Author(s) : W. Jank, D. Lellouch, R. Matthews, E. Pagiola, J. Zoll **Library:** KERNLIB
Submitter : **Submitted:** 28.08.1984
Language : Assembler or C **Revised:**

Restart of Next Event

This interface routine allows the user to restart his program at the entry point `QNEXTE`, provided he has initiated it at this same entry point.

For first entry, `QNEXTE` remembers all necessary internal Fortran parameters, such as registers, trace-back, stack pointers, signal mask, whatever is needed on a given machine, and then calls a user-supplied routine `QNEXT`.

On any subsequent entry, `QNEXTE` resets all internal parameters so as to cancel all open `CALL`s below its own level, and then transfers again control to `QNEXT`. If in `QNEXT` a `RETURN` statement is reached this will lead back to the routine which did the first call to `QNEXTE`, usually the `MAIN` program.

Structure:

(Pseudo) `SUBROUTINE` subprogram
User Entry Names: `QNEXTE`
Internal Entry Names: `QNEXTD` (on Vax)
External References: User-supplied `SUBROUTINE` subprogram `QNEXT` (Z041)

Usage:

```
CALL QNEXTE
```

will transfer control to the routine `QNEXT` supplied by the user, via a `CALL QNEXT` (no parameter list).

Notes:

`QNEXT` is a user routine which cannot be loaded implicitly from a library. If to be used at all, it has to be loaded explicitly, either from a load file (such as produced by the compiler) or by some form of `INCLUDE` from a user library.

Because `QNEXTE` is referenced by some general packages, whose user may not want to supply a `QNEXT`, the reference from `QNEXTE` to `QNEXT` has been made 'weak' (to avoid the 'missing external' message from the loader) on the Vax (and probably also on some other machines in the future). In this case `QNEXTE` has a call to a Fortran dummy routine `QNEXTD` to print a message if it is reached without the user having supplied a routine `QNEXT`.

On most UNIX machines the loader is not able to start a module with missing externals; in this case, the user is obliged to provide a routine `QNEXT`, to stop the run, for example.

Examples:

Schema of Fortran `CALL` levels :

```
MAIN     CALL QNEXTE       ...      EVLOOP   CALL MATCH
QNEXTE   CALL QNEXT         .       MATCH    CALL TEST
QNEXT    CALL EVLOOP     ...        TEST     CALL QNEXTE
```

The last `CALL QNEXTE` abandons the current event.

•