

**David Cole**

---

**From:** David Cole  
**To:** Bill Gates  
**Cc:** Brad Silverberg; David Cole; Paul Maritz; Steven Sinofsky  
**Subject:** Chicago and Monday's meeting  
**Date:** Friday, April 09, 1993 9:50AM

This mail is suppose to serve as the framework for our Monday review. It's a bit long, but I hope you get a chance to read it before me meet. It lays out our motivations and some history of the project so you can understand where things are. If we are in sync on this, then we can focus on the details at Monday's meeting and make forward progress.

1) What kind of release should Chicago be?  
Chicago must be a compelling release that the bulk of the active Win 3.1 user base will upgrade to. We really don't want the active market split between Win 3.1 and Chicago. To do this it needs to have a broad set of compelling features, much more than Win 3.1 did relative to Win 3.0. A good chunk of Win 3.1's success was due to Win 3.0's short comings. The feedback we are getting is that users are generally happy with Win 3.1, so it'll be harder to move them off it. Also, we don't believe there is a magic bullet feature that will appeal to the broad masses and get them to upgrade the way doublespace is doing for MS-DOS 6.

2) Content required for a compelling release  
There is nothing new in this list, it's our top 10 list, but it's worth reminding you what we are shooting for at the high level so we can be in sync.

- Easy to use, intuitive shell. Exploit OLE 2 for app/shell interoperability.
- Plug and play for increased ease of use dealing with devices, new industry standard to put ourselves in the leadership position.
- Complete and integrated os; windows utilities to manage the system, never need to return to DOS, protect mode file sys and network paths for better performance
- Universal network client; peer sever, device sharing, system management, great Novell connectivity
- great 32-bits; preemptive multitasking, robustness; complete 32-bit API, async input.
- mobile computing platform and features; RNA, file sync.
- runs great in 4 megs. This is the sweet spot in the market, very big upgrade revenue opportunity here.
- high performance, meet or beat Win 3.1 in 4 megs or more.
- robust; bad apps don't crash the system, good error handling
- compatible with Win 3.1 apps, drivers, VxDs, and dos apps and drivers
- it ships as soon as possible.

There are lots of secondary things we are doing: fixing the core product so we can get Chicago into all international markets quickly; improved setup to provide a safe and reliable upgrade path; running Win 3.1 in a full screen VM as a compatibility safety valve, device independent color, new printing subsystem system, etc. You've seen the spec.

3) Some quick project history  
Getting this product out quickly is serious business for us. The original RTM goal we established was Dec 93. I don't think anyone believed this date, but we built our feature set and schedule for that goal. As expected the minimum compelling feature set could not be completed and tested in time. The team was not making the optimistic progress planned for in the schedule.

With a little bottoms up planning and some goal setting, we established April 94 as the serious ship date. We have a series of milestones that get us to that date (shown further below in this mail), the team is driving toward

Page 1

MS 0073067  
CONFIDENTIAL

Plaintiff's Exhibit

5549

Comes V. Microsoft

RBC 04581

1 milestone at a time. The development schedules did not have enough buffer time, so we started cutting features in every area of the product so that April was reasonably low risk. On Monday, I'll have a list of the things we've cut so far.

However, the problem came when we got to the great "32-bit feature". We couldn't cut enough out to make April 94 low risk. This forced us (me anyway) to really evaluate whether we wanted a good 32-bit system or not. Could we make Chicago compelling enough without great "32-bits"? (I've been known to whine, moan and complain that 32-bits is just a strategy tax for Chicago so I wanted to pursue this. Reduced or no 32-bit support lowers the risk to making April 94, so it was worth thinking about) The conclusion we all reached was that great 32-bit support in Chicago provides a gigantic opportunity for us that we must take advantage of.

#### 4) Why do we want great 32-bits?

Rather than just pay our taxes, we should deliver a product that capitalizes on the industry buzz about 32-bits. This is the feature which will make Chicago really compelling. Customers are asking for it. They think 32-bits means real multitasking, power, robustness, performance. Who are we to tell them they don't. They want this from Windows. Our competitors have it. The industry expects Chicago to have it. I now think 32-bits is the feature that will put Chicago over the top as a compelling upgrade to Win 3.1. There is some magic dust associated with 32-bits, we must exploit that.

Maybe I'm preaching to the choir here Bill, but I want to make the point clear that my personal position has changed on the 32-bit issue and I now think it's worth some risks to make it happen. The secondary benefits of great 32-bit support is of course the internal efficiency here at MS, groups can just focus on this and support both NT and Chicago much easier than doing separate 16 and 32-bit versions.

We want 32-bits to be great in Chicago, not just a sticker on the box that in reality is klunky. (We had talked about some klunky plans where all 32-bit API calls are blocked at the thunk layer, saving us the work of serializing the core. However, this would mean 32-bit apps don't run if some 16-bit app is in a core component.)

To help us there we will put the shell on 32-bits. That is key to getting apps to making the 32-bit system small enough, fast enough and high quality. This will also help move apps to 32-bits. Yes, we will have to work harder to make the 4 meg goal, but that's what we will do because that's where there's a gigantic upgrade revenue opportunity.

We are still pursuing preempting 16bit apps too. Most of the work to do this already needs to happen for great 32-bit support. There are issues we still must work through, but this is the goal. This isn't something we would slip the April 94 date to make however.

#### 5) The schedule and milestones

We are very milestone driven, the team is currently focused on M4. Here's how it lays out. We will have a complete list of what features go in which milestone for you Monday.

M3: March 93. Early PDK sent to 40 ISVs for an early look. Basically had Win32s level support for 32-bits. We kept the shell out of this release. DONE.

M4: July 93. PDK release. We will send to 2000 or so ISVs, IITVs, and a few end users. This will be the real 32-bit PDK, for both apps and device drivers. Preemption will be in for Win32c apps. The shell will be included here too.

M5: Oct 93. Beta 1. Using the PDK, we'll move the shell to 32-bits and debug, tune the 32-bit system. We will expand our end user sites as well. The core of the product is pretty much complete for this date.

M6: Early Feb 94. Beta 2 All applets are done, drivers are in. This will be a big beta release.

RTM: April 94.

#### 6) Risks

Making the 32-bit system complete, small, fast, and high quality is a big challenge. We've made some mistakes in the past in the kernel area and progress has been slower than expected. Getting preemption into a system that's been designed for non-preemption is hard. Lots of unexpected things happen. We are working through the problems as we get to them. Putting the shell on 32-bits will help us here.

High quality developer have been added to the team and we'll continue to add them. On Monday, karst will take you through the history of the team, mistakes, successes, where we are, and what's left to do. Davidw will move to this area when he's finished with the DIC plumbing, that should be in about 2 weeks. Remaining work there will be done by a good GDI guy, and program management will get as much as possible from 3rd parties. Lins will cover the bases on graphics.

Size is also a risk worth mentioning. We'll have to work damn hard and be creative to make 4 megs. But it's something that we must do. We're watching lego closely and will explore pcode for making stuff smaller. There are also challenges in the mail area which we'll talk about Monday.

We are continuing to go through all areas of the product and cutting anything we view as fat. This is essential to reducing risk in all other areas.

#### 7) Wrap up

The schedule risks are not caused by us adding features. We've done just the opposite. The team is trying to be very realistic about the schedule and ship as soon as possible with a compelling product. We've really added no risk, we've just identified them early. Since a lot is riding on Chicago, understanding risks, communicating them up the management chain, and of course trying to reduce them is a key part of my job. Chicago is not out of control. If it were, you wouldn't hear about this sort of thing until much later in the cycle.

On Monday, here's the additional data we'll have for you:

- detailed content for each of the milestones.
- list of the features we've already cut
- kernel team history, status, future
- what personnel we have covering the important areas
- robustness data; GP fault data from PSS, our plan
- mail store size data

We'll have guys there to talk about some of the technical issues with getting the 32-bit stuff done as well.