

To: Bobmu, Markz, Steveb, Nathanm,
Johnsa

date: 6/14/90

Fm: Paul Maritz

What should we do about PM vs. Windows?

In an effort to get myself to think more clearly, and to get us to a decision more quickly, I have tried to set down what I see as the key issues and possibilities. I think the PM vs. Windows decision is actually orthogonal to a lot of the other issues we have been discussing (how many packages, etc), and is a decision which we need to make quickly.

Please keep this memo limited in its distribution.

PM and Windows:

I agree with the premise that we can have only one long term window manager asset. Thus (no surprise), I believe that there are two basic paths we can be on:

- (i) A "PM" path which is:
- move PM as the native display manager to RISC/NT, and use Porthole as a migration tool.
 - use the establishment of PM on RISC as the signal to the world that PM is our long term asset and we expect the world to make a transition.
 - restrict the evolution of Windows on x86 to be "limited" (ie. do not do major functional enhancements to Win API) as another signal that PM is our long term asset.

The message to the respective PM, Windows developers is then:
PM developers: "You have smooth waters ahead of you, you lack a low end platform today, but within 2 years hardware advanced will have taken care of you - ie. 386/4MB will be low-end."

Windows developers: "You have turbulence ahead of you, you will be able to sell on large segment of market for next 2-3 years, but there is major market segment of the future (RISC) which you need to convert to PM api's for. We will give you a porting layer (Porthole), but for new function you need to make the switch."

Another way of looking at this is:

X0195488
EXH. 10 DATE 02/11/02
WITNESS M. A. H. vel 0
SUSAN MELIE

Mainline ISV's: "You have to maintain 2 source bases for foreseeable future - Windows and PM. And you should get ready for time when PM will offer something important that Windows will not - ie. RISC".

Corporate Developers: "Develop for OS/2. Things may look a little bleak now, but we will fix the software problems and the hardware cost problems will fix themselves - just like Windows few years ago."

- (ii) A "Windows" path which is:
- move the Windows API to RISC/NT via 2 paths:
 - a. one which allows 16:16 Windows apps written in C to be very easily moved in source compatible way to RISC (this path may be handled entirely by "smart" compiler tools).
 - b. one which allows Windows apps to be converted first to 0:32 C code, and then moved in source compatible way to RISC/NT (on to the "merged API" - see below).
 - release OS/2 2.0 and position it as a good deployment platform for those who have PM Apps - ie. it:
 - will not disadvantage end-users because it is "Windows Plus" (for Windows 3.x apps)
 - and it runs the PM Apps.
 - announce that MS is going to move long term to a "merged 32bit API" for display management, that this merged API will be called Windows 4, and that it will be:
 - available on all MS 386 and RISC platforms,
 - That both existing Window and PM Apps will require modification to use it, but that it will be highly compatible with Windows 3.x, while obtaining the advantages in the PM "technology" (ie. a PM app will not "lose" functionality).

The message to the PM/Windows developers then becomes:

Windows developers: "You have smooth waters ahead of you. We will have a new 32bit "merged" API that you convert to in order to get new function (beziers, paths, areas, OO libraries, etc), but the porting path is straight forward."

X0195489

PM developers: "You have turbulent waters ahead. We will give you a good deployment platform in OS/2 2.0, but beyond that you have a major conversion effort to get to the new merged API. You will need to convert, because there is a new important platform (RISC) which you don't get to otherwise. Likewise there are functions (OO libs) that you will not get without converting".

Or from the ISV/Corporate developer view:

ISV's: "Write like mad for Windows, be prepared for the merged API Win-32 which has new functions. You can decide whether to incorporate the new functions before or after you go to RISC (by virtue of our 2 paths), but eventually you should move to the new merged API to get the new function (which is after all relatively painless). We will run your Windows apps on OS/2 2.0".

Corporate Developers:

"Sorry, we told you to write for PM and now you have a conversion effort ahead. But this is better than having your API not be that which is mainstream asset."

2. Background Data:

In choosing which of the above paths. the following factors are pertinent:

- (i) Do we have to choose only one path - could we not offer both the Windows and the PM path?

We cannot - because there will be tremendous pressure to use the set that gains majority market share as an "asset" and keep that asset competitive. This will ensure that the other API set suffers - from an evangelical point view, from an investment point of view, from a management point of view.

Can we keep one around as a "sop" (ie. have it be available, but sitting on disk most of the time). We could do this but it would be rightly preceived as just that "a sop", or at best a migration aid. New function will make their way into the "asset" first and the other will become increasingly incompatible. If we can keep the other API set around at little cost we should, but we should not perform unnatural

X0195490

technical acts to do so.

(ii) What is effect of competition?

If it were not for the fact that I fear greatly Sun/SPARC, either strategy would in fact be workable. We have no credible competition on the x86 - both SCO ODT and UNIX Lite are either in their infancy or vaporware, and are resource hogs - so it will be several (2-3) years before they are threats. By then we could have established either Windows or PM as our asset.

So Sun/SPARC is the competition. [Paradoxically, it is the very weakness/fragmentation in the UNIX camp that is giving Sun/SPARC their franchise. When will Scott McNealy realize that OSF is the best thing that ever happened to him, and cease offering it technology?] So how best to counter Sun/SPARC is a key determinant of which path to pick.

I will look to the technical people to provide a definitive technical answer but I think it is true that the Windows path (as I have defined it, ie. both 16:16 and 0:32 route) provides a smoother path.

Perhaps more important though is the positioning/message at this point. We need to be locking ISVs tightly into our asset, and DISCOURAGING THEM FROM BEING PLATFORM

INDEPENDENT.

The PM path, of necessity, is a multi platform message. It encourages ISVs to become platform independent and spend cycles on that, rather than setting up the vision of something that will have 80% market share and hence they should be as early and competitive on the 80% share platform. Having a single, credible API set to sell is the key issue.

Single is addressed by picking one of the above paths. What makes an API set credible?

Technical considerations:

PM is more advanced/cleaner than Windows, but not dramatically so.

32bit - 32bit is important in long term, though surprisingly unimportant in the near term as I

X0195491

believe that people will use "extender techniques" to get most of what they want. Either strategy yields 32bits.

Provides access to key hardware platforms - RISC is the key, either strategy can yield it with time. The Windows path does so quicker (but not dramatically so).

Marketing considerations:

Present Market Share: Windows has it.

Future Market Share: We say PM will have it, but credibility (because of Windows present market share) is not high.

Message: The Windows path offers a single "message" (ie. write for Windows), the PM path is "do this now, this later" - it is more complex and hence less credible.

(iii) Is getting a RISC PC out the most effective way to combat SPARC then?

Would not getting Win API's on NT/386 be the most effective means?

No (at least I believe not). SPARC is getting its beach heads by virtue of:

- a. getting into certain accounts because the accounts are hiring "rocket scientists" who are numerically sophisticated folks who want the large address space and horse power of a Sun Workstation,
- b. the increasing cheapness/power of the Sun offerings.

While we can position 386/486 to address a., Sun's increasing lead in b. will keep them growing in these accounts.

Thus I agree that getting a RISC based PC is a necessary and time critical goal.

(iv) What is the effect of IBM?

X0195492

IBM would readily buy into the PM path - no problems there.

If we pick the Windows path, then there is the risk that they might feel morally obliged to pick the PM path anyway. This would create competing platforms, and would likely have the effect of keeping ISV's in a platform neutral stance for longer (they would be lobbying ISVs as well).

(v) MS credibility?

With ISV's:

ISV's have to be mercenaries and will follow the path of greatest market share. If NeXT suddenly got hot, they (including MS Apps) would swing round and write for NeXT, if NeXT got cold, they would abandon it (vide Aldus), etc.

Some ISV's (esp. Lotus) will team up with Sun to have an orgy of MS bashing, but the net effect will be that they will write for whatever API set looks credible in its claim to significant market share. They will prioritize which market segments based on probably size and level of investment required.

So the conclusion is that don't be constrained on this account.

With Corporations and "Corporate ISV's" (who sell MIS type solutions):

These are the guys who will feel most abused by a switch away from PM.

On the other hand, our current stance of selling Windows like crazy with one hand, while promising PM with the other is not very credible either. One can argue that "coming clean" would be viewed as more credible.

3. Implications of choosing the "Windows Path".

The net position of all this, is that we should choose the "Windows Path" (by this is meant the set of steps laid out in 1. above).

X0195493

What then are the implications of this, and when do we have to take the steps involved (privately and publicly)?

Implications and Steps:

- (i) OS/2 2.0 must be able to run Windows apps well.

Otherwise it is not Windows Plus and not a good deployment platform.

- (ii) Do we release OS/2 2.0?

- a. Release it all?

Yes, otherwise you cut those how have invested in PM at the knees. It will also provide the OS/2 base API's in 16 and 32bit form (which is something we do wish to preserve).

- b. With a 32bit PM API?

Yes, there is nothing much to be gained by not doing it, so we should probably release it.

- (iii) Getting Windows onto RISC.

This must be done quickly and in a manner that present Windows apps can follow just as quickly. We should then re-focus our resources onto:

- a. getting Windows ported to NT/RISC asap (refocus current Portable PM team),
- b. defined the technology that allows a 16:16 C Windows app to be recompiled for RISC (this is key).

- (iv) What do I do with ancilliary OS/2 investment?

- OS/2 1.2 & 2.0 Disk Driver work (WAMIC) - continue it.
- OS/2 1.2 & 2.0 Generic Printer driver work and other driver acquisition work? Look it we can leverage

X0195494

Windows drivers, failing that continue the investment.

(v) Internal Morale:

This is containable if we act quickly and have a strategy that makes sense to people.

Right now, unrest is building because people:

- are no longer confident that they are working on things that make sense,
- they are worried about our relation to IBM (after having being told for so long to bend over backwards for IBM, people are worried when they see us and IBM vaguely threatening each other),
- the marketing folk worry that they are selling a phony story.

It is containable at this point, but it is going to get worse fairly quickly.

(vi) IBM:

We have need to alert them to what we are doing. This will probably cause an upheaval, but I think that at this point that would help.

(v) Public announcement:

I think we should articulate this as a complete strategy fairly soon as well probably in the fourth quarter. I would not even wait to have complete spec's on the "merged API". IT would throw an enormous bucket of cold water on OS/2 development, and cause a lot of MS questioning. This is best catered by having an even stronger Windows story to keep even those who are most pissed off loyal. We could thus also announce the RISC strategy with some hooplah (it really would be great to have Compaq publicly signed up).

(vi) Base consistency:

CONFIDENTIAL

X0195495

In all this, we have assumed that when Win Apps get moved to RISC they (i) use MS-DOS style INT 21 services for base functions (it is fairly trivial to emulate these on top of OS/2 base services), (ii) they would use a new set (OS/2 API's) when running in "new 32bit mode".

Because of this (and other issues like device driver models), there are a set of questions that need to get addressed at looking what could be our "three" possible platforms (steady state):

286 + 386: DOS 6 + Windows

386: NT + Windows

RISC: NT + Windows

X0195496