

Re: Report enhancements

Source: <http://coding.derkeiler.com/Archive/Cobol/comp.lang.cobol/2004-11/0083.html>

From: Chuck Stevens (charles.stevens_at_unisys.com)

Date: 11/02/04

Date: Tue, 2 Nov 2004 09:11:28 -0800

My point about PERFORM UNTIL EXIT is that I can certainly see the potential of syntactic/semantic ambiguities that could present problems for the implementor, which potential doesn't exist for the PERFORM UNTIL FALSE case. I don't dispute that, upon extensive analysis, one might find that there is no ambiguity after all, and that no implementor would have a problem providing the syntax with no ambiguities whatever.

I thought I made it clear both in this forum and in the proposal itself that the heart of the proposal wasn't the **syntax** but rather the **functionality**. If J4 decides this is worth pursuing, I have no doubt whatever that the appropriate syntax will be debated at rather more length than it deserves, and very little doubt that the final syntax won't be PERFORM UNTIL EXIT. The syntax was chosen because **I** was writing the proposal, and **I** thought it was the best of the available choices for stated reasons.

Can you (or Robert) demonstrate that there is no circumstance in the standard, no individual's choice of parsing mechanism in existing compilers and no existing implementor extension for which PERFORM ... UNTIL EXIT might present a semantic problem?

I understand the point that he's **probably** right, but as a first "Here's a functionality I think is useful" stab at a proposal, I don't think the PERFORM UNTIL FALSE is so horrible a first choice as to sink the entire ship!

–Chuck Stevens