

Der XIO-Befehl für Dateioperationen#

Table of Contents

- [Der XIO-Befehl für Dateioperationen](#)
- [DAS ÖFFNEN EINER DATEI:](#)
- [DATEI SCHLIESSEN:](#)
- [DER XIO BEFEHL](#)
- [DATEI UMBENNENEN \(RENAME\):](#)
- [DATEI LÖSCHEN \(ERASE\):](#)
- [DATEI ZURÜCKHOLEN \(UNERASE\):](#)
- [DATEI SICHERN \(PROTECT\):](#)
- [DATEI FREIGEBEN \(UNPROTECT\):](#)
- [DISKETTE FORMATIEREN \(FORMAT\):](#)
- [FILEPOSITION SETZEN \(POINT\):](#)
- [FILEPOSITION ERMITTELN \(NOTE\):](#)

von Peter Bee

Wer stand nicht schon einmal vor dem Problem, daß er ein Programm geschrieben hat, will es abspeichern, aber die Diskette ist voll. Jetzt habe ich noch zwei, aber was ist auf der einen drauf? Und wie formatiere ich die andere?

Besitzer des BIBO-DOS sind da gut dran. Aber vielleicht brauchen Sie diese Funktionen auch einmal in einem Ihrer Programme. Zum Glück haben die Programmierer des ATARI BASIC auch an diese Möglichkeit gedacht und haben uns armen, gequälten Usern einige sehr starke Befehle zur Hand gegeben.

Fast alle Funktionen des BIBO-DOS, oder des ATARI DOS 2.0 oder 2.5, können auch vom BASIC aus ausgeführt werden. Das ATARI BASIC bietet Ihnen dazu fünf verschiedene Befehle:

OPEN, CLOSE, XIO, NOTE und POINT

Zusammen mit den BASIC Befehlen

PRINT, INPUT, PUT und GET

können Sie bereits vom BASIC aus viele Probleme im Zusammenhang mit Disketten-Operationen lösen.

Die Anwendung dieser Befehle ist sehr leicht. Wir werden Ihnen an Hand einiger Beispiele die Anwendung dieser Befehle erklären. Für alle nachfolgenden Beispiele kann eine IOCB-Nummer von 1 bis 7 verwendet werden.

DAS ÖFFNEN EINER DATEI:#

Bevor wir mit den I/O Befehlen arbeiten können, müssen wir dem Betriebssystem mitteilen, was wir vor haben. Das machen wir über den OPEN Befehl. Die Syntax ist recht einfach:

```
OPEN #IOCB, AUX1, AUX2, "Dn:NAME.EXT"
```

IOCB gibt die von Ihnen gewünschte Datenkanalnummer an. In der Regel werden Sie die Nummer 1 oder 2 benutzen. Sie können aber aus insgesamt 7 verschiedenen Kanalnummern wählen. Das Betriebssystem stellt insgesamt 8 Datenkanäle zur Verfügung, aber einer wird ständig vom EDITOR des BASIC benutzt. Also bleiben Ihnen 7 Kanäle übrig.

AUX1 gibt die Art des Zugriffs auf die zu öffnende Datei an. Beim BIBO-DOS gibt es 6 verschiedene Befehle:

4	Die Datei kann nur gelesen werden.
6	Das Directory der Diskette soll gelesen werden.

Beispiel:

```
10 DIM FILE$(20)
20 OPEN #1,6,0,"D:*.*)"
30 INPUT #1,FILE$
40 IF FILE$(5,5)="F" THEN 70
50 PRINT FILE$
60 GOTO 30
70 CLOSE #1
80 PRINT FILE$
```

7	Das Spezial-Directory mit Anzeige der gelöschten und fehlerhaften Dateien soll gelesen werden. Dieses ist eine Spezialfunktion des BIBO-DOS und nicht im DOS 2 oder 2.5 implementiert. Anwendung wie bei zuvor beschrieben.
8	Die Datei soll geschrieben werden. Eine Datei mit gleichen Namen wird gelöscht.

Bei diesem Befehl gibt es eine Besonderheit. Sie können das DOS.SYS einfach durch einen OPEN Befehl auf eine Diskette schreiben. Probieren Sie mal:

Bei DOS 2.0 und DOS 2.5

```
OPEN #1,8,0,"D:DOS.SYS"
```

Beim BIBO-DOS

```
OPEN #1,8,0,"D:BDOS.SYS"
```

Anschließend bitte nicht vergessen:

```
CLOSE #1
```

9	Append Modus. Es sollen Daten an eine vorhandene Datei angehängt werden.
12	Update Modus. Es können von der angegebenen Datei Daten gelesen und in diese Datei Daten geschrieben (angehängt) werden.

AUX2 wird weder vom BIBO-DOS noch vom DOS 2 oder DOS 2.5 benutzt und ist immer 0 (Null).

Dn:

gibt die Laufwerksnummer an, mit der gearbeitet werden soll. Beispiele: D: oder D1: stehen für Laufwerk 1, D2: für Laufwerk 2 usw.

NAME . EXT

stellt den Namen der Datei dar, mit der Sie arbeiten wollen. Beispiel: TEST.BAS

DATEI SCHLIESSEN:#

Nach beendeter Arbeit müssen wir den geöffneten Datenkanal wieder schließen. Dieses schließen ist sehr wichtig. Denn das DOS trägt das Öffnen der Datei in die Directory ein. Beenden wir die Arbeit ohne den Datenkanal zu der Datei wieder zu schließen, kann es passieren, daß wir nie wieder an die Daten der Datei herankommen. Die Syntax hierzu ist:

```
CLOSE #IOCB
```

IOCB entspricht der Datenkanalnummer die wir bereits beim Öffnen der Datei benutzt haben. Ein Beispiel für die Anwendung des CLOSE Befehles finden Sie oben, in unserem kleinen Beispielprogramm. Nach dem schließen des Datenkanals ist kein Zugriff auf die Datei mehr möglich. Es muß ein neuer Datenkanal geöffnet werden.

Alleine mit den Befehlen OPEN, CLOSE, PRINT und INPUT können Sie nun schon einiges anfangen. Beispielsweise können Sie eine kleine Adressdatei aufbauen. Wie Sie feststellen werden, ist das Programm wirklich nur ein Demoprogramm. Es soll Ihnen als Anregung zum weiterprogrammieren dienen. Sie finden es als Datei in der Hauptdirectory unter dem Namen "ADRESS.BAS".

DER XIO BEFEHL#

Mit dem XIO Befehl sind wir in der Lage, DOS Befehle direkt auszuführen. Wir haben mit Hilfe des XIO Befehles direkten Zugriff auf bestimmte DOS Routinen. Da wir in diesem Artikel alle drei DOS Arten besprechen wollen, werden wir die Spezialfunktionen des BIBO-DOS besonders hervorheben. Wie Sie gleich sehen werden, ist die Anwendung des XIO Befehles sehr leicht. Sie müssen nur die einzelnen Befehle und ihre genaue Funktion kennen. Dann können Sie vom BASIC aus Dateien umbenennen, löschen, zurückholen usw. Wichtig ist, daß der Datenkanal, mit dem Sie mit dem XIO Befehl arbeiten wollen, vor Aufruf des XIO Befehles geschlossen sein muß. Sie können das an unseren Beispielprogrammen deutlich sehen. Aber beginnen wir mit dem Befehl 32, DATEI UMBENNEN:

DATEI UMBENNENEN (RENAME):#

Die Anwendung ist dieselbe wie beim DOS. Sie rufen die RENAME Funktion mit dem XIO Befehl 32 auf. Danach kommt die Datenkanalnummer, mit der Sie arbeiten wollen, gefolgt von zwei in dieser Funktion unwichtigen Bytes. Anschließend der alte Dateiname, gefolgt von dem neuen. Getrennt werden die beiden Namen durch ein Komma. Der neue Name darf auf keinen Fall auf der Diskette schon einmal vorhanden sein!

```
XIO 32, #IOCB, 0, 0, "Dn:ALT.EXT, NEU.EXT"
```

Nach Ausführung der Funktion wird der Datenkanal automatisch vom DOS wieder geschlossen. Sie brauchen sich darum also nicht zu kümmern!

DATEI LÖSCHEN (ERASE):#

Auch hier ist die Anwendung wieder dieselbe wie beim DOS. Sie rufen die ERASE Funktion mit dem XIO Befehl 33, gefolgt von der Datenkanalnummer, und den zwei in dieser Funktion unwichtigen Bytes. Anschließend der Name der Datei, die gelöscht werden soll.

```
XIO 33, #IOCB, 0, 0, "Dn:NAME.EXT"
```

Auch nach Ausführung dieser Funktion wird der Datenkanal automatisch vom DOS wieder geschlossen.

DATEI ZURÜCKHOLEN (UNERASE):#

Das BIBO-DOS bietet Ihnen die Möglichkeit, eine aus Versehen gelöschte Datei wieder "zurückzuholen". Ein zurückholen einer Datei ist aber nur möglich, wenn der Directory-Eintrag noch vorhanden ist und die Daten der Datei noch nicht durch eine neue Datei überschrieben wurden. Der XIO Befehl für diese Funktion ist 34:

```
XIO 34, #IOCB, 0, 0, "Dn:NAME.EXT"
```

Wichtig ist, daß der angegebene Name mit dem der gelöschten Datei übereinstimmt!

DATEI SICHERN (PROTECT):#

Mit dem XIO Befehl 35 können Sie eine Datei, oder mehrere Dateien, gegen ein unbeabsichtigtes löschen schützen. Beispiel:

```
XIO 35, #IOCB, 0, 0, "Dn:NAME.EXT"
```

Auch nach Ausführung dieser Funktion wird der Datenkanal automatisch vom DOS wieder geschlossen.

DATEI FREIGEBEN (UNPROTECT):#

Das genaue Gegenteil des Befehles 35 stellt der XIO Befehl 36 dar. Mit diesem Befehl geben Sie eine Datei, oder mehrere Dateien, frei. Dadurch können diese Dateien gelöscht oder überschrieben werden. Beispiel:

```
XIO 36, #IOCB, 0, 0, "Dn:NAME.EXT"
```

Auch hier wird nach Ausführung der Funktion der Datenkanal automatisch vom DOS wieder geschlossen.

DISKETTE FORMATIEREN (FORMAT):#

Als letzter der XIO Befehle haben wir einen etwas komplizierteren Befehl. Es ist auch der einzige, bei dem AUX2 eine Rolle spielt. Es ist der FORMAT Befehl 254. Sehen wir uns zuerst die Syntax an:

```
XIO 254, #IOCB, 0, AUX2, "Dn: "
```

Mit diesem XIO Befehl kann eine Diskette in Laufwerk n formatiert werden. Wie das BIBO-DOS benutzt, gibt es hier eine Besonderheit. Der Wert in AUX2 bestimmt die Dichte (DENSITY) der zu formatierenden Diskette. Dies gilt aber nur beim BIBODOS. Hier die Werte für AUX2:

AUX2	0 Single Density
AUX2	1 Double Density. Funktioniert natürlich nur auf erweiterten Diskettenlaufwerken.
AUX2	2 Medium Density (DOS 2.5 kompatibel).
AUX2	128 Clear Disk. Die Diskette wird nicht formatiert, es werden nur die VTOC, die Directory-Sektoren und die Bootsektoren neu geschrieben.

Soll mit dem BIBO-DOS die Ramdisk formatiert werden, ist der Wert in AUX2 nicht von Bedeutung. Die Ramdisk wird immer in der konfigurierten Größe formatiert. Insofern verhält sich der XIO Befehl 254, genau wie alle anderen, genauso wie sein Gegenstück "I" im DUP-Menü des BIBO-DOS.

Wie versprochen, erhalten Sie ein Demo Programm in ATARI BASIC. An Hand dieses Programmes können Sie die Anwendung aller besprochenen Befehle erkennen. Sicher ist das Programm sehr umständlich geschrieben, man könnte es auch viel kürzer schreiben. Aber darum geht es gar nicht. Wichtig ist nur die Anwendung der OPEN und der XIO Befehle. Und wenn jemand etwas daraus lernt, hat es sich schon gelohnt. Sie finden das Programm übrigens als Datei in der Hauptdirectory der Magazin Diskette unter dem Namen "MINIDOS.BAS". Laden Sie es bitte vom Atari BASIC aus. Wenn Sie es vom Inhaltsverzeichnis der Magazin Diskette aus starten, arbeiten einige Funktionen nicht!

Als letztes kommen wir nun zu zwei BASIC Befehlen, mit denen Sie in der Lage sind, Daten direkt auf die Diskette zu schreiben, oder direkt von der Diskette zu lesen. Sie können mit diesen Befehlen die Position von einzelnen Daten auf der Diskette bestimmen, bzw. ermitteln.

FILEPOSITION SETZEN (POINT):#

```
XIO 37,#IOCB,0,0,"Dn:"  
POINT #IOCB,X,Y
```

Diese Funktion wird mit dem Basic-Befehl POINT aufgerufen. Der IOCB-Kanal muß bereits für das zu bearbeitende File geöffnet sein. Wird der POINT-Befehl verwendet, enthält X die Sektornummer und Y die Position in diesem Sektor.

Soll die XIO-Funktion verwendet werden, müssen folgende Speicherstellen die erforderlichen Werte enthalten:

```
POKE 846+IOCB*16,Position  
POKE 845+IOCB*16,Sector (High-Byte)  
POKE 844+IOCB*16,Sector (Low-Byte)
```

FILEPOSITION ERMITTELN (NOTE):#

```
XIO 38,#IOCB,0,0,"Dn:"  
NOTE #IOCB,X,Y
```

Diese Funktion wird mit dem Basic-Befehl NOTE aufgerufen. Der IOCB-Kanal muß auch hier bereits geöffnet sein. Wenn der NOTE-Befehl verwendet wird, erhält man in der Variablen X die Sektornummer und in der Variablen Y die Position in diesem Sektor zurück.

Bei der XIO-Funktion können die gleichen Speicherstellen wie beim POINT-Befehl ausgelesen werden:

```
Position=PEEK(846+IOCB*16)  
Sektor=PEEK(844+IOCB*16)+256*PEEK(845+IOCB*16)
```