From: Peter Kukol
Sent: Wednesday, October 16, 1996 10:45 AM
To: Bill Gates; Bob Muglia
Cc: Aaron Contorer; Steven Sinofsky; Brad Silverberg; Paul Maritz; Jon DeVaan; Patrick Dussud; Ben Slivka; Russ Arun
Subject: RE: Java <=> C++

We can certainly look into this in detail, but off the top of my head it should not be that hard. In an earlier life I was involved in a project where some 250K lines of old commercial Pascal code (pretty nasty stuff - things like nested procedures, reference parameters and sets used all over the place) was converted to C, and the process took a team of 2 just a few months (most of the time spent filling in the gaps (where the translator could not handle something) and cleaning up the worst inefficiencies introduced by the translation) - and the result become the fastest and most robust of its category on the market at the time. The API's used by the code in question can be harder to convert automatically, but QPW (and I have to assume WP as well) was written on top of a platform-independent framework so the underlying API porting job may have been relatively easy.

There are numerous C/C++ constructs that don't translate easily to Java (many pointer constructs (including C++ member pointers), local object destructors (using finalizers for this may not always work and try / finally may have to be used instead), unions, reference parameters, goto's, pre-processor tricks, and so on), but some of these may not come up much in a particular piece of code and those that have been used a lot may be auto-convertible (esp. if efficiency is not a great concern because you count on removing the worst bottle-necks by hand after the conversion is done).

Bottom line: if I were to hazard a wild guess, given a reasonably clean piece of C/C++ code and a translator that can be tailored to how the code is written one might be able to convert anywhere from 60 to 90+ % of the code to Java with very little manual intervention.

Incidentally, we've been doing quite a bit of research on how to make Java run as fast as C++ across the board - this requires pretty advanced program analysis and optimization work but should be quite feasible (in fact, it might be possible to make Java run faster in many cases - after all, it's a much cleaner language plus the code generator can easily work on much larger parts of the program).

Thanks
Peter

From: Bob Muglia
Sent: Wednesday, October 16, 1996 6:18 AM
To: Bill Gates
Cc: Aaron Contorer; Steven Sinofsky; Brad Silverberg; Paul Maritz; Jon DeVaan; Peter Kukol; Patrick Dussud
Subject: RE: Java < = > C + +

I will discuss with Peter and Patrick. The only data I have on this is that the Trident guys have had some interest in doing a Java version and it was not felt that they'd get too much leverage from their C++ codebase.

We will do native Java in 97.

bob

ATTORNEYS ONLY

From: Bill Gates
Sent: Tuesday, October 15, 1996 1:51 PM
To: Bob Muglia
Cc: Aaron Contorer; Steven Sinofsky; Brad Silverberg; Paul Maritz; Jon DeVaan
Subject: Java < = > C + +

MSS 0065248
CONFIDENTIAL

I want some expert in the tools group to write something about how hard it is to convert C code to Java and vice-versa. I understand fully that some crazy code can do such strange things with pointers that you can't figure out how to convert it. However for a lot of code determining what is going on with the pointers is easy enough that you should be able to map C++ to Java. Mapping the other way is less interesting to me.

This comes up because I am reading about all of Wordperfect and Quattro pro being available as part of the Corel Java suite. I am amazed they were able to get their suite moved across to Java so quickly and I am sure

GOVERNMENT EXHIBIT
1341

they must have had automatic conversion to máke it work.

This is all somewhat interesting because Microsoft can't figure out how to run things like IE4, Trident and Office97 on platforms like Mac and Win 3.1 and yet people who do Java applications seem to make us look like fools - particularly with the upcoming Java native code compilers (which for some stupid reason is not an explicit part of our plan - we will be forced to do it).

Ironically our original application strategy was based on the portability of Pcode to many platforms - we ran Multiplan on the VAX, UNIX, Datapoint, TI 9900 and Commodore 64 among other platforms.

ATTORNEYS ONLY

MSS 0065249
CONFIDENTIAL