QL Today DEUTSCH

Jahrgang 4
Ausgabe 6
März/April
2000

ISSN 1432-5446

Das Magazin über QL, QDOS, Sinclair Computer, SMSQ...

Mit dem QL ins "Web"?

Wir nähern uns!

Mehr in der nächsten Ausgabe!

Inhalt

2 Kleinanzeigen
2 Impressum
3 Editorial
4 & Neuheiten
5 Programmieren mit den Basic-Erweiterungen von EASYPTR - Teil 2
Eine Serie von Wolfgang Uhlig
9 Werbung Jochen Merz Software
10 Eindhoven 2000
12 Werkzeug zur Umwandlung eines gewöhnlichen QL-Textes ins HTML-Format
Gerhard Plavec
16 Sein oder nicht sein... läuft er noch?
Jochen Merz
17 Q40 News
Claus Graf
17 JMS - In letzter Minute

Kleinanzeigen

Da QL Today eine der wichtigsten Quellen für QL-Neuigkeiten ist, werben auch die meisten QL-Händler hierin. Nun stellt sich natürlich die Frage, warum nur QL-Händler inserieren dürfen, warum nicht auch QL-User, die ihre eigenen Programme, Hardware oder Entwicklungen verkaufen möchten.

Anfangs konnte man selbstgeschriebene Software von Clubs vertreiben lassen, beispielsweise dem deutschen QL User Club. Aber warum soll man es denn nicht selbst vermarkten können? Es gibt hier keinen Unterschied zwischen privaten und kommerziellen Anzeigen, nur mit dem QL und Drumherum sollte es schon etwas zu tun haben.

Bis zu 50 Worte im englischen oder deutschen Teil kosten DM 5,- (oder 3 Internationale Antwortscheine), bis zu 100 Worte kosten DM 10,- (oder 6 Internationale Antwortscheine). Soll die Anzeige im deutschen und englischen Teil erscheinen, ist der Preis zu verdoppeln.

Verkauf

SINCLAIR QL deutsch, komplett mit Bernsteinmonitor. TrumpCard 896Kb, FLP2 Level-Chip, 2*3,5"-Floppy, 122Mb-HardDisk + QuBide.
Preis 199,- DM VB
Andreas Schulz
Frankampstr. 170
45891 Gelsenkirchen
Tel.: 0209/799098

QL Today DEUTSCH

ISSN 1432-5446

Herausgeber:
Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Deutschland

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
Email: QLToday@j-m-s.de

QL Today erscheint alle zwei Monate, Erscheinungsdatum der ersten Ausgabe ist der 15. Mai. Das Abo beginnt mit der aktuellen Ausgabe zum Zeitpunkt der Bestellung. Preisinformationen und Antragsformulare sind bei Jochen Merz Software erhältlich.

Ihre Kommentare, Vorschläge und Artikel sind herzlich willkommen. SIE machen **QL Today** möglich. Wir verbessern das Magazin wo immer möglich, um Ihren Vorstellungen gerecht zu werden. Artikel sollten auf 3,5" Diskette (DD oder HD) eingeschickt werden. Das Format sollte ASCII, Quill oder Text87 (Druckertreiber angeben!) sein. Bilder sollten im _SCR-Format geschickt werden, GIF und TIF ist auch möglich. BITTE senden Sie auch einen Ausdruck der Bilder. Wenn ein Bild an einer bestimmten Stelle plaziert werden soll, geben Sie es bitte auch an. Natürlich können auch alle Bilder/Artikel auf elektronischem Weg übersandt werden, also in die Box oder per E-Mail.

Redaktionsschluß für Artikel und Werbung:

Ausgabe 1:	30. April
Ausgabe 2:	30. Juni
Ausgabe 3:	30. August
Ausgabe 4:	30. Oktober
Ausgabe 5:	30. Dezember
Ausgabe 6:	28. Februar

QL Today behält sich vor, eingeschicktes Material nicht zu veröffentlichen. **QL Today** ist unter keinen Umständen für die Richtigkeit der abgedruckten Artikel und Programmen haftbar, ebenso nicht für aus fehlerhaftem Material hervorgerufene Datenverluste, Unbenutzbarkeit oder ähnliche Probleme, die aus Artikeln in **QL Today** herrühren könnten. Die Meinung in diesem Magazin entspricht der des jeweiligen Autors und nicht notwendigerweise der des Herausgebers.

Dieses Magazin unterliegt dem Copyright und jegliches hierin veröffentlichte Material darf nicht ohne schriftliche Erlaubnis von **QL Today** reproduziert, übersetzt oder sonstwie verbreitet werden. Allen Copyrights und Trademarks wird hiermit Rechnung getragen.

Editorial

Jochen Merz

Liebe Leser,

diese Zeilen schreibe ich auf meinem Laptop mit QPC2 während ich von Oostende nach Dover düse. "Düse" ist genau das richtige Wort, hier fährt ein Katamaran - richtig schön schnell, wirklich beeindruckend. Wer noch niemals mit so einem Teil gefahren ist, sollte es auf seinem nächsten England-Trip mal ausprobieren. Preiswerter als Euro-Tunnel und auf jeden Fall schneller und interessanter als die Fähren.

Stichwort "nächster England-Trip": DAS QL-2000-Treffen hat nun ein definitives Datum, das Wochenende am 14./15. Oktober. Alle QLer sollten kommen, es ist ein wirklich großes Zwei-Tage-Treffen. Wer noch nie in England war, sollte diese Chance nicht verstreichen lassen - wer weiß, wann mal wieder so etwas auf der Tagesordnung steht. Selbst eine größere Zahl Amerikaner fliegt für das Treffen aus Amerika nach England. Veranstaltungsort ist Portsmouth - an der schönen Südküste Englands zwischen Brighton und Southampton. Hier dürfte auch zu dieser Zeit noch mildes Klima herrschen. Also - wer Zeit hat und sowieso schon immer mal nach England wollte: Nichts wie los! Wer nicht alleine fahren möchte, kann ja mal in die QL User Liste schreiben (Email an **ql-users@nvg.ntnu.no**, Betreff **subscribe**). Sicherlich ist er nicht alleine und es könnte sich die eine oder andere Fahrgemeinschaft bilden. Aber vorher stehen ja auch noch einige andere Treffen an, vielleicht ist dort auch schon etwas zu klären - bis Oktober ist es ja noch lange hin.

Aus England werde ich einen Q40 mitbringen - er wartet dort schon auf mich. Freue mich schon drauf, dann habe auch ich endlich mal die neuen Farbtreiber - mit vielen, vielen Farben, Bildschirmhintergrund und dergleichen. Der Q40 wird dann ruck-zuck den Mailbox-Rechner ersetzen. Das ist nicht nur für mich von Vorteil (die Mailbox empfängt nämlich auch meine Faxe, und Fax-Darstellung ist so ziemlich das Langsamste was es unter QDOS und SMSQ/E gibt), sondern natürlich auch für alle, die sich dort normal einloggen - sei es, um Updates herunterzuladen, sei es, um nach Neuigkeiten zu schauen, Nachrichten zu lesen und zu hinterlassen oder nach neuester Public-Domain-Software zu schauen. Es ist glücklicherweise dank Wettbewerb im Telefonmarkt zu Zeiten, in denen ein Ferngespräch abends preiswerter als ein Ortsgespräch ist, nicht allzu kostenintensiv, mal ab und zu in die Box zu schauen. Doch je schneller sie reagiert, umso mehr Spaß macht die Nutzung.

Gleichzeitig mit der Tatsache, daß ich Q40 selber habe (und damit vernünftigen Support bieten kann) und der Tatsache, daß die Farbtreiber für Q40 endlich fertig sind (und damit Q40 ordentlich ausgeliefert werden kann) nehme ich Q40 mit in mein Programm auf.

Ich denke mal, daß ich zu den nächsten Treffen (insbesondere Eindhoven) mit Q40 fahren werde - insofern wundert Euch bitte nicht, wenn die Mailbox zu Zeiten, in denen ich auf Reise bin, nicht antwortet (vielleicht findet sich ja eine Notlösung, mal sehen). Über Q40 wurde ja in einigen vergangenen QL Todays Deutsch und Englisch bereits ausführlich berichtet, daher werde ich jetzt hier keine Details auflisten.

Positiv zu vermelden sei noch, daß Tony Tebby auf dem Eindhoven-Treffen im Februar einen Q60 ans Laufen brachte!

Die Mailbox (genauer gesagt, beide): In den vergangenen Monaten etwas vernachlässigt, aber seit zwei Wochen wieder aufgepeppt und aktualisiert. Ich habe festgestellt, daß es nicht möglich ist, zu viele Dinge nebeneinander laufen zu lassen - die Zeit reicht einfach nicht! QL Today kostet mich fast einen ganzen Monat an Zeit jeden zweiten Monat, d.h. rund 50% der Zeit, die ich vorher hatte, stehen nicht mehr zur Verfügung. Zudem ist der QL-Markt leider

auch nicht mehr so erträglich wie es wünschenswert wäre, d.h. Geld muß logischerweise auch mit anderen Aktivitäten verdient werden. Um es kurz zu machen: Ich werde wieder verstärkt die Mailboxen aktualisieren und die JMS-Internet-Präsenz relativ statisch gestalten. Ich habe daher auf den Webseiten nur grundsätzliche Informationen belassen - alle sich öfter verändernden Dinge (wie z.B. Versionsnummernlisten) werden nur noch in der Mailbox veröffentlicht, sonst wird es einfach zu viel und nur "halber Kram", und das stört auch mich! Wir haben in der QL-Welt ja auch eine wirklich exzellente Website rund um den QL - die von Thierry Godefroy - da brauche ich die Arbeit nicht auch noch doppelt machen. Ja, und da ich mir nun viel Mühe mit der Box gebe hoffe ich, daß dies auch Anklang findet und ich öfter Besucher in der Mailbox sehe! Wer lernen will: Ab Mai 2000 kann man für 5,- DM im Monat den ganzen Sonntag kostenlos telefonieren - und natürlich auch die JMS-Mailbox besuchen! (Danke für den Hinweis, Dietrich!).

Auch bei Software hat sich viel getan! Komisch, es war auch in den vergangen Jahren so, daß im Frühjahr die meisten Neuigkeiten herauskamen. Liegt vielleicht daran, daß die Winterzeit ideal ist, um sich vor den Rechner zu setzen, oder?
Joachim van der Auwera (PROGS) hat nun ProWesS für kostenlos und frei verfügbar erklärt. Das ist prima und fördert sicherlich die Entstehung von neuen ProWesS-Anwendungen, da jetzt keine weiteren kostenpflichtigen Voraussetzungen gegeben sein müssen. Das neue Agenda-Programm von Wolfgang Lenerz kommt ja beispielsweise in zwei Versionen - hier hat der Anwender die Wahl, ob er lieber das Programm im gewohnten WMAN-Look nutzen möchte oder aber mit skalierten Schriften unter ProWesS. Da beide Versionen mit der gleichen Termin-Datenbank arbeiten, kann jederzeit gewechselt werden - kein Problem!

Neu und ebenfalls erhältlich ist Suqcess von Wolfgang Uhlig - wir hatten einen Bericht in der vergangenen Ausgabe von QL-Today.
An SMSQ/E hat sich ebenfalls viel getan: Die aktuelle Version ist 2.96 - dort sind einige kleine Bugs behoben (nichts Dramatisches), aber neu ist der Bildschirmtreiber. Unterschiede im Aussehen werden Sie nicht bemerken, sofern Sie nicht die neuen Fähigkeiten nutzen (Kompatibilität ist nach wie vor sehr wichtig, sonst wäre der neue Farbtreiber schon längst fertig gewesen!). Auch auf Systemen, die noch nicht "viel mehr" Farben darstellen können, können die Farb-Werte nun so wie bei Q40 (und zukünftigen Systemen) angegeben werden - sie werden dann in 4 bzw. 8 Farben umgerechnet. Eine auf jedem System nutzbare Neuerung durch den neuen Grafiktreiber ist die Möglichkeit des Hintergrund-Musters bzw. -Bildes. Mehr dazu in dieser oder der nächsten Ausgabe - je nachdem, wie schnell der neue Treiber verfügbar ist.

So, jetzt ist das Editorial viel länger als geplant geworden - und viel länger als jemals zuvor. Es gibt jedoch zur Zeit so viel Aktivität im QL-Markt, die alle Aufmerksamkeit verdient.
Es wäre auch noch viel länger geworden, hätte ich jetzt auf dem Rückweg nicht den Laptop zur Seite legen müssen. Ich hatte Glück und frühzeitig eine Seacat nach Calais bekommen (sehr klein) und da es sehr heftigen Seegang gibt - hoffe ich, es ohne "unangenehme Nebenerscheinungen" zu schaffen. Nur noch einen Satz: Q40 habe ich dabei!

Mit der nächsten Ausgabe starten wir in den fünften Jahrgang von QL Today - hoffentlich auch weiterhin mit Euch als Lesern und Autoren!

Bis dahin grüßt herzlichst

Euer Jochen Merz

Programmieren mit den Basic-Erweiterungen von EASYPTR - Teil 2

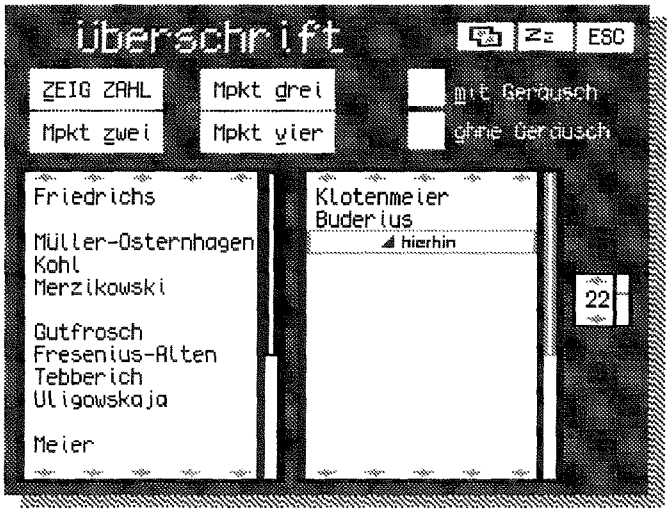
Eine Serie von Wolfgang Uhlig

Im heutigen und nächsten Teil meiner Serie werde ich mich damit beschäftigen, was man mit Tabellen in EASYPTR alles machen kann. Heute sind erst einmal normale, zweidimensionale Tabellen an der Reihe, also eigentlich keine "richtigen" Tabellen, sonder nur jene, die aus einer Spalte und mehreren Zeilen bestehen. Diese Zeilen werde ich im folgenden Felder nennen. Sie können also heute lernen,

- wie man dimensionierte Tabellen in Anwendungsfenstern anzeigt,
- wie man die Felder der Tabelle abfragt
- wie man den Inhalt von Feldern in ein anderes Anwendungsfenster verschiebt (drag and drop)
- wie man dafür sorgt, dass ein gewünschtes Feld am Anfang steht

Das ist eine ganze Menge und die Vorarbeit ist auch nicht gerade wenig, aber der Erfolg am Ende wird im richtigen Verhältnis dazu stehen. Außerdem habe ich viel Denkarbeit schon im Vorhinein für Sie geleistet und viele frustrierende Erfahrungen und endlose Ausprobiererei werden Ihnen deswegen erspart bleiben.

Erst einmal müssen wir unser Menü vom letzten Mal verändern und auch noch einen neuen Sprite herstellen. Um einen Eindruck zu haben, wie das in etwa aussehen wird, schauen Sie sich einmal folgendes Bild an:



Sie sehen, ich habe das Menü ein wenig schmaler gemacht und die Menüpunkte etwas zusammengerückt. Einer davon hat einen neuen Namen bekommen, nämlich "ZEIG ZAHL". Dazu gekommen sind, und das ist wichtig für unseren heutigen Lehrgang, drei Anwendungsfenster, zwei große für Namen und ein kleines für Zahlen. Bitte versuchen Sie nicht, irgendeinen tieferen Sinn zu entdecken, die Namen und Zahlen sind völlig fiktiv und haben nur den Zweck, Ihnen bestimmte Funktionen und Prozeduren von EasyPtr deutlich zu machen.

Wenn Sie Anwendungsfenster in EasyMenu anlegen, die für zwei- oder dreidimensionale Tabellen sein sollen, ist folgendes zu beachten:

- Die zweite Dimension der Tabelle, also in unserem Fall die Anzahl der Buchstaben, darf nicht breiter sein als das Anwendungsfenster, außerdem muss man noch einen Eingabepuffer von einem Buchstaben dazurechnen, wenn man in den Feldern editieren möchte.
- Die Höhe darf nicht kleiner sein als ein Buchstabe (oder Zahl) hoch ist.
- Dazu kommt, dass, wenn mehr Felder vorhanden sind als senkrecht in das Fenster passen, automatisch Rollbalken (und -pfeile) erstellt werden, die das Fenster oben und unten um jeweils 8 Pixel verkleinern und um 14 Pixel auf der rechten Seite verschmälern.

Wie berechnet man das also? Auf dem QL sind Buchstaben und Zahlen 6 Pixel breit. Wenn wir also eine Tabelle planen von 20 Namen mit maximal 18 Buchstaben, also z.B. **DIM namen$(19,18)** rechnen wir:

18 Buchst. x 6 Pixel
= 108 + 6 Pixel Eingabepuffer + 14 Pixel für den Rollbalken
= 128 Pixel

Legen Sie sich also zwei Anwendungsfenster mit der Breite von 128 Pixeln an. Warum zwei, werden Sie nachher sehen. Die Höhe spielt hier keine Rolle, denn es werden wie gesagt, automatisch Rollpfeile angelegt, wenn die Tabelle zu groß ist.

In dem kleinen Zahlenfenster wollen wir immer nur eine einzige Zahl zwischen 1 und 52 sehen, darum spielen sowohl Höhe als auch Breite eine Rolle. Mit **DIM zahl$(51,2)** dimensionieren wir unsere Tabelle als Zeichenkettentabelle, da EasyPtr keine reinen Zahlentabellen zeichnen kann (ein manchmal etwas ärgerliches Manko, finde ich übrigens). Ein QL-Buchstabe ist 9 Pixel hoch und die Rollpfeile nehmen 8 Pixel in Anspruch, also bekommen wir für die Höhe:

9 Pixel + 8 Pixel oben und 8 Pixel unten
= 25 Pixel
und für die Breite
2 Buchst. x 6 Pixel
= 12 + 14 Pixel für den Rollbalken
= 26 Pixel
Hierbei ist **nicht** mit einem Eingabepuffer von 6 Pixel gerechnet, weil wir die Zahlen nicht editieren, sondern nur anzeigen wollen. Ein Anwendungsfeld von 26x25 ist also ausreichend, um Zahlen bis 99 anzuzeigen.

Bitte zeichnen Sie sich ein solches in Ihr Menü, das nun ungefähr so aussehen sollte wie das oben gezeigte. Kleine Abweichungen sind natürlich völlig okay, wichtig ist jedoch die Reihenfolge der Anwendungsfenster, weil sich die Programmierung darauf bezieht. Also die beiden großen sind links Nummer 1 und rechts Nummer 2 und das kleine für die Zahlen ist Nummer 3.

Hängen Sie nun die neue Menüdefinition an in Ihrer Datei "Projektmenüs". Dort haben Sie nun zweimal das Hauptmenü, das ist eigentlich überflüssig. Starten Sie also einfach den Appendix-Manager "Appman", der zum EasyPtr Programmpaket gehört und löschen Sie dort das alte Menü. Sollten Sie sich nicht sicher sein, wie man mit Appman umgeht, lassen Sie alles wie es ist und verändern Sie im Programm die Zeile

```
MDRAW #3,1
```

in

```
MDRAW #3,2
```

dann wird Ihr neues Menü aufgerufen.

Da ich Ihnen zeigen möchte, wie man aus einer Tabelle den Inhalt eines Feldes "schnappt" und ihn in ein anderes Feld "zieht" (das berühmte "drag and drop"), möchte ich Ihnen noch vorschlagen, einen kleinen Sprite anzufertigen, der diesen Vorgang symbolisiert. Ich habe mir nicht so wahnsinnig viel Mühe gegeben und mein Sprite ist denn auch nur eine kleines Schild mit der Aufschrift "hierhin". Sie dürfen jedoch Ihrer Fantasie freien Lauf lassen und sich wunderschöne Sprites anfertigen, selbst animierte sind schließlich möglich! Wenn Sie Ihren Sprite fertig haben, hängen Sie ihn bitte an in Ihrer Datei "Projektmenüs". Alle Vorbereitungen sind damit abgeschlossen und wir können uns ans Programmieren machen. Im Vorspann und im Hauptprogramm sind ein paar Zeilen dazugekommen und einiges hat sich verändert. Benutzen Sie Ihr altes Programm und fügen Sie folgende Zeilen hinzu bzw. verändern Sie:

```
161 drag%=0     :REMark ***** Variable für die
    Drag and Drop Prozedur
162 puffer$=""    :REMark ***** für dieselbe
    Prozedur
171 REMark ***** Felder dimmen in Extraprozedur
172 DIMMEN
173 :
230  =-4:ZEIGE_ZAHL
271  =2^16 TO 2^32:
272       awfnum=taste
273       position=MAWNUM(#3,awfnum)-1
274       IF awfnum=3
275        NEXT haupt
276       ELSE
277        FELD_VERSCHIEBEN
278       END IF
```

Danach führen Sie ein RENUM durch und Sie sollten nun die gleiche Nummerierung haben wie im untenstehenden Programm.

```
100 REMark ***** ein kleines EasyPtr-Übungs-
    programm
110 :
120 LRESPR win1_projektmenüs
130 OPEN #3,con
140 MDRAW #3,1
150 :
160 REMark ***** eventuelle Variablen
    definieren/setzen
170 angekreuzt=0
180 drag%=0:REMark ***** Variable für die Drag
    and Drop Prozedur
190 puffer$="":REMark ***** für dieselbe
    Prozedur
200 :
210 REMark ***** Felder dimmen in Extraprozedur
220 DIMMEN
230 :
240 REMark ***** hier beginnt das Hauptprogramm
250 REPeat haupt
260 taste=MCALL(#3,taste,0)
270 SELect ON taste
280  =-1:ENDE
290  =-4:ZEIGE_ZAHL
300  =-5:REMark ***** PROZ_2
310  =-6:REMark ***** PROZ_3
320  =-7:REMark ***** PROZ_4
330  =-8,-9:ANKREUZEN
340  =2^16 TO 2^32:
350       awfnum=taste
360       position=MAWNUM(#3,awfnum)-1
370       IF awfnum=3
380        NEXT haupt
390       ELSE
400        FELD_VERSCHIEBEN
410       END IF
420  =REMAINDER :NEXT haupt
430 END SELect
440 END REPeat haupt
450 REMark ***** und hier ist das Hauptprogramm
    zuende
```

(Zeile 460 bis 690 wie im alten Programm)

Was ist dazugekommen? In Zeile 180 und 190 definieren wir zwei Variablen, die später gebraucht werden. Welche Funktion sie haben, siehe dort. In Zeile 220 wird die neue Prozedur "DIMMEN"

aufgerufen, in ihr werden die Tabellen dimensioniert, die in unsere neuen Anwendungsfenster kommen sollen, siehe dort. In Zeile 290 habe ich dem Menüpunkt -4 eine neue Prozedur zugewiesen, in ihr werden wir später die Zahl aus dem kleinen Zahlenfenster auslesen. Ab Zeile 340 geht es nun richtig ans Eingemachte.

Um Menüpunkte, (also eigentlich Felder in Tabellen) in Anwendungsfenstern manipulieren zu können, muss man sie natürlich erstmal identifizieren. Sie beginnen mit 2^16 und enden mit 2^32. Da das sehr unhandliche Zahlen sind, gibt es die Funktion **MAWNUM**. Ich habe lange gebraucht um sie zu verstehen, weil sich mir die Anleitung einfach nicht erschließen wollte. Inzwischen geht es einigermaßen und ich will hier die wichtigsten Dinge darstellen. Normalerweise bekommt man mit
num=MAWNUM(#kanal,taste)
die Nummer "num" des angeklickten Menüpunktes in einem Anwendungsfenster. Der Kanal ist der Hauptfensterkanal, das Trennzeichen ist ein Komma und "taste" ist in unserem Fall einfach das, was wir von **MCALL** geliefert bekommen. Auch hat man die Möglichkeit, Koordinaten zurückzubekommen, z.B. mit
num=MAWNUM(#kanal,taste,xpos%,ypos%)
xpos% und ypos% geben uns dann Spalte und Zeile des angewählten Menüpunktes an. Das ist jedoch mehr für dreidimensionale Tabellen interessant. Dazu mehr in der nächsten Folge.
Ganz wichtig ist aber folgendes: "taste" ist nach dem Ausführen dieser Funktion nicht länger mehr der von MCALL gelieferte Wert von "taste", sondern hat sich verändert in die reine Anwendungsfensternummer!!
Ich fand das am Anfang schwer zu begreifen, aber es hat immense Vorteile, wenn man bestimmen will, wo denn überhaupt geklickt wurde. Ab Zeile 350 habe ich mir das zunutze gemacht: Ich habe zuerst die Variable "taste" kopiert in die Variable "awfnum" (Anwendungsfensternummer). Das stimmt zwar in diesem Moment noch nicht, aber schon eine Zeile später, nämlich nach
position=MAWNUM(#3,awfnum)-1
ist "awfnum" tatsächlich zur Nummer des Anwendungsfensters geworden und 'taste' kann trotzdem in evtl. anderen Funktionen weiter benutzt werden. Wir können nun in Zeile 370 und 380 einfach sagen: Wenn in einen Menüpunkt des Anwendungsfensters 3 geklickt wird, soll gar nichts passieren, wir gehen gleich wieder in die nächste Hauptschleife. Was man sehr wohl mit der Maus machen kann, ist, sich mit Hilfe der Rollpfeile die gewünschte Zahl einzustellen. Diese auszulesen, geschieht mit der letzten Variante von MAWNUM, benutzt auch in der Prozedur "ZEIGE_ZAHL", die ganz am Ende des Programms steht und mit dem freien Menüpunkt -4 aufgerufen wird:
zahlpos=MAWNUM(#3\3)-1
In diesem Fall wird ein Backslash als Trennzeichen benutzt und die reine Anwendungsfensternummer. Ausgewertet wird in dieser Variante normalerweise der Menüpunkt im Anwendungsfenster, der links oben steht. In unserem Fall, wo wir ja sowieso nur eine zweidimensionale Tabelle haben und auch nur einen Wert sehen, ist das natürlich eben dieser Wert! (Warum -1?: siehe übernächster Absatz) Nun werden Sie fragen, wozu soll ich das denn noch mal auswerten, ich sehe die Zahl doch?! Das hier ist ja nur ein Beispiel, es könnte aber auch sein, dass Sie die Zahl in einer Formel oder anderen Berechnung im Programm nötig haben und dann müssen Sie das so machen.

Gut, nun noch einmal zurück zu Zeile 360. MAWNUM liefert uns also normale kleine Zahlen für unsere Felder zurück. Also 1 für das erste Feld, 2 für das zweite usw. und 20 für das zwanzigste. Da eine QL-Tabelle immer bei 0 anfängt müssen wir, um den richtigen Platz in der Tabelle zu bestimmen, die von MAWNUM gelieferte Nummer um eins verringern. Daher:
position=MAWNUM(#3,awfnum)-1

Wir haben nun bestimmt, welcher Menüpunkt angeklickt wurde und welche Position im Feld das ist. Wir wissen außerdem, in welchem Anwendungsfenster das Klicken stattgefunden hat. Wenn wir im Fenster 3 sind, geschieht weiter nichts, wenn wir aber ins Fenster 1 oder 2 geklickt haben, soll etwas passieren. Damit beschäftigt sich die Prozedur "FELD_VERSCHIEBEN", dazu kommen wir gleich.

Da wir es mit Tabellen zu tun haben, müssen die natürlich erstmal dimensioniert und eventuell gefüllt sein, und sie müssen letztlich auch gezeichnet werden. Das tun wir in der Prozedur "DIMMEN", die noch vor dem Hauptprogramm aufgerufen wurde. Das Dimensionieren der Tabellen dürfte Ihnen bekannt sein, interessant für unseren Lehrgang ist der Befehl **MAWDRAW**. Wir sehen ihn hier in seiner einfachen Form:
MAWDRAW #kanal,awfnum,feld$
wobei "kanal" der Hauptkanal ist, "awfnum" das Anwendungsfenster und "feld$" die Zeichenkettentabelle. Mit dem wesentlich umfangreicheren MAWDRAW-Befehl für dreidimensionale Tabellen werden wir uns im dritten Teil dieser Serie beschäftigen. MAWDRAW zeichnet also schlicht und

einfach das gewünschte Feld in das gewünschte Anwendungsfenster.

```
700 :
710 DEFine PROCedure DIMMEN
720 REMark ***** ein Feld für die 20 Namen
    dimensionieren
730 DIM namenfeld$(19,18)
740 RESTORE
750 FOR i%=0 TO 18
760  READ a$
770  namenfeld$(i%)=a$
780 END FOR i%
790 DATA "Friedrichs", "Klotenmeier",
    "Müller-Osternhagen", "Kohl"
800 DATA "Merzikowski", "Buderius",
    "Gutfrosch", "Fresenius-Alten"
810 DATA "Tebberich", "Uligowskaja", "Müller",
    "Meier"
820 DATA "Schulze", "Bäcker", "Fleischer",
    "Schuhmacher"
830 DATA "Beckenblubber", "Becker", "Graf",
    "Dracula"
840 :
850 REMark ***** ein Feld, in das später
    selektierte Namen
860 REMark ***** kommen, nun aber noch leer:
870 DIM selektion$(19,18)
880 :
890 REMark ***** ein Zahlenfeld von 1 bis 52
900 DIM zahlenfeld$(51,2)
910 FOR i%=0 TO 51:zahlenfeld$(i%)=i%+1:
920 :
930 REMark ***** jetzt alle Felder zeichnen:
940 MAWDRAW #3,1,namenfeld$
950 MAWDRAW #3,2,selektion$
960 MAWDRAW #3,3,zahlenfeld$
970 :
980 END DEFine DIMMEN
```

So, nun aber endlich zu unserer Prozedur "FELD_VERSCHIEBEN", in der wir Felder von einem Fenster ins andere oder zurück oder im selben Fenster verschieben können.

```
990 :
1000 DEFine PROCedure FELD_VERSCHIEBEN
1010 IF drag%
```

drag% ist eine Variable, die mir, wenn sie '1' ist, zeigt, ob zur Zeit ein Feld (in unserem Fall ein Familienname) im Puffer ist, also noch irgendwo hingesetzt werden muss. Nun gibt es zwei Möglichkeiten: erstens, im Feld ist an der angegebenen Position schon ein Feldinhalt (Familienname). Das überprüfen wir mit der Funktion **MTEXT$**.

MTEXT$ ist eine einfache Funktion (siehe Easy-Ptr-Handbuch), mit der man Texte aus Menüpunkten oder Infoobjekten auslesen kann. Geht es um Menüpunkte, ob nun freie oder welche im Anwendungsfenster, setzt man für 'num' einfach den Wert ein, der von MCALL zurückgegeben wird, in unserem Fall also 'taste'. Bei Infoobjekten muss man das Pärchen aus Infofensternummer und Infoobjektnummer angeben und 'num' weglassen.

```
1020 IF MTEXT$(taste)<>""
```

Diesen Text wollen wir da lassen und nicht überschreiben, also:

```
1030 NEXT haupt
```

Oder aber es ist leer:

```
1040 ELSE
```

dann überprüfen wir erst, in welchem Anwendungsfenster wir sind und setzen dann das entsprechende Feld der entsprechenden Tabelle auf den Pufferinhalt.

```
1050   IF awfnum=1
1060    namenfeld$(position)=puffer$
1070   ELSE
1080    selektion$(position)=puffer$
1090 END IF
```

zum Schluss zeichnen wir den Inhalt des Puffers mit dem Befehl **MAWITEM** in den Menüpunkt, machen den Puffer leer und setzen drag% auf sein Gegenteil (also 0 in diesem Fall). Zum Schluss setzen wir den Sprite wieder zurück mit **SPRS** und schließen damit die Verschiebeaktion ab.

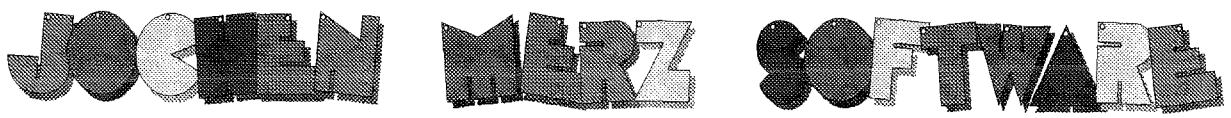
```
1100 MAWITEM #3,taste,,puffer$
1110 puffer$=""
1120 drag%=NOT(drag%)
1130 SPRS #3,0
1140 END IF
```

Der Befehl **MAWITEM** unterscheidet sich in nichts von dem in der vorigen Folge behandelten Befehl **MITEM**, außer der Tatsache, dass er für Menüpunkte im Anwendungsfenster gilt. **SPRS** ist ein einfacher Befehl, dessen Beschreibung im Handbuch jedoch seine Tücken hat. Dort steht zwar, dass man den Namen einer angehängten Spritedefinition nehmen kann, es zeigt sich jedoch in der Praxis, dass das nicht immer erkannt wird. Vor allem nach dem Kompilieren führt das zu endlosen Frustrationen (wenn man das nicht weiß, aber Sie wissen's ja jetzt!). Ich kann nur dringend empfehlen, das zu benutzen, was im Handbuch als 'adr' bezeichnet wird. Das ist nicht etwa eine Speicheradresse, sondern schlicht und einfach die Nummer des Sprites in der Anhängedatei. In unserem Beispiel also die 3, weil es der dritte angehängte Sprite ist. Eine 0 setzt zurück. Mit SPRS #3,3 setzt man also diesen Sprite als Zeiger für das Programm, mit SPRS #3,0 stellt man den ursprünglichen Zeiger wieder her. So einfach ist das!

Nun müssen wir uns mit dem anderen Fall beschäftigen, nämlich wenn noch gar kein Feld angewählt ist und der Puffer dementsprechend leer.

```
1150 ELSE
```

In diesem Fall wollen wir natürlich, dass sich in



Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
http://www.j-m-s.com/smsq/index.htm

Jochen Merz Software Programme und Versionsnummern

QPC1 + QPC2 + SMSQ/E Software QL-Emulator für PC's	[V1.54]	DM 249,-
QPC1 + QPC2 + SMSQ/E für SMSQ/E-Besitzer	[V1.54]	DM 199,-
QPC2 Upgrade von QPC1	[V1.54]	DM 79,90
SMSQ/E für ATARI oder QXL oder (Super)GoldCard	[V2.97]	DM 249,-
QD 98 Pointer-Environment-Editor	[VA.06]	DM 119,-
QD 98 Upgrade von V9	[VA.06]	DM 39,90
QD 98 Upgrade von älteren Versionen	[VA.06]	DM 49,90
NEU! AGENDA für WMAN or ProWesS	[V1.08]	DM 59,90
NEU! SUQCESS Datenbankprogramm für das Pointer Environment		DM 79,90
QMAKE Pointer-MAKE für GST/Quanta Assembler	[V4.26]	DM 45,-
BASIC Linker	[V1.16]	DM 49,-
WINED Floppy/Harddisk Sektor- & Datei-Editor	[V1.21]	DM 49,-
FiFi II File-Finder - Sehr nützlich!	[V4.22]	DM 49,-
FiFi II Upgrade von älteren Versionen	[V4.22]	DM 19,-
EPROM Manager	[V3.02]	DM 49,-
QSpread99 Tabellenkalkulation	[V2.08]	DM 139,-
QSpread99 Upgrade	[V2.08]	DM 39,90
QPAC I Utility Programme	[V1.07]	DM 59,-
QPAC II Dateien, Jobs & andere Dinge	[V1.39]	DM 119,-
QPAC II Update	[V1.39]	DM 16,-
QTYP II Buchstabierüberprüfer	[V2.17]	DM 79,-
QMENU Programmieranleitung/Beispiele für Menü Erweiterung	[V7.04]	DM 41,90
QMENU Upgrade	[V7.04]	DM 18,-
QPTR Pointer Toolkit	[V0.30]	DM 89,-
QPTR Update	[V0.30]	DM 16,-
DISA Interaktiver Disassembler	[V3.04]	DM 89,-
DISA Upgrade	[V3.04]	DM 33,-
typeset-ESC/P2 text87 Treiber für alle ESC/P2 Drucker (incl. Stylus)		DM 65,-
CueShell	[V2.14]	DM 90,-
CueShell für QPC	[V2.14]	DM 40,-
SER Mouse Software Maustreiber für serielle Mäuse		DM 25,-
EASYPTR Part 1&2	[V3.10]	DM 89,-
QDOS/SMSQ Reference Manual		DM 79,-

LIEFER- und ZAHLUNGSBEDINGUNGEN

Versandkosten **[Deutschland]** DM 8,99 (wenn Rechnungsbetrag unter DM 50,- dann nur DM 5,99). Bei Rechnungsbeträgen über DM 500,- kostet es DM 18,99 **[Europa]** DM 14,50 (wenn Rechnungsbetrag unter DM 50,– dann nur DM 9,50). Alle Preise inkl. 15% MwSt. Irrtum und Preisänderung vorbehalten. Verrechnungs-, Euroschecks und Kreditkarten werden akzeptiert. Bankeinzug möglich.

EUROCARD. VISA AMEX MasterCard Cards

dem Feld ein Eintrag befindet. Leere Felder ignorieren wir (siehe Zeile 1270, 1280)

```
1160 IF MTEXT$(#3,taste)<>""
```

Nun also wieder schauen, in welchem Anwendungsfenster wir sind. Dann den entsprechenden Feldtext in den Puffer schieben und das Feld auf 'nichts' setzen:

```
1170    IF awfnum=1
1180     puffer$=namenfeld$(position)
1190     namenfeld$(position)=""
1200    ELSE
1210     puffer$=selektion$(position)
1220     selektion$(position)=""
1230    END IF
```

Das leere Feld muss natürlich auch noch gezeichnet werden

```
1240    MAWITEM #3,taste,,""
```

drag% wird nun sein Gegenteil (also 1) und der Zeiger wird auf unseren eigenen Sprite gesetzt

```
1250    drag%=NOT(drag%)
1260    SPRS #3,3
```

War in dem angeklickten Feld gar kein Inhalt, springen wir gleich in die nächste Schleife.

```
1270   ELSE
1280    NEXT haupt
1290   END IF
1300 END IF
1310 END DEFine FELD_VERSCHIEBEN
```

Das war es schon! Ist doch gar nicht so schwer, oder? Natürlich kann man Dinge anders machen, so ist es vielleicht praktischer, in den Zeilen 1050 - 1090 und 1170 - 1230 statt IF-ELSE eine SELect-Abfrage zu machen. Ab drei Anwendungsfenstern wäre das sicher die elegantere Lösung.

Auch könnte man die Aktionen der Prozedur FELD_VERSCHIEBEN auch einfach im Hauptprogramm unterbringen. Letztlich ist das Geschmacksache. Da ich sehr viel in QD programmiere, benutze ich so viele Prozeduren wie möglich. Man kann in QD so wunderbar einfach zu Prozeduren und Funktionen springen, da wäre man dumm, wenn man das nicht ausnutzen würde.

Zum Schluss folgt noch die Prozedur, mit der man das Ergebnis des dritten Anwendungsfenster auswertet.

```
1320 :
1330 DEFine PROCedure ZEIGE_ZAHL
1340 LOCal zahl
1350 zahl=MAWNUM(#3\3)
1360 idem=ITEM_SELECT('DIE ZAHL','die
     Zahl ist '&zahlenfeld$(zahl-1),
 'aha',,,,,2)
1370 END DEFine ZEIGE_ZAHL
```

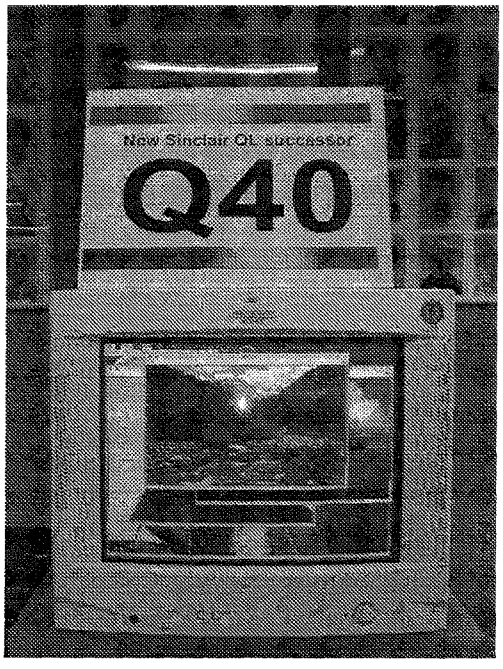
Das soll es für heute gewesen sein. Zugegeben eine nicht so wahnsinnig einfach zu verdauende Mahlzeit. Probieren Sie es aus, es lohnt sich!

Ich hoffe, es hat Ihnen Spaß gemacht und Sie freuen sich schon auf die nächste (und wahrscheinlich letzte) Folge, in der wir uns mit dreidimensionalen Tabellen beschäftigen, Menüpunkte in unterschiedlichen Status setzen, linke und rechte Maustaste abfragen und noch etliches mehr.

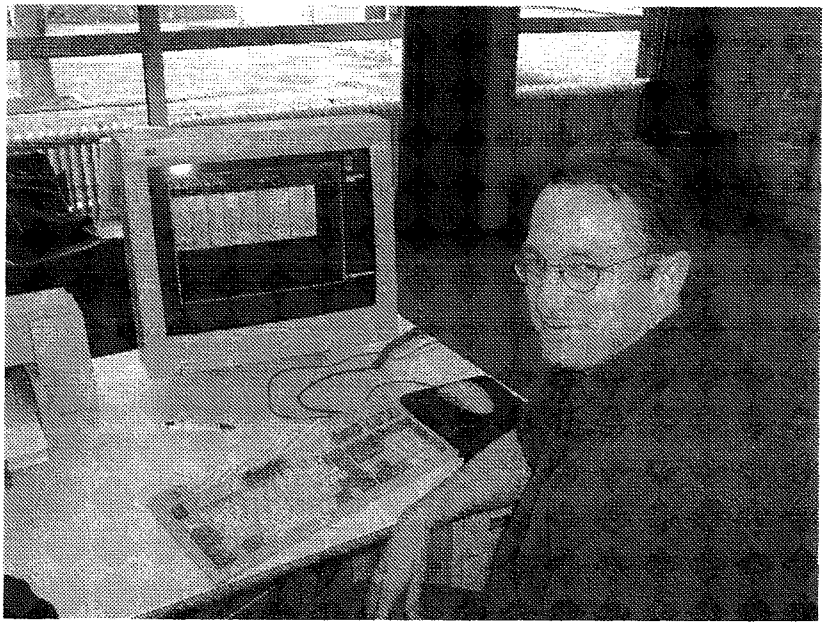
Bis dann, lang lebe der QL!

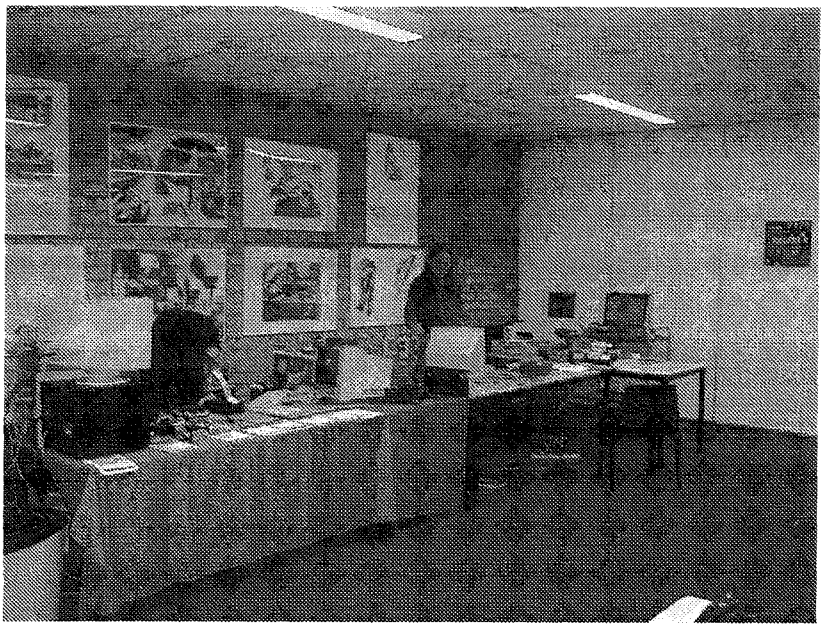
Eindhoven 2000

Im Februar fand das erste Eindhoven-QL-Treffen im Jahr 2000 statt. Die englischen Händler waren wieder dort, Jochen Merz Software, die "Gebrüder Graf" (Q40). Tony Tebby kam auch, zusammen mit anderen QLern aus Frankreich. Man konnte ihn den ganzen Tag heftig an der Tastatur arbeiten sehen, schließlich wollte er auch Q60 ans Laufen bekommen (was ihm auch noch an diesem Tag gelang) und ein paar andere ungelöste Probleme in den Griff bekommen bzw. life vor Ort sehen. Bei Jochen Merz Software gab es einige Updates sowie zwei neue Programme: Agenda und Suqcess (Premiere!). Über beides war ja in den vergangenen Ausgaben von QL Today zu lesen.
Beide Autoren waren ebenfalls anwesend: Wolfgang Lenerz hat seinen Laptop



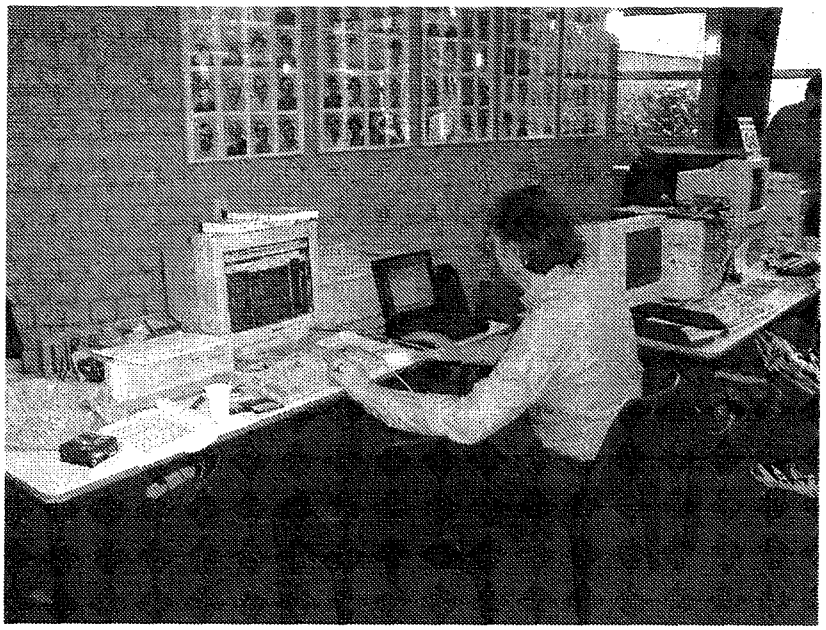
jedoch den größten Teil der Zeit Tony Tebby als Debug-Computer überlassen. Wolfgang Uhlig demonstrierte Suqcess (das es übrigens auch in Niederländisch, nicht nur in Deutsch und Englisch, gibt) - sehr beeindruckend, nicht nur, weil es auf einem 21" Monitor demonstriert wurde.
QBranch hatte die gesamte Hard- und Softwarepalette mitgebracht (Roy und Tony kamen in einem kleinen Transporter, da paßt viel rein - zu ihrem ganzen Equipment auch noch Weinvorräte für die nächsten Monate). Sowohl Roy als auch TF waren





den ganzen Tag damit beschäftigt, Q40-Systeme zusammenzubauen (so auch eins für Jochen Merz, der damit in Zukunft seine Mailbox betreiben will). Es waren auch wieder Besucher aus Holland (logisch), Deutschland, Frankreich und Belgien anwesend. Interessant auch: Physiklehrer Jens Wildgruber führt an seinem QL nicht nur einen "Neid"-Flatscreen vor, sondern zeigt auch fast auf jedem Treffen interessante physikalische Versuchsanordnungen, die er seinen Schüler-Jahrgängen am QL mit I²C-Interface vorführt.

Sich dies anzuschauen ist auch jedes Mal für Nicht-Physiker sehr interessant, denn Jens erklärt sehr gut. Das Bewunderungs-Highlight auf diesem Treffen war wieder Q40. Es ist auch sehr beeindruckend (und sehr ungewohnt), in einem "normalen" QPAC2-ähnlichen Fenster ein wie echt aussehendes Bild von Flußläufen und dergleichen zu bewundern (65536 Farben). Möchte man eigentlich sofort mitnehmen. Die Geschwindigkeit ist schon beeindruckend, aber mittlerweile glaube ich, daß dies nicht ganz so wichtig ist (QL-Programme



sind so optimiert, daß man keine 700MHz-Systeme benötigt um einen Brief zu schreiben) - nein, es ist die Farbe, die anzieht. Vielleicht ist's das Ungewohnte, wer weiß. Nun ja, sehr bald werde auch ich die Farben ausprobieren dürfen (Q40 steht ja an) und auch QXL- und QPC2-Besitzer dürfen sich bald freuen. Aurora-User werden auch nicht vergessen, keine Sorge! Schade, daß nicht mehr Besucher kamen. Dies ist leider bei Winter-Treffen öfter so, vielleicht schreckt das Wetter hier zu sehr ab. Es war mild, regnete etwas, doch die Sonne war auch oft zu sehen. Leider gab's am Tag vorher selbst in Holland noch Glatteis, so daß einige Niederländer schon am Tag vorher absagten. Wäre schön, wenn sich auf zukünftigen Treffen wieder mehr QLer einfinden würden. Trotz Wetter-Handicap war's ein relativ gut besuchtes und auf jeden Fall schönes Treffen, das wie jedes Eindhoven-Treffen mit dem obligatorischen Chinesischen Buffet ein paar Straßen weiter endete.

Werkzeug zur Umwandlung eines gewöhnlichen QL-Textes ins HTML-Format

Gerhard Plavec

Den letzten Gerüchten nach zu urteilen, dürfte es nicht mehr allzulange dauern, bis auch der QL endlich Zugang zum Internet findet. Wer Internet sagt, meint eigentlich HTML ;-)
Der Vorteil von HTML ist, daß es Texte System-übergreifend darstellen kann. Hatte die Umstellung von Schreibmaschine auf Computer schon ein Umdenken gefordert (Seitenumbrüche wurden sinnlos, Zeilen justieren, Mittel- oder Rechtsbündig schreiben wurde automatisiert, etc...) so fordert HTML neuerliches Umdenken. Man muß sich daran gewöhnen, daß eine jede Seite nicht nur auf jeden Computer anders aussieht - je nach Bildschirmgröße, Zeichensatz, Schriftgröße, etc...) - sondern von jedem Browser - ohne dem eine HTML-Seite überhaupt nicht sinnvoll dargestellt werden kann - anders dargestellt wird. Der Text wird für gewöhnlich unter Verwendung der vorhandenen Zeichensätze und ggf. der voreingestellten Schriftgrößen in das Browserfenster eingepaßt. Dafür ist es relativ unwichtig, in welchem System bzw. auf welchem Computer der Browser läuft.
Nun darf gefragt werden, wie denn das Problem mit den Umlauten und sonstigen Sonderzeichen unter HTML gelöst wird; dies stellt ja meistens die unangenehmste Hürde bei dem Wechsel zwischen den diversen Systemen dar, da sich diese gerade darüber nicht einigen konnten (sogar DOS und Windows - beide von Microsoft (Winzigweich) - benützen nicht die gleiche Sonderzeichenbelegung). HTML kennt zwar auch eine numerische Zeichenbelegung (Unicode genannt), die sogar - zumindest bei den gewöhnlichen Zeichen «a-z» etc. - mit den üblichen Zeichen übereinstimmen; darüberhinaus gibt es diese aber auch für alle Sonderzeichen, diese wiederum sind HTML-spezifisch (z.B. ist dem «ü» der Unicode "ü" zugewiesen). Es handelt sich dabei um den sogenannten "Zeichensatz iso-8859-1" bzw. "Western Latin-1" der in HTML Version 3.2 definiert wurde.
Interessanter finde ich, daß auch "benannte Zeichen" definiert wurden, die mir recht hilfreich erscheinen. Benannte Zeichen haben immer den gleichen Aufbau: Zuerst wird ein kaufmännisches Und (&) notiert, dahinter den Namen des benannten Zeichens, und abschließend ein Semikolon (;). So muß beispielsweise die Zeichenfolge **ü** notiert werden, um ein kleingeschriebenes "ü" zu erzeugen. QL-mäßig interessiert uns darüberhinaus noch der zusätzliche Zeichensatz - unter HTML 4.0 definiert - der z.B. sämtliche griechische Zeichen beinhaltet (da der QL-MGG Zeichensatz zumindest einzelne griechische Zeichen enthält). Das große Alpha-Zeichen wird unter HTML 4.0 mit **Α** beschrieben oder mit dem Unicode **Α** bezeichnet, das kleine Alpha-Zeichen, das beim QL verwendung findet wird **α** oder **α** notiert. Nachteil: die wenigsten aktuellen Browser sind derzeit in der Lage diese Zeichen umzusetzen... macht aber nichts, sie schreiben dann halt einfach **α** in den Text; wer will, versteht. Mehr zu diesem Thema gibt es im Internet unter:
http://www.netzwelt.com/selfhtml/the.htm
Soviel zur Theorie.

Was wir nun brauchen ist ein kleines Programm, das uns unsere QL-Texte in eine entsprechende HTML-Form bringt. Das Programm habe ich in Assembler geschrieben und als «Filter» konzipiert. Ein Filter ist ein Programm, das mit EXEC oder EX aufgerufen wird und zwei Parameter enthält. Diese Parameter können sowohl Devices wie CON, SER, etc... als auch Filenamen sein. Der erste Parameter stellt die Quelle und der zweite das Ergebnis dar.

Beispiele:
`EX win1_tools_Q2H_exe,CON,SCR`
zeigt direkt am Bildschirm, was mit den Zeichen, die man eintippt passiert.

Benützt man den QemuLator (Win oder Mac) kann direkt in ein Windows- bzw. Mac-Verzeichnis (win2_) geschrieben werden:
`EX win1_tools_Q2H_exe, win1_texte_brief_9.2.2000_txt,"win2_html_brief_9-2-2000.html"`

Mit QPC oder einem QL wird man das Ergebnis vorzugsweise auf eine Diskette im PC-Format speichern.
`EX win1_tools_Q2H_exe, win1_texte_brief_9.2.2000_txt, "flp1_9-2-2000.htm"`

Programmbeschreibung

Der Vorteil bei einem Filter ist, daß ich mich nicht darum kümmern muß, Devices bzw. Files zu öffnen, das erledigt das Betriebssystem für mich. Ich kann mir die Channel-IDs gleich nach dem Start vom Stapel holen. Dann schreibe ich erstmals einen HTML-Header ("‹html›‹header›‹title› ...") in den Ausgangs-Channel. Um einerseits den Prozeß zu beschleunigen und andererseits die Bearbeitung von Dateien, die wesentlich größer sein können als der verfügbare freie Speicher, zu ermöglichen, hole ich mir jeweils die Daten in 64 Bytes große Happen. Für jedes Zeichen (Byte) schaue ich in einer Tabelle nach, ob und wodurch es zu ersetzen ist. Die Tabelle kann leicht von jedermann nach eigenen Bedürfnissen verändert werden, danach muß das Programm selbstverständlich neu assembliert werden.

Besonderheiten

Leider kennt HTML keinen Tabulator, so wird CHR$(9) durch 3 Leerzeichen ersetzt. Programme mit vielen Tabs sollte man daher vorerst durch ein anderes Filter laufen lassen, daß die Tabs durch entsprechen viele Spaces ersetzt. Der Zeilensprung wird durch ‹BR› ergänzt, so daß das Ergebnis sowohl HTML-tauglich als auch am QL noch halbwegs lesbar ist. Das seltsamste QL-Zeichen ist das seitenverkehrte S mit einem Punkt darunter (Code#181) , ich habe mir erlaubt dieses in HTML als Euro-Zeichen darzustellen.
Das Programm eignet sich lediglich, um gewöhnliche QL-Texte in HTML umzuwandeln. Um Quill-Doc-Files sowie Files anderer Editoren umzuwanden, müssen diese zuerst exportiert werden. Sollte dabei allzuviel Information verlorengehen (Schriftgröße, Farbe, Zeichensatz, etc...) gibt es zwei Möglichkeiten: Die einfachere bestünde darin, das File zu exportieren, umzuwandeln und dann mit einen der unzähligen HTML-Editor unter Windows/MacOS/etc. Relativ leicht wieder in Form zu bringen oder mein Programm dem entsprechenden Editor so zu adaptieren, daß dies alles automatisch stattfindet - viel Vergnügen!

Das Programm und die Source sind unter
http://altern.org/kuelat/d_prog.html
zu finden.

```
*
* Umwandlung einer QL-Textdatei mit QL-MGG Zeichensatz
*  nach HTML-Bestimmungen
*
* Aufruf: EX dev1_Q2H_exe,dev2_QUELLE_txt,dev3_ZIEL_html
*
*   G.Plavec (Freeware) Version vom 24. Feber 2000
*

HTM_JOB EQU     -1      Laufender Job
ERR.NC  EQU     -1      Nicht Vollständig
ERR.EF  EQU     -10     Ende der Datei
MT.FRJOB EQU    $05     Erzwinge die Löschung des Jobs
IO.FSTRG EQU    $03     Hole String
IO.SSTRG EQU    $07     Sende String
UT.ERRO EQU     $CA     Fehlermeldung an Channel 0
UT.MTEXT EQU    $D0     Sende Message
BUF_LEN EQU     $40     Buffer-Länge


start   BRA.S   init            Sprung zum Code

        DS.B    4
        DC.W    $4AFB           Kennung für einen Job
        DC.B    0,3,"Q2H"       Job Name
        EVEN
        DS.W    0               Ende Kennung

init    MOVE.W  (SP)+,D7        Anzahl der Channels
        SUBQ.W  #2,D7           es müssen 2 sein
        BNE     suizid

        MOVE.L  (SP)+,A4        hole die Eingangs- und
        MOVE.L  (SP)+,A0        Ausgangschannel-IDs

* HTML-Preambel senden
        MOVE.W  UT.MTEXT,A2     Sende Message
        LEA     PRE(PC),A1      Pointer auf Preambel
        JSR     (A2)

        TST.L   D0
        BNE     suizid          ... oops

htm_holen
        LEA     BUFFER(PC),A6   Buffer Pointer
        MOVEQ   #IO.FSTRG,D0    hole eine Anzahl an Bytes
        MOVEQ   #BUF_LEN,D2
        MOVEQ   #10,D3          warte ein Fünftel einer
Sekunde
        MOVE.L  A6,A1       Buffer Pointer
        EXG     A0,A4       auf Eingangs-Channel-ID setzen
        TRAP    #3

        EXG     A4,A0       Channel-ID auf Ausgang setzen
        MOVE.L  D0,D4       Error Flag retten
        MOVE.W  D1,D6       Anzahl der zu bearbeitenden
```

```
Bytes
        BEQ.S   htm_test        ... keine

        MOVE.L  A6,A1           Arbeitsregister
        MOVEQ   #0,D5           bereits bearbeitete Bytes

htm_loop
        MOVEQ   #0,D0
        MOVE.B  (A1)+,D0        hole ein Zeichen
        LEA     MGG_TABL(PC),A5 Tabellenanfang
        ADD.L   D0,A5           Zeichen addieren
        ADD.L   D0,A5           2x denn jedes Zeichen
                        ; belegt ein Wort in der Tabelle
        MOVE.W  (A5),D0         in HTML umwandeln ?
        BEQ.S   htm_skip        ... nein

* Ausgabe der bisherigen Bytes
        MOVEQ   #-1,D2          Zähler wieder auf 0
        EXG     D2,D5           bereits bearbeitete Bytes
        EXG     A6,A1           Puffer alt<->neu
        TST.B   D2              gibt's überhaupt was ?
        BEQ.S   htm_weiter      auf Null prüfen

        JSR     ausgabe

        BNE.S   suizid

* Ausgabe des HTML-Zeichens
htm_weiter
        LEA     HTML(PC),A1     Zeiger auf Länge der Ausgabe
        ADD.W   (A5),A1         Zeiger auf HTML-String
        MOVE.W  UT.MTEXT,A2
        JSR     (A2)

        MOVE.L  A6,A1           hier geht's weiter (Zeiger)
        TST.L   D0              außer es gab ein Fehler
        BNE.S   suizid

htm_skip
        ADDQ.W  #1,D5
        SUBQ.W  #1,D6           die Zähler stellen
        BGT.S   htm_loop        ... bis Null

        MOVE.L  A6,A1           ab Buffer-Anfang
        MOVE.W  D5,D2           Anzahl der bearbeiteten
Bytes
        JSR     ausgabe

        BNE.S   suizid          ... oops

htm_test
        MOVE.L  D4,D0           teste den Lesevorgang
        BEQ.S   htm_holen       ... OK, weitere Bytes holen

        ADDQ.L  #-ERR.NC,D4     "not complete" ist der
                                ; Regelfall
        BEQ.S   htm_holen       ... weitere Bytes holen

        CMP.L   #ERR.EF-ERR.NC,D4     Dateiende = fertig
        BNE.S   suizid          ... oops

* HTML-Postambel senden
        MOVE.W  UT.MTEXT,A2
        LEA     post,A1         Zeiger auf Postambel
        JSR     (A2)

suizid
        MOVE.W  UT.ERRO,A2      Fehlermeldung ausgeben
                                ; Channel#0
        JSR     (A2)

        MOVE.L  D0,D3           Fehler returnieren
        MOVEQ   #MT.FRJOB,D0 Job zwangsweise löschen
        MOVEQ   #HTM_JOB,D1  und zwar sich selbst (Suizid)
        TRAP    #1

*******************************************************
* Unterprogramm: ausgabe
*
* A1.L = Erstes zu sendende Byte (wird aktualisiert)
* D2.W = Anzahl der zu sendenden Bytes -> D1.W
* A0.L = Ausgangs-ChannelID
* D0.L/D3.W werden gesetzt bzw. ausgewertet
*
ausgabe MOVEQ   #IO.SSTRG,D0    einen String senden
        MOVEQ   #-1,D3          ... ewig warten
        TRAP    #3

        TST.L   D0              alles klar ?
        RTS

*******************************************************

HTML

PRE     DC.B    0,49,"<HTML><HEAD><TITLE> QL-Text
                      </TITLE></HEAD><BODY>"
        EVEN
POST    DC.B    0,14,"</BODY></HTML>"
        EVEN

NL      DC.B    0,5,"<BR>",10   10
        EVEN
LF      DC.B    0,4,"<P>",10    13
        EVEN
SPACE   DC.B    0,6," "    32
        EVEN
TAB     DC.B    0,18,"   "
        EVEN
AMP     DC.B    0,5,"&amp;"     38
        EVEN
LT      DC.B    0,4,"&lt;"      60
        EVEN
POUND   DC.B    0,7,"&pound;"   96
        EVEN
COPY    DC.B    0,6,"&copy;"    127
        EVEN
auml    DC.B    0,6,"&auml;"    128
        EVEN
atilde  DC.B    0,8,"&atilde;"  129
        EVEN
aring   DC.B    0,7,"&aring;"   130
        EVEN
eacute  DC.B    0,8,"&eacute;"  131
        EVEN
ouml    DC.B    0,6,"&ouml;"    132
        EVEN
otilde  DC.B    0,8,"&otilde;"  133
        EVEN
oslash  DC.B    0,8,"&oslash;"  134
        EVEN
uuml    DC.B    0,6,"&uuml;"    135
        EVEN
ccedil  DC.B    0,8,"&ccedil;"  136
        EVEN
ntilde  DC.B    0,8,"&ntilde;"  137
        EVEN
aelig   DC.B    0,7,"&aelig;"   138
        EVEN
oe      DC.B    0,6,"&#156;"    139
        EVEN
```

```
aacute  DC.B    0,8,"&aacute;"  140
        EVEN
agrave  DC.B    0,8,"&agrave;"  141
        EVEN
acirc   DC.B    0,7,"&acirc;"   142
        EVEN
euml    DC.B    0,6,"&euml;"    143
        EVEN
egrave  DC.B    0,8,"&egrave;"  144
        EVEN
ecirc   DC.B    0,7,"&ecirc;"   145
        EVEN
iuml    DC.B    0,6,"&iuml;"    146
        EVEN
iacute  DC.B    0,8,"&iacute;"  147
        EVEN
igrave  DC.B    0,8,"&igrave;"  148
        EVEN
icirc   DC.B    0,7,"&icirc;"   149
        EVEN
oacute  DC.B    0,8,"&oacute;"  150
        EVEN
ograve  DC.B    0,8,"&ograve;"  151
        EVEN
ocirc   DC.B    0,7,"&ocirc;"   152
        EVEN
uacute  DC.B    0,8,"&uacute;"  153
        EVEN
ugrave  DC.B    0,8,"&ugrave;"  154
        EVEN
ucirc   DC.B    0,7,"&ucirc;"   155
        EVEN
szlig   DC.B    0,7,"&szlig;"   156
        EVEN
cent    DC.B    0,6,"&cent;"    157
        EVEN
yen     DC.B    0,5,"&yen;"     158
        EVEN
acute   DC.B    0,7,"&acute;"   159
        EVEN
gAuml   DC.B    0,6,"&Auml;"    160
        EVEN
gAtilde DC.B    0,8,"&Atilde;"  161
        EVEN
gAcirc  DC.B    0,7,"&Acirc;"   162
        EVEN
gEacute DC.B    0,8,"&Eacute;"  163
        EVEN
gOuml   DC.B    0,6,"&Ouml;"    164
        EVEN
gOtilde DC.B    0,8,"&Otilde;"  165
        EVEN
gOslash DC.B    0,8,"&Oslash;"  166
        EVEN
gUuml   DC.B    0,6,"&Uuml;"    167
        EVEN
gCcedil DC.B    0,8,"&Ccedil;"  168
        EVEN
gNtilde DC.B    0,8,"&Ntilde;"  169
        EVEN
gAElig  DC.B    0,7,"&AElig;"   170
        EVEN
goe     DC.B    0,6,"&#140;"    171
        EVEN

* Diese Zeichen können erst ab HTML 4.0 dargestellt werden

alpha   DC.B    0,7,"&alpha;"   172
        EVEN
delta   DC.B    0,7,"&delta;"   173
        EVEN
theta   DC.B    0,7,"&theta;"   174
        EVEN
lambda  DC.B    0,8,"&lambda;"  175
        EVEN
mu      DC.B    0,4,"&mu;"      176
        EVEN
pi      DC.B    0,4,"&pi;"      177
        EVEN
phi     DC.B    0,5,"&phi;"     178
        EVEN
iexcl   DC.B    0,7,"&iexcl;"   179
        EVEN
iquest  DC.B    0,8,"&iquest;"  180
        EVEN

* Ein MGG-QL zeigt hier ein seitenverkehrtes S mit einem
* Punkt darunter:
euro    DC.B    0,6,"&euro;"    181
        EVEN
sect    DC.B    0,6,"&sect;"    182
        EVEN
curren  DC.B    0,8,"&curren;"  183
        EVEN
laquo   DC.B    0,7,"&laquo;"   184
        EVEN
raquo   DC.B    0,7,"&raquo;"   185
        EVEN
deg     DC.B    0,5,"&deg;"     186
        EVEN
divide  DC.B    0,8,"&divide;"  187
        EVEN
larr    DC.B    0,6,"&larr;"    188
        EVEN
rarr    DC.B    0,6,"&rarr;"    189
        EVEN
uarr    DC.B    0,6,"&uarr;"    190
        EVEN
darr    DC.B    0,6,"&darr;"    191
        EVEN

* Ersatzzeichen für undefinierte codes

oops    DC.B    0,1,"!"
        EVEN

MGG_TABL
        DC.W    oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, TAB-HTML, NL-HTML, oops-html, oops-html,
LF-HTML, oops-html, oops-html
        DC.W    oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html
        DC.W    0,0,0,0,0,0,AMP-HTML,0,0,0,0,0,0,0,0,0
        DC.W    0,0,0,0,0,0,0,0,0,0,0,0,LT-HTML,0,0,0

        DC.W    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        DC.W    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        DC.W    POUND-HTML,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
        DC.W    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,COPY-HTML

        DC.W    auml-HTML, atilde-HTML, aring-HTML,
eacute-HTML, ouml-HTML, otilde-HTML, oslash-HTML,
uuml-HTML, ccedil-HTML, ntilde-HTML, aelig-HTML, oe-HTML,
aacute-HTML, agrave-HTML, acirc-HTML, euml-HTML
        DC.W    egrave-HTML, ecirc-HTML, iuml-HTML,
iacute-HTML, igrave-HTML, icirc-HTML, oacute-HTML,
ograve-HTML, ocirc-HTML, uacute-HTML, ugrave-HTML,
ucirc-HTML, szlig-HTML, cent-HTML, yen-HTML, acute-HTML
```

```
        DC.W    gAuml-HTML, gAtilde-HTML, gAcirc-HTML,
gEacute-HTML, gOuml-HTML, gOtilde-HTML, gOslash-HTML,
gUuml-HTML, gCcedil-HTML, gNtilde-HTML,
gAElig-HTML,goe-HTML, alpha-HTML, delta-HTML, theta-HTML,
lambda-HTML
        DC.W    mu-HTML, pi-HTML, phi-HTML, iexcl-HTML,
iquest-HTML, euro-HTML, sect-HTML, curren-HTML,
laquo-HTML, raquo-HTML, deg-HTML, divide-HTML, larr-HTML,
rarr-HTML, uarr-HTML, darr-HTML

        DC.W    oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html
        DC.W    oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html

        DC.W    oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html
        DC.W    oops-html, oops-html, oops-html,oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html, oops-html, oops-html, oops-html,
oops-html, oops-html
        EVEN

BUFFER
* BUF_LEN-8 bytes reicht, da die Ein- und Ausgabe channel
IDs überschrieben werden können.
        DATA    BUF_LEN-8

        end
```

Sein oder nicht sein... läuft er noch?

Jochen Merz

Vergangene Woche fragte mich ein QLer, wie er denn feststellen könnte, ob ein Job mit einem bestimmten Namen gerade läuft oder nicht - wenn er nur so "auf gut Glück" RJOB nimmt kann es ja eine Fehlermeldung geben, wenn der Job zur Zeit nicht läuft.

Nun, es gibt verschiedene Möglichkeiten dieses Problem anzugehen. Wer's einfach mag, der listet einfach alle JOBS in eine Datei und schaut sie durch - mit INSTR ist ja ganz leicht herauszufinden, ob es den Job gibt oder nicht. Natürlich ist diese Lösung nicht perfekt, aber es funktioniert in den meisten Fällen.

Toolkit 2 bietet eigene Funktionen um alle Jobs im System durchzugehen bzw. alle Tochterjobs eines bestimmten Jobs durchzugehen. Hierzu dient die Funktion NXJOB. Sie will zwei Parameter: die Job-ID des aktuellen Jobs und die Job-ID[1] des Jobs, von dem gestartet werden soll. Der aktuelle Job ist am Anfang 0 (also SuperBASIC) und wir wollen alle Jobs durchsuchen, daher ist der Start-Job immer 0 (SuperBASIC stellt auch die Spitze des Job-Baumes dar - alle später gestarteten Jobs sind immer Töchter von Job 0).

Ruft man nun diese Funktion in einer Schleife auf und nimmt als aktuellen Job die Nummer, die NXJOB beim vorherigen Mal zurückgegeben hat, so findet man der Reihe nach (wie JOBS sie auflistet) alle Jobs. Auch diese Lösung ist nicht perfekt, sie hat im Prinzip das gleiche Manko wie die erste Lösung, obschon sie schöner ist.

Sowohl Job-Baum "durchsuchen" als auch eine Jobs-Liste als Datei "durchsuchen" sind Aktionen, die von anderen Aktionen durch das Multitasking unterbrochen werden können und sicherlich auch werden. Ein Job, der in der Jobliste gelistet wurde, kann im nächsten Moment schon verschwunden sein und dann, wenn der RJOB-Befehl ausgeführt wird, schon gar nicht mehr existieren. Oder ein anderer Job gleichen Namens kann in der Zwischenzeit den vorherigen ersetzt haben.

Letzteres ist im zweiten Fall nicht so tragisch, da Jobs über die ID identifiziert werden - und die ist beim Ersetzen eines Jobs durch einen anderen Job gleichen Namens glücklicherweise nicht gleich - aber auch hier würde RJOB eine Fehlermeldung ergeben, weil es den Job mit der gewünschten ID nicht mehr gibt. Auch kann die ID beim Durchsuchen des Job-Baums mit NXJOB auf einmal nicht mehr gültig sein, weil der Job in der Zeit, in der Schleife durchlaufen wird, entfernt wurde. In diesem Fall gibt NXJOB keine ID zurück sondern -1 ... so weiß man, es ist etwas passiert und man scannt den Baum am besten noch einmal.

Selbst wenn wir eine Funktion schreiben die den Baum in einem Rutsch ununterbrechbar durchsucht, dann kann theoretisch immer noch zwischen den beiden Befehlen "Suchen (wie auch immer es heißen mag)" und RJOB der Job verschwinden.

Sie sehen, in BASIC gibt es keine Möglichkeit, hier absolut fehlerfrei den Job zu entfernen - Sie kommen um ein WHEN ERRor zum Abfangen dieser Möglichkeit nicht umhin.

In Maschinencode gäbe es eine Lösung - man müßte den Prozessor vor dem Durchsuchen des Job-

1: Job-ID: eine Zahl, die einen Job im laufenden System identifiziert. Die Job-ID besteht aus einer Job-Nummer und einem Zähler ("Tag" genannt). Der Zähler wird mit jedem neu angelegten Job eins hochgezählt. So wird sichergestellt, daß ein Job, der gelöscht wird und durch einen Job mit der gleichen Jobnummer belegt wird, eine andere Identitätsnummer erhält und nicht mit dem neuen Job verwechselt wird.

Baums bis nach Entfernen des gewünschten Jobs in den Supervisor-Modus versetzen, dann kommen während dieser Zeit keine anderen Jobs dran.
Hier ein paar Zeilen, mit denen das Durchsuchen nach Jobs in BASIC realisiert werden kann:

```
100 DEFine FuNction job_exist(jobname$)
110 LOCal id
120 id=0
130 REPeat
140  id=NXJOB(id,0)
150  IF id<=0:RETurn id
160  IF (jobname$ INSTR JOB$(id))=1:RETurn id
170 END REPeat
180 END DEFine
```

Diese Funktion schaut nach, ob ein Job mit dem angegebenen Namen(sanfang) existiert und gibt die entsprechende Job-ID zurück. Wenn -11 zurückgegeben wird, dann ist beim Scannen etwas schiefgegangen. Wenn 0 zurückgegeben wird gibt es den Job nicht - versuchen Sie bitte nicht, Job 0 mit RJOB zu löschen, das kann bei älteren QDOS-Systemen tödlich sein! Neuere SMSQ/E ignorieren den Wunsch, Job 0 zu killen.

Ein "relativ" sicherer Weg, einen gewünschten Job (z.B. "Fred") zu entfernen, ist also

```
id=job_exist("Fred")
IF id>0: RJOB id,0
```

Wenn Sie wissen möchten, ob es mehrere Jobs gleichen Namens(anfangs) gibt, dann nehmen Sie einfach die leicht abgewandelte Funktion:

```
200 DEFine FuNction job_exnum(jobname$)
210 LOCal id,ct
220 id=0:ct=0
230 REPeat
240  id=NXJOB(id,0)
250  IF id<0:RETurn id
260  IF id=0:RETurn ct
270  IF (jobname$ INSTR JOB$(id))=1:ct=ct+1
280 END REPeat
290 END DEFine

PRINT job_exnum("QD")
```

zeigt Ihnen an wie viele QD's momentan laufen - wenn mehr als einer läuft ist es riskant, einfach zu entfernen - man könnte ja den Falschen löschen - besser mit Menü nachfragen!
Und wie gesagt, an den Anfang des Programms in WHEN ERRor den Test auf Fehler -2 (ungültige Job ID) mit einbauen, damit das Programm im ungünstigen Falle eines Falles nicht mit einer Fehlermeldung abbricht.
Hiermit sind dem geneigten Leser sicherlich genügend Denkanstöße für das sichere Löschen eines oder mehrerer gezielter Jobs gegeben.

Q40 News

Claus Graf

Bildbetrachter PQIV

Für den Q40 gibt es ein Programm zum Anschauen von Bildern: pqiv. Dieses Programm nutzt die Bildschirmmodi mit 65535 Farben des Q40, daher benötigt es mindestens SMSQ/E in der Version 2.95. Im Augenblick können folgende Bilddateien gelesen und angezeigt werden:

- PCX
- PNG
- PIC
- GIF

Das Programm ist für das Pointer Environment geschrieben und läßt sich hoffentlich ohne Anleitung bedienen. Als nächstes ist eine Möglichkeit zum Speichern geplant, so daß PCX, PNG oder GIF in PIC umgewandelt werden können und Speichern als Hintergrundbild. Mit dem SBasic-Befehl BGIMAGE kann man sich dann ein tolles Hintergrundbild auf den Q40 Bildschirm zaubern. PQIV kostet in der aktuellen Version nichts und kann von der Q40 Website
http://www.q40.de
heruntergeladen werden.

JMS - In letzter Minute

Es gibt doch noch ein paar Neuerung seit Schreiben des Editorials: Wir haben neue Email-Adressen für Mails an QL-Today!

QLToday@j-m-s.de für die Editor der deutschen Ausgabe.
QLToday@j-m-s.com für die Editor der englischen Ausgabe.
Bitte nutzen Sie in Zukunft diese Email-Adressen,

In dieser Ausgabe war leider kein Platz mehr für die Zusammenfassung der englischen Ausgabe. Wir reichen sie in der nächsten Ausgabe nach.

Die Mailbox ist deutlich aufgeräumter - alter Kram ist raus, einfach mal reinschauen: **0203 502013** für 14.400 und ZyXel 16.800 und **0203 502014** für bis zu 33.600er Modems.

Um jederzeit über News informiert zu werden empfehle ich, sich in die QL News Liste einzutragen. Dies geht über die J-M-S Website
http://www.j-m-s.com/smsq/index.htm
Sie erhalten dann eine Email wenn neue Versionen von JMS-Programmen in der Mailbox zu finden sind.

QL-Treffen 2000

Auch in diesem Jahr stehen wieder QL-Treffen auf dem Plan:

Sonntag, 16. April - Davyhulme, UK
Quanta AGM - findet in Nord-England statt und wird eigentlich nur der Vollständigkeit halber erwähnt - deutschsprachige Leser werden wohl nicht hinfahren. Details im englischen QL Today.

Samstag, 13. Mai - Eindhoven
Natürlich wieder im St.-Joris College, gleicher Platz wie immer. Beginn um 10 Uhr, Ende gegen 16 Uhr.

Samstag, 20. Mai - Vermont
Wer noch nie in den USA war oder wer wieder einen Grund sucht hinüber zu fliegen, der hat ihn nun gefunden: QL-Treffen ist angesagt! Diesmal ziemlich nord-östlich gelegen stehen viele Flughäfen zur Verfügung, Kanada (und die Niagara-Fälle) sind auch nicht weit und in der Gegend gibt es viel zu sehen.
Nach dem Motto "Wer früh bucht, bezahlt am wenigsten" sollten sich Interessenten möglichst bald ein Flugticket besorgen und auch ein Hotelzimmer besorgen. Bill Cable von NESQLUC, die das diesjährige Treffen auch wieder ausrichten, hat preiswerte Hotelzimmer reserviert - auch hier gilt: Wer zuerst kommt hat die besten Chancen - daher nichts wie los mit der Planung!

Geplant:
QL-Treffen in Italien, vermutlich Ende September.
QL-Treffen in Österreich, vermutlich Anfang Oktober.

Sa. & So., 14. und 15. Oktober
"Das" QL-2000-Treffen in England!
Bitte unbedingt vormerken!
Datum ist bestätigt!

... und in der nächsten Ausgabe von QL Today Deutsch kommt der lang ersehnte nächste Teil der Serie über QL-EPROMs von Frank Dibowski!