	Ungültiges Indexdateiformat. Bitte neu indizieren
1145	Länge der Zeichenfolge nicht zulässig.	Muß ein Zeichen-oder numerisches Schlüsselwort sein.
1147	SAVE TO TEMP nicht zugelassen.	Ziel wird bereits in einer Beziehung benutzt.
1148	Die Zahlen der Spalten müssen übereinstimmen	Benutzung des Ausdrucks hier nicht sinnvoll.
1149	Nach Dateinamen muß die Liste der Spalten/Feldnamen folgen.	Kein Speicherplatz für Pufferspeicher.
1150	Innerhalb einer Zeichenfolge nur "+" oder "-" zugelassen.	Kein Speicherplatz für internen Speicher.
1151	ALTER bei Sicht nicht möglich: <Sichname>	Kein Speicherplatz für Dateinamen.
1152	INSERT fehlerhaft/falsch	Kein Zugriff auf ausgewählte

Appendix C: dBASE error numbers that represent different errors
in FoxPro

		Tabelle.
1153	Numerischer Wert zu klein	Datei kann nicht auf ein anderes Laufwerk bewegt werden.
1154	Keine aktivierte Zeile vorhanden: <Tabellenname/Sichtname>	Ungültiger Pufferzeiger-Aufruf.
1155	Name bereits als Tab./Datei-/Unterverz.-Name vorhanden.	Ungültiger Zeiger-Aufruf.
1156	Namen mit mehr als 8 Zeichen nicht zugelassen.	Feldnamen nicht eindeutig.
1157	Operator vor CHAR, LOGICAL, DATE-Spalte nicht zulässig.	Datei kann nicht aktualisiert werden.
1160	Zugeordnete Unterabfrage in dieser Form nicht zulässig.	Kein Speicherplatz für "<Dateiname>"
1161	Katalogtabellen sind schreibgeschützt: <Tabellenname>	Zu viele Datensätze für BROWSE/EDIT IN Test- Software.
1162	Tabellenkoordinaten nicht numerisch.	Prozedur "<prozedur>" nicht gefunden.
1163	Formelfunktionen benötigen alle SELECT-Spalten.	BROWSE-Tabelle geschlossen.
1164	Alle SELECT-Spalten müssen in	BROWSE-Struktur verändert.

Appendix C: dBASE error numbers that represent different errors
in FoxPro

	einer Aggregatsfunktion erfaßt.	
1201	Diesen Befehl können Sie nur für andere Benutzer vergeben.	Zu viele Namen benutzt.
1202	Benutzerkennung zweimal vorhanden.	Programm zu groß.
1217	Dateiname und vorhandenes Synonym sind identisch.	Fehler bei Schablonencode in GET-Anweisung.
1220	Interner Fehler # 3 im SQL Dienstprogramm.	Ungültiges Zeichen im Befehl.
1221	Interner Fehler # 4 im SQL Dienstprogramm.	Befehl nicht vollständig.
1223	Tabelle im SQL-Katalog nicht gefunden.	Ungültige Variablenreferenz.
1225	DBCHECK und RUNSTATS müssen mit Datenbestandsdateien verwendet werden.	Muß eine Speichervariable sein.
1226	Datei ist verschlüsselt.	Muß eine Dateivariable sein.
1229	Katalogtabelle SYSDBS fehlt.	Zu wenige Argumente.
1230	Datei in dBASE/SQL nicht zugelassen: <Dateiname>	Zu viele Argumente.

Appendix C: dBASE error numbers that represent different errors
in FoxPro

1231	Datei in aktivierter Datenbank nicht gefunden.	Fehlender Operand.
1245	Interner SQL-Fehler #29	Fehler bei der Etikettenfelddefinition.
1282	Fehler bei Dateiverschlüsselung.	Nicht genügend Speicherplatz.
1249	Datei SYSDBS im SQLHOME-Verzeichnis kann nicht geöffnet werden.	Zu viele aktive READs.

**Anhang D:
Gleiche Fehlermeldungen mit unterschiedlichen Nummern**

dBAS E	FoxPr o	Fehlermeldung
29	1705	Zugriff auf Datei verweigert.
180	1621	Für dieses Menü wurden keine Blöcke definiert.
207	19, 114	.MDX Datei stimmt mit dB-Datei nicht überein.
244	1211	IF/ELSE/ENDIF-Fehler.
245	1211	If/ELSE/ENDIF-Fehler.
247	1213	Falscher Aufbau der CASE-Struktur.
249	1211	IF/ELSE/ENDIF-Fehler.
250	1213	Falscher Aufbau der CASE-Struktur.
304	1649	Kein Druckauftrag (PRINT JOB) für Druckende abgesetzt.

305	1649
341	1214
1275	1282

Kein Druckauftrag (PRINT JOB) für Druckende abgesetzt.

ENDTEXT ohne TEXT.

Nicht genügend Speicherplatz.

Appendix E: Bibliotheken mit Netzwerk- und Sicherheitsfunktionen

NETWARE.PLB

Microsoft Corporation

Wird mit FoxPro MS-DOS 2.5 installiert

Sie finden NETWARE.PLB im dem Hauptverzeichnis, in dem Sie FoxPro 2.5 installiert haben. Die Bibliotheksfunktionen werden im Hilfesystem dokumentiert, wählen Sie dazu das Thema Transaktionen unter **Allgemeine Themen**. Mit Hilfe dieser Bibliothek können Sie das Novell Netware Transactional Tracking System aus FoxPro heraus aufrufen.

TTSAVAIL()	Prüft, ob Novell Netware TTS (Transactional Tracking System) geladen ist.
TTSATTRIB()	Legt fest, daß eine Datei in eine Transaktion eingebunden wird.
BEGINTRAN()	Legt den Beginn einer Transaktion fest.
COMMIT()	Schreibt Änderungen in die an einer Transaktion beteiligten Dateien.
RLLBACK()	Nimmt beim Scheitern einer Transaktion alle Änderungen, die in den an einer Transaktion beteiligten Dateien vorgenommen wurden, zurück.

FPNET

Platinum Software
17 Thorburn Road

North Potomac, Maryland 20878

301-330-5118

Untersützt Versionen: 2.5

Bindery- und Bindery-Sicherheit

N_AddID()	Fügt Benutzerkennung/Paßwort in Bindery ein.
N_ChangeID()	Verändert das Paßwort für eine Anwendung.
N_DelID()	Löscht den Eintrag für Benutzerkennung/Paßwort einer Anwendung
N_FindIDs()	Gibt eine Liste der in der Bindery gespeicherten Benutzerkennungen für Anwendungen aus.
N_FullName()	Ermittelt den vollständigen Netznamen eines Arbeitsplatzrechners
N_Groups()	Gibt aus, zu welchen Gruppen ein Benutzer gehört.
N_IsEquiv()	Überprüft Sicherheitsäquivalente.
N_IsMember()	Überprüft, ob Benutzer zur Gruppe gehört.
N_LastLog()	Zeigt Datum und Uhrzeit des letzten Einloggens an.
N_LoginID()	Gibt den Login-Namen eines Arbeitsplatzrechners aus.

N_Members()	Gibt eine Liste der Mitglieder einer Gruppe aus.
N_ObjList()	Gibt eine Namensliste der Bindery-Objekten aus.
N_Operator()	Gibt eine Namensliste der Consolen aus.
N_PQList()	Gibt eine Namensliste der Druckschlangen aus.

Appendix E: Network and Security Libraries

N_PropVal()	Listet die Eigenschaften eines Bindery-Objekts auf.
N_SecEqs()	Gibt eine Liste aller Sicherheitsäquivalente aus.
N_SList()	Gibt eine Liste der Fileserver-Namen aus.
N_UserList()	Zeigt eine Liste aller aktuell eingeloggten Benutzer mit Login-Zeit und Verbindungsnummer an.
N_VerId()	Überprüft das Anwendungspaßwort eines Benutzers in der Bindery.
N_VerPswd()	Überprüft, ob das Netz-Login-Paßwort eines Benutzers korrekt eingegeben wurde.

Dateiverwaltungssystem

N_AddTrust()	Legt Zugriffsrechte für Verzeichnis-Trustee fest.
N_ClrConn()	Hebt logische Verbindung zu einem Fileserver auf.
N_Date()	Ermittelt das aktuelle Systemdatum vom Fileserver.
N_DelTrust()	Hebt die Trustee-Rechte eines Benutzers für den Zugriff auf ein Verzeichnis auf.
N_DownServ()	Deaktiviert einen Fileserver.

Appendix E: Network and Security Libraries

N_Flag()	Zeigt/verändert die Attribute einer Gruppe von Dateien.
N_GetPath()	Gibt den Suchpfad für die genannte Laufwerkskennung aus.
N_IsMap()	Überprüft, ob Laufwerkskennung einem Netzlaufwerk/-verzeichnis zugeordnet ist.
N_IsOkLog()	Überprüft, ob Login in Fileserver möglich ist.
N_Map()	Ordnet Netzlaufwerk/-verzeichnis einem bestimmten Laufwerksbuchstaben zu.
N_MapRem()	Hebt die Laufwerkszuordnungen für ein Netzlaufwerk/-verzeichnis auf.
N_NoLogin()	Sperrt einen Fileserver für Logins.
N_NWRevDat()	Gibt Revisionsdatum der NetWare-Software aus
N_NWVers()	Gibt Version- und Revisionsnummer der NetWare-Software aus.
N_OkLogin()	Ermöglicht Login in einen Fileserver.

Appendix E: Network and Security Libraries

N_RenDir()	Benennt ein Verzeichnis um.
N_ServerID()	Ermittelt die Verbindungskennung (ID-Nummer) eines Fileservers.
N_ServFrID()	Gibt den Namen des Fileservers für eine bestimmte Verbindung aus.
N_ServName()	Gibt den Namen des Fileservers aus, der einer bestimmten Laufwerkskennung zugeordnet ist.
N_SetDate()	Setzt das aktuelle Systemdatum im Fileserver.
N_SetTime()	Setzt die aktuelle Systemuhrzeit im Fileserver.
N_Time()	Ermittelt die aktuelle Systemuhrzeit vom Fileserver.

Appendix E: Network and Security Libraries

Nachrichtendienste

N_BFConGrp()	Sendet eine EMail-Nachricht von einer Console zu einer Gruppe.
N_Brd2Cons()	Sendet eine EMail-Nachricht zu einer Console.
N_BrdFCons()	Sendet eine EMail-Nachricht von einer Console.
N_Cast()	Setzt den Broadcast-Modus auf einem Arbeitsplatzrechner.
N_GetBMode()	Gibt den Broadcast-Modus eines Arbeitsrechners aus.
N_GetBMsg()	Gibt eine EMail-Nachricht aus, die in einem Fileserver gespeichert ist.
N_LogMsg()	Zeichnet eine Nachricht in der Datei NET\$LOG.MSG auf.
N_SendMsg()	Sendet eine Nachricht an einen Netzwerkbenutzer.
N_SMsgGrp()	Sendet eine Nachricht an eine Gruppe.

Drucken

N_Capture()	Aktiviert/deaktiviert LPT-Umleitung.
-------------	--------------------------------------

Appendix E: Network and Security Libraries

N_GetPStat()	Gibt alle verfügbaren Daten über die Umleitung von Druckdaten aus.
N_PJobList()	Gibt alle verfügbaren Informationen zu Druckaufträgen, die sich in der Druckerwarteschlange befinden, aus.
N_PJobMove()	Verändert die Position eines Druckauftrags in der Warteschlange.
N_PJobRem()	Löscht einen Druckauftrag aus der Warteschlange.
N_SetPStat()	Legt Umleitungsparameter für einen LPT-Anschluß fest.

Semaphor- und TTS-Funktionen

N_FlagTran()	Zeigt an, ob eine Datei in eine Transaktion eingebunden ist.
N_Semaphor()	Sperrt eine Semaphor-Zeichenfolge bzw. hebt die Sperre auf.
N_SemClose()	Schließt ein Semaphor.
N_SemOpen()	Öffnet ein Semaphor.
N_SemSig()	Signalisiert ein Semaphor (Sperre einer Funktion aufheben).

Appendix E: Network and Security Libraries

N_SemVal()	Ermittelt den Wert eines Semaphor.
N_SemWait()	Wartet auf ein Semaphor (eine Funktion sperren).
N_TTSBeg()	Beginnt eine explizite Transaktion.
N_TTSEnd()	Beendet eine explizite oder implizite Transaktion.
N_TTSAvbl()	Überprüft, ob TTS verfügbar ist.
N_TTSOff()	Deaktiviert TTS auf Standard-Server.
N_TTSON()	Aktiviert TTS auf Standard-Server.
N_TTSZap()	Bricht eine impliziet oder explizite Transaktion ab.

Appendix E: Network and Security Libraries

Arbeitsplatzrechner und Verbindungsdienste

N_Attach()	Stellt Anbindung an Fileserver her.
N_Detach()	Hebt die Anbindung an Fileserver auf.
N_DfServer()	Legt fest, welcher Fileserver Standard-Server ist.
N_GetMATyp()	Gibt die Typbezeichnung des Arbeitsplatzrechners aus.
N_GetOSTyp()	Ermittelt den Namen des auf dem Arbeitsplatzrechner installierten Betriebssystems.
N_GetOSVer()	Ermittelt die Versionsnummer des auf dem Arbeitsplatzrechner installierten Betriebssystems.
N_GetSMTyp()	Gibt die Kurzbezeichnung für den Rechnertyp des Arbeitsplatzrechners aus.
N_IsAttach()	Ermittelt, ob Arbeitsplatzrechner an einen Fileserver angebunden ist.
N_IsLogged()	Ermittelt, ob Benutzer gerade eingeloggt ist.
N_IsNWBoss()	Ermittelt die Consolen-Rechte eines Arbeitsplatzrechners.
N_LocDrv()	Gibt die Anzahl lokaler Laufwerke aus.

Appendix E: Network and Security Libraries

N_LogDate()	Gibt das Login-Datum eines Arbeitsplatzrechners aus.
N_LogTime()	Gibt die Login-Uhrzeit eines Arbeitsplatzrechners aus.
N_LoginFS()	Führt Login zu einem Fileserver aus.
N_Logout()	Führt Logout von allen Fileservern aus.
N_LogoutFS()	Führt Logout von einem Fileserver aus.
N_StaAdr()	Ermittelt die physikalische Knotenadresse eines Arbeitsplatzrechners.
N_StaID()	Ermittelt die Kennnummer (ID) eines Arbeitsplatzrechners.

Informationen zur Hardware des Arbeitsplatzrechners

N_DosVer()	Gibt DOS-Versionsnummer aus.
N_DskFree()	Ermittelt, wieviel freier Speicherplatz auf einem Laufwerk verfügbar ist.
N_DskSpace()	Ermittelt die Gesamtspeicherkapazität eines Laufwerks.

Appendix E: Network and Security Libraries

N_IsDisk()	Überprüft, ob es sich um gültiges Laufwerk oder gültige Diskette handelt.
N_IsMouse()	Überprüft, ob Maus in Betrieb ist.

Appendix E: Network and Security Libraries

GPLIB

George Sexton [73237,1665]

Public domain (kostenlos). Erhältlich in CompuServe, FoxForum. Die Datei heißt GPLIB.ZIP

Unterstützt derzeit nur FoxPro 2.0

AboutGPLIB()	Gibt Versionsnummer u.a. Statusinformationen von GPLIB aus.
AMPM()	Gibt Uhrzeit als Zeichenfolge, an die AM bzw. PM angehängt wird, aus.
ColdBoot()	Führt Kaltstart eines Rechners aus.
CtlAltShft()	Liefert True, falls STRG, ALT oder UMSCHALT gedrückt wird.
ElapseTime()	Berechnet die Zeitdifferenz (in Minuten) aus zwei Uhrzeit-Zeichenfolgen.
FileCount()	Gibt an, wieviele Dateien in einem Verzeichnis einer Suchmaske entsprechen.
FindFirst()	Führt die DOS-Befehle find first/find next aus.
FKillAll()	Löscht alle Dateien, die Suchmaske (mit Platzhaltern) entsprechen.
Flag()	Setzt DOS-Dateiattribute.

Appendix E: Network and Security Libraries

GenError()	Generiert einen FoxPro-Fehler.
IsDiskIn()	Ermittelt, ob es sich eine Diskette im genannten Laufwerk befindet.
Make_Dir()	Erstellt ein Verzeichnis.
Math_Chip()	Liefert True, falls ein 80x87-Coprozessor installiert ist.
MReset()	Setzt Maustreiber zurück und ermittelt Anzahl der Maustasten.
Num_Serial()	Ermittelt die Anzahl serieller Schnittstellen.
N_AcctList()	Gibt eine Liste der in einem Server registrierten Benutzer aus.
N_Attach()	Stellt Anbindung an Fileserver her.
N_Brd2Cons()	Sendet Broadcast-Nachricht an Server-Console.
N_BrdFCons()	Sendet Broadcast-Nachricht von Server-Console.
N_Capture()	Aktiviert/deaktiviert Protokollfunktion.

Appendix E: Network and Security Libraries

N_Cast()	Ermittelt, ob Arbeitsplatzrechner EMail-Nachrichten akzeptiert.
N_Date()	Liefert aktuelles Systemdatum vom Fileserver.
N_Detach()	Hebt Anbindung an Fileserver auf.
N_DfServer()	Setzt andern Server als Standard-Server ein.
N_FlushLPT()	Leert Puffer der genannten LPT-Schnittstelle.
N_FullName()	Gibt vollständigen Benutzernamen aus.
N_GetBMode()	Gibt aktuellen Broadcast-Modus eines Arbeitsplatzrechners aus.
N_GetBMsg()	Lädt eine in einem Fileserver gespeicherte Nachricht.
N_GetGroup()	Gibt eine Liste der in einem Fileserver registrierten Gruppen aus.
N_GetMaTyp()	Gibt vollständige Rechnertypbezeichnung aus.
N_GetOSTyp()	Ermittelt Typ des Betriebssystems.

Appendix E: Network and Security Libraries

N_GetOSVer()	Ermittelt Versionsnummer des Betriebssystems.
N_GetPath()	Gibt Suchpfad für ein bestimmtes Laufwerk aus.
N_GetPStat()	Gibt aktuelle Einstellungen der Protokollfunktion aus.
N_GetQList()	Gibt Liste der Warteschlangen eines Servers aus.
N_GetSList()	Gibt Liste der in einem Fileserver registrierten Server aus.
N_GetSMTyp()	Gibt Kurzform der Rechnertypbezeichnung aus.
N_GetWild()	Gibt eine Liste aller Bindery-Objekte eines Servers aus.
N_IsAttach()	Ermittelt, ob Verbindung zu Server besteht.
N_IsEquiv()	Gibt Sicherheits-Äquivalente aus.
N_IsLogged()	Ermittelt, ob genannter Benutzer in Netzwerk eingeloggt ist.

Appendix E: Network and Security Libraries

N_IsMap()	Ermittelt, ob Laufwerk einem NetWare-Fileserver zugeordnet ist.
N_IsMember()	Ermittelt, ob Benutzer Mitglied der genannten Gruppe ist.
N_IsNWBoss()	Ermittelt, ob Arbeitsplatzrechner als Console fungiert.
N_LastLog()	Gibt Datum und Uhrzeit des letzten Logins eines Benutzers aus.
N_LocDrv()	Gibt Anzahl logischer lokaler Laufwerke aus.
N_LogDate()	Gibt Login-Datum aus.
N_LoginFS()	Führt Login in einen Fileserver aus.
N_LoginID()	Gibt die NetWare-Benutzerkennung (ID) aus.
N_Logout()	Führt Logout von allen Fileservern aus.
N_LogoutFS()	Führt Logout vom genannten Fileserver aus.
N_LogTime()	Ermittelt Uhrzeit des Logins.

Appendix E: Network and Security Libraries

N_Map()	Ordnet ein Laufwerk einem Netzwerklaufwerk/-verzeichnis zu.
N_MapRem()	Hebt die Zuordnungen eines Laufwerks auf.
N_MaxConns()	Liefert die maximale Anzahl von NetWare-Verbindungen.
N_NoLogin()	Sperrt einen Fileserver für Logins.
N_NukeSta()	Hebt die Verbindung mit der genannten Nummer auf.
N_OkLogin()	Ermöglicht Login in einen Fileserver.
N_PJobList()	Legt ein Array an, das die Druckaufträge der genannten Warteschlange speichert.
N_PJobMove()	Verändert Position eines Druckauftrags in einer Warteschlange.
N_PJobRem()	Löscht einen Druckauftrag aus einer Warteschlange.
N_PwdDate()	Gibt das Datum aus, an dem das Benutzerpaßwort verfällt.
N_Semaphor()	Perform Netware semaphore locking

Appendix E: Network and Security Libraries

N_SendMsg()	Sendet eine Nachricht an einen anderen Netzwerkbenutzer.
N_ServerID()	Gibt die Kennnummer (ID) des genannten Servers aus.
N_ServFrID()	Ermittelt Servername anhand der Verbindungskennung (ID).
N_ServName()	Return Fileserver name for a mapped drive
N_SetPStat()	Verändert die aktuellen Einstellungen der Protokollfunktion.
N_ShellVer()	Gibt Versionsnummer der Shell aus.
N_StaAddr()	Gibt physikalische Adresse des Arbeitsplatzrechners aus.
N_StaID()	Ermittelt die Nummer des NetWare-Arbeitsplatzrechners.
N_SYSTime()	Synchronisiert die Systemuhren von Arbeitsplatzrechner und Fileserver.
N_Time()	Ermittelt die aktuelle Systemuhrzeit vom Fileserver.

Appendix E: Network and Security Libraries

N_VerPwd()	Überprüft die Eingabe eines Benutzerpaßworts.
Rem_Dir()	Löscht ein Verzeichnis.
SetMLimit()	Definiert Arbeitsbereich der Maus.
SetMPos()	Positioniert den Mauszeiger.
ShftPrtScr()	Löst Hardware-Interrupt 5 (Umschalt Druck) aus.
TrimLen()	Liefert die Länge einer gekürzten Zeichenfolge.
UniqueName()	Liefert einen eindeutigen Dateinamen.

NetLib

Pinnacle Publishing

PO Box 8099

Federal Way, Washington 98003,0099

800-788-1900

206-251-1900

Unterstützte Versionen: FoxPro 2.0

Eine Version für FoxPro 2.5 für MS-DOS soll im 2. Quartal 1993, die Version für FoxPro für Windows im 4. Quartal 1993 erscheinen.

Allgemeine Netzwerkfunktionen

Appendix E: Network and Security Libraries

N_ADDR()	Ermittelt physikalische Netzwerk-Adresse.
N_APPEND()	Hängt Daten an .TXT oder >SDF-Datei an.
N_AREDIRECT()* **	Speichert umgeleitete Geräteeinheiten und npaths in einem Array.
N_SERVER()*	Gibt Liste aller angebundenen Server aus.
N_ATTACH()*	Stellt Anbindung an genannten Server her.
N_CHECKF()	Ermittelt, bei welchem Arbeitsplatzrechner RLOCK/FLOCK für eine Datei in Kraft ist.
N_CHECKR()	Ermittelt, bei welchem Arbeitsplatzrechner Datensatz gesperrt ist.
N_CHECKS()	Ermittelt, bei welchen Arbeitsplatzrechnern Semaphor gesperrt ist.
N_CHECKU()	Ermittelt, welcher Arbeitsplatzrechner .DBF-Datei bearbeitet.
N_CONNECT()*	Initialisiert Verbindung zu Arbeitsplatzrechner(n).
N_DATE()*	Ermittelt das aktuelle Systemdatum vom Server.

Appendix E: Network and Security Libraries

N_DETACH()*	Hebt Verbindung zu genanntem Server auf.
N_DISCON()*	Hebt Verbindung zu Arbeitsplatzrechner(n) auf.
N_ENVLEN()	Gibt die Länge der in der Umgebungsvariablen gespeicherten Zeichenfolge aus.
N_ENVSIZ()	Ermittelt verfügbare Größe für Umgebungsvariable.
N_ERROR()	Liefert den Fehlercode der zuletzt ausgeführten NetLib-Funktion.
N_FATTR()	Ermittelt oder setzt Dateiattribute.
N_FCOPY()	Kopiert eine Datei.
N_FDRIVE()	Gibt die in einer Dateibezeichnung enthaltenen Laufwerkskennung aus.
N_FEXT()	Gibt die in einer Dateibezeichnung enthaltene Dateinamenserweiterung aus.
N_FMAP()	Ermittelt Dateinamen mit Suchpfad für angegebenen Handle.
N_FNAME()	Gibt den in einer Dateibezeichnung enthaltenen

Appendix E: Network and Security Libraries

	Dateinamen aus.
N_FPATH()	Gibt den in einer Dateibezeichnung enthaltenen Suchpfad aus.
N_FSPEC()	Ermittelt vollständigen Suchpfad mit Laufwerkskennung für genannte Datei.
N_FULLNAME()*	Ermittelt vollständigen Benutzernamen.
N_FVOL()	Ermittelt Namen des Netzwerk-Volumes.
N_ISEXCL()	.T. = aktuelle Datei ist für exklusive Nutzung geöffnet.
N_ISFLOCK()	.T. = aktuelle Datei ist gesperrt.
N_ISRLOCK()	.T. = genannter Datensatz ist gesperrt.
N_ISSLOCK()	.T. = Semaphore ist gesperrt.
N_JOURNAL()	Öffnet Datei für Journal-Funktion.
N_LOGIN()*	Verbindet Objekt mit Standard-Server und führt Login aus.

Appendix E: Network and Security Libraries

N_LOGMSG()*	Schreibt Kommentar in System-Log
N_LOGOUT()*	Logout aus genanntem Server.
N_MAPDRIVE()*	Erstellt, ermittelt oder löscht eine Laufwerkszuordnung.
N_MLOCK()	Sperrt mehrere Datensätze in einer Datei.
N_MODENV()	Setzt neue Umgebungsvariable.
N_NDX()	Ermittelt Nummer und Name der Indexdatei(en)
N_NETNAME()**	Gibt Eintrag der NetBIOS-Namenstabelle aus.
N_PRTSC()	Gibt Bildschirminhalt mit Hilfe von INT5 aus.
N_READONLY()	Öffnet eine Datei nur für Leseoperationen.
N_READY()	.T. = Netzwerk ist in Betrieb.
N_RECV()*	Empfängt Nachricht.

Appendix E: Network and Security Libraries

N_REDIRECT()***	Setzt, ermittelt oder löscht die Parameter für die Umleitung einer Einheit.
N_RESTATTR()	Stellt Attribute der Bildschirmanzeige wieder her.
N_RIGHTS()*	Ermittelt Zugriffsrechte auf ein Verzeichnis.
N_SAVEATTR()	Speichert Attribute der Bildschirmanzeige.
N_SECONDS()	Ermittelt vom Server, wieviele Sekunden seit 12 a.m. vergangen sind.
N_SEND()*	Sendet Nachricht an einen oder mehrere Arbeitsplatzrechner.
N_SERIAL()*	Gibt Seriennummer des Servers aus.
N_SERVER()*	Ermittelt oder setzt Server.
N_SERVNUM()*	Ermittelt Verbindungsnummer des Servers.
N_SETLOG()*	Ermöglicht oder unterbindet Login in einen bestimmten Server.
N_SETTIME()	Beendet Wartestatus, ruft Ereignis-Prozedur auf.

Appendix E: Network and Security Libraries

N_SLOCK()	Sperrt Semaphor.
N_SOFTSCR()	Gibt Bildschirminhalt über Software-Emulation aus.
N_STAMAX()	Gibt Nummer des höchsten Arbeitsplatzrechners aus.
N_STANUM()	Gibt Nummer des aktuellen Arbeitsplatzrechners aus.
N_SUNLOCK()	Hebt Semaphor-Sperre auf.
N_TIME()*	Ermittelt aktuelle Systemuhrzeit vom Server.
N_USE()	Öffnet Datenbankdatei.
N_VERSION()*	Gibt die Versionsnummer u.a. Statusinformationen des Servers aus.
N_WHERE()	Ermittelt, an welchem Arbeitsplatzrechner Benutzer eingeloggt ist.
N_WHOAMI()	Ermittelt Benutzererkennung (ID).

Appendix E: Network and Security Libraries

Druckfunktionen

N_BANNER()*	Legt fest, ob Seite mit Druckererkennung bei jedem Druckauftrag gedruckt wird.
N_CLASS()*	Legt Druckklasse des Dokuments fest.
N_COPIES()*	Legt fest, wieviele Kopien gedruckt werden.
N_PRINTER()	Ermittelt die Nummer des angesteuerten Druckers.
N_SPLFRM()*	Legt den Namen des Druckformats fest.
N_SPLLPT()*	Legt LPT-Nummer für Empfang von Spoolerdaten fest.
N_SPLQUEUE()*	Legt Warteschlange für Spooler fest.
N_SPLSRV()*	Legt Druck-Server fest.
N_SPLTABS()*	Legt Tabweite fest.
N_SPLTMO()*	Legt Drucker-Timeout fest.

Appendix E: Network and Security Libraries

N_SPOOL()	Startet oder stoppt Spooler.
-----------	------------------------------

Verschlüsselungsfunktionen

N_CODELVL()	Ermittelt Verschlüsselungsstatus.
N_DECODE()	Entschlüsselt Datei.
N_DECODEST()	Entschlüsselt Zeichenfolge
N_ENCODE()	Verschlüsselt Datei.
N_ENCODEST()	Verschlüsselt Zeichenfolge.
N_SETKEY()	Legt Code für Verschlüsselung fest.

Netware-Bindery-Funktionen

N_B_CREATE()*	Erstellt ein Bindery-Objekt.
N_B_DEL()*	Löscht ein Binder-Objekt.
N_B_ID()*	Gibt Binder ID des genannten Objekts aus.

Appendix E: Network and Security Libraries

N_B_ISMEMBER() *	.T. = Objekt ist Element der Objektmenge.
N_B_LINK()*	Bindet Objekt in Menge ein.
N_B_MEMBERS() *	Speichert die Namen der Mengenelemente im genannten Array.
N_B_NAME()*	Gibt den Namen des genannten Bindery-Objekts aus.
N_B_PASSWORD()*	.T. = Paßwort ist gültig.
N_B_PRCREATE() *	Legt Bindery-Eigenschaft fest.
N_B_PRSCAN()*	Speichert die Namen aller Eigenschaften, die mit dem genannten Array verbunden sind.
N_B_PRREAD()*	Liefert den Wert der genannten Eigenschaft.
N_B_PRTYPE()*	Liefert den Typ der genannten Eigenschaft.
N_B_PRWRITE()*	Modifiziert die Eigenschaft eines Bindery-Eintrags.

Appendix E: Network and Security Libraries

N_B_SCAN()*	Durchsucht Bindery nach übereinstimmenden Objekten und schreibt deren Namen in ein Array.
N_B_TYPE()*	Liefert Typ des genannten Bindery-Objekts.
N_B_UNLINK()*	Hebt Mengenzugehörigkeit eines Objekts auf.

*nur für Novell Netware

**nur für NetBIOS

***nur für NetBIOS und Banyan VINES

Appendix E: Network and Security Libraries

Inkompatible Befehle und Funktionen in Clipper 5.x

Anhang F:
Inkompatible Befehle und Funktionen in Clipper 5.x

**Präprozessor-
direktiven**

#command
#define
#error
#ifdef
#ifndef
#include
#stdout
#undef
#xcommand

Operatoren

:=
++
--

Variablentypen

STATIC
LOCAL

Datentypen

Codeblocks
NIL

Klassen

Error
Get
TBrowse
TBColumn

Befehle

DELETE TAG

GO
INDEX
SEEK
SET INDEX
SET ORDER

Funktionen

AEVAL()
ALERT()
ARRAY()
ATAIL()
BREAK()
BROWSE()
DBAPPEND()
DBAPPEND()
DBCLEARFIL()
DBCLEARIND()
DBCLEARREL()
DBCLOSEALL()
DBCLOSEAREA()
DBCOMMIT()
DBCOMMITALL()
DBCREATE()
DBCREATEIND()
DBDELETE()
DBEVAL()
DBFILTER()
DBGOBOTTOM()
DBGOTO()
DBGOTO()
DBGOTOP()
DBRECALL()
DBREINDEX()

DBRELATION()
DBRLOCK()
DBRLOCKLIST()
DBRSELECT()
DBRUNLOCK()
DBSEEK()
DBSELECTAR()
DBSETDRIVER()
DBSETFILTER()
DBSETINDEX()
DBSETINDEX()
DBSETORDER()
DBSETRELAT()
DBSKIP()
DBSTRUCT()
DBUNLOCK()
DBUNLOCKALL()
DBUSEAREA()
DEVOUT()
DEVOUTPICT()
DEVPOS()
DISPBEGIN()
DISPBOX()
DISPCOUNT()
DISPEND()
DISPOUT()
ERRORBLOCK()
EVAL()
EXP()
FIELDBLOCK()
FIELDGET()
FIELDNAME()
FIELDPOS()

Programm-Konverter

FIELDPUT()
FIELDWBLOCK()
HEADER()
MEMVARBLOCK()
NOSNOW()
ORDBAGEXT()
ORDBAGNAME()
ORDCREATE()
ORDDESTROY()
ORDFOR()
ORDKEY()
ORDLISTADD()
ORDLISTCLEAR()
ORDLISTREBUI()
ORDNAME()
ORDNUMBER()
ORDSETFOCUS()
OS()
OUTERR()
OUTSTD()
QOUT()
RDDLIST()
RDDNAME()

Funktionen

RDDSETDEFAULT
()
READMODAL()
RECNO()
SETBLINK()
SETCANCEL()
SETCOLOR()
SETCURSOR()

SETKEY()
SETMODE()
SETPOS()
SETPRC()
VALTYPE()
VERSION()

GET System

GETACTIVE()
GETAPPLYKEY()
GETDOSETKEY()
GETPOSTVALIDA
TE()
GETPREVALIDAT
E()
GETREADER()
READFORMAT()
READKILL()
READUPDATED()

Anhang G: Tastenbelegung

Tastenbezeichnungen für ON KEY LABEL KEY in FoxPro

Tastatur	Tastenbezeichnung
Nach links	LEFTARROW
Nach rechts	RIGHTARROW
Nach oben	UPARROW
Nach unten	DNARROW
Pos1	HOME
Ende	END
BildAuf	PGUP
BildAb	PGDN
Entf	DEL

Rücktaste	BACKSPACE
Leertaste	SPACEBAR
Einfüg	INS
Tab	TAB
Umschalt Tab	BACKTAB
Eingabetaste	ENTER
{	LBRACE
}	RBRACE
F1 bis F12	F1, F2 ...
Strg+F1 bis Strg+F12	Ctrl+F1, Ctrl+F2 ...
Umschalt+F1 bis	Shift+F1, Shift+F9...

Umschalt+F9	
Umschalt+F11, Umschalt+F12	Shift+F11 ...
Alt+F1 bis Alt+F12	Alt+F1, Alt+F2 ...
Alt+0 bis Alt-9	Alt+0, Alt+1 ...
Alt+A bis Alt+Z	Alt+A, Alt+B ...
Alt+BildAuf	Alt+PGUP
Alt+BildAb	Alt+PGDN
Strg+Nach links	Ctrl+LEFTARROW
Strg+Nach rechts	Ctrl+RIGHTARROW
Strg+Pos1	Ctrl+HOME
Strg+Ende	Ctrl+END

Strg+BildAuf	Ctrl+PGUP
Strg+BildAb	Ctrl+PGDN
Strg+A bis Strg+Z	Ctrl+A, Ctrl+B ...
Rechte Maustaste	RIGHTMOUSE
Linke Maustaste	LEFTMOUSE
Maus	MOUSE
Esc	ESC

Appendix G: Key assignments

INKEY() Codes in FoxPro, dBASE und Clipper Summer '87

Funktions- und Cursorsteuertasten

	FoxPro	dBASE E	Clipper S'87		FoxPro	dBASE E	Clipper S'87
Strg-End	373	23	23	Rücktaste	127	127	8
Strg-Pos1	375	26	29	Strg-Rückt	NA	-401	127
Strg-Nach links	371	1	26	Entf	339	NA	7
Strg-BildAb	374	30	30	Nach unten	336	24	24
Strg-BildAuf	388	31	31	Ende	335	2	6
Strg-Eingabe	10	-402	10	Pos1	327	26	1
Strg-Nach rechts	372	6	2	Einfg	338	NA	22
Eingabe/Enter	13	13	13	Nach links	331	19	19
Esc	27	27	27	BildAb	337	3	3

Appendix G: Key assignments

F1	315	28	28	BildAuf	329	18	18
F2	316	-1	-1	Nach oben	333	4	4
F3	317	-2	-2	Umschalt- Tab	15	-400	271
F4	318	-3	-3	Tab	9	9	9
F5	319	-4	-4	Nach oben	328	5	5
F6	320	-5	-5				
F7	321	-6	-6				
F8	322	-7	-7				
F9	323	-8	-8				
F10	324	-9	-9				

Appendix G: Key assignments

Alt- und Strg-Tastenkombinationen

	FoxPr o	dBAS E	Clipper S'87		FoxPr o	dBAS E	Clipper S'87
Alt-A	286	-435	30	Strg-A	1	1	1
Alt-B	304	-434	48	Strg-B	2	2	2
Alt-C	302	-433	46	Strg-C	3	3	3
Alt-D	288	-432	32	Strg-D	4	4	4
Alt-E	274	-431	18	Strg-E	5	5	5
Alt-F	289	-430	33	Strg-F	6	6	6
Alt-G	290	-429	34	Strg-G	7	7	7
Alt-H	291	-428	35	Strg-H	8	8	8
Alt-I	279	-427	23	Strg-I	9	9	9
Alt-J	292	-426	36	Strg-J	10	10	10

Appendix G: Key assignments

Alt-K	293	-425	37	Strg-K	11	11	11
Alt-L	294	-424	38	Strg-L	12	12	12
Alt-M	306	-423	50	Strg-M	13	13	13
Alt-N	305	-422	49	Strg-N	14	14	14
Alt-O	280	-421	24	Strg-O	15	15	15
Alt-P	281	-420	25	Strg-P	16	16	16
Alt-Q	272	-419	16	Strg-Q	17	17	17
Alt-R	275	-418	19	Strg-R	18	18	18
Alt-S	287	-417	31	Strg-S	19	19	19
Alt-T	276	-416	20	Strg-T	20	20	20
Alt-U	278	-415	22	Strg-U	21	21	21

Appendix G: Key assignments

Alt-V	303	-414	47	Strg-V	22	22	22
Alt-W	273	-413	17	Strg-W	23	23	23
Alt-X	301	-412	45	Strg-X	24	24	24
Alt-Y	277	-411	21	Strg-Y	25	25	25
Alt-Z	300	-410	44	Strg-Z	26	26	26

Appendix G: Key assignments

Funktionsastenkombinationen mit Umschalt, Alt und Strg

	FoxPr o	dBAS E	Clipper S'87
Alt-F1	104	NA	-30
Alt-F2	105	NA	-31
Alt-F3	106	NA	-32
Alt-F4	107	NA	-33
Alt-F5	108	NA	-34
Alt-F6	109	NA	-35
Alt-F7	110	NA	-36
Alt-F8	111	NA	-37
Alt-F9	112	NA	-38
Alt-F10	113	NA	-39

Appendix G: Key assignments

Strg-F1	94	-10	-20
Strg-F2	95	-11	-21
Strg-F3	96	-12	-22
Strg-F4	97	-13	-23
Strg-F5	98	-14	-24
Strg-F6	99	-15	-25
Strg-F7	100	-16	-26
Strg-F8	101	-17	-27
Strg-F9	102	-18	-28
Strg-F10	103	-19	-29

Appendix G: Key assignments

Umschalt-F1	84	-20	-10
Umschalt-F2	85	-21	-11
Umschalt-F3	86	-22	-12
Umschalt-F4	87	-23	-13
Umschalt-F5	88	-24	-14
Umschalt-F6	89	-25	-15
Umschalt-F7	90	-26	-16
Umschalt-F8	91	-27	-17
Umschalt-F9	92	-28	-18
Umschalt-F10	93	-29	-19

Appendix G: Key assignments

Anhang I: Reservierte Wörter in FoxPro

Falls in einem dBASE IV-Programm ein reserviertes Wort aus FoxPro als Arrayname verwendet wird, gibt FoxPro die Fehlermeldung "Attempt to use FoxPro function as an array." If you use a reserved word as a UDF name, FoxPro will call the internal command or function instead, with varying error messages.

#ENDIF	_PDSETUP	AINS	BELL	CMONTH
#IF	_PECODE	ALEN	BETWEEN	CNT
#ITSEXPRESSION	_PEJECT	ALIAS	BG	CNTBAR
#READCLAUSES	_PEPAGE	ALL	BLANK	CNTPAD
#REGION	_PFORM	ALLTRIM	BLINK	COL
#SECTION	_PLENGTH	ALTERNATE	BLOCKSIZE	COLOR
#WNAME	_PLINENO	AMERICAN	BOF	COLOUR
.AND.	_POFFSET	AMPM()	BORDER	COLUMN
.F.	_PPITCH	AND	BOTTOM	COMMAND
.NOT.	_PQUALITY	ANSI	BOX	COMMIT
.OR.	_PRETEXT	ANSITOOEM	BR	COMPACT
.T.	_PSCODE	APP	BREAK	COMPATIBILITY
@FUNCTION	_PSCODE	APPEND	BRITISH	COMPATIBLE
@PROCEDURE	_PSPACING	APPLICATION	BROWSE	COMPILE
_ALIGNMENT	_PWAIT	ARRAY	BRSTATUS	COMPLETED
_BOX	_RMARGIN	AS	BUILD	COMPRESS
_CALCMEM	_TABS	ASC	BY	CONFIRM
_CALCVALUE	_TALLY	ASCAN	CALCULATE	CONSOLE
_CLIPTEXT	_TEXT	ASCENDING	CALL	CONTINUE
_CUROBJ	_THROTTLE	ASIN	CANCEL	CONVERT
_DBLCLICK	_TRANSPORT	ASORT	CAPSLOCK	COPY
_DIARYDATE	_WINDOWS	ASSIST	CARRY	COS
_DOS	_WRAP	ASUBSCRIPT	CASE	COUNT
_GENGRAPH	ABS	AT	CATALOG	CREATE
_GENMENU	ACCEPT	ATAN	CDOW	CTOD
_GENPD	ACCESS	ATC	CDX	CURDIR
_GENSCRN	ACOPY	ATCLINE	CEILING	CURRENCY
_GENXTAB	ACOS	ATLINE	CENTER	CURSOR
_INDENT	ACTIVATE	ATN	CENTURY	CYCLE
_LMARGIN	ADD	ATN2	CGA	DATABASE
_MAC	ADDITIVE	AUTO	CHAIN	DATABASES
_MLINE	ADEL	AUTOMATIC	CHANGE	DATE
_PADVANCE	ADIR	AUTOSAVE	CHR	DAY
_PAGENO	AELEMENT	AVERAGE	CHRSAW	DB
_PBPAGE	AFIELDS	AVG	CHRTRAN	DB4
_PCOLNO	AFONT	BAR	CLEAR	DBASEII
_PCOPIES	AFTER	BEFORE	CLOCK	DBDATE
_PDRIVER	AGAIN	BEGIN	CLOSE	DBF

DBMEMO3	EJECT	FLUSH	HISTORY	LOGERRORS
DDEAbortTrans	ELSE	FONTMETRIC	HOURS	LOGOUT
DDEAdvise	ELSEIF	FOOTER	IF	LOOKUP
DDEEnabled	EMPTY	FOPEN	IIF	LOOP
DDEExecute	ENABLE	FOR	IMPORT	LOWER
DDEInitiate	ENCRYPT	FORM	IN	LPARTITION
DDELastError	ENCRYPTION	FORMAT	INDEX	LTRIM
DDEPoke	END	FOUND	INDEXES	LUPDATE
DDERequest	ENDCASE	FOXPLUS	INFORMATION	MACKEY
DDESetOption	ENDDO	FOXPRO	INKEY	MACRO
DDESetService	ENDFOR	FOXQ	INLIST	MACROS
DDESetTopic	ENDIF	FOXSWAP	INPUT	MARGIN
DDETerminate	ENDPRINTJOB	FPUTS	INSERT	MARK
DEACTIVATE	ENDSCAN	FREAD	INSMODE	MASTER
DEBUG	ENDTEXT	FREEZE	INSTRUCT	MAX
DECIMALS	ENVIRONMENT	FRENCH	INT	MCOL
DECLARE	EOF	FROM	INTENSITY	MDOWN
DEFAULT	ERASE	FSEEK	INTO	MDX
DEFINE	ERROR	FSIZE	ISALPHA	MDY
DELETE	ESCAPE	FULLPATH	ISCOLOR	MEMLINES
DELETED	EVALUATE	FV	ISDIGIT	MEMO
DELIMITED	EXACT	FW	ISLOWER	MEMORY
DELIMITER	EXCEPT	FW2	ISMARKED	MEMOWIDTH
DELIMITERS	EXCLUSIVE	FWRITE	ISUPPER	MEMVAR
DESC	EXIT	GATHER	ITALIAN	MENU
DESCENDING	EXP	GB	JAPAN	MENUS
DESIGN	EXPORT	GENERAL	JOIN	MESSAGE
DEVELOPMENT	EXTENDED	GERMAN	KEY	MESSAGES
DEVICE	EXTERNAL	GET	KEYBOARD	MIN
DIF	FCHSIZE	GETBAR	KEYCOMP	MINIMIZE
DIFFERENCE	FCLOSE	GETDIR	LABEL	MLINE
DIMENSION	FCOUNT	GETENV	LAST	MOD
DIR	FCREATE	GETEXPR	LASTKEY	MODAL
DIRECTORY	FEOF	GETFILE	LEDIT	MODIFY
DISABLE	FERROR	GETFONT	LEFT	MODULE
DISKSPACE	FFLUSH	GETPAD	LEN	MONO43
DISPLAY	FGETS	GETS	LEVEL	MONTH
DISTINCT	FIELD	GO	LIB	MOUSE
DMY	FIELDS	GOMONTH	LIBRARY	MOVE
DO	FILE	GOTO	LIKE	MOVER
DOHISTORY	FILER	GR	LINE	MRKBAR
DOS	FILES	GROUP	LINENO	MRKPAD
DOUBLE	FILL	GROW	LINKAGE	MROW
DOW	FILTER	HAVING	LIST	MULTI
DTOC	FIND	HEADER	LKSYS	MULTILOCKS
DTOR	FIXED	HEADING	LOAD	NDX
DTOS	FKLABEL	HEIGHT	LOCATE	NEAR
ECHO	FKMAX	HELP	LOCFILE	NETWORK
EDIT	FLOAT	HELPFILTER	LOCK	NEXT
EGA25	FLOCK	HIDE	LOG	NOALIAS
EGA43	FLOOR	HIGHLIGHT	LOG10	NOAPPEND

NOCLEAR	OPTIMIZE	PUTFILE	RUN	STUFF
NOCLOSE	OR	PV	SAFETY	STYLE
NOCONSOLE	ORDER	QUERY	SAME	SUBSTR
NODEBUG	OS	QUIT	SAMPLE	SUM
NODELETE	OTHERWISE	RAND	SAVE	SUMMARY
NOEDIT	OVERWRITE	RANDOM	SAY	SUSPEND
NOEJECT	PACK	RANGE	SCAN	SYLK
NOENVIRONMENT	PAD	RANK	SCATTER	SYS
NOFLOAT	PADC	RAT	SCHEME	SYSTEMU
NOFOLLOW	PADL	RATLINE	SCOLS	SYSTEMTIRC
NOGROW	PADR	RB	SCOREBOARD	SYSTEM
NOINIT	PAGE	RDLEVEL	SCREEN	TAB
NOLGRID	PALETTE	READ	SCROLL	TABLE
NOLINK	PANEL	READERROR	SDF	TABS
NOLOCK	PARAMETERS	READKEY	SECONDARY	TAG
NOMARGIN	PARTITION	RECALL	SECONDS	TALK
NOMENU	PATH	RECCOUNT	SECTION	TAN
NOMODIFY	PAUSE	RECNO	SEEK	TARGET
NOMOUSE	PAYMENT	RECORD	SELECT	TEXT
NONE	PCOL	RECSIZE	SELECTION	TEXTMERGE
NOOPTIMIZE	PDOX	REDIT	SEPARATOR	TIME
NOOVERWRITE	PDRIVER	REFERENCE	SEQUENCE	TIMEOUT
NOREFRESH	PDSETUP	REFRESH	SET	TITLE
NORGRID	PFS	REGION	SHADOW	TITLES
NORM	PI	REGIONAL	SHADOWS	TO
NORMAL	PICTURE	REINDEX	SHOW	TOP
NOSAVE	PLAIN	RELATION	SIGN	TOPIC
NOSHADOW	PLAY	RELATIVE	SIN	TOTAL
NOSHOW	POINT	RELEASE	SINGLE	TRANSACTION
NOT	POP	REMARK	SIZE	TRANSFORM
NOTAB	POPUP	RENAME	SKIP	TRAP
NOTE	POPUPS	REPLACE	SKPBAR	TRBETWEEN
NOTIFY	PRECISION	REPLICATE	SKPPAD	TRIM
NOUPDATE	PREFERENCE	REPORT	SNAP	TXTWIDTH
NOWAIT	PREVIEW	REPROCESS	SNAPCODE	TYPE
NOWINDOW	PRIMARY	RESET	SNAPMACRO	TYPEAHEAD
NOZOOM	PRINT	RESOURCE	SOFTSEEK	UDFPARMS
NPV	PRINTER	REST	SORT	UNION
NUMBER	PRINTJOB	RESTORE	SOUNDEX	UNIQUE
NUMLOCK	PRINTSTATUS	RESUME	SPACE	UNLOCK
OBJECT	PRIVATE	RETRY	SQL	UNPACK
OBJNUM	PRMBAR	RETURN	SQRT	UPDATE
OCCURS	PRMPAD	RG	SROWS	UPDATED
ODOMETER	PRODUCTION	RIGHT	STATUS	UPPER
OEMTOANSI	PROGRAM	RLOCK	STD	USA
OF	PROJECT	ROLLBACK	STEP	USE
OFF	PROMPT	ROUND	STICKY	USED
ON	PROPER	ROW	STORE	USER
ONKEY	PROW	RPD	STR	USERS
ONLY	PUBLIC	RTOD	STRTRAN	USING
OPEN	PUSH	RTRIM	STRUCTURE	VAL

VALID
VALUE
VALUES
VAR
VARREAD
VERSION
VGA
VGA25
VGA50
VIEW
WAIT
WBORDER
WCHILD
WCOLS
WEXIST
WFONT
WHEN
WHERE
WHILE
WIDTH
WINDOW
WINDOWS
WITH
WK
WK1
WK3
WKS
WLAST
WLCOL
WLROW
WMAXIMUM
WMINIMUM
WONTOP
WORKAREA
WOUTPUT
WPARENT
WR
WR1
WRAP
WREAD
WRK
WROWS
WTITLE
WVISIBLE
XLS
YEAR
YMD
ZAP
ZOOM

Anhang J: Bibliotheken mit BIN-Funktionen

FoxPro unterstützt die Funktionen BIN2I(), BIN2L(), BIN2W(), I2BIN und L2BIN nicht. Es sind eine Reihe von Bibliotheken von Drittherstellern erhältlich, die diese Funktionen unterstützen.

phDbase

A fuzzy search developer library
Korenthal Associates
511 Avenue of the Americas
Suite 400
New York, NY 10011
(212) 242-1790

INDEX

