QHELP: An On-Line Help System

David J. Greer
Robelle Consulting Ltd.

SUMMARY

QHELP is a software tool that provides an interactive help facility for on-line programs. QHELP is designed to be easy to set up and maintain, efficient at run time, and convenient for the end users. QHELP eliminates duplication by allowing the user manual and the help text to be the same file. A series of keywords defines a "tree" which is used to allow the end user easy access to any part of the help text.

Contents

INTRODUCTION

Most application subsystems and software utilities would benefit from an on-line HELP facility for users. QHELP is such a facility, and it does not require a great deal of system or programmer resources. QHELP makes it easy to maintain help text in an external QEDIT file.

For utility tools (i.e., QEDIT, SUPRTOOL, etc.), the HELP file would primarily explain the commands available in the tool, but would also give examples, uses, sample results and news of recent changes. One reason we developed QHELP was to allow Robelle software products to have a greatly expanded "HELP" capability.

For application programs, a HELP file would document what ach module in an application is supposed to do, in language the end user can understand. The benefits of using HELP files are: (1) they can be part of the specifications during the design stage of a project; (2) they assist in user training, eliminating many questions; (3) they are more accessible than written documentation and are much easier to keep up-to-date.

Ad hoc HELP systems tend to skimp on information for the user and are often out of date because they are not easy to maintain. A standard HELP system should be flexible, easy to program, low in overhead and hierarchical (many key levels), with many indices for quick retrieval by the user.

QHELP provides an easy means for the COBOL programmer (or the SPL or FORTRAN programmer) to code the HELP capability into an application program. The structure of the HELP text gives the user access to the information which is immediately necessary to the task at hand. In addition, QHELP is designed to consume a minimum of system resources when it isn't being used.

Why Not Use the MPE HELP Facility?

The MPE HELP facility has several problems that make it awkward to use. One problem is that the HELP text can only be organized on two levels. Many applications will be easier for the user to understand if the help material can be decomposed into more than two levels of detail. "Subdivision" should make it easier to maintain the documentation (increasing the probability that maintenance will actually be done).

The MPE HELP system is not designed to be used from a COBOL program. Without writing some SPL interfaces, it is not possible for the COBOL programmer to access the MPE HELP facility. In addition, the MPE HELP files must be updated with user documentation every time an update is done to MPE.

With the MPE HELP facility, HELP text cannot be broken down into different files. Ideally, every programmer should maintain the HELP documentation for each module/program that he writes. This is done most easily by having one HELP file per module/program; but it should be easy to connect all of the HELP files and make them look like an integrated whole to the user.

Is QHELP Any Better?

QHELP solves all of the problems mentioned above. Within QHELP, entries can be organized into as many as ten levels of keyword indices. QHELP keeps the help files open and saves indices at every level of the HELP file so that searches are fast. QHELP uses QEDIT work files, which consume less disc space and are more efficient than regular KEEP files. QHELP provides modular files (similar to $INCLUDE) so that a system of HELP files can be easily integrated into one HELP package. Finally, the programmatic interface to QHELP is flexible. The *FIND* and *UNFIND* commands, for example, allow an application subprogram to position the HELP file pointers to the HELP text for that subprogram, without knowing anything about

"higher" levels in the HELP file (i.e., QHELP provides relative indexing).

QHELP is distributed to all users of Robelle software products as part of the QLIB contributed library (no extra charge). Users are authorized to merge QHELP into their own software, without restriction. QHELP works only with QEDIT files. There are four reasons for this: (1) QEDIT files provide data compression (reducing the cost of the help facility); (2) QEDIT files can be easily and efficiently modified (if you have QEDIT); (3) QEDIT files provide fast random access (as required for the recursive tree structure); and 4) we may sell more copies of QEDIT by tying QHELP to it.

STRUCTURE OF A QHELP FILE

The basic structure of a QHELP file consists of a key (e.g., QEDIT), some text about the key, and, optionally, lower level keys (e.g., NEWS, Add, Change . . .). The structure is recursive: lower level keys can also have text and more keys. A new key is specified by the \BEGINKEY command. The end of a key is indicated by the \ENDKEY command. Everything between the \BEGINKEY command and the \ENDKEY command is considered to be part of the key.

The following is an extract from an early HELP file for QEDIT:

```
\BEGINKEY QEDIT

                QEDIT Capsule Summary

/S LANG=COB         Select COBOL as the current 'language'.
/TEXT SRC2=SRC1     Make a new QEDIT file SRC2 and copy SRC1 into it.
/OPEN SRC2          Select SRC2 as the current workfile for editing.
/LIST 3/13          Display lines from SRC2 (current workfile).
/LIST SRC1 5/10     List lines 5 thru 10 of SRC1 (not current file).
/LIST \BAL-DUE\     List all lines of the workfile with BAL-DUE in them.
/MOD [ ];D 63.5     Modify FIRST and LAST ([ ]) lines, delete line 63.5.
/ADD 5              Add new lines to workfile at or after line 5.
/ADD 50 = 70/78     Make a copy of lines 70 thru 78 after line 50.
/ADD 40 < 20/30     Move lines 20 thru 30 to after line 40.
/ADD 20 = TXT4      Copy all of the file TXT4 after line 20.
/C "XY"ABC"@        Change each string "XX" to "ABC" in ALL lines.
/:COBOL *           Compile the current workfile, listing on terminal.
/:PREP              Preps $OLDPASS into $NEWPASS; see /S OPT MAX.
/:RUN               Runs $OLDPASS (result of :PREP); see /S OPT LIB.
/:STREAM *          Stream the /OPEN workfile as a batch job.
Special keys:       Control-Y (stop commands), Control-X (cancel input),
                    Control-S (suspend listing until Control-Q).
/HELP INTRO         Print more HELP (try /H NEWS, /H ADD, etc.).
/EXIT               Leave QEDIT and return to MPE for :BYE.

\BEGINKEY INTRO
              QEDIT:  Introducing Terms

  Term:                Explanation:
Workfile?              File built via NEW/TEXT that QEDIT edits (code=111)
                       QEDITSCR is the default, temporary file if none spec.
Current Workfile?      The QEDIT file that is currently /OPEN for editing.
External File?         Not the current workfile; QEDIT can /LIST any file.
Language?              Each file has 'lang' attached to it (COB,RPG,FTN,SPL,
                       JOB,TEXT for >80 columns,COBX to use comment field).
Command?               Defined by first letter (A=add), lower-case is okay,
                       and semicolon(;) sets off multiple commands per line.
        ADD       DELETE   GALLEY   LIST     OPEN     REPLACE  USE
        BEFORE    EXIT     HELP     MODIFY   PROC     SET      VERIFY
        CHANGE    FIND     KEEP     NEW      Q        TEXT     ZAVE
Command Option?     Second char modifies command action (LQ, AJ, KJ)
        T:  template option (prints column headings)
        Q:   quiet option (without linenumbers or without printing)
```

```
        J:   justified (/ADD), jumping (/L), window only (/K)
MPE Command?        Needs a colon(:), /:LISTF, /:STREAM, /:RUN, /:PREP.
Rangelist?          List of ranges (5/7,9) or string match ("bb" [range]).
Range?              Lines: 5, 5/7, @=ALL, [=FIRST, *=curr, 5/=5/LAST.
String?             Char in quotes("bb",-bb-,\bb\) to find or change.
Window?             Where/How to match strings: /S W=(1/50,SMART).
User Manual?        Enter /GALLEY QMANUAL.DOC.ROBELLE LP to print manual.

\ENDKEY INTRO
\BEGINKEY NEWS

     ...  NEWS about the latest version of QEDIT.

\ENDKEY NEWS

\BEGINKEY A

     ...  The ADD command is described here.

\ENDKEY A

     .
     .
     .

     The rest of the QEDIT commands are entered here.

\BEGINKEY S

     ...  A general description of the SET command goes here.

\BEGINKEY GENERAL

     ...  This key belongs to the SET command.  It describes all
          parts of the set command except for /SET OPTION.

\ENDKEY GENERAL

\BEGINKEY OPTION

     ...  Each of the /SET OPTION parameters is described here.

\ENDKEY OPTION

\ENDKEY S

\BEGINKEY Z

     ...  The ZAVE command is the last one in the HELP file.

\ENDKEY Z

\ENDKEY QEDIT
```

There are several important points to note: 1) The \BEGINKEY command and the \ENDKEY command may be spelled in any combination of upper- and lower-case letters. Within QHELP, all key names are upshifted. 2) The key for this HELP file is QEDIT (the first line in the file must contain a \BEGINKEY command). The keys that belong within QEDIT are INTRO, NEWS, A, . . . , Z. 3) Each \BEGINKEY command has a matching \ENDKEY command. The key name on the \BEGINKEY command and \ENDKEY commands must be identical.

Any key may have sub-keys associated with it. For example, the SET command above contains "GENERAL" and "OPTION" as keys. These two keywords are then associated with the SET command rather than with QEDIT.

Using QHELP with QGALLEY

Any lines in the HELP file that start with a backslash, but do not contain one of the HELP commands (BEGINKEY, ENDKEY, BEGININDEX, ENDINDEX), are retained, but ignored, by the QHELP system. This allows you to embed QGALLEY (a modified version of GALLEY) commands in the HELP file to specify formatting. Thus, the HELP file can be used as part of the user manual.

Example:

```
\BEGINKEY QEDIT
\FORMAT
    (text)
\IMAGE
    (list of commands)
\ENDKEY QEDIT
```

Any text between one \ENDKEY and the next \BEGINKEY will be ignored by QHELP, but will be picked up by QGALLEY and PROSE. A useful technique is to "include" the help file into the file for the user manual, thus separating the title page and the table of contents from the actual text. The OUTQ command of QGALLEY should be used to "cleanup" any help files that you have modified (in order to adjust lines so that they again fit neatly within the margins). Then, the file created by OUTQ can be processed through the QHELP compiler.

INTERACTING WITH QHELP

Assuming that you have RUN QHELP.QLIB and asked for the help file named QEDIT.HELP.ROBELLE, what happens next? QHELP prints the initial block of text from the help file (the text between \BEGINKEY QEDIT and \BEGININDEX INTRO, the first nested key word):

```
:HELLO user.account
:RUN QHELP.QLIB.ROBELLE

QHELP/QLIB/ROBELLE Consulting Ltd.(C) 1981
(Version 0.2)

Enter HELP Filename? QEDIT.HELP.ROBELLE

                QEDIT Capsule Summary

/S LANG=COB         Select COBOL as the current 'language'.
/TEXT SRC2=SRC1     Make a new QEDIT file SRC2 and copy SRC1 into it.
      .
      .
      .
/HELP INTRO         Print more HELP (try /H NEWS, /H ADD, etc.).
/EXIT               Leave QEDIT and return to MPE for :BYE.

LO: Keywords Under: QEDIT

INTRO,NEWS,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
```

```
>NEWS
```

QHELP prompts with a ">" character for the user to enter a keyword or a special command. If the user enters a valid keyword, all of the information for that keyword is displayed, along with a list of any sub-keys. In the example above, the user may type NEWS to get information on new features of QEDIT.

The user need not type the entire key word, since QHELP normally matches on any leading sub-string of the key (this feature can be disabled with a *SET* command). In this example, the user could have typed N, NE, or NEW to get NEWS (notice one disadvantage of this feature: the user cannot get to the N keyword for the \N command; some care must be given to the naming and ordering of key words).

Whenever QHELP is printing text, the user may strike Control-Y to interrupt the printout. QHELP will usually prompt for another key word.

The user leaves QHELP by typing 'exit', or by typing a circumflex (^), instead of a key name. Because the user may be several levels deep into the HELP file structure, he may have to type 'exit' several times (once for each level). Alternatively, several circumflexes (^) may be typed in a row; QHELP will return one level for each circumflex (e.g., entering >^^ exits two levels).

There are several special commands that may be entered when the user interacts with QHELP. One of these is the circumflex (^), which is used to exit the interact mode of QHELP. The question mark (?) prints information on the QHELP special commands. Another special command is the dollar sign ($), which is exactly the same as the QHELP *SET* command (e.g., $ LP ON equals SET LP ON).

Sometimes it is desirable to print all of the information under a specific level. The "at" sign (@) is a special

QHELP command (and key word) to do this.

The following example illustrates the use of these special commands. Starting at the node for QEDIT, output is directed to the line printer. Next, all of the information under QEDIT is printed on the line printer; then output is directed back to the terminal.

```
LO: Keywords Under: QEDIT

INTRO,NEWS,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
>$ LP ON

LO: Keywords Under: QEDIT

INTRO,NEWS,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
>@

LO: Keywords Under: QEDIT

INTRO,NEWS,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
>$ LP OFF
```

Suppose you want to see the help text associated with the key word GENERAL, nested under the key word S (for \SET command). One way to get this would be to enter the S keyword, wait for the S text and sub-keys to print, then enter the GENERAL key word, read the desired material and, finally, enter an "exit". Or, at the entry level to QHELP (i.e., level 0), you could enter both the key word and the sub-key, separated by commas; QHELP will follow the indicated path through the HELP file tree and print only the lowest level (then exit back to the original level).

```
>S,GENERAL
```

Suppose you want to see everything on the SET command. You can use the multiple key word option plus the "@" option to request all information under the S key word (Control-Y will bring you back to the original level immediately):

```
>S,@
```

USING QHELP FROM COBOL

QHELP is designed to make "help" information available to the end user with the minimum amount of involvement by the applications programmer. The COBOL programmer invokes QHELP by using a series of commands that are passed to the QHELP procedure through a standard communication area. The layout of the communication area is:

```
01  QHELP-AREA.
    05  QHELP-COMMAND              PIC X(80).
    05  QHELP-RESULT               PIC S9(4) COMP.
        88  QHELP-OK               VALUE ZEROS.
        88  QHELP-MISSING-KEY      VALUE 6.
    05  QHELP-BUFFER-LENGTH        PIC S9(4) COMP VALUE 200.
    05  FILLER                     PIC X(200).
```

To make maintenance easier, this area is normally declared in the COPYLIB, and copied into the source program with the COPY statement. All QHELP commands are moved into the QHELP-COMMAND buffer before calling QHELP, and QHELP returns the status of each call in the QHELP-RESULT variable. Like IMAGE, QHELP must be opened and initialized before it can be used. The *OPEN* command is used to start up the QHELP system, and to tell QHELP the name of the QHELP file. The following COBOL fragment opens a HELP file called EXAMPLE.HELP and initializes the QHELP system.

```
MOVE "OPEN EXAMPLE.HELP"          TO QHELP-COMMAND.
CALL "QHELP" USING QHELP-AREA.
IF NOT QHELP-OK THEN
    DISPLAY "ERROR:  CANNOT OPEN HELP FILE"
    MOVE FALSE                    TO QHELP-INITIALIZED-FLAG
ELSE
    MOVE TRUE                     TO QHELP-INITIALIZED-FLAG.
```

The applications program must check the user commands to see if the user requests help. When the user asks for help, the applications program must make one more call to QHELP, using the *PRINT* command.

```
MOVE "PRINT"                          TO QHELP-COMMAND.
CALL "QHELP" USING QHELP-AREA.
IF NOT QHELP-OK THEN
    DISPLAY "ERROR:  FAILURE OF QHELP ", QHELP-RESULT
    PERFORM END-OF-PROGRAM.
```

On the terminal, the result of the *PRINT* command would be as described above under "Interacting With QHELP".

The following example program demonstrates the use of QHELP. The program prompts the user for a command, and executes a separate module for each command. One of the valid commands is "HELP", and when the user types "HELP", QHELP is called.

```
$CONTROL SOURCE,ERRORS=5,LIST
 IDENTIFICATION DIVISION.
 PROGRAM-ID.  EXAMPLE.
 AUTHOR.      DAVID GREER, ROBELLE CONSULTING LTD.

 ENVIRONMENT DIVISION.

 DATA DIVISION.
 WORKING-STORAGE SECTION.
 01  TRUE                          PIC X VALUE "T".
 01  FALSE                         PIC X VALUE "F".

 01  QHELP-INITIALIZED-FLAG        PIC X.
     88  QHELP-INITIALIZED         VALUE "T".

 01  ACCEPT-BUFFER                 PIC X(80).
     88  ANSWER-SPACES             VALUE SPACES.

 01  ACCEPT-BUFFER-1 REDEFINES ACCEPT-BUFFER.
     05  ACC-ONE                   PIC X.
         88  ACC-COMMAND-EXIT      VALUE "E".
         88  ACC-COMMAND-ADD       VALUE "A".
         88  ACC-COMMAND-CHANGE    VALUE "C".
         88  ACC-COMMAND-DELETE    VALUE "D".
         88  ACC-COMMAND-HELP      VALUE "H".
     05  FILLER                    PIC X(79).


 01  QHELP-AREA.
     05  QHELP-COMMAND             PIC X(80).
     05  QHELP-RESULT              PIC S9(4) COMP.
         88  QHELP-OK              VALUE ZEROS.
         88  QHELP-MISSING-KEY     VALUE 6.
     05  QHELP-BUFFER-LENGTH       PIC S9(4) COMP VALUE 200.
     05  FILLER                    PIC X(200).

 PROCEDURE DIVISION.
 00-MAIN                            SECTION.

     PERFORM 05-INITIALIZE
        THRU 05-INITIALIZE-EXIT.

     IF QHELP-INITIALIZED THEN
         MOVE SPACES                TO ACCEPT-BUFFER
         PERFORM 10-MAIN-PROCESSING
            THRU 10-MAIN-PROCESSING-EXIT
           UNTIL ACC-COMMAND-EXIT.

     PERFORM 95-FINISH-UP
        THRU 95-FINISH-UP-EXIT.
```

```
 00-MAIN-EXIT.  GOBACK.

$PAGE "[05]  INITIALIZE"
 05-INITIALIZE                      SECTION.

     MOVE "OPEN EXAMPLE.HELP"    TO QHELP-COMMAND.
     CALL "QHELP" USING QHELP-AREA.
     IF NOT QHELP-OK THEN
         DISPLAY "ERROR:  CANNOT OPEN HELP FILE"
         MOVE FALSE              TO QHELP-INITIALIZED-FLAG
     ELSE
         MOVE TRUE               TO QHELP-INITIALIZED-FLAG.

 05-INITIALIZE-EXIT.  EXIT.

$PAGE "[10]  MAIN PROCESSING"
 10-MAIN-PROCESSING             SECTION.

     MOVE SPACES                TO ACCEPT-BUFFER.

     PERFORM 10-10-GET-COMMAND
       UNTIL NOT ANSWER-SPACES.

     IF ACC-COMMAND-HELP THEN
         PERFORM 10-20-CALL-QHELP
     ELSE
     IF ACC-COMMAND-ADD THEN
         PERFORM 20-PROCESS-ADD-COMMAND
            THRU 20-PROCESS-ADD-COMMAND-EXIT
     ELSE
     IF ACC-COMMAND-CHANGE THEN
         PERFORM 30-PROCESS-CHANGE-COMMAND
            THRU 30-PROCESS-CHANGE-COMMAND-EXIT
     ELSE
     IF ACC-COMMAND-DELETE THEN
         PERFORM 40-PROCESS-DELETE-COMMAND
            THRU 40-PROCESS-DELETE-COMMAND-EXIT
     ELSE
      IF NOT ACC-COMMAND-EXIT THEN
         DISPLAY "ERROR:  UNKNOW COMMAND NAME".

     GO TO 10-MAIN-PROCESSING-EXIT.

 10-10-GET-COMMAND.
     DISPLAY "ENTER COMMAND NAME".
     ACCEPT ACCEPT-BUFFER.

 10-20-CALL-QHELP.
     MOVE "PRINT"               TO QHELP-COMMAND.
     CALL "QHELP" USING QHELP-AREA.
     IF NOT QHELP-OK THEN
         PERFORM 99-FATAL-ERROR
            THRU 99-FATAL-ERROR-EXIT.

 10-MAIN-PROCESSING-EXIT.  EXIT.

$PAGE "[20]  PROCESS ADD COMMAND"
 20-PROCESS-ADD-COMMAND         SECTION.

     DISPLAY "PROCESSING FOR THE ADD COMMAND".

 20-PROCESS-ADD-COMMAND-EXIT.  EXIT.

$PAGE "[30]  PROCESS CHANGE COMMAND"
 30-PROCESS-CHANGE-COMMAND      SECTION.
```

```
        DISPLAY "PROCESSING FOR THE CHANGE COMMAND".

    30-PROCESS-CHANGE-COMMAND-EXIT.  EXIT.

$PAGE "[40]  PROCESS DELETE COMMAND"
    40-PROCESS-DELETE-COMMAND       SECTION.

        DISPLAY "PROCESSING FOR THE DELETE COMMAND".

    40-PROCESS-DELETE-COMMAND-EXIT.  EXIT.

$PAGE "[95]  FINISH UP"
    95-FINISH-UP                    SECTION.

        MOVE "CLOSE"                TO QHELP-COMMAND.
        CALL "QHELP" USING QHELP-AREA.
        IF NOT QHELP-OK THEN
            DISPLAY "ERROR:  UNABLE TO TERMINATE QHELP SYSTEM".

    95-FINISH-UP-EXIT.  EXIT.
$PAGE "[99]  FATAL ERROR"
    99-FATAL-ERROR                  SECTION.

        DISPLAY " ".
        DISPLAY "ERROR:  FATAL TERMINATION OF THE PROGRAM EXAMPLE".

        STOP RUN.

    99-FATAL-ERROR-EXIT.  EXIT.
```

This example does not cover all of the QHELP commands available to the COBOL programmer. It is possible, with the *FIND* and *UNFIND* commands, to position the internal QHELP pointers to various levels of the help tree. These commands could be used to position the HELP file to the HELP text associated with the add command, when processing the add command in the example above.

There is also a QHELP set command, which allows various QHELP defaults to be overridden. These include options such as the number of lines on a screen, and the type of pattern matching that QHELP should use when looking for keywords.

ADVANCED QHELP FILE STRUCTURE

The basic file organization of QHELP files is one file with many \BEGINKEY and \ENDKEY commands, where \BEGINKEY commands can be nested up to a level of 10. For very large systems this is too cumbersome. It is necessary to have separate HELP files for each major system module.

QHELP permits a single "logical" HELP file to be separated into several physical files (which can also act as stand-alone "logical" files). This is done by including an entire \BEGINKEY - \ENDKEY pair in one file. In the QEDIT example above, assume that all of the information having to do with the set command was to end up in a file called SET.HELP.ROBELLE. The primary QEDIT HELP file would then include the following entry:

```
\BEGINKEY SET [SET.HELP.ROBELLE]
\ENDKEY SET
```

Any key can be in another file, but nesting of files is only permitted to five levels. The [>]. indicates that the key is located in the file specified inside the [>]. The file SET.HELP.ROBELLE is the file which contains all of the HELP information about the QEDIT SET command. This file is a regular QHELP file, which starts with \BEGINKEY SET and ends with \ENDKEY SET.

The layout of the SET file would look like this:

```
\BEGINKEY SET

    ...  General information on what the /SET command does.

\BEGINKEY GENERAL


    ...  This key belongs to the SET command.  It describes all
         parts of the set command except /SET OPTION.
```

```
\ENDKEY GENERAL
\BEGINKEY OPTION
     ...  Each of the /SET OPTION parameters is described here.
\ENDKEY OPTION
\ENDKEY SET
```

The SET HELP file can also be used as an individual HELP file, in addition to acting as the entry for the SET key in the QEDIT HELP file.