PLAINTIFF'S EXHIBIT 1927
Comes v. Microsoft

From: Paul Maritz
To: David Cutler
Cc: Jim Allchin
Subject: RE: Chicago question
Date: Thu, Dec 9, 1993 8:13AM

I will stop round. Two points to state them again:
1. This has NO impact on ISVs (Lotus, Wordperfect, etc.) - they can write programs to their hearts content.
2. This does have a potential to deter likes of WABI - ie. Win32 cloners. It could allow us to make a business decision on this. It also could allow us to make Win32 a source level (aka POSIX) standard vs. a binary standard.

There is a difference.

> From: David Cutler
> To: Paul Maritz
> Cc: Jim Allchin
> Subject: RE: Chicago question
> Date: Thursday, December 09, 1993 7:14AM
>
> This is bullshit and you know it. It is a proprietary move aimed at making Win32 unclonable - as if anyone could ever figure that out anyway. Having to decrypt a critical part of the image to get it started adds significant time to startup - something I understand you complain about constantly. Taking real page faults makes it even slower.
>
> This ain't magic - Chicago won't be able to do it any faster. In fact why don't you get a Chicago system and put it beside your NT system and see what it "feels" like today and see if you want it to be slower.
>
> d
>
>> From: Paul Maritz
>> To: David Cutler; Jim Allchin
>> Cc: Lou Perazzoli
>> Subject: RE: Chicago question
>> Date: Thursday, December 02, 1993 6:52PM
>>
>> The performance issue is something to be concerned about. However it is NOT intended to keep ISVs out. ISVs will be able to continue to continue as before. It is intended to give us some barriers vs outright cloners of Windows.
>>
>> From: David Cutler
>> To: Jim Allchin; Paul Maritz
>> Cc: Lou Perazzoli
>> Subject: FW: Chicago question
>> Date: Wednesday, December 08, 1993 4:16PM
>>
>> As markl states below this will have a large effect on APP start up performance and will create more dirty pages which has a bad effect on size. Cairo has some images with a very large number of DLLs. Initiation of these images would be very slow.

I also agree with markl that this will immediately be picked up by the press for what it really is - a way to keep everyone out. This is not an "openess" message.

d

From: Mark Lucovsky
To: David Cutler; Lou Perazzoli
Subject: FW: Chicago question
Date: Wednesday, December 08, 1993 4:01PM

Dave, Loup,

Bens is going to meet with paulma/billg to move on this real soon (as early as 12/20). His current idea is to scramble the idata section using compression and encryption. This is "to prevent viruses". NT would support new and old style images. Chicago would only support new images (invalidating old image format).

This is of course fucked for performance/size... We would always dirty all of idata, and load time would slow significantly.
Of course this is nothing compared to the message we project to our isv friends.

-markl

From: Paul Maritz <paulma@microsoft.com>
To: markl
Cc: loup
Subject: RE: Chicago question
Date: Friday, August 06, 1993 2:14PM

See comments below.

> From: Mark Lucovsky
> To: Paul Maritz
> Cc: Lou Perazzoli
> Subject: Chicago question
> Date: Thursday, August 05, 1993 1:30PM
>
> Paul,
>
> I installed chicago M4 beta a few days ago. I have had significant problems running simple Win32 apps on chicago. I am working closely with the chicago people to solve these problems.
>
> I would really appreciate your input on a few areas with respect to chicago.
>
> 1) Why did we do such a drastic change to the user interface. We have shipped millions of copies of windows. Everyone knows how to use it and finds it a frienly and simple environment.
> My six year old daughter is comfortable with launching apps, closing apps, minimizing, moving windows... on all machines running windows and windows NT. I have not been in on any of the reasons for making the shell changes that we have made, but I don't understand the motivation for changing something as drastically as we have with the chicago shell. I would think that incremental changes like groups within groups would

> have been plently. I won't go on about this, as I would really like to understand more about why we choose to make this change, and what impact we think this change will have on the millions and millions of current windows users.

We had been hoping to do something that would be a mid-point between where we are now, and where want to get to several years from now when the paradigms of applications/data structures/navgiating over tree structured name space will be replaced by "documents and queries" (ie. user does not explictly start/stop apps, instead uses compound documents that start the component apps for him, and you find the documents by querying over their properties/contents). We have also received lots of ffedback on the defficiencies of the Win3.1 shell (why do we have two disjoint name spaces - file manager, program manager; why is the control panel an app and not a program group, inconsistent useage of drag/drop, inconsistent handling of properties on entities, etc., etc.)

That being said, we are receiving tons of feedback that the shell as implemented in.M4 is worst of both worlds (different for no gain). As a result we are going going to have an intensive effort to focus on this over next 2 months - particularly focussing on ease of use for the non-sophisticated user.

> 2) What is the real target for chicago ? I know the simple 4mb low end x86 part of it, but as far as applications, are we expecting real live 32bit apps to run on chicago ? What level of compatability are we going to commit to between chicago and NT ? Are we expecting ISVs to write to a SINGLE Win32 API so that their applications run on chicago, NT, Cairo ? I am a little confused. I don't know what story we plan to tell ISVs, but I certainly get the feeling that chicago will be whatever is easy, or can be done in their scheduled timeframe (I have heard developers state this). Again, I would like to hear your version of what chicago is supposed to be, what problems it is meant to solve, and how you think ISVs benifit from the several Win32 API sets/subsets that are in the pipline. This will help me alot as I develop my relationship with the chicago team.

Our basic plan is to ensure that an ISV can write a Win32 app that will run unmodified on Chicago, NT3.1, and Cairo - with following caveats:
- Chicago is subset of WinNT3.1 Win32 API set (ie. will be api's that work only on NT - eg. more advanced GDI transforms),
- there may be some new Win32 API's (eg. device independent color mapping, file sync for portables, etc.) introduced on Chicago that may not be on NT until WinNT 3.1A or until Cairo (depending on what can be done when for NT),
- Cairo must be superset of WinNT 3.1 and Chicago (ie. any Chicago or NT3.1 app must run)
- Cairo will further extend API's (in compatible way) beyond Chicago and NT3.1.

However in all cases it should be possible for an ISV to write the vast majority of his app in a generic way, and dynamically decide to call the extra API's.

> -markl

>>> I won't go over any of the problems I am having with respect to running Win32 apps on this. I am working
>>
>> PS: the Chicago guys know that M4 is early drop - in particular Win32 apps support is not targetted to be really decent until M5 (October).