

QLib_run v3.35mod
=====

Eureka ! Ça y est... QLiberator peut enfin profiter pleinement du traitement des erreurs. Deux petits POKE et adieux les vilains bogues !

La version "3.35mod" de QLib_run corrige deux bogues de la version 3.35 qui empêchait la récupération du code d'erreur avec ERNUM (ce dernier était en fait équivalent à ERLIN !). Elle remplacera avantageusement toute version antérieure (compatibilité ascendante totale).

Mais en quoi consiste donc la correction de ces bogues ?

Il faut tout d'abord savoir que ERLIN et ERNUM font partie intégrante du code de QLib_run (même si vous possédez une version du QDOS qui intègre ces instructions, c'est le code contenu dans QLib_run qui sera exécuté !). Ceci permet de faire profiter les versions AH et JM du QDOS de ces instructions (du moins pour un programme compilé).

De plus lorsque QLib_run rencontre une erreur, il stocke son code et le numéro de ligne à laquelle elle s'est produite dans deux variables du pseudo-entête Super Basic (pour assurer une parfaite compatibilité, QLiberator crée ce pseudo-entête qui contient l'équivalent des variables Super Basic de l'interpréteur). Ces deux variables doivent normalement (et par analogie au Super Basic des versions JS et au-delà), se trouver aux offsets \$C2 et \$C6 par rapport à la base du job (dont l'adresse est contenue dans le registre A6). \$C2(A6) est un mot long contenant le code de la dernière erreur et \$C6(A6) est un mot contenant le numéro de ligne.

Or la version 3.35 de QLib_run stocke le code d'erreur sur un mot long en \$C4(A6) et le numéro de ligne en \$C6(A6) sur un mot. Résultat: le mot de poids faible du code d'erreur est écrasé par le numéro de ligne (le mot de poids fort contenant toujours \$FFFF pour une erreur QDOS) ! Donc, non seulement le code d'erreur n'est pas stocké au bon endroit, mais en plus il est irrécupérable (même par une extension "maison") ! D'autre part et pour tout arranger, le "ERNUM" de QLib_run allait chercher le code d'erreur en \$C6(A6) sur un mot, renvoyant en fait le numéro de ligne...

Le remède consiste donc à modifier les instructions incriminées pour les forcer à écrire et lire le code d'erreur au bon endroit. Le petit programme Super Basic suivant permet de réaliser ces modifications:

Eureka! That is ... QLiberator can finally take full advantage of error handling. Two small POKES and goodbye villainous bugs!

The "3.35mod" version of QLib_run fixes two bugs of version 3.35 which prevented the error code from being recovered with ERNUM (the latter was actually equivalent to ERLIN!). It will replace any previous version (total upward compatibility).

But what is the correction of these bugs?

First of all, ERLIN and ERNUM are an integral part of the QLib_run code (even if you have a version of the QDOS that includes these instructions, the code in QLib_run will be executed!).

This allows the AH and JM versions of the QDOS to benefit from these instructions (at least for a compiled program).

In addition, when QLib_run encounters an error, it stores its code and the line number where it occurred in two variables of the Super Basic pseudo-header (to ensure perfect compatibility, QLiberator creates this pseudo-header, equivalent to the interpreter's Super Basic variables). These two variables should normally be (and by analogy with the SuperBASIC in JS versions and beyond), offset to the \$ C2 and \$ C6 offsets from the job base (whose address is contained in the A6 register). \$ C2 (A6) is a long word containing the code of the last error and \$ C6 (A6) is a word containing the line number.

Now, version 3.35 of QLib_run stores the error code on a long word in \$ C4 (A6) and the line number in \$ C6 (A6) on a word. Result: The low-order word of the error code is overwritten by the line number (the high-order word always contains \$ FFFF for a QDOS error)! So, not only is the error code not stored in the right place, but it is irretrievable (even with a "home" extension)! On the other hand and to fix everything, the "ERNUM" of QLib_run would look for the error code in \$ C6 (A6) on a word, in fact returning the line number ...

The remedy is therefore to modify the offending instructions to force them to write and read the error code in the right place. The following small Super Basic program makes these changes possible:

```
100 REMark Charger QLib_run en mémoire:
110 L=10552:REMark L=10624 pour QLib_run v3.36
120 AD=ALCHP(L):LBYTES "flp1_QLib_run",AD
130 :
140 REMark Placer $C4 à l'offset $0D10:
150 POKE_W AD+3344,196
160 :
170 REMark Placer $C2 à l'offset $146E:
180 POKE_W AD+5230,194
190 :
200 REMark Sauver la nouvelle version de QLib_run:
210 SBYTES "flp1_new_QLib_run",AD,L:RECHP AD
220 :
230 REMark C'est tout !
```

T. Godefroy. 1991