ZPCheckZC

COLLABORATORS			
	TITLE : ZPCheckZC		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 ZPCheckZC **1**

1.1 main . . . 1
1.2 system_requirements . . . 2
1.3 parameters . . . 2
1.4 usage_example . . . 4
1.5 online_help . . . 5
1.6 arexx . . . 5
1.7 history . . . 6
1.8 mui . . . 7

Chapter 1

ZPCheckZC

1.1 main

ZPCheckZC überprüft ZCONNECT(R) Datenpuffer. Ein Rückgabewert von 0 (RETURN_OK) signalisiert einen augenscheinlich korrekten Puffer, der Wert 5 (RETURN_WARN) steht für einen fehlerhaften Puffer. Dies macht das Einbinden in OS-Skripten einfach. ZPCheckZC kann außerdem korrekte Teile fehlerhafter Puffer rekonstruieren. Zur größeren Bequemlichkeit ist auch eine grafische MUI-Benutzeroberfläche vorhanden. Das Programm kann auch über ARexx genutzt werden.

Systemvoraussetzungen

Online-Hilfe zum Programm

Parameter und Tooltypes

Entwicklungsgeschichte

Beispiel eines Programmlaufes

Magic User Interface

Liste der ARexx-Kommandos

Ralph Seichter Telefon +49-2233-70293
Deutscher Ring 6 zodiac@darkness.gun.de
D-50354 Hürth
Bundesrepublik Deutschland

Stuntz.

DISCLAIMER: THIS MATERIAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE USE, RESULTS, AND PERFORMANCE OF THIS MATERIAL IS ASSUMED BY YOU AND IF THE PRODUCT SHOULD PROVE TO BE DEFECTIVE, YOU ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR OTHER REMEDIATION. Oder auf gut Deutsch: Wenn etwas in die Hose geht, ist der Benutzer und niemand sonst angeschmiert! :)

1.2 system_requirements

Systemvoraussetzungen
==========

OS 2.04 (Kickstart 37.175) oder besser. Die Oberfläche benötigt außerdem das

Magic User Interface
(MUI) von Stefan Stuntz. Wenn Sie die integrierte

Online-Hilfe
nutzen wollen, benötigen Sie außerdem die amigaguide.library.

1.3 parameters

Parameter und Tooltypes
==========

Die folgenden Parameter können Sie entweder beim Aufruf in einer Shell angeben oder als Icon-ToolTypes einstellen. Beachten Sie bitte, daß auch im Falle eines Shell-Starts die Icon-ToolTypes ausgewertet werden, damit Sie über die ToolTypes ihre bevorzugten Standardwerte eintragen können.

```
ZPCheckZC FILE/K,PATH/K,PUBSCREEN/K,MATCHABS/K,MATCHBET/K,MATCHEMP/K,
          OUTPUT/K,BUFFERS/K/N,SPLIT/K/N,SPLITKB/K/N,ANDMATCH/S,GUI/S,
          NOCHKABS/S,NOCHKEDA/S,NOCHKMID/S,PICKY/S,REPAIREOL/S
```

FILE ist der Name der zu überprüfenden Datei. Falls sie keinen Namen angeben, öffnet ZPCheckZC einen Dateiauswahl-Requester. Die Option PICKY bestimmt, welche Fehler als "fatal" eingestuft werden. Normalerweise wird versucht, so lange wie möglich weiterzuarbeiten, aber wenn PICKY angegeben wird, wird jeder Fehler als fatal interpretiert und die Bearbeitung sofort unterbrochen. Ansonsten gibt es nur einen fatalen Fehler, nämlich einen völlig unsinnigen oder fehlenden Wert für die Kopfzeile "LEN:".

BUFFERS legt fest, wieviele 1024-Byte-Blöcke im RAM reservert werden

(Standardwert 64). Wenn PATH (Zielpfad) spezifiziert wird, wird versucht, den Puffer zu reparieren. Man sollte übrigens nie PATH und PICKY zusammen verwenden, da es keinen Sinn macht, die Reparatur schon beim ersten Fehler zu unterbrechen. Falls das aktuelle Verzeichnis Ausgabepfad sein soll, muß PATH="" angegeben werden.

Die Angabe von SPLIT=<n> hat die Generierung einer Reihe von Dateien zur Folge (datei.001, datei.002, etc.), die jeweils höchstens <n> Nachrichten umfassen. Normalerweise wird nur eine Ausgabedatei für korrekte Nachrichten (datei.001) und eine für defekte Nachrichten (datei.BAD) erzeugt. Mittels SPLITKB kann man einen Schwellenwert für die Dateigröße definieren. So wird SPLITKB=100 beispielsweise eine neue Ausgebedatei angelegen, sobald die aktuelle größer als 102400 Bytes wird. SPLIT und SPLITKB können kombiniert werden: Bei z.B. SPLIT=30 SPLITKB=70 wird eine neue Ausgabedatei geöffnet, wenn die aktuelle 30 Nachrichten enthält oder mehr als 70 KB groß wird. Es ist zu beachten, daß SPLITKB niemals innerhalb einer Nachricht trennt.

MATCHABS, MATCHBET und MATCHEMP können benutzt werden, um Muster für den Absender, Empfänger und Betreff von Nachrichten zu definieren. Alle Muster des AmigaOS, "Wildcards" eingeschlossen, werden unterstützt. Wenn mehr als ein Muster spezifiziert wird, wird eine Nachricht erkannt, wenn sie auf wenigstens ein Muster paßt (logisches ODER). Durch ANDMATCH legt man fest, daß sie auf alle Muster passen muß (logisches UND).

OUTPUT dient zur Spezifikation eines alternativen Ausgabefensters oder einer Ausgabedatei. OUTPUT="CON:////Titel/WAIT/SCREEN%s" würde die Ausgabe in das angegebene Konsolenfenster umleiten (ein %s Platzhalter wird zur Laufzeit mit dem Namen des Screens belegt, auf dem sich das GUI befindet). Durch den Parameter GUI ist es möglich, auch bei einem Start aus einer Shell heraus die grafische Benutzeroberfläche zu verwenden. Sie können durch PUBSCREEN den Namen eines Public Screens bestimmen, auf dem dann das Fenster erscheinen soll.

Die Parameter NOCHKABS, NOCHKEDA und NOCHKMID schalten die Überprüfung der jeweiligen Kopfzeilen für den Absender, das Erstellungsdatum und die Nachrichte-ID aus. WARNUNG: Wenn diese Testkriterien abgeschaltet werden, läßt ZPCheckZC möglicherweise Nachrichten als "in Ordnung" durchgehen, die in Wirklichkeit fehlerhaft sind. VERWENDEN SIE DIESE PARAMETER DAHER MIT GROESSTER VORSICHT! Wenn ZPCheckZC Nachrichten als fehlerhaft beanstandet, entsprechen sie nicht dem ZCONNECT-Standard und sollten daher keinesfalls ins Netz eingespielt werden. Was sie privat damit anfangen, ist Ihre Sache, aber schicken sie bitte keine fehlerhaften Daten ins Netz.

Normalerweise erwartet ZPCheckZC, daß jeder Zeile im Nachrichtenkopf mit einem CR-LF-Paar endet. Fehlt diese Sequenz, wird ein Fehler gemeldet und die entsprechende Header-Zeile ignoriert. Mit REPAIREOL können sie diese defekte Zeilen reparieren lassen (sofern dies noch möglich ist).

ZPCheckZC wird teilweise Nachrichten als fehlerhaft beanstanden, bei denen auf den ersten Blick keine Unkorrektheit zutage tritt. Beispiele für derartige Kopfzeilen sind:

```
ABS: puhvogel@foo_bar.ZER.sub.org
MID: 1994Jun28.151046@tferi4
MID: uN/XOMD21CaszE@robot.gaga.do.main
MID: j48dlnglJD3@!wow.bag.ger.edu
```

Im ersten Fall wurde ein unzulässiges Zeichen im Domain-Teil der Adresse benutzt; '_' ist nicht gestattet. Fall Nummer zwei hat keine vollständige Domain (mindestens ein '.' muß im Domain-Teil vorhanden sein). Im dritten Fall ist der Schrägstrich '/' im lokalen Teil der ID unzulässig. Alle drei Fälle würden wahrscheinlich nicht zu Problemen führen, aber eben nur wahrscheinlich -- die Angaben entsprechen nicht den Vereinbarungen. Auch im letzten Fall gibt es einen Fehler, das '!' im Domain-Teil. Dies ist leider ein typischer Fall von Konfigurationsfehler, in dem ein Sysop einem Point den Namen "!wow" gegeben hat, ohne darauf zu achten, daß er ein unerlaubtes Symbol verwendet. Wer sich für die Details interessiert, dem sei hier die Lektüre der ZCONNECT-Dokumentation empfohlen.

1.4 usage_example

```
Beispiel eines Programmlaufes
=============================

        [1] Shell $ ZPCheckZC FILE=dh2:data/foo PATH=t: BUFFERS=128
        ZPCheckZC 36.14 (28.9.94) Copyright © Ralph Seichter
        19904-20083: missing "BET:"
        21201-21472: missing header data
        29857-30278: missing "ABS:" "MID:"
        41468-42005: invalid "EDA:"
        42300-43001: futuristic "EDA:"
        46743-46997: missing line terminator
        48002-48503: invalid length, exiting
        File "dh2:data/foo" has errors.
```

Wie man sieht wurde ZPCheckZC ohne die Option PICKY gestartet, sonst wäre die Bearbeitung schon nach dem ersten Fehler unterbrochen worden. In jeder Statuszeile ist zunächst das Offset-Intervall aufgeführt, in dem der Fehler aufgetreten ist (Offset = Anzahl der Bytes vom Dateianfang). Danach folgt die Fehlerursache im Klartext.

Der erste Fehler dieses Beispiels sagt aus, daß zwischen Byte 19904 und 20083 der Datei "foo" der Kopfeintrag "BET:" fehlt (dieser Eintrag darf in keinem Fall fehlen). Der zweite Fehler meldet, daß für einen der wichtigen Kopfeinträge keine Daten angegeben wurden. Fehler Nummer drei entspricht Nummer eins, nur daß hier "ABS:" und "MID:" fehlen.

Fehler vier und fünf melden ungültige Einträge für das Erstellungsdatum der Nachricht. Der erstere der beiden sagt aus, daß ein völlig unsinniger Eintrag vorliegt, während der letztere eine Nachricht bemängelt, die laut Eintrag angeblich aus der Zukunft stammt. Das weist meist auf eine falsch eingestellte Systemuhr hin, überprüfen Sie bitte zunächst Ihre eigene. Die folgende Meldung gibt an, daß mindestens eine Zeile des Nachrichtenkopfes nicht wie vorgeschrieben mit einem CR-LF-Paar beendet wurde. Der letzte Fehler schließlich bedeutet, daß die Länge der aktuellen Nachricht nicht in Ordnung ist. Dies ist derzeit der einzige Fehler, der ZPCheckZC zum Abbruch zwingt (fehlt der Eintrag ganz, wird natürlich auch unterbrochen).

Wenn Sie ZPCheckZC mit der grafischen Oberfläche gestartet haben, wird als Ausgabekanal ein CON: Fenster geöffnet. Sofern der Switch WAIT benutzt wird, müssen sie das Fenster von Hand schließen. Das ist sinnvoll, da sonst die Ausgabe nur für einen kurzen Moment sichtbar ist und dann verschwindet,

denn der Ausgabekanal wird natürlich bei Testende geschlossen. Sie können die Ausgabe aber selbstverständlich auch in eine Datei umleiten.

1.5 online_help

```
                Online-Hilfe zum Programm
=========================
```

Sofern Sie das Dokument ZPCHeckZC.guide im Suchpfad von AmigaGuide oder im gleichen Verzeichnis wie das Programm haben, können sie jederzeit durch das Drücken der Help-Taste die Beschreibung aller
Parameter
erhalten, falls
die amigaguide.library bei Ihnen installiert ist. Es kann vorkommen, daß die Hilfsseite erst nach dem zweiten Druck auf die Help-Taste erscheint; es scheint sich dabei um ein Problem der amigaguide.library zu handeln.

1.6 arexx

```
                ARexx-Kommandos
===============
```

ZPCheckZC verarbeitet in der vorliegenden Version neben den von jeder MUI-Applikation unterstützten Standard-Kommandos die folgenden Befehle (Sie erhalten eine Liste, wenn sie den ARexx-Befehl "HELP <Datei>" an ZPCheckZC schicken):

CHECK: Startet die Überprüfung mit den aktuellen Einstellungen. Falls Fehler im Puffer auftreten, wird RETURN_WARN (5) zurückgeliefert, ansonsten wird RETURN_OK (0) in der ARexx-Variablen RC zurückgegeben.

Allgemein gilt für RC, daß RETURN_OK (0) "kein Fehler" signalisiert. Werte größer als Null liefert das Programm, Werte kleiner als Null werden von
MUI
geliefert. Derzeit sind nur die folgenden MUI-Fehler definiert:

- -1 Unzulässige Kommando-Definition (ein interner Fehler).
- -2 Nicht genügend Speicher.
- -3 Unbekanntes ARexx-Kommando.
- -4 Unzulässige Parameter für ARexx-Kommando.

GET: Aktuelle Einstellungen als String in RESULT zurückliefern.

SAVE: Aktuelle Einstellungen im Icon speichern.

SET (Template wie beim CLI-Aufruf): Einstellen der
Parameter
. Falls Sie
Strings wie Dateinamen oder -Pfad oder numerische Werte weglassen, bleiben sie unverändert bestehen. Alle anderen Einstellungen müssen Sie nach Wunsch

angeben (damit beispielsweise NOCHKEDA nicht abgeschaltet wird, wenn es vorher angeschaltet war). Auf diese Weise kann man z. B. mittels "SET" ohne Parameter alle "Work Modes" Checkbox-Gadgets abschalten. SET PUBSCREEN gibt Ihnen die Möglichkeit, das GUI auf einen anderen Screen zu bringen.

1.7 history

Entwicklungsgeschichte
======================

36.1 (02.02.94): Erste öffentliche Freigabe.

36.2 (09.02.94): BUFFERS und Reparaturfunktion via PATH hinzugefügt.

36.3 (16.02.94): Fehlerkorrektur: V36.2 meldete leere ROT-Einträge als fehlerhaft, was natürlich nicht korrekt war.

36.4 (22.02.94): Fehlerkorrektur: V36.3 behandelte Nachrichten der Länge 0 als fehlerhaft und stoppte die Bearbeitung. V36.4 arbeitet weiter und meldet nur eine Warnung. Leere Ausgabedateien werden gelöscht.

36.5 (19.04.94): SPLIT/K/N und den Dateiauswahlrequester hinzugefügt.

36.6 (25.04.94): Umstellung auf schnellere Datei-E/A-Funktionen.

36.7 (26.04.94): Fehlerkorrektur: Frühere Versionen konnten ziemlich aus dem Tritt kommen, wenn die Eingabedatei unerwartet früh endete.

36.8 (03.05.94): MATCHABS/K, MATCHBET/K, MATCHEMP/K, und SPLITKB/K/N hinzugefügt. Nachrichten ohne Rumpf werden nicht mehr gemeldet, da dies zu Verwirrung bei einigen Benutzern führte.

36.9 (06.05.94): ANDMATCH/S und die grafische Benutzeroberfläche wurden hinzugefügt (und die deutschsprachige Dokumentation).

36.10 (19.05.94): OUTPUT/K zu ZPCheckZC hinzgefügt. Das GUI kann nun auch von der Workbench gestartet werden.

36.11 (01.07.94): NOCHKABS/S, NOCHKEDA/S und NOCHKMID/S hinzugefügt. Checker und Oberfläche sind jetzt eine Einheit. Umstellung auf MUI.

36.12 (12.07.94): Fehler beim Testen von Netzadressen behoben (ZPCheckZC meldete beim Auftreten von '@' im Realnamen irrtümlich den Lokalteil der Adresse als fehlerhaft). Die im GUI eingestellten Vorgaben können jetzt als Icon-Tooltypes gespeichert werden. ARexx-Unterstützung. Der Parameter FILE wurde im Template durch FILE/K ersetzt.

36.13 (28.08.94): Neuer Parameter PUBSCREEN/K.

36.14 (28.09.94): Im GUI wird jetzt bei der Angabe des Ausgabefensters ein %s Platzhalter für den Namen des Public Screens unterstützt.

36.15 (21.11.94): Es wird nun peinlich genau darauf geachtet, daß jede Header-Zeile mit einem CR-LF-Paar endet.

36.16 (24.11.94): Defekte Header-Zeilen können jetzt mittels REPAIREOL repariert werden.

1.8 mui

This application uses

MUI - MagicUserInterface

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US$ 20.-

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY