

Should GoTo Become a Deprecated Statement?

(Original creator: eknochs)

I recently read an article that referred to "Structured Programming". That struck a nostalgic chord with me as that programming method was being taught, and strongly encouraged, by my first I.T. employer, when I was a COBOL programmer. Three historical papers by some computing heroes of their day preface the following discussion:

- Böhm, C. and Jacopini, G. (May 1966). "[Flow diagrams, Turing machines and languages with only two formation rules](#)"
- Edsger Dijkstra (March 1968). "[Go To Statement Considered Harmful](#)"
- Donald E. Knuth (December 1974). "[Structured Programming with go to Statements](#)"

The first paper from 1966 introduced the notion that you can write programs (or express an algorithm) using only 3 control structures, which are: (1) sequentially executing subprograms, (2) selection of one subprogram over another based on a Boolean value, and (3) the iteration of a subprogram while a Boolean value remains true. That's the basis of Structured programming. The second paper from 1968 drew on the experiences of structured and unstructured programming and concluded that the go to statement is evil (my interpretation). The third paper from 1974 tried to find a middle ground, i.e. do stick to structured programming, but if go to statements solve a problem that the language cannot address with existing statements, or are used at the correct level of abstraction, they aren't so evil.

I haven't read all 3 papers completely, and have used other writers' summaries and interpretations, but only where I saw consistency in those writings. Well, Uniface is a mature language, and so it comes as no surprise that there is a **GoTo** proc statement in its vocabulary. For completeness, I'll add that the library reference gives a polite suggestion that **while** and **repeat** statements are better for structured programming. The **GoTo** statement references a label; not the type that is displayed along with fields, but at least the **GoTo** target is not a statement line number as some languages used. If you turn on compiler information messages, you might also discover unused labels and obsolete proc statements (that are never executed by virtue of being bypassed by **GoTo**). Over the years you will have seen new proc statements introduced that assist with meeting the requirements for subprogram definition, sequential processing, selection and iteration. In the past, the absence of statements like **selectcase**, **for**, **forentity**, **callfieldtrigger** and so on, legitimised some use of the **GoTo** statement. It is possible to write good structured code today with **GoTo** statements, though it is easier to write "spaghetti code". Thus the question now becomes: is there any need to use **GoTo** at all these days? If the answer is NO, then should the **GoTo** statement become a deprecated statement for the reason of "Obsolete feature"? [poll id="7"]