

GetVer

Jörg Schuchardt

Copyright © Copyright 1994-1996 by J. Schuchardt

COLLABORATORS

	<i>TITLE :</i> GetVer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Jörg Schuchardt	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GetVer	1
1.1	GetVer 4.00 Anleitung	1
1.2	Haftungsausschluß des Autors	2
1.3	Shareware-Bestimmungen für GetVer	2
1.4	Kopier- und Verbreitungsbestimmungen für GetVer	3
1.5	Sinn und Zweck des Programms	4
1.6	Vorzüge gegenüber dem DOS-Befehl VERSION	5
1.7	Systemvoraussetzungen	5
1.8	Installation des Programms	5
1.9	Funktions- und Vorgehensweise	6
1.10	Der Aufruf von der Shell	6
1.11	Das Argument <Namensmuster Dateiname>	7
1.12	Das Argument DEF DEFAULT	7
1.13	Das Argument FULL	8
1.14	Das Argument ALL	8
1.15	Das Argument SORT	8
1.16	Das Argument FORMAT	9
1.17	Das Argument FORCE	9
1.18	Das Argument CMP COMPARE	9
1.19	Das Argument PATH	10
1.20	Das Argument FILE	10
1.21	Das Argument BRIEF	11
1.22	Das Argument REC RECURSIVE	11
1.23	Das Argument DATE	12
1.24	Das Argument COM COMMENT	13
1.25	Das Argument LIB	13
1.26	Das Argument DEV	14
1.27	Das Argument DTYPE	14
1.28	Das Argument BUF BUFFER	14
1.29	Die Argumente VER VERSION und REV REVISION	15

1.30	Das Argument SYSTEM	15
1.31	Das Argument INFO	15
1.32	Der Aufruf von der Workbench	16
1.33	Das Graphical User Interface (GUI)	17
1.34	Woher man Updates bekommt	20
1.35	Die Zukunft von GetVer	20
1.36	Danksagungen	21
1.37	Index	21

Chapter 1

GetVer

1.1 GetVer 4.00 Anleitung

GetVer 4.00

Copyright © 1992-1996 by Jörg Schuchardt

- SHAREWARE -

1. Rechtliches

[Haftung](#)

[Shareware](#)

[Copyright](#)

2. Allgemeines

[Beschreibung](#)

[Warum nicht VERSION?](#)

[Systemvoraussetzungen](#)

[Installation](#)

[Funktionsweise](#)

3. Bedienung

[CLI-Aufruf](#)

[Workbench-Aufruf](#)

[Das GUI](#)

4. Sonstiges

[Updates](#)

[History](#)

[Die Zukunft](#)

[Danksagungen](#)

1.2 Haftungsausschluß des Autors

ACHTUNG!

DIE ZUM GETVER-PACKET GEHÖRENDE SOFTWARE WIRD IHNEN "SO WIE SIE IST" ZUR VERFÜGUNG GESTELLT UND BEINHALTET WEDER AUSDRÜCKLICH, NOCH STILLSCHWEIGEND IRGENDWELCHE GARANTIEN.

DER AUTOR ÜBERNIMMT KEINERLEI HAFTUNG FÜR JEDLICHE ART VON SCHÄDEN ODER VERLUSTEN DIE DURCH DIREKTE ODER INDIREKTE BENUTZUNG DER SOFTWARE AUFTRETEN. DIE ANWENDUNG DER SOFTWARE GESCHIEHT AUF EIGENE VERANTWORTUNG.

1.3 Shareware-Bestimmungen für GetVer

SHAREWARE

Ab der Version 3.00 ist GetVer ein SHAREWARE-Programm, das ohne Schlüsseldatei einige wenige Einschränkungen enthält:

- Vor jeder Ausgabe erscheint der Text

"GetVer is SHAREWARE! Please register..."

Erst nach Ablauf von ca. 10 Sekunden verschwindet dieser Text wieder und GetVer beginnt mit der eigentlichen Arbeit.

- Die **Puffergröße** ist auf einen erheblich kleineren Wert festgelegt, was die Arbeitsgeschwindigkeit beeinträchtigt.

- Das Speichern der ermittelten Versionsnummer als **Kommentar**, das Anpassen des **Dateidatums** an das ermittelte Erstellungsdatum sowie das Einstellen eines **Ausgabemediums** in der GUI ist nicht möglich.

Wenn Sie GetVer nach einer Testphase von 14 Tagen weiterhin benutzen, müssen sie sich registrieren lassen. Nach erfolgter Registrierung erhalten Sie ihre persönliche Schlüsseldatei, die die vorgenannten Einschränkungen aufhebt.

Die Registrierungsgebühr beträgt:

Innerhalb Deutschlands: 10,- DM

Außerhalb Deutschlands: 10,- US\$

Andere Währungen werden nicht akzeptiert!

Legen Sie einfach den entsprechenden Betrag als Geldschein in das ausgefüllte und gefaltete Registrierformular, stecken Sie dieses so in einen Briefumschlag, daß das Geld von außen nicht erkennbar ist, und schicken Sie es an meine Hausanschrift.

Wenn Sie Ihre Schlüsseldatei schnell und sicher per E-Mail bekommen

möchten, dann senden Sie Ihren öffentlichen PGP-Schlüssel an meine Netzadresse.

Sie erhalten Ihre persönliche Schlüsseldatei, so wie in der Registrierungsmeldung vermerkt, zugeschickt.

Durch die Registrierung erwerben Sie lediglich das Recht zur fortgesetzten Nutzung des Programms GetVer sowie zur Nutzung Ihrer persönlichen Schlüsseldatei, welche nicht an Dritte weitergegeben oder -veräußert werden darf. Die Schlüsseldatei bleibt weiterhin mein Eigentum!

Meine Anschrift ist: E-Mail:

Jörg Schuchardt ODIN@liteline.mcnet.de

Fichtenweg 4

37181 Hardegsen

1.4 Kopier- und Verbreitungsbestimmungen für GetVer

COPYRIGHT

GetVer 4.00 darf nur in dem nachfolgend aufgeführten Umfang und in keiner Weise verändert weitergegeben werden:

Catalogs (dir)

English (dir)

GetVer.catalog

Português (dir)

GetVer.catalog

Deutsch (dir)

GetVer.guide GetVer.guide.info

GetVer.history GetVer.history.info

GetVer.register GetVer.register.info

Install.info

English (dir)

GetVer.guide GetVer.guide.info

GetVer.history GetVer.history.info

GetVer.register GetVer.register.info

Install.info

MWB-Icons (dir)

GetVer0 GetVer0.info

GetVer1 GetVer1.info

GetVer2 GetVer2.info

GetVer3 GetVer3.info

GetVer4 GetVer4.info

GetVer5 GetVer5.info

GetVer6 GetVer6.info

Info! Info!.info

ImageCopy_GVO

S (dir)

GetVer.paths

Deutsch.info English.info

GetVer GetVer.info

Install MWB-Icons.info

Es ist nicht erlaubt, die GetVer-Schlüsseldatei zu verkaufen, verschenken, verleihen oder auf anderem Wege in den Besitz Dritter zu bringen.

Es ist nicht gestattet, die Software in eine PD-Serie aufzunehmen, wenn nicht mindestens eine der folgenden Voraussetzungen zutrifft:

- Es handelt sich um die Aminet Serie

(einschl. Aminet CD-ROM, Aminet Set CD-ROM und Aminet Disketten)

- Es handelt sich um die Meeting Pearls CD-ROM Serie

- Eine Diskette dieser Serie kostet nicht mehr als 5,- DM(*)

- Ein CD-ROM dieser Serie kostet nicht mehr als 25,- DM(*)

(*) oder den entsprechenden Betrag in einer anderen Währung

Eine irgendwie geartete Verbreitung der Software mit kommerziellen Produkten ist nur mit der ausdrücklichen, schriftlichen Genehmigung des Autors gestattet.

Diese kann ebenfalls bei der unter [Shareware](#) angegebenen Adresse schriftlich erfragt werden.

1.5 Sinn und Zweck des Programms

Beschreibung

GetVer ist ein Ersatz für den Standard DOS-Befehl VERSION. Es ermittelt die Versionsnummer eines Programms, einer Library, eines Devices, eines Handlers oder eines Datatypes, auf Disk oder im Speicher. Je nach Wunsch vergleicht es Versionen und sortiert und/oder formatiert seine Ausgabe. Es kann die gefundene Versionsnummer als Dateikommentar anhängen und das Dateidatum auf das Compilierungsdatum setzen, sofern dieses im Versionsstring enthalten ist. Desweiteren benutzt GetVer die üblichen DOS-Joker und ist dadurch sehr flexibel.

Nun wird sich manch einer vielleicht fragen:

"Warum soll ich GetVer benutzen? Es gibt doch schon einen DOS-Befehl dafür. **Warum nicht VERSION?**"

1.6 Vorzüge gegenüber dem DOS-Befehl VERSION

Warum nicht VERSION?

Der VERSION-Befehl hat einige Unzulänglichkeiten, als da wären:

- es kann immer nur eine Datei getestet werden
- keine Vergleichsfunktion
- kein Aufruf von der Workbench möglich

All diese Mängel sind durch GetVer beseitigt, und die sich dadurch ergebenden neuen Möglichkeiten durch ergänzende Funktionen implementiert worden. Dabei erkennt GetVer wesentlich mehr Versionsnummern, da das Verfahren der Versionsstringerkennung großzügiger arbeitet (s. **Funktionsweise**).

1.7 Systemvoraussetzungen

Systemvoraussetzungen

Da einige neue Betriebssystemfunktionen verwendet werden, ist GetVer nur unter OS 2.0 und höher funktionsfähig!

Der RAM-Bedarf ist abhängig von den Funktionen, die GetVer ausführen soll, von der Anzahl der Dateien, deren Version ermittelt werden soll und natürlich von der eingestellten Puffergröße. Für einen reibungslosen Betrieb sollten aber mindestens 100 KByte RAM zur Verfügung stehen.

1.8 Installation des Programms

Installation

Das GetVer 4.00-Archiv beinhaltet ein Install-Skript für den Installer von Commodore. Um es zu starten öffnen Sie das Verzeichnis, das mit der gewünschten Sprache benannt ist, und klicken Sie dann einfach doppelt auf das Installpiktogramm.

Benutzer der MagicWB von Martin Huttenloher, die eines der im Verzeichnis "MWB-Icons" enthaltenen Piktogramme verwenden wollen, beachten bitte, daß sie das gewünschte Piktogramm doppelt anklicken müssen BEVOR sie die Installation starten! Es erscheint dadurch als Programmpiktogramm im "GetVer"-Fenster.

Registrierte Anwender kopieren ihre Schlüsseldatei als "GetVer.key" nach ENVARC: (wenn die Schlüsseldatei sofort aktiv sein soll, auch nach ENV:) oder in das Verzeichnis, in dem auch GetVer selbst liegt. Beachten Sie für die CLI-Variante bitte, daß das Programm in einem Verzeichnis stehen sollte, das mittels PATH in den Suchpfad eingebunden ist.

1.9 Funktions- und Vorgehensweise

Funktionsweise

GetVer prüft keine .info-Dateien und standardmäßig (ohne **FORCE** - Option) nur ausführbare Dateien (Hex-Code "000003F3" als erste Bytes - also auch Libraries, Devices, Fonts, ...).

Es durchsucht den Programmcode nach der Zeichenfolge "\$VER:", die bei neueren Dateien standardgemäß den Versionsstring einleitet.

Parallel dazu wird der Code nach dem Namen der Datei durchsucht.

Wurde kein "\$VER:" gefunden, dann wird versucht den Versionsstring mit Hilfe eines Musters zu identifizieren, das etwa so aussieht:

```
"#?Dateiname#?[0-9].#[0-9]#?"
```

Verständlicher Weise kann dies nicht in jedem Fall zum Erfolg führen und GetVer gibt dann schon mal einen String aus, der offensichtlich nicht der Versionsstring ist.

So geschehen z. B. bei dem Programm "RKick", das kein "\$VER:" aber als Syntaxbeschreibung für den CLI-Aufruf den Text "RKick [1.3|2.x]" enthält. Da er dem Muster entspricht, interpretiert GetVer diesen Text fälschlicher Weise als Versionsstring und gibt ihn aus.

Sorry! Aber um eine möglichst effiziente Erkennungsroutine zu realisieren, die auch Versionsstrings älterer Dateien erkennt, ist dieses Manko in Kauf zu nehmen. Glücklicherweise tritt dieser Umstand ja auch sehr selten auf.

1.10 Der Aufruf von der Shell

CLI-Aufruf

Für die CLI- bzw. Shell-Variante von GetVer gelten folgende Konventionen:

Format

```
GETVER {<Name>} [DEF|DEFAULT] [FULL] [ALL] [SORT] [FORMAT]
[FORCE] [BRIEF] [CMP|COMPARE [PATH=<path>] [FILE=<file>]]
[REC|RECURSIVE] [DATE] [COM|COMMENT=<mode>] [LIB|DEV|DTYPE]
[BUF|BUFFER=<n>] [VER|VERSION=<n>] [REV|REVISION=<n>]
[SYSTEM] [INFO]
```

Schablone

```
SOURCE/A/M,DEF=DEFAULT/S,FULL/S,ALL/S,SORT/S,FORMAT/S,FORCE/S,
CMP=COMPARE/S,PATH/K,FILE/K,BRIEF/S,REC=RECURSIVE/S,DATE/S,
COM=COMMENT/K,LIB/S,DEV/S,DTYPE/S,BUF=BUFFER/K/N,
```

VER=VERSION/K/N,REV=REVISION/K/N,SYSTEM/S,INFO/S:

Die Bedeutung und Anwendung der einzelnen Argumente ist dabei folgendermaßen festgelegt:

Dateiname REC|RECURSIVE

DEF|DEFAULT DATE

FULL COM|COMMENT

ALL LIB

SORT DEV

FORMAT D|TYPE

FORCE BUF|BUFFER

CMP|COMPARE VER|VERSION

PATH REV|REVISION

FILE SYSTEM

BRIEF INFO

1.11 Das Argument <Namensmuster|Dateiname>

CLI-Argument Namensmuster|Dateiname

Der Name der Datei, deren Versionsnummer ermittelt werden soll, ggf. mit Angabe des Verzeichnisses.

Bei Dateinamen mit den Endungen ".library", ".device", "-handler" und ".datatype" sucht GetVer automatisch in vorgegebenen Standardverzeichnissen (LIBS:, DEVS:, L:, SYS:Classes/Datatypes) wenn die entsprechende Datei im angegebenen Pfad nicht gefunden wurde.

Es können ein oder mehrere Dateinamen, mit oder ohne Jokerzeichen angegeben werden.

Beispiel:

```
> GetVer asl.library
```

Gibt den Versionsstring der asl.library in LIBS: aus.

1.12 Das Argument DEF|DEFAULT

CLI-Argument DEF|DEFAULT

Unterbindet das Lesen der Tooltypes des Programmicons. Die darin festgelegten Parameter werden somit nicht als Voreinstellungen übernommen.

Beispiel:

```
> GetVer asl.library DEFAULT
```

1.13 Das Argument FULL

CLI-Argument FULL

Die normale Ausgabe des Versionsstrings endet hinter der Versionsnummer. Meistens stehen dahinter aber noch Informationen, wie z. B. das Erstellungsdatum. Mit FULL werden diese Informationen mit ausgegeben.

Beispiel:

```
> GetVer LIBS:asl.library FULL
```

Zeigt die Version und das Erstellungsdatum der asl.library.

1.14 Das Argument ALL

CLI-Argument ALL

Bei Verwendung des Jokerzeichens im Dateinamen, prüft GetVer nur die erste mit der Jokervorgabe übereinstimmende Datei. Mit ALL werden alle mit der Vorgabe übereinstimmenden Dateien geprüft.

Beispiel:

```
> GetVer LIBS:a#? ALL
```

Zeigt die Versionsstrings aller ausführbaren Dateien in LIBS:, die mit "a" anfangen.

1.15 Das Argument SORT

CLI-Argument SORT

Bei Überprüfung mehrerer Dateien wird normalerweise in der Reihenfolge vorgegangen, in der die Dateien auf dem Datenträger gefunden werden. Durch SORT werden die Dateien in alphabetischer Reihenfolge geprüft und ausgegeben.

Bei der Angabe mehrerer Dateien in verschiedenen Pfaden, wird vorrangig nach dem Pfad sortiert und innerhalb dieses nach dem Dateinamen. Auf diese Weise stehen alle Dateien eines Verzeichnisses immer beieinander.

Beispiel:

```
> GetVer LIBS:[alblc]#? C:List ALL SORT
```

Listet die Versionen aller ausführbaren Dateien in LIBS:, die mit "a", "b" oder "c" beginnen und von "C:List" in alphabetisch sortierter Form.

1.16 Das Argument FORMAT

CLI-Argument FORMAT

Normalerweise wird einfach nur der gefundene Versionsstring ausgegeben. Bei manchen Dateien lautet dieser aber z. B. einfach nur "Version 1.3" und gibt so keine Auskunft darüber, welche Datei getestet wurde. Mittels FORMAT wird zuerst der Dateiname ausgegeben, dann der Pfad in dem die Datei gefunden wurde und erst dann der Versionsstring. Desweiteren wird die Ausgabe zur besseren Übersicht formatiert.

Diese Option wird bei Verwendung des **CMP|COMPARE** -Arguments automatisch aktiviert!

Beispiel:

```
> GetVer LIBS:xpr#? ALL FORMAT
```

Zeigt die Versionen aller mit "xpr" beginnenden Libraries unsortiert, aber formatiert.

1.17 Das Argument FORCE

CLI-Argument FORCE

Ohne dieses Argument überprüft GetVer nur ausführbare Dateien (Hex-Code "000003F3" als erste Bytes einer Datei - also auch Libraries, Devices, Fonts, ...). Durch Angabe von FORCE kann man auch andere Dateien überprüfen lassen (z. B. ASCII-Dateien).

Beispiel:

```
> GetVer GetVer.guide FORCE
```

Erzwingt das Lesen des "GetVer.guide"s, das ja keine ausführbare Datei ist.

1.18 Das Argument CMP|COMPARE

CLI-Argument CMP|COMPARE

Um festzustellen, ob neue Libraries, Devices, Datatypes oder andere Dateien neuer sind als die, welche man schon besitzt, kann man das Argument COMPARE benutzen. Es liest zuerst die Datei aus dem angegebenen Pfad und dann aus SYS:Classes/DataTypes, LIBS:, DEVS: oder L:, je nachdem ob die Datei ein Datatype, eine Library, ein Device oder ein Handler ist, und listet beide auf.

Zur leichteren Identifizierung der neuesten Dateien, wird die

größte Versionsnummer in Farbe 2 (Weiß), alle anderen in Farbe 3 (Blau) ausgegeben.

Dieses Argument aktiviert automatisch die **FORMAT** -Option!

Beispiel:

```
> GetVer DF0:Libs/reqtools.library COMPARE
```

Gibt die Versionen der "reqtools.library" in DF0:Libs/ und in LIBS: aus.

Hinweise:

- Bei gleichzeitiger Verwendung mit den Argumenten **LIB** , **DEV** oder **DTYPE** wird die Datei im angegebenen Pfad mit der entsprechenden Datei im Speicher verglichen.
- Um die Vergleichspfade zu ändern verwenden Sie die Argumente **PATH** oder **FILE** .

1.19 Das Argument PATH

CLI-Argument PATH

DIESES ARGUMENT IST EINE UNTERFUNKTION VON **CMPICOMPARE** UND STEHT SOMIT NUR IM ZUSAMMENHANG MIT DIESER OPTION ZUR VERFÜGUNG!

Wenn man als Vergleichspfad nicht die Standardpfade LIBS:, DEVS:, L: oder SYS:Classes/DataTypes benutzen will, sondern ein beliebiges anderes Verzeichnis, dann gibt man dies hinter dem Schlüsselwort PATH ein.

Mehrere Pfade werden durch Kommata voneinander getrennt.

Beispiel:

```
> GetVer xpkNUKE.library COMPARE PATH DF1:Libs,LIBS:compressors
```

Gibt die Versionen der "xpkNUKE.library" im aktuellen Pfad, DF1:Libs und LIBS:compressors aus.

1.20 Das Argument FILE

CLI-Argument FILE

DIESES ARGUMENT IST EINE UNTERFUNKTION VON **CMPICOMPARE** UND STEHT SOMIT NUR IM ZUSAMMENHANG MIT DIESER OPTION ZUR VERFÜGUNG!

Wenn man als Vergleichspfad nicht die Standardpfade LIBS:, DEVS:, L: oder SYS:Classes/DataTypes benutzen will, sondern beliebige andere Verzeichnisse, dann kann man diese mit einem Texteditor in eine Datei schreiben. Jedes Verzeichnis steht dabei alleine in einer Zeile.

Beispiel:

```
> GetVer SuperTool COMPARE FILE S:GetVer.paths
```

Gibt die Versionen des Programms "SuperTool" im aktuellen Pfad und in allen, in der Datei "S:GetVer.paths" stehenden Verzeichnissen aus.

1.21 Das Argument BRIEF

CLI-Argument BRIEF

Wenn GetVer eine Datei in dem angegebenen Pfad nicht finden kann, dann erscheint der Text "'Datei' nicht in 'Pfad' gefunden." Da diese Ausgaben bei Verwendung der Vergleichsfunktion für jeden Vergleichspfad erscheinen, in dem die Datei nicht gefunden wurde, erhält man mitunter recht lange Listen in denen eben diese Meldung vorherrscht. Daher bietet das Argument BRIEF die Möglichkeit diese Meldungen heraus zu filtern, so daß Ausgaben nur noch für existierende Dateien erfolgen.

Beispiel:

```
> GetVer #? COMPARE FILE S:GetVer.paths BRIEF
```

Gibt die Versionen aller existierenden und ausführbaren Dateien im aktuellen Pfad und in allen, in der Datei "S:GetVer.paths" stehenden Verzeichnissen aus.

1.22 Das Argument REC|RECURSIVE

CLI-Argument REC|RECURSIVE

Bei Benutzung des Arguments RECURSIVE durchsucht GetVer nicht nur das genannte Directory, sondern auch alle darin enthaltenen Unterverzeichnisse nach dem angegebenen Dateinamen oder Namensmuster.

Beispiel:

```
> GetVer SYS:Tools/A#? ALL RECURSIVE
```

Gibt die gefundenen Versionsstrings aller mit "A" beginnenden Dateien in SYS:Tools und allen darin enthaltenen Unterverzeichnisse aus.

1.23 Das Argument DATE

CLI-Argument DATE

FUNKTION IN DER UNREGISTRIERTEN VERSION NICHT AKTIV!

Bei vielen Dateien werden neben der Versionsnummer auch noch weitere Informationen gespeichert (die man mit der Option **FULL** anzeigen lassen kann). Enthalten diese zusätzlichen Informationen ein Datum, so wird, bei Verwendung des Parameters DATE das Dateidatum auf das gefundene Datum gesetzt.

Für die Datumserkennung gelten folgende Konventionen:

1. Die Position von Tag, Monat und Jahr ist im Prinzip beliebig.

Die Angaben werden in dieser Reihenfolge gelesen und werden gegebenenfalls vertauscht:

· Tag gegen Monat, wenn der Monat größer 12 und der Tag kleiner 13 ist.

· Tag gegen Jahr, wenn der Tag größer 31 und das Jahr kleiner 32 ist.

2. Der Monat darf als Zahl, Wort oder Kurzwort (Jan, Feb, etc.) in der deutschen oder englischen Schreibweise enthalten sein.

3. Die Trennung zwischen den einzelnen Angaben darf erfolgen durch Leerzeichen (" "), Punkte ("."), Bindestriche/Minuszeichen ("-") und Schrägstriche ("/").

Gültige Formate wären also:

17.12.94

17 Dec 94

Dez 17 94

December-17-1994

etc.

Stimmt eine evtl. enthaltene Datumsangabe nicht mit diesen Konventionen überein, wird kein Datum erkannt und das eingestellte Dateidatum wird nicht verändert.

Wenn der Versuch, ein gefundenes Datum als Dateidatum zu speichern, aus irgendeinem Grund scheitert, dann erscheint ein Requester mit vier Knöpfen, die, von links nach rechts betrachtet, folgende Funktionen haben:

· Fortsetzen der Bearbeitung

· Unterdrückung des Requesters bei weiteren Fehlversuchen

· Deaktivieren der DATE-Funktion

· Abbrechen der Bearbeitung

Beispiel:

```
> GetVer xpr#.library ALL DATE
```

Stellt bei allen XPR-Libraries in LIBS:, in denen ein Datum erkannt wurde, dieses als Dateidatum ein.

1.24 Das Argument COM|COMMENT

CLI-Argument COM|COMMENT

FUNKTION IN DER UNREGISTRIERTEN VERSION NICHT AKTIV!

Dieses Argument speichert eine gefundene Versionsnummer in der Form "V #.#" als Kommentar zu einer Datei. Daraufhin kann man also einfach mit dem LIST-Befehl die Versionsnummern aller Dateien anzeigen lassen.

Die Funktion kennt drei Modi:

KEEP Die Versionsnummer wird nur dann als Kommentar gespeichert, wenn noch kein Kommentar existiert.

ADD Die Versionsnummer wird als Kommentar gespeichert. Ein evtl. schon existierender Kommentar wird dahinter gehängt.

FORCE Die Versionsnummer wird als Kommentar gespeichert. Ein evtl. schon existierender Kommentar wird gelöscht!

Jeder andere Modusname (z. B. OFF) deaktiviert eine, in den Tooltypes eventuell definierte Voreinstellung.

Wenn der Versuch, einen Kommentar zu speichern, aus irgendeinem Grund scheitert, dann erscheint ein Requester mit vier Knöpfen, die, von links nach rechts betrachtet, folgende Funktionen haben:

- Fortsetzen der Bearbeitung
- Unterdrückung des Requesters bei weiteren Fehlversuchen
- Deaktivieren der COMMENT-Funktion
- Abbrechen der Bearbeitung

Beispiel:

```
> GetVer LIBS: COMMENT=FORCE
```

Listet die gefundenen Versionen aller in LIBS: stehenden Dateien und speichert die Versionsnummern als Kommentar zu den jeweiligen Dateien.

1.25 Das Argument LIB

CLI-Argument LIB

Mit LIB wird die angegebene Datei als Library interpretiert und nicht von Disk gelesen, sondern aus dem Speicher.

Beispiel:

```
> GetVer exec.library LIB
```

Zeigt die Version der im Speicher stehenden "exec.library".

1.26 Das Argument DEV

CLI-Argument DEV

Mit DEV wird die angegebene Datei als Device interpretiert und nicht von Disk gelesen, sondern aus dem Speicher.

Beispiel:

```
> GetVer timer#? DEV
```

Zeigt die Version eines im Speicher stehenden, mit "timer" beginnenden Devices (also in der Regel des "timer.device"s).

1.27 Das Argument DTYPE

CLI-Argument DTYPE

Mit DTYPE wird die angegebene Datei als Datatype interpretiert und nicht von Disk gelesen, sondern aus dem Speicher.

Beispiel:

```
> GetVer Amiga#? DTYPE
```

Zeigt die Version eines im Speicher stehenden, mit "Amiga" beginnenden Datatypes (hier vermutlich des "Amigaguide.datatype"s).

1.28 Das Argument BUF|BUFFER

CLI-Argument BUF|BUFFER

FUNKTION IN DER UNREGISTRIERTEN VERSION NICHT AKTIV!

Normalerweise arbeitet GetVer mit einem Puffer von 2 KBytes Größe, in den die Dateidaten eingelesen und nach dem Versionsstring durchsucht werden. Durch Angabe von BUFFER=n kann man diese Größe verändern, wobei n eine Zahl zwischen 1 und 20 sein kann und die Puffergröße in KByte angibt.

Durch eine Vergrößerung des Puffers wird die Ausführungszeit teilweise extrem verkürzt.

Beispiel:

```
> GetVer LIBS:#? ALL SORT FORMAT BUFFER=20
```

Zeigt die Versionen aller in LIBS: stehenden, ausführbaren Dateien in sortierter und formatierter Form und setzt dazu den Puffer auf 20 KBytes.

1.29 Die Argumente VER|VERSION und REV|REVISION

CLI-Argumente VER|VERSION und REV|REVISION

Die Versions-/Revisionsnummer, die die zu prüfenden Dateien mindestens haben sollten. Sind die gefundenen Nummern der Datei kleiner, dann wird das Bedingungskennzeichen auf 5 (WARN) gesetzt. Anderenfalls bleibt es auf 0.

Bei mehreren zu durchsuchenden Dateien darf keine einzige Versions- oder Revisionsnummer kleiner als die angegebene sein, damit das Bedingungskennzeichen auf 0 bleibt.

Ist auch nur eine der beiden Zahlen angegeben, dann entfällt die "Datei: in x Bytes keinen Versionsstring gefunden"-Ausgabe, die sonst angezeigt wird, wenn keine Versionsnummer gefunden wurde.

Durch Anhängen von "VERSION=0" kann man so auf einfache Weise diese Ausgaben herausfiltern.

Beispiel:

```
> GetVer LIBS:asl.library VERSION=39
```

Zeigt die Version der asl.library und liefert WARN (5) wenn sie kleiner ist als 39.

1.30 Das Argument SYSTEM

CLI-Argument SYSTEM

Um die Versionen von Kickstart und Workbench zu ermitteln geben Sie den Parameter SYSTEM an.

Beispiel:

```
> GetVer SYSTEM
```

zeigt die Version des Kickstarts und die der Workbench an.

1.31 Das Argument INFO

CLI-Argument INFO

Zeigt Informationen über GetVer an.

Achtung!

Dieses Argument muß allein stehen!

Beispiel:

```
> GetVer INFO
```

1.32 Der Aufruf von der Workbench

Workbench-Aufruf

Es gibt zwei Möglichkeiten GetVer von der Workbench zu aktivieren:

1. Wählen Sie zuerst das GetVer-Piktogramm aus¹. Erweitern Sie dann die Auswahl² um die Piktogramme der Dateien, deren Versionsnummer Sie wissen möchten. Halten Sie die SHIFT-Taste gedrückt und doppelklicken Sie eines der ausgewählten Piktogramme.

2. Doppelklicken Sie das GetVer-Piktogramm. Es erscheint ein AppIcon, das mit "GetVer 4.00" unternitelt ist und genauso aussieht, wie das GetVer-Piktogramm. Wählen Sie nun die Piktogramme der Dateien aus², deren Versionsnummer sie wissen möchten. Ziehen Sie dann die ausgewählten Piktogramme³ auf das GetVer-AppIcon und lassen Sie die Auswahl taste los, wenn sich die Spitze des Mausfeils auf dem AppIcon befindet.

Um das AppIcon wieder zu entfernen, doppelklicken Sie noch einmal das GetVer-Piktogramm oder klicken Sie das AppIcon doppelt an und wählen Sie aus dem Menü des sich daraufhin öffnenden GetVer-Einsteller-Fensters den Punkt PROJEKT/BEENDEN.

Bei beiden Varianten beginnt GetVer nun mit der Überprüfung der von Ihnen ausgewählten Dateien und gibt das Ergebnis auf das, im GUI unter **Ausgabespezifikation** eingestellte Medium aus. Normalerweise ist hier ein Console-Fenster eingestellt, daß Sie entweder durch einen Klick auf das Schließsymbol in der linken oberen Ecke oder durch drücken von <CTRL-\> wieder schließen können.

Workbench-Argumente

Die Workbench-Variante akzeptiert die meisten beim **CLI-Aufruf** angegebenen Argumente, so wie dort beschrieben, als Tooltypes des Programmicons. Die mitgelieferten Programmicons enthalten folgende Tooltypes:

(FULL)

ALL

SORT

FORMAT

(FORCE)

(RECURSIVE)

(CMP)

(PATH=<path>)

(FILE=<file>)

(BRIEF)

(COMMENT=<mode>)

(DATE)

(LIB|DEV|IDTYPE)

BUFFER=20

VERSION=0

OUTPUT=CON:8/24/784/320/

GetVer 4.00 Copyright © 1992-1995 by J. Schuchardt/

AUTO/WAIT/SCREEN=Workbench

In Klammern eingeschlossene Tooltypes gelten dabei als inaktiv.

Das OUTPUT-Argument ist hier nur der Übersichtlichkeit wegen in drei Zeilen geschrieben.

¹²³ Handbuch zur Workbench,

¹ "Piktogramme/Arbeiten mit Piktogrammen/Auswahl eines Piktogramms"

² "Piktogramme/Arbeiten mit Piktogrammen/Auswahl mehrerer Piktogramme"

³ "Piktogramme/Arbeiten mit Piktogrammen/Ziehen eines Piktogramms"

1.33 Das Graphical User Interface (GUI)

Das "Graphical-User-Interface" (GUI)

Um die Parameter, die durch die **Tooltypes** vorgegeben werden, vorübergehend zu ändern, ohne die Tooltypes manipulieren zu müssen, genügt ein Doppelklick auf das AppIcon. In dem daraufhin erscheinenden Fenster kann man die Parameter folgendermaßen anpassen (die nachfolgend großgeschriebenen Funktionsnamen entsprechen den Argumenten des **CLI-Aufruf s**):

EINLESEN

Schalterfunktionen

Alle passenden - **ALL**

Nicht-Ausführbare - **FORCE**

Rekursiv - **RECURSIVE**

Vergleichen - **COMPARE**

Durch Anwählen dieser Checkbox werden gleichzeitig zwei Stringgadgets aktiviert. In dem ersten können Sie die Namen der Verzeichnisse eintragen, in denen die zu vergleichenden Dateien gesucht werden sollen. Mehrere Verzeichnisse werden dabei auch hier durch Kommas ohne Leerzeichen getrennt. In dem zweiten Stringgadget können Sie den Namen der Datei eintragen, die eine Liste der

zusätzlichen Vergleichspfade enthält. Die mit einem Fragezeichen beschrifteten Knöpfe am Ende dieser Gadgets, rufen bei Betätigung einen ASL-Filerequester auf, der Ihnen bei der Wahl der Verzeichnisse bzw. der Datei behilflich ist.

Das Anwählen der Checkboxes de-/aktiviert die jeweilige Funktion.

Speicherresidente Dateien prüfen

Quelle - **LIB**

DEV

DTYPE

Durch Klicken auf dieses Cycle-Gadget kann eingestellt werden, ob auf Disk gespeicherte Dateien überprüft werden sollen oder im Speicher befindliche.

Ist letzteres der Fall, dann kann man mit Hilfe der Radio-Buttons bestimmen, welche Art speicherresidenter Dateien verwendet werden soll (Libraries, Devices oder Datatypes).

Die Auswahl der Radio-Buttons mit der Tastatur ist mit der Leertaste möglich.

AUSGEBEN

Schalterfunktionen

Gesamten Text - **FULL**

Sortiert - **SORT**

Formatiert - **FORMAT**

Nur bei Erfolg - **VERSION**

Diese Checkbox ist angewählt, wenn in den Tooltypes eine Mindestversionsnummer vorgegeben ist. Da von der Workbench der Rückgabewert nicht ausgewertet wird, hat diese Funktion hier ausschließlich eine Filterwirkung und entspricht genau dem setzten der Versionsnummer auf Null (VERSION=0). Dadurch wird der Hinweis "...keine Versionsnummer gefunden" nicht ausgegeben.

Verkürzen - **BRIEF**

Datieren - **DATE**

FUNKTION IN DER UNREGISTRIERTEN VERSION NICHT AKTIV!

Das Anwählen der Checkboxes de-/aktiviert die jeweilige Funktion.

Speichern der Versionsnummer als Kommentar

Kommentar - **COMMENT**

FUNKTION IN DER UNREGISTRIERTEN VERSION NICHT AKTIV!

Hiermit wird die COMMENT-Funktion aktiviert und einer der drei möglichen Modi KEEP (Wenn Neu), ADD (Anhängen) und FORCE (Ersetzen) ausgewählt.

Auswahl des Ausgabemediums

Ausgabespezifikation

FUNKTION IN DER UNREGISTRIERTEN VERSION NICHT AKTIV!

Wem weder das, durch das Tooltype OUTPUT vorgegebene Fenster, noch das Standardfenster des Programms als Ausgabemedium zusagt, der kann es hier nach seinen eigenen Wünschen spezifizieren.

Die Ausgabe läßt sich auf jedes verfügbare Gerät (CON:, PRT:, DHx:, DFx:, NIL:, etc.) umleiten. Bei der Umleitung in eine nicht existierende Datei wird diese beim ersten Aufruf erzeugt. Danach, oder wenn die Datei aus einem anderen Grund bereits existierte, wird sie um die neuen Ergebnisse ergänzt.

Die Puffergröße einstellen

Puffergröße (KB) - **BUFFER**

FUNKTION IN DER UNREGISTRIERTEN VERSION NICHT AKTIV!

Die zu benutzende Puffergröße kann entweder direkt in das kleine Stringgadget eingegeben werden, oder durch verschieben des schwarzen Knopfes innerhalb des Slider-Gadgets verändert werden. Dazu klicken Sie auf das kleine, schwarze Kästchen in dem langen, waagerechten Rahmen und bewegen die Maus bei gedrückter Taste nach links oder rechts. Das Kästchen folgt innerhalb des Rahmens Ihren Bewegungen und repräsentiert dabei die Größe des Pufferspeichers in KBytes: ganz links bedeutet 1 KB, ganz rechts hingegen 20 KB. In dem kleinen Stringgadget wird die aktuelle Puffergröße angezeigt.

Alle Funktionen, bis auf die Wahl, welcher Art die im Speicher zu suchenden Dateien sein sollen (s. o. "Einlesen/Speicherresidente Dateien prüfen"), können auch durch Druck auf die Taste, deren Buchstabe unterstrichen ist, eingestellt werden.

Das Menü des Einsteller-Fensters

Information...

Gibt eine kurze Information über das Programm und dessen

Status aus.

Beenden

Entfernt das AppIcon und beendet das Programm vollständig.

Jedem einigermaßen erfahrenen Amiga-User sollte die Bedeutung der Gadgets und Ihre Anwendung aber auch ohne Erklärungen schnell klar werden.

1.34 Woher man Updates bekommt

Wie man ein Update bekommt

Ab der Version 4.00 wird GetVer im Aminet veröffentlicht.

Anwender, die DFÜ betreiben und einen Zugang zum Aminet haben, können die neueste Version aus diesem Netz beziehen.

Aber auch alle anderen Anwender können von diesem Weg der Veröffentlichung profitieren: über die, alle zwei Monate erscheinende, Aminet-CD oder die neue Aminet-Disk-Serie.

1.35 Die Zukunft von GetVer

Was bringt die Zukunft?

Ich spiele mit dem Gedanken, die in dieser Version neu hinzugekommene Funktion der Definition variabler Vergleichspfade mittels Vergleichspfad-Datei, ein wenig intelligenter zu machen. Das soll dann so aussehen, daß ich in der Datei in der ersten Zeile die dateitypische Endung angebe und in der darauf folgenden Zeile den Pfad, mit dem verglichen werden soll, wenn die zu überprüfende Datei die zuvor genannte Endung hat.

Beispiel:

.library

LIBS:;LIBS:compressors

.device

DEVS:

.

.

.

etc.

Was sagen die User dazu? Ist eine solche Funktion eine weitere Vergrößerung von GetVer wert?

1.36 Danksagungen

Danksagungen

Ich möchte es nicht versäumen, einigen Personen meinen Dank auszusprechen, die an der Entwicklung von GetVer, mehr oder weniger maßgeblich, beteiligt waren oder mich bei der Realisierung unterstützt haben:

Volkmar Mai, der schon nach der ersten Veröffentlichung gute Ideen zur Verbesserung des Programms beigesteuert hat.

Maik Wieland, der, neben ein wenig Beta-Testing, freundlicherweise auch DFÜ-Nachrichten von Anwendern früherer Versionen an seiner E-Mail-Adresse entgegen genommen und an mich weiter geleitet hat.

Jürgen Urbanek, dessen Ideen die meisten Änderungen seit der Version 3.00 ausmachen.

Nils Heeren, der für die Übersetzung der Catalog-Datei ins Portugiesische verantwortlich zeichnet und dabei gleich ein wenig Beta-Testing betrieben hat.

Maxon Computer GmbH, für den problemlos laufenden MaxonC++ 3 Compiler.

Und natürlich allen registrierten Anwendern.

1.37 Index

INDEX

A **ALL**

Ausgabespezifikation

B **BRIEF**

BUFIBUFFER

C **CMPICOMPARE**

COMICOMMENT

D **DATE**

DEFIDEFAULT

DEV

DTYPE

F **FILE**

FORCE

FORMAT

FULL

I INFO

L LIB

N NAME

P PATH

R REC|RECURSIVE

REV|REVISION

S SORT

SYSTEM

T Tooltypes

V VER|VERSION
