

VersCheck_deutsch

Michaela Prüß

Copyright © 1991-1999 by Michaela Prüß, All Rights Reserved

COLLABORATORS

	TITLE : VersCheck_deutsch		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Michaela Prüß	August 26, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	VersCheck_deutsch	1
1.1	Inhalt	1
1.2	Copyright	1
1.3	History	2
1.4	Important	3
1.5	Support	3
1.6	Install	3
1.7	Usage	3
1.8	Benutzung von der WB	4
1.9	ThirdPartyProd	5
1.10	ProgInfo	5
1.11	Future	6
1.12	Bugs	6
1.13	CheckGuides	6
1.14	ToDo	8
1.15	Locale	8
1.16	Index	9

Chapter 1

VersCheck_deutsch

1.1 Inhalt

VersCheck Guide (14.5.1999) [Index](#)

Inhaltsverzeichnis

[Copyright, Nutzungsbedingungen Support und Fehlermeldungen Programm-History](#)

[Wichtig, unbedingt Lesen Installation des Programmes](#)

[Benutzung des Programms im CLI Benutzung von der Workbench Verwendete Fremdprodukte](#)

[Über VersCheck \(Hintergründe\) Zukunft und Planungen Bekannte Fehler und Probleme](#)

[Die Vergleiche und Listen!](#)

[Offene Punkte \(To Do\)](#)

[Sprachverwaltung \(Locale\)](#)

1.2 Copyright

VersCheck: Copyright

VersCheck ist Copyright 1998, 1999 by Michaela Prüß

LibGuide, DevsGuide und DTypesGuide sind COPYRIGHT © Heiko Schröder (age@thepentagon.com) 1996-1999

COPYRIGHT © Stuart Kelly (stuart@dkelly1.demon.co.uk) By Stuart Kelly Copyright 1999 ©

Die vorliegende Version ist Freeware.

Das Programm darf nicht kommerziell benutzt werden, ansonsten ist die Nutzung frei. Das Aminet hat die Erlaubnis zur Verbreitung über alle Server, Mirror's und die AmiNet-CD's.

Diese Software wird verbreitet, wie sie ist, ohne eine Garantie jeglicher Art. In keinem Fall, weder direkt noch indirekt ist die Autorin haftbar für Schäden oder Datenverluste die durch die Verwendung dieser Software entstehen. Die Benutzung erfolgt ausschließlich auf eigenes Risiko des Benutzers. Jede Änderung des Archives oder Teilen davon ist verboten. Das Archiv darf auf elektronischem Weg via Mailbox oder Internet verbreitet werden.

[Inhalt Updates, Fehler CheckGuides](#)

1.3 History

HISTORY VersCheck

V 1.10 / 30.05.1999 - Problem bei der Anzeige des Versionsstring mit c:version beseitigt. Die CPU/FPU wird falsch angezeigt. Das ist kein Fehler in VersCheck sondern das Problem tritt beim Compilieren des C-Codes auf. Leider finde ich im Moment die genaue Ursache nicht so das ich keinen Bugbericht an Volker senden kann. Aber ich suche weiter.

V 1.9 / 30.04.1999 - norsk-Catalog zugefügt Option -4 and -5 für MUI und Handler aufgenommen. Interne Erweiterung für Namensvergleich mit * und ? Im \$VER wurden Zeichen über 127 ge- filtert. Jetzt wird 32-127 und alles über 159 akzeptiert.

V 1.8 / 15.04.1999 - Bereinigung der internen V1.7 und weitere Test's der Funktionen für die Class-Guide. Das in der 1.7 noch auskommentierte WB-Interface ist jetzt fertig und einsetzbar. Die aus diesen Erweiterungen resul- tieren Änderungen im Archiv, Installer- script und scripts-Verzeichnis. Kleine Fehlerbeseitigung: Die letzte Zeile der Lokalefiles wurde nicht gelesen, dadurch gab es englische Tasten in einem deutschen Requester.

V 1.7 / 14.04.1999 - Version für die ClassGuide. *** Nur für interne Benutzung ***

V 1.6 / 12.04.1999 - Verzeichnis Script und / oder die Locale-Dateien werden jetzt auch gefunden wenn sie im Startdir des Programmes liegen (PROGDIR:). Die Filetypenerkennung beherrscht jetzt die iff-Header und div. andere Typen. Die Daten habe ich aus Gateway-5 übernommen. Noch immer fehlende getargs-Funktion ist drin (endlich). Bei -f wird jetzt der absolute Pfad angezeigt, nicht nur ein Teilpfad. Fehlerkorrektur: Die Logik für die HTML- Seiten war verdreht (Ja=nicht anlegen, Nein=Anlegen). Jetzt OK.

V 1.5 / 06.04.1999 - Merge-Modus (-m) geändert. Es wird auch dann kein Header angezeigt wenn -1, -2 oer -3 gesetzt sind. Nur mit -s kann der Header geschaltet werden. Soll mehrfache Header vermeiden. - Neue Option -f (fullpath) zugefügt. - Fehler im Installerscript gefixed. Unab- hängig von der Einstellung JA/NEIN wurde das Verzeichnis Scripts immer kopiert. - Anzeige -v(erbose) jetzt ordentlich. - Mehrere kleinere Änderungen.

V 1.4 / 01.04.1999 - Neue Filetypen zugefügt. - Neue Option -n unterdrückt die Requester. - Neue Option -m fügt die neuen Ausgaben an vorhandene Dateien an. Das ist für die CheckGuides. - Installerscripte in deutsch und englisch sind jetzt fertig. Für eine leichtere und fehlerfreie Installation.

V 1.3 / 01.04.1999 - Bessere Aufsplittung des Versionsstrings. Leerzeichen werden jetzt berücksichtigt. - Prüfung einzelner Files zugefügt. Ist ein Verzeichnis angegeben, wird das gesamte Verzeichnis geprüft. Ist ein Filename angegeben, wird die Version des Files angezeigt. Die Angabe von -1/-2/-3 mit einem einzelnen File vergleicht das File mit den CheckGuides. In diesem Fall werden alle Ausgabedateien in der Ram-Disk angelegt. Nur das HTML-Update-Formular wird nicht erzeugt. - Readme-Files durch AmigaGuide ersetzt. - Versionstest der Localedateien zugefügt.

V 1.2 / 27.03.1999 - Requester zugefügt. Zeigt Info's über den Check in Reqtools-Requestern, wenn die reqtools.library installiert ist. Anderenfalls wird nur eine Ausgabe in den CLI getätigt. - HTML-Formular (Update-Formular) integriert. - Bei Angabe von -x werden jetzt .elf, .vos und .ppc zur Vergleichsliste hinzugenommen. Grund: Bisher gab es nur ein oder zwei Library's, die einen Namen wie xxxxxx.elf benutzten. Inzwischen hat z.B. Andreas Kleinert auch Datentypen mit folgender Syntax erstellt: akXXXX.datatype, akXXXX.elf, akXXXX.vos Andere Autoren werden wohl ähnlich handeln. VersCheck ist jetzt auf jeden Fall in der Lage das zu verarbeiten.

V 1.1 / 13.03.1999 - Die Headerinformationen werden jetzt in die Ausgabedateien geschrieben.

***** Es war geplant V1.0 ins Aminet zu senden. Jedoch macht *** ** wuarchive ein newfs (Zitat Homepage: 112 G takes a while). *** ** So ist diese Version nur auf meiner Homepage erschienen. *** *****

V 1.0 / 10.03.1999 - Neue Funktion um das eigene Hauptverzeichnis zu erkennen. - Wegen Probleme mit unbekannter Ursache wurden spezielle Debug-Funktionen integriert (normalerweise sind diese auskommentiert und werden nicht in einem veröf- fentlichten Stand compiliert). - Bugfix: Sprachtexte werden jetzt in jedem Fall korrekt geschlossn. - Funktion -1, -2 und -3 können jetzt ohne Angabe eines Verzeichnisses benutzt werden, es werden dann Standardvorgaben verwendet. Die Defaultwerte sind: -1 = Libs; / -2 = Devs; / -3 = Sys:Classes/Datatypes

V 0.9 / 04.03.1999 - Letzte Beta-Version Version für die 'CheckGuide' zum Test

V 0.4 / 03.03.1999 - Neucompilierung (Intern) - Preversion vbcc 0.7 gegen Releaseversion 0.7 getauscht. Neucompiling von allen Lib's, Library's und Programmen mit neuen, aktuellen Includes und Libs von vbcc. - Umfangreiche Testreihe.

V 0.3 / 28.02.1999 - Pre-Version (Intern) - Interne Sprache ist jetzt englisch. - Texte in Dateien ausgelagert.

V 0.2 / 27.02.1999 - Beta-Version (Intern) - Fehler in Rekursion ExAll() korrigiert (unnötig hoher Stackverbrauch) - Ausgabefiles (Ram:) für Check. - Ausgabe zentralisiert und optimiert. - CPU-Check verändert - Fehler (GURU) in Versionstextaufteilung korrigiert. - Internes Flag (-v) eingefügt (Verbose Info's)

V 0.1 / 23.02.1999 - Erste Version (Intern)

1.4 Important

Wichtige Hinweise:

Das Programm braucht einiges an Stack. Wenn der Stack nicht reicht wird es zum GURU kommen (oder auch nicht, dank MCP). Die benötigte Größe des Stack hängt von der Verzeichnistiefe in 'Libs:' ab. StackSnoop hat als höchstens Wert bei mir 74966 gezeigt. Ich denke mal so 100.000 sollten als Minimum gesetzt sein.

1.5 Support

Support:

Michaela Prüß Berlin

eMail:

Michaela@Flagranti.net

Supportseite im Internet (aktuelle Version und andere Projekte):

<http://www.flagranti.net/amiga>

1.6 Install

Installation:

Das Programm VersCheck kann überall liegen. Für den Versionsvergleich mit den Testlisten muß im aktuellen Verzeichnis das Unterverzeichnis "scripts" mit den Listen vorhanden sein bzw. das Verzeichnis muß sich dort befinden, wo das Programm selber liegt. Beides ist gleichermaßen möglich.

Als reiner Versionslister/-tester wird dieses Verzeichnis nicht benötigt und das Programm kann von überall aus gestartet werden.

Installation der Sprachtexte

1.7 Usage

Benutzung:

VersCheck [[-<0loptlhl>] [<pathname><filename>]] | <?> opt = <123>afmnpvs[x<123>] hlp = ?h 0 = 0 = [-0s <.suffix> -0l <cmplist> -0n <name outfile>]

-1 = Versionsvergleich <dir> mit "scripts/LibList". Fehlt die Angabe <dir>, wird Libs: benutzt. -2 = Versionsvergleich <dir> mit "scripts/DeviceList". Fehlt die Angabe <dir>, wird Devs: benutzt. -3 = Versionsvergleich <dir> mit "scripts/DTypesList". Fehlt die Angabe <dir>, wird Sys:Classes/Datatypes benutzt.

Diese Funktionen wurden für das Prüfsystem von 'age@thepentagon.com' geschrieben.

Werden -1 bis -3 mit einem Filenamen statt einem Verzeichnis gestartet wird nur dieses eine File geprüft. Bei Angabe eines Verzeichnisses wird immer -a gesetzt. Außerdem wird in beiden Fällen -s (ausgenommen Modus -m) gesetzt.

----- Um weitere Listen zu aktivieren ohne das dazu ein Update von VersCheck erforderlich ist wurden diese speziellen Options ge- schaffen:

-0s = Versionsvergleich, freie Definition. Festlegung des Suffix. -0l + " " " . Festlegung der Liste in scripts. -0n + " " " . Festlegung der Namensendung für die Ergebnislisten. (Z.B.: -0n CLASS erzeugt Ram:NewCLASS, Ram:AltCLASS) -0s, -0l, -0n MÜSSEN immer als complete Gruppe angegeben sein. Es gibt keine voreingestellten Default-Werte! -----

-f = Zeit den Pfad zu jeder geprüften Datei. Sinnvoll bei längeren Prüfläufen ohne das man dabei ist.

-n = Unterdrückung der Requester beim Versionstest.

-m = Ausgabedaten anfügen, statt überschreiben für die CheckGuides.

-p = Keine Programme, dafür IFF Bilder. -P = Keine Programme, dafür alle Bilder.

-a = Alle Unterverzeichnisse ebenfalls bearbeiten.

-s = System-Informationen (OS-Version/CPU) anzeigen. Die System-Info zeigt installierte PPC-CPU's und MMU's an, wenn die mmu.library (Thor) bzw. die ppc.library (Phase5) installiert sind.

-v = Zusätzliche Anzeigen für Testzwecke.

-x = WarpOS-Programme und PowerUP-Programme wie Amiga-Programme behandeln. -x1 = Nur PowerUP-Programme wie Amiga-Programme behandeln. -x2 = Nur WarpOS-Programme wie Amiga-Programme behandeln. -x3 = Identisch mit '-x'.

dir = Verzeichnis mit dem der Prüflauf beginnt. Unterverzeichnisse werden nur bei -a berücksichtigt.

file = Einzelnes File von dem die Version angezeigt werden soll.

File oder Dir müssen alternativ benutzt werden. Die Optionen können einzeln oder in Gruppen gesetzt werden. -1sav ist identisch mit -1 -s -a -v oder -1 -sav usw. Nur bei -x gilt die Regel: Muß einzeln oder am Ende stehen, maximal gefolgt von 1,2 oder 3 Die drei Definitionsgruppen (0) müssen jeweils einen einzelnen Aufruf haben.

1.8 Benutzung von der WB

Benutzung von der Workbench

Die Steuerung erfolgt mittels Tool-Types aus dem Icon heraus.

Normalerweise gibt das Programm ja seine Daten im CLI aus. Als Ersatz wird hier kurz nach dem Start ein Fenster geöffnet. Fehlermeldungen werden per Reqtools-Requester ausgegeben.

Tool-Types im Programmicon:

WINDOW=CON:0/10/640/180/VersCheck - Ausgabefenster für die Vergleiche. CON: = Console 0/ = Erste Spalte von links 10/ = Erste Zeile von oben 640/ = Breite des Fensters 180/ = Höhe des Fensters VersCheck = Fenstertitel

DEFAULT_PATH=Sys: Pfadname von wo der Scan startet, Ausgenommen der COMPARE-Funktionen.

ALLDIRS Wenn aktiv, dann werden alle Unterverzeichnisse ebenfalls gescannt.

COMPARE_LIBS Aktiviert den Vergleich mit den Daten der LibGuide.

COMPARE_DEVS Aktiviert den Vergleich mit den Daten der DevsGuide.

COMPARE_DTYPES Aktiviert den Vergleich mit den Daten der DTypesGuide.

COMPARE_MUI Aktiviert den Vergleich mit den Daten der MUIGuide.

COMPARE_HANDLER Aktiviert den Vergleich mit den Daten der HandlerGuide. Es darf immer nur ein COMPARE aktiviert sein.

MERGE_OUTPUT Ergebnisdateien in der Ram-Disk (falls vorhanden) nicht überschreiben sondern neue Daten anfügen.

REQUESTER Anzeige des Ergebnisses bzw. Abfrage HTML-Generierung per Requester (entfernen wenn nicht gewünscht).

SHOW_FULLPATH Pfadnamen komplett anzeigen.

SHOW_SYSTEMINFO Systeminformationen anzeigen.

ACCEPT_ELF PPC-Programme (.elf) wie normale Amigaprogramme behandeln und in den Lauf einbeziehen.

ACCEPT_WOS WarpOS-Programme (.ehf) wie normale Amigaprogramme behandeln und in den Lauf einbeziehen.

CMP_SUFFIX=.library Legt ein Suffix fest das geprüft werden soll (ClassesGuide).

CMP_LIST=LibsList Name der Vergleichsdatei (ClassesGuide)

CMP_OUTNAME=Libs Namenszusatz Ergebnisdateien (ClassesGuide) Durch Benutzung des CMP_OUTNAME wird der Vergleich aktiviert. Die beiden anderen Tool-Types wirken auch auf die anderen Vergleichsscans.

ONLY_PICTURES Nur Bilder suchen.

ONLY_IFFPICTURES Nur IFF-Bilder suchen.

VERBOSE Zusätzliche Anzeigen aktivieren.

1.9 ThirdPartyProd

Fremdprodukte

Das Programm nutzt ggf. einige Fremdsoftware:

- 1.) Befehl VERSION von AmigaDOS. VORSICHT mit den sogenannten "verbesserten Ersetzungen". Meist lesen diese nur den \$VER-String, also sehr viel weniger als das Originalprogramm.
- 2.) ppc.library (sofern vorhanden) zur Ermittlung der Daten von installierten PowerPC-Prozessoren. Original von Phase-5. (Die Emulation von Frank Wille ist noch NICHT getestet!).
- 3.) mmu.library V0.24+ (sofern vorhanden) zur Ermittlung der MMU-Daten. Original von Thor, siehe Aminet. PPC-Besitzer lesen bitte den Anhang **Bekannte Fehler**.
- 4.) reqtools.library für Requester. Copyright by © Nico François and Magnus Holmgren.

Fehlen die angegebenen Libs wird der jeweilige Check ausgelassen. Falls die mmu.library nicht läuft (PPC-Systeme), wird die Anzeige ebenfalls einfach weggelassen. Ist der Befehl Version nicht installiert, werden diese Daten nicht zum Vergleich benutzt was die Erfolgsquote senkt. Falls die reqtools.library nicht installiert ist, erscheinen die Hinweise nur im CLI. Die Abfrage der HTML-Seite wird mit JA angenommen und eine HTML-Update-Seite generiert.

Diese AmigaGuide wurde mit Guided von Linus McCabe erstellt.

1.10 ProgInfo

Programmfunktion und Hintergrund

Das Programm selber sucht nach dem "\$VER:"-String und zerlegt ihn. Außerdem führt es den Befehl VERSION aus und zerlegt dessen Ausgabe ebenfalls.

Es wird nicht sortiert, sondern direkt das Filesystem gescannt und dann jeweils sofort verglichen (es wird ExAll() verwendet!). Dies spart Zeit, besonders bei den Systemen mit sehr vielen Dateien.

Diese Daten sind sicherlich für diverse Zwecke nutzbar. Aber der eigentliche Verwendungszweck ist der Vergleich mit den Daten der Libs-, DT- und Device-Checks.

Ich finde die Idee dieser Test's gut und sehe auch, wieviel Zeit und Energie darin investiert wird, das ganze zu Katalogisieren und zu Pflegen.

Es gibt aber ein Problem, das recht komplex ist und sich nur sehr schwer mit CLI-Batches, ARexx oder ähnlichen Dingen lösen läßt. Gerade bei älteren Lib's sind häufig Probleme vorhanden. Versionsnummern die z.B. "1.-1" lauten, falsche oder merkwürdige Angaben im Namensbereich der Version und zu guter letzt noch Versionsnummern die mal x.x, dann V x.x oder Vx.x lauten.

Dieses Programm "versucht" diese Daten identifizieren und dann das Ergebnis mit den Vergleichsdateien zu prüfen. "Versucht" deshalb, weil ich natürlich erstmal nur mit meinen Daten testen kann. Die Erkennung wird sicherlich nicht gleich perfekt sein, aber das wird mit jeder Version verbessert.

Sollte es zu ganz unsinnigen Anzeigen kommen, bitte eine Kopie des Files an mich senden. Wenn es sich um geschützte Software handelt, dann bitte vorher Kontakten, denn dann ist es besser nur einen Ausschnitt zu senden da ich keine Raubkopien haben möchte.

1.11 Future

Zukunft

Neben Verbesserung im Vergleich und laufender Wartung werden gerne auch Wünsche für andere Einsatzzwecke berücksichtigt.

Das Programm DVC, das vor einiger Zeit diese Tests durchführte, hat einige Funktionen, die für mich unverzichtbar geworden sind. Z.B. das man Dateien auf ihre Version testen kann und ein Requester gleich nachfragt ob man sie installieren möchte. Leider ist DVC längst nicht mehr auf dem laufenden Stand. Zudem fehlt mir z.B. eine Möglichkeit neuere Dateien sofort zu installieren (vorherige Version evtl. Sichern) und ebenso, bei alten Versionen keine Abfrage zu bringen. Daher soll eine Copy-Funktion, ähnlich DVC, hinzukommen die diese Dinge verbessert.

Die reine Erkennung der Dateien an ihrer Endung ist ein Schwachpunkt von DVC. So wird das ReadMe xxx.library von DVC nach Libs: kopiert. Nicht so sinnvoll.

Weitere Dateitypen sollen erkannt werden.

Ein Brainfile für die Copyfunktion ist in Vorbereitung. Eine Anti-Liste mit aktuellen Warnungen (z.B. Viren) ist auch geplant.

1.12 Bugs

Bekannte Fehler

Die aktuelle mmu.library arbeitet mit PPC-System nicht zusammen und kann auf diesen daher keine Anzeige bringen. Laut Autor ist das Problem mangelnde Kooperationsbereitschaft der Firma Phase-5 (Info's über die MMU-Verwaltung auf PPC-Systemen). Das wird sich aber wohl bald ändern (Dank Sam Jordan/Haage & Partner).

Probleme mit anderen Programmen

Beim Test kam es mehrfach zum Rechnerstillstand bei umfangreicheren Verzeichnissen. Gründe:

- 1.) Output in einem KingCon-CLI-Fenster (CON: und AUX: von KingCon). Ursache unklar, gleiches passiert auch bei Benutzung von vim und anderen Programmen, war aber nicht immer so. Vermutlich ein Konflikt zwischen KCon und einer nicht identifizierten, anderen Software. Abhilfe: Tritt nicht auf, wenn der Verbose-Modus (-v) aktiv ist.
- 2.) Stillstand beim Versionstest mpegavideo.datatype. Ursache ist C:Version. Aus unerfindlichen Gründen stürzte der Befehl hierbei ab. Ist aber sehr wahrscheinlich ein Hardware-Problem auf meinem Rechner. Abhilfe: Tritt nicht auf, wenn das Verzeichnis statt auf der Platte in der Ram-Disk liegt. Nach der Installation von WarpOS und einem decrunch aller Files in Sys: war das Problem weg. Ich weiß nicht, warum. Aber ich denke, es lag an einigen gecrunchten Files. WARUM CRUNCHET JEMAND ÜBERHAUPT? Festplatten sind inzwischen extrem billig und gecrunchte Files machen nur das System langsamer.

1.13 CheckGuides

Anhang für die Vergleichslisten

(siehe Ende dieser Seite!)

- LibGuide - DTypesGuide - DevsGuide - ClassesGuide

Der Vergleich:

Jede Datei wird zunächst anhand der "Magic-Number" identifiziert. Die "Magic-Number" sind die ersten 4 Bytes (evtl. 8, hier aber belanglos), einer binären Datei. Daran kann man identifizieren, was es ist. Z.B. ein Amigaprogramm, genau wie ein MS-Dos-Programm usw.! Da sowohl Libs, als Devs und DT's den Magic-Code eines ausführbaren Programmes haben, werden im Moment nur Dateien geprüft, die eine solche Amigaprogrammkennung tragen. Also eine Textdatei "Libs:Irgendwas.Library" wird garnicht erst geprüft. Aus diesem Grund werden auch ELF/EHF (PowerUP/WarpOS) übergangen, wenn sie nicht per '-x'-Option aktiviert werden.

Je nach Check wird auch noch ".Library", ".Device" oder ".Datatype" geprüft. Bei Bedarf könnte das natürlich ausgedehnt werden (z.B. MUI '.mcc' und '.mcp'). Aber diese Anforderung behalte ich dem Autor der CheckGuides vor. Lediglich -x bewirkt hier das zusätzlich .wos, .elf und/oder .ppc aktiviert werden können.

1.14 ToDo

ToDo (CheckGuides):

- Versionsnummer alternativ scannen [suche '(DATUM)'] für die ganz mies programmierten Teile. Nur ein Fall bekannt.
- Versionsunsinn "%ld" etc. identifizieren. Nur ein Fall bekannt.

ToDo (VersCheck):

- Weitere Filetypen erkennen/anzeigen.
- Copy-Funktion mit Requester/Suffix-Datei/Sicherungen u.v.m.
- Alles, was die Praxis als sinnvoll erscheinen läßt.

1.15 Locale

Information über locale in VersCheck:

Das Programm benutzt keine wirklichen Locale-Dateien. Es hat eine eigene Sprachverwaltung.

Das Programm wurde für zwei unterschiedliche Anwendungen geschaffen. Erstens, um in den CheckGuides den Vergleich zwischen deinem Rechner und den Listen durchzuführen. Zweitens (in Zukunft), um Vergleiche und Updates von Programmen und Systemfiles, ähnlich DVC, zu ermöglichen. Allerdings mit einigen weiteren Funktionen und einem konfigurierbarem Suffixfile.

Das Programm ist im AmiNet verfügbar, und es wird mit den CheckGuides ausgeliefert. Was, wenn eine Version bereits installiert ist und eine andere bei den Guides liegt wobei die beiden Version in den Texten nicht zusammenpassen?

Um dieses Problem zu vermeiden und die Anpassung an weitere Sprachen zu vereinfachen (kein besonderes Wissen notwendig), habe ich die Programmtexte in einfache Ascii-Dateien ausgelagert. Zum Ändern oder Anpassen ist lediglich ein Editor wie Ed, Edit, GoldEd, CygnusEd, emacs, vi usw. nötig. Man sollte sich lediglich zuerst die Programmausgaben ansehen, damit man ein Gefühl dafür bekommt und nicht eine sehr merkwürdige Ausgabe produziert.

Zeile 1 bis 3 sind Kommentare. Bitte so belassen! Die Zeilen ab 4 sind die Programmausgaben von VersCheck. VersCheck sucht zunächst im aktuellen Verzeichnis sowie dem Programmstart-Verzeichnis nach einer Datei mit dem Namen VersCheck.<Locale>.

Dabei wird der Wert für <Locale> aus ENV:Language gelesen. Dies sollte von der s:startup-sequence zur Bootzeit korrekt gesetzt worden sein.

Wird kein passendes File gefunden, dann sucht VersCheck das File Locale:catalogs/<Locale>/VersCheck.catalog Für eine dauerhafte Installation sollte das richtige File entsprechend kopiert werden. Für die CheckGuide können die Files bleiben, wo sie sind.

Zur Zeit sind nur drei Sprachfiles vorhanden:

VersCheck.deutsch (Locale:catalogs/deutsch/VersCheck.catalog) VersCheck.english (Locale:catalogs/english/VersCheck.catalog)
VersCheck.norsk (Locale:catalogs/norsk/VersCheck.catalog)

Falls Du eine Anpassung an eine weitere Sprache vornimmst, bitte schick eine Kopie an mich, damit ich die Übersetzung in den Release-Stand übernehmen kann.

eMail: Michaela@Flagranti.Net

Niemals sollten Zeilen eingefügt oder gelöscht werden. Das führt zu falschen Anzeigen!

Die maximale Länge pro Zeile ist 70, mehr wird abgeschnitten.

Leere oder fehlende Zeilen werden durch den internen, englischen Text ersetzt

Übersetzungen:

english : von Michaela Prüß (28.02.1999) deutsch : von Michaela Prüß (28.02.1999) norsk : von Roger Hågensen (20.04.1999)

1.16 Index

B

[Bekannte Fehler und Probleme Bekannte Fehler](#)

[Benutzung des Programms im CLI Benutzung von der Workbench](#)

C

[CheckGuides Copyright, Nutzungsbedingungen](#)

D

[Die Vergleiche und Listen!](#)

I

[Index Inhalt](#)

[Installation der Sprachtexte Installation des Programmes](#)

O

[Offene Punkte \(To Do\)](#)

P

[Programm-History](#)

S

[Sprachverwaltung \(Locale\) Support und Fehlermeldungen](#)

U

[Updates, Fehler](#)

V

[Verwendete Fremdprodukte](#)

W

[Wichtig, unbedingt Lesen](#)

Z

[Zukunft und Planungen](#)

Ü

[Über VersCheck \(Hintergründe\)](#)
