

>From dennisad Wed Jul 3 12:29:39 1991
To: billg bradsi darrylr mikemap
Cc: aarong appsdm bobmu davidcol davidw spastat
Subject: Metafile-able TextOuts
Date: Wed Jul 03 12:28:17 1991

As you may recall, I have been working with several people in both Systems and Apps to define a specification for a better TextOut.

The specification called for TextOut to accept high-resolution (65536 dpi) placement units that would allow us to maintain identical line breaks across all output devices. The API would contain logic that correctly formatted the line of text (which might contain multiple fonts/styles) according to known typographic principles. The "metafile-able" part of this falls out of the high-resolution placement units combined with the formatting logic; this results in great looking text from a metafile, no matter what the original target device resolution was.

The plan was to contract this work to an outside consultant. Unfortunately, this consultant is now occupied with other critical work (relating to TrueType) and is not available to complete the API work until later... this is "later" enough that we cannot put the new TextOut API into Windows 3.1.

However, the efforts that went into the spec for the API will not be wasted. EliK met with members of the NT team (MarkCl, AsmusF, KirkO, KentD and HockL) to discuss this issue.

The end result is that the NT team has reviewed our spec and is considering how well it does or does not fit within their product plans. I am lobbying with them to utilize the text layout methodologies outlined in the LLM APIs.

/Dennis

Mail-Flags: 0001
From mikemap Fri Jul 05 09:38:39 1991
To: bobga
Subject: Saleries this review period.
Date: Fri, 05 Jul 91 09:38:38 PDT

Until we get teh data sorted thru that you presented in your letter, lets stick with the Redmond salary grid plus 15% . This should cause your averages to be slightly below 4% for merit increases.

Mail-Flags: 0001
From mikemap Fri Jul 05 09:40:42 1991
To: cathyw davidpr susanb
Subject: Wed staff meeting
Date: Fri, 05 Jul 91 09:40:39 PDT

Susan please plan to demo the cool new stuff you have at the staff meeting next wednesday. anywhere from 1 to 1.5 hours.

Mail-Flags: 0001
From mikemap Fri Jul 05 10:45:34 1991

Plaintiff's Exhibit

7591

Comes V. Microsoft

MS 5048929
CONFIDENTIAL

Mail-Flags: 0001
From mikemap Fri Jul 05 15:18:59 1991
To: bobga chasst chrisgr chrisp darrylr gregs jeffr peteh susanb tandyt
vijayv
Subject: Inserting Audio and other objects
Date: Fri, 05 Jul 91 15:18:38 PDT

In many ways, Rob hits upon a valid point. We should try to change the INSERT OBJECT command to something like INSERT OTHER OBJECTS or INSERT ADDITIONAL OBJECTS so that the user does not have to try to understand why a graph or other hard menu item is not an object.

Chrisgr, could the UITF look at this quickly.

>From robg Fri Jul 5 13:13:50 1991
To: darrylr
Subject: Inserting Audio and other objects
Cc: aarong billg bradsi chrisdo ericle jeffh jeffr jimall mikemap paulma
paulo peteh raleighr reuelr robg tonyw
Date: Fri Jul 5 13:13:41 1991

Thanks for forwarding your message on audio OLE server ideas. I forwarded it to my guys (Also I emphasized the need for my guys and spag to coordinate-- just like you I often try to make sure that multiple groups coordinate plans so we don't duplicate effort).

Your message did raise an important UI question regarding whether it's better to have one menu item for each insertable object or whether it's better to have a general purpose "Insert Object." I think this issue is somewhat subtle, hence the following discourse and rather broad cc list. Like many discussion of objects the following might seem a little theoretical or maybe even theological, but please bear with me.

My sense is the proper UI for inserting objects varies app to app, based on how important inserting external objects is. The apps that are most likely to be insert/embed intensive are presentation, communication, and print oriented tools such as Word, Powerpoint, Mail, and Voodoo/Publisher. (Analytical tools such as Project, Excel doubtless need to support insertion too but they don't strain the model as much).

As I'm sure you know, both Winword 2 and Voodoo implement the "Insert Object" model for OLE. However beyond this there are many inconsistencies. For instance Voodoo has a few "Import" menu items (actually under the file menu) that in fact insert objects (first doing conversion, hence the word "Import." Further Voodoo treats the "Font Effects" OLE server as a core tool that in fact does NOT show up in the "Insert Objects" dialog box. Winword 2 has a full pulldown dedicated to "Insert" with about 15 individual items (up from about 12 in winword 1). The data types that you "Import" in Barney are "Inserted" in WinWord. Further many of the 14 items besides "insert object" really are about inserting objects, just not OLE objects. This means that, while not being really explicit about it, we force the user to understand the internal-plumbing of who wrote the code associated with the object -- not very user friendly.

I'm not arguing against "Insert" -- I actually am a huge fan of having an "Insert" pulldown in a word processor (an old idea that dates back to "Cashmere" 6 or 7 years ago); Nor am I arguing against having lots of

menu items in such a pulldown -- Winword supports lots of different kinds of objects and that's great. Nor of course am I arguing against OLE -- implemented properly it will be a huge competitive advantage for Windows relative to other systems (and maybe even a competitive advantage for MS apps too). My specific issues are:

1. We should have a consistent UI model for objects regardless of whether they are implemented internally in the App or via OLE. Long term we will move more and more stuff to OLE implementations -- this should be as seamless as possible to the user.
2. I don't want to seem too theological but I do think that it would be a useful exercise for us to build a "conceptual hierarchy" of the object types in Windows and in our applications. This kind of abstraction would be super useful so that individual apps designers could build apps that "felt" consistent to the user, even if some had more menu items than the other and of course individual apps had non-overlapping feature sets.

In particular one thing that is often confusing is the difference between compound objects and what I call "media elements." In my vernacular objects are made up of one or more fundamental media elements, plus of course a bunch of code that links these component elements in space and/or time. Audio is a fundamental media element, as are text, and graphics (one could argue that vector graphics and bitmaps are two different media elements at least from a technology standpoint). Pretty much every other object type is built on these fundamental primitives.

3. Based on this view, audio is not just one object to be buried in an "Insert Objects" dialog box, it's a fundamental building block that will be useful for many different object types such as voice, music, and motion video. It's great that in the age of OLE we now know how to implement object-specific functions (methods) once and have apps share them. However they may be no correlation between this implementation and how important the associated object is to the app. In particular I really do believe that for e-mail audio-related objects will soon become very important; audio insertion in e-mail needs to be handled explicitly, not just via "Import Object."
4. The "Insert Object" menu in Voodoo, Winword 2, and in the OLE "standard" should probably be renamed "Insert OLE Object" or "Insert External Object" or somesuch. In the short-run at least there will be significant behavioral differences between an "external OLE" insert and an insert of one of the apps' inherent object types. (for instance the ole insert brings up a separate app with its own menus.) When something's different it's better to explain explicitly rather than to pretend it's not. Of course the best thing in the long term would be to make it totally transparent whether the methods to insert/edit/render the object was part of the app or part of an external library, but we're not there yet. Maybe Win4...

Rob

Mail-Flags: 0001
From mikemap Fri Jul 05 15:21:07 1991
To: chrisp

MS 5048931
CONFIDENTIAL