

# **EXHIBIT C**

Defendant's  
Exhibit  
22

**Unknown**

**From:** bradc[SMTP:bradc@DAWGHAUS.ITG.microsoft.com]  
**Sent:** Monday, April 03, 1995 5:24 PM  
**To:** bradc; bradsi; MCalkins  
**Cc:** BFRANK; bradstr; BRUCEB; DAVEL; GLENN; TODDTR  
**Subject:** RE: Windows 95 Logo Program Requirements

March 31, 1995

Dear Mark,

Thanks for your note and for voicing your concerns to us regarding the Windows 95 Logo program. I am sorry I did not get back to you sooner but as you can imagine things are very busy.

To give you some background, the requirement to run on Windows NT has been a part of the Windows Logo test since the Windows NT debut in August, 1993. At that point in time, Microsoft required testing on Windows NT for software products to receive the Windows Logo.

As Windows 95 was being created we tried to reflect the feedback from customers and developers. One piece of feedback we heard from developers was that they wanted to be able to leverage their investments across a broader spectrum of users. Similarly customers want to be able to use the same application on their notebook that they use on their engineering workstation. Consequently, Microsoft architected Windows 95 and Windows NT to be able to run a common set of applications and we have tried to make it as easy as possible for you to write win32 apps that will run on both Windows 95 and Windows NT.

The purpose of the logo is to help customers identify products designed for Windows 95 so given the above customer criteria it makes sense that the logo requirements for Windows 95 keep the requirement for applications to also run on Windows NT.

Our Windows 95 Logo requirements do state that we will grant applications exceptions to the requirement to run on Windows NT if the application relies on functionality that is significantly different in architecture between the two platforms and that functionality cannot easily degrade gracefully. If in our discussions we decide that this exception was to be granted to Novell, you would need to make the difficult business decision to release a product to market that did not support the same platforms that many of the leading applications will.

Before I respond to your technical points, it is important to reinforce that the Win32 API was designed to address the feedback above and make it easy for ISVs to create a single application that works on both Windows 95 and Windows NT. It is important that your development teams read the Win32 API documentation and base their design and development on the information found there. Your development teams should not exploit some observed functionality as it appears they are doing from several of your questions. Exploiting such features will most likely cause these applications to break in future releases of the operating system as we can only commit to supporting documented behavior as we move forward.

EXHIBIT # 9  
WITNESS Chase  
DATE 4/29/99  
ZOLA SPENCER (206) 502-1000

Here are responses to your specific issues:

- 1. RE: Relying on Windows 95 functionality to enhance product functionality.

You mention two specifics here, there Task Bar and Plug and Play

HIGHLY  
CONFIDENTIAL

MSC 00700613

messages. We do not believe the Task Bar should be relied upon being there for an SDI application to easily switch between documents. The dilemma you point out with Windows NT is really no different from the one you will face on Windows 95 if an end user chooses to hide the task bar.

We are confused by your Plug and Play comment. Applications that support Plug and Play respond to messages sent to it by the system. On Windows 95 your application will receive these messages and respond accordingly. On Windows NT your application will just never receive these messages, so it is a non-issue.

2. RE: Memory Mapped Files.

To resolve this, WordPerfect can use the standard Win32 calls CreateFileMapping and MapViewOfFile and not assume anything about where it gets loaded. This easily enables you to work on both Windows 95 and Windows NT and ensures that your application will be well behaved on future releases of Windows. If it is important to be able to have the same address mapped for a file in different processes, you could look at using some conditional code to have your application call MapViewOfFileEx on Windows NT.

3. RE: Memory Management in DLLs.

It appears your analysis is incorrect here. Having a pointer in a shared data object is equally dangerous on Windows NT and Windows 95. On both platforms, the DLL is possibly unable to load at the same base address. If the DLL is loaded at different base addresses in different processes, pointers inside that shared data object are invalid. This is equally true on NT and Win95. There is literally no difference here. It is not easier to share memory between processes on Windows 95 that it is on Windows NT.

4. RE: Registry.

RegSaveKey and the related APIs are not designed to be binary compatible between Windows 95 and Windows NT. It is true that the data employed by these APIs is not the same between the platforms but it wasn't meant for this. The data is meant to be black box and not portable. This should not cause problems for mainstream applications. We can discuss this issue in more detail in a follow-up conference call, if you would like.

5. RE: Unicode and ANSI OLE.

Both Windows NT and Windows 95 have ANSI APIs. You should be calling the ANSI APIs on both systems for most APIs. They translate to Unicode when calling OLE on both systems. This is documented. Again, this is a case where we have given developers an easy way to work on both systems.

6. RE: Other Small Differences.

With our cross platform story we never claimed that there would be zero work to target two platforms, but have achieved a straightforward and easy way to work on both. There are some subtle differences between the platforms. We have documented what we are aware of in the SDK and will continue to evolve this documentation based up ISV feedback such as yours.

7. RE: Maintain two development and testing environments.

Any incremental increased cost in targeting both platforms seems justified by the fact that it that it would be a great deal less work than maintaining two different products as you say you plan to do with your separate Windows NT release. With the Windows 95 M8 release, we have a common Win32 SDK so some of the above issues should be resolved.

In closing, Microsoft wants to be flexible with the logo if a product can not run on Windows NT because of significant architectural differences between the Windows NT and Windows 95. If an application runs into significant architectural differences then our policy is not to withhold the Logo for that application. If the developer codes something incorrectly however, then the problem needs to be corrected before the Logo is issued. This is simply a question of meeting customer expectations with respect to support between the two operating systems.

At this point in time, we do not believe the issues you raise constitute significant enough architectural issues between the Windows NT and Windows 95 to warrant an exception being granted.

I would be glad to have a conference call between our teams should you have any additional questions. Brad Struss will be glad to set this up for you if you wish.

Mark, we appreciate your continued support of Windows and in particular Windows 95. I believe it is the best thing for your customers to build applications that will run on both Windows 95 and Windows NT. Of course this is your call and as you know, participation in the Windows 95 Logo Program is optional and by no means required to ship a great Windows 95 application.

Please let us know if you would like to speak further and if there is any thing else we can do to be of help.

Sincerely,

Brad Chase  
General Manager  
Personal Systems Division

---

From: Mark Calkins[SMTP:MCalkins@WordPerfect.com]  
Sent: Thursday, March 06, 1919 12:26 PM  
To: brad; bradsi  
Cc: BFRANK; bradstr; BRUCEB; DAVAL; GLENM; TODDTR  
Subject: Windows 95 Logo Program Requirements

Dear Brad Silverberg and Brad Chase,

The Novell Applications Group requests that Microsoft drop the requirement to provide compatibility with Windows NT to participate in the Windows 95 logo program.

We make this request based on the technical differences between Windows 95 and Windows NT. Listed below, we have outlined the problems an ISV will encounter attempting to develop a single application to run under both operating systems.

Novell does plan to support Windows NT with our applications. We were pleased to participate in your latest press release on Windows NT and show support with a future version of WordPerfect for Windows NT. We have looked extensively at supporting Windows NT with other applications, such as PerfectOffice. Requesting this change in the Windows 95 logo program does not mean we will not support Windows NT, but we do believe that it imposes additional burdens on us to compete in the Windows 95 applications market.

There is no question that time to market is critical to the success of any application. We believe that Windows 95 offers a major opportunity to the entire PC software industry. Delivering applications in a timely

manner is critical to the potential success of an application. And the ability to participate in the Windows 95 logo program obviously adds to the recognition that an application has met the operating system vendor's requirements for successful operation on that new platform.

In some respects, we are a little surprised that Microsoft is requiring dual support for both Windows 95 and Windows NT for the Windows 95 logo program. For example, it is our belief that the success of the Macintosh has been due in part to the common user interface of Macintosh applications. Apple was able to get the entire Macintosh development community to use Macintosh OS user interface controls. We see the same potential with Windows 95. Microsoft has the opportunity to get the entire ISV community to support the Windows 95 user interface and environment for common consistency between applications. But the problem is that many of those OS facilities either do not exist or they work differently under Windows NT. One prime example is Windows 95 Plug and Play feature. We think Plug and Play represents a huge breakthrough in the industry, yet any application that supports it has to jump through hoops to code around it for Windows NT.

It is our assertion to Microsoft that having to also support Windows NT in the same timeframe to use the Windows 95 logo is unfair and could be viewed in some respects as illegal. While we could debate the level of success that is expected for Windows NT on the desktop, there is no question that analysts industry wide see a huge difference in market acceptance between the two operating systems for desktop PCs.

#### Windows 95 and Windows NT 3.5 Portability Issues

Let us outline the technical reasons for our request that you disconnect the requirement to support Windows NT from the Windows 95 logo program.

1. Relying on Windows 95 functionality to enhance product functionality. In Windows 95 there exists potential to rely on the OS for certain functionality. For example, the Task Bar under Windows 95 influences the design of an SDI application because it provides for easy switching of tasks and switching between documents. If an ISV relies on that capability and does not put code in to switch between documents easily (even under SDI), users would not have the same experience with the product under Windows NT. Therefore, extra work needs to be taken to give it the same functional level under Windows NT as the product running under Windows 95. The same is true for other key features in Windows 95, such as Plug and Play. It is expected that there could be many such instances with this type of impact in developing for both Windows 95 and Windows NT.
2. Memory Mapped Files. In Windows NT, a memory mapped file is only accessible to processes that have called CreateFileMapping and MapViewOfFile for that particular file. In addition the file's memory region can be based at different virtual addresses in different processes. In Windows 95, once a program creates a memory mapped file, that memory region is accessible to all programs. Thus, a Windows 95 memory mapped file is always at the same virtual address in all processes.
3. Memory Management in DLLs. There is a difference in the way the SHARED data in DLLs is handled between Windows 95 and Windows NT 3.5. Initializing a SHARED variable with a pointer to another SHARED variable will work in Windows 95, but not in Windows NT 3.5. It appears that it is easier to share memory between processes in Windows 95 than Windows NT.
4. Registry. The registry file format is different between Windows 95 and Windows NT. This means that it is not possible to do a

MSC 00700616

HIGHLY  
CONFIDENTIAL

RegSaveKey/LoadKey/RestoreKey from a Windows 95 machine to a Windows NT machine, or vice-versa. Windows 95 provides some System Administration capability through system policies. We have not seen any information regarding system policies on Windows NT. Any ISV use of system policies would appear to be a problem for NT.

5. Unicode and ANSI OLE. The level of integration with OLE 2.0 between the two operating systems is different. In Windows NT, the application needs to supply Unicode strings for many APIs. With Windows 95, the application must use ANSI. Between the two environments it becomes difficult for an application to operate the same way using ANSI or Unicode with the OLE system.

6. Other Small Differences. There will be numerous subtle differences that have to be programmed around. For example we have already had seen a difference in our code that adds menu items. We modified the code to work on NT, and when we moved to Windows 95 we found that the code didn't work. This stems from the internal Unicode under NT to the ANSI API set under Windows 95. We were able to come up with an easy fix that worked on both platforms but there was effort involved. We found setting a global hook worked on NT, but brought Windows 95 to its knees. There will be differences in memory management, in addition to memory mapped files, that will require special attention. We have even noticed that while we are developing we have behavior differences in many APIs themselves. When found, we have reported them to Microsoft.

Common controls are supposed to behave the same under both operating systems but the ISV will need to validate the current DLLs for consistency in behavior. Windows 95 help system running on Windows NT, for example, appears to have a problem of inconsistency in behavior and functionality. Basically it requires a lot of validation to feel comfortable that Microsoft has handled all the issues of common sub-systems between the two platforms since they are not based on the same OS model.

7. Maintain two development and testing environments. There will be additional effort required in maintaining two environments, testing, programming around subtle differences, etc.; costs such as having to purchase two different operating systems, buying hardware capable of developing for NT (not only for developers but for testing as well), and licensing of development tools.

For example, the ISV must use the Windows 95 SDK to get certain headers pertaining to the Common control set and Plug and Play messages, etc. VC++ 2.0 release does not have the changes in to support Windows 95 (headers). Under Windows 95 you must load the SDK and VC++ to get the environment setup for proper Windows 95 development. Patches are made available to keep VC++ 2.0 working under Windows 95.

While there are theoretical solutions to these problems around, they pose unfair additional burden on the development process and require more resources in order to get the application code common between both platforms.

#### Conclusion

Based on these technical problems and challenges, we request that Microsoft drop the requirement to concurrently support Windows 95 and Windows NT to participate in the Windows 95 logo program.

We would like to propose that we set up a conference call to discuss this issue. We will work through our Windows 95 contact, Brad Struss, to find a time that is mutually compatible for all to review this with you.

MSC 00700617

Thank you for taking the time to review and evaluate our request.

Very sincerely,  
Mark Calkins  
Vice President and General Manager  
Novell Business Applications

MSC 00700618