Fortran-like CoCo Scribe Notes

Following are scribe notes from the technical portion of the discussion on the Fortran-like Conditional Compilation (CoCo) facility. This discussion was held in the Courtyard Room at the Sunrise Suites Hotel, 4575 Boulder Highway, Las Vegas, NV from 8:00 p.m. until 9:35 p.m. on Wednesday, February 12, 1997.

References: X3J3 / 97-111, Part 3, Conditional Compilation

Michael Hennecke	Would it be acceptable for a Fortran-like CoCo to have macro expansion?
Richard Bleikamp	A Fortran-like CoCo needs to have a simplistic, portable macro expansion capability, at least one tenth as powerful as what is in cpp.
Kurt Hirchert	We need to have CoCo interpolate values into output lines. The Fortran-like CoCo has a very powerful computational capability but we can't use it in anything. We need to plug computed values into generated source code and it is desirable to flag the lines of the input code where the interpolated values are used.
Walt Brainerd	CoCo should have the capability to get information off of the command line and out of environmental variables. It does not have CHARACTER variables right now. It should have CHARACTER variables.
John Cuthbertson	Macro expansion is like statement functions. I find it strange that we want to include a feature which is equivalent to a feature which is obsolescent.
Keith Bierman	Formerly, the straw votes indicated that we did not want macro expansion. A while back, we went through some code from ISVs which were significant commercial applications. We found out that almost all of this code was based on the full capabilities of the cpp macro facility.
Craig Dedo	I would like the Fortran-like CoCo to have the Fortran PRINT and STOP statements instead of the current ERROR statement.
Wolfgang Walter	Why did David use an ERROR statement instead of the Fortran PRINT and STOP statements?
Craig Dedo	David wanted the absolute minimalist approach.
Question	What does a macro expansion or interpolation capability give you that symbolic constants do not?
Kurt Hirchert	This capability allows you to change the names of procedures in the source code.
Tony Warnock	I want the absolute minimum tool that is necessary to get the job done. We can always expand it later. We should put stuff into this version of CoCo knowing that it is going to be added to. We should make it a really simple facility that can get the job done.
Michael Hennecke	I would use many different macro expansion capabilities beyond the simple INTEGER and LOGICAL variables which are supported right now. This

should not be simple text replacement. Many complex codes use arbitrary text.

Kurt Hirchert — We considered a proposed requirement on TYPE aliasing. A CoCo interpolation or macro expansion capability could be used to achieve TYPE aliasing in a different way.

Reva Sacks — While I like full macro expansion, judging by the help desk inquiries, our customers are not using full macro expansion very much. From looking at Larry Rolison's paper, I am impressed by how hard full macro expansion is. I am very worried about the time delay in getting CoCo out if we include full macro expansion.

Straw Vote — Do we want macro expansion at all in a Fortran-like CoCo?
10 Yes 4 No 7 Undecided

Straw Vote — Assuming a Fortran-like CoCo has some level of macro expansion capability, what level of macro expansion do you want?
1 Full
9 Limited
8 Minimal
5 Undecided

Tony Warnock — There are lots of reasons to want a CHARACTER data type. Right now, I have a program which reads a parameter file and writes into a file. If I had a CHARACTER data type in a pre-processor, I could have the pre-processor read the parameter file and insert the CHARACTER data directly into the source file. There are reasons why this is useful. For example, there are different names for system calls on a PC than on a Sun. On different systems, I use different names for scratch files.

Michael Hennecke — With the way that CoCo is defined right now, we need to document the meaning of INTEGER variables in comments. When a processor has trouble with a variable, the programmer has trouble detecting its meaning. The current examples of the use of INTEGER variables are very clumsy. I do not want to have to use numbers to identify the different machines that I am compiling for.

Walt Brainerd — You could use various LOGICAL variables to specify the target platforms.

Michael Hennecke — I am troubled by the definition of the scope of the CoCo variables. Right now, the CoCo document defines the scope to be the program. This contrasts with cpp defining the scope to be the source file. The current CoCo definition could have some serious undesirable interactions.

Straw Vote: — Do you want a CHARACTER data type in a Fortran-like CoCo facility?
15 Yes 1 No 7 Undecided