

MICROSOFT CONFIDENTIAL

TO: Bill Gates, Steve Ballmer, Cameron Myhrvold, Alistair Banks
FROM: Tandy Trower
SUBJECT: Style guidelines feedback
DATE: 4/11/89
CC: Mika Maritz, Paul Maritz, Peter Neupert, Jeff Raikes, Jody Snodgrass, Ellen Bierman

This is a summary of the details of my 4/4 meeting with Eric Meyers (user interface manager), Nolan Larsen (Director, Mac product development) and Chuck Middleton (manager, PC product development) of WordPerfect and their feedback on the style guide.

Lotus had been in contact with them over issues on the style guide. WordPerfect shares many of the same concerns that Lotus has. Like Lotus, their chief concern is transition from their DOS based products. Unlike Lotus, they do care about compatibility with their Mac products, but it is a far lesser priority.

I have attempted to include an assessment of whether some of the interface issues involve potential changes to OS/2s APIs. We need to evaluate those as well as the impact to the PM shell interface. I did not assess the impact on Windows. (Jody, please review and provide me with that information.)

In addition to addressing these concerns in our style guide, there is also the issue of how we get IBM (CUA) to buy in on these changes. In the past, we have convinced CUA to follow our some of our conventions. If we now go back and ask them to adopt positions they originally had, we may lose credibility.

I did not commit to any specific changes. I promised to review the feedback internally and get back to WordPerfect with a response.

The Major Issues

They highlighted the top five issues they have. (All of these have been also raised as issues by Lotus.) While we could decide to accommodate some of these in the style guide, it is unlikely any of them could be reflected in the PM shell for 1.2. They all have a compatibility impact on current Windows/PM applications.

1. The style guide should not require a standard use for Shift, Ctrl and Alt function keys. The major concern is that window management function keys assignments (Alt + F4 through Alt + F10, Ctrl + F4 through Ctrl + F10) rob them of many of the keys used in their DOS applications. Further, they question the value of most of these function keys, since there already is a keyboard interface (menu mnemonics) for these operations. *disagree*

They would also be happy to see F10 removed as a the menu key assignment. They are much happier *agree* with using Alt.

Recommendation:

I agree that many of window manipulation functions do not require accelerators. Some of them like Close and Maximize, have some value (Lotus agrees). The issues are interface compatibility and an implementation change. The Ctrl assignments are part of the MDI document window defintion which

X 193792
CONFIDENTIAL

Plaintiff's Exhibit
7512
Comes V. Microsoft

EXHIBIT
Meyers

an application must explicitly implement. Removing the assignments would be a style guide issue only. The Alt assignments; however, are defined by the system; an application window that enables a system menu gets these by default. The developer can remove and override the assignments. Therefore, Alt is a style and PM shell interface issue. If this is truly the long term style direction the system should be changed. An application should not have to include code to turn off a default system feature to comply with a standard convention.

Giving back most of the window manipulation command accelerators would probably resolve their biggest complaint. However, their interest goes further than this. WordPerfect would like to have us avoid recommendations for any standard accelerator conventions for common commands text formatting (Ctrl+B, Ctrl+I, Ctrl+U) or Open and Save commands (the ones our apps currently use are terrible). The underlying issue for the style guide is that there are a common set of operations for which we should provide guidelines. If they determine not to support these for compatibility purposes, that would be their choice; but we provide guidance to those for whom DOS app compatibility is not an issue.

Personally, I would be happy to tell IBM to give back F10 and only support Alt as the menu key, but I doubt that we could be successful with this. Snorkel (CUA 1.5/2.0) is now recognizing Alt. Perhaps once all these apps ship and teach Alt as the menu key, we will be able to convince IBM.

2. The Style Guide should allow the Enter key to be used for sequencing through dialog controls. The default pushbutton should not have to be highlighted when the user first opens the dialog. Their goal is to allow the user to use Enter to move through the relevant fields in the dialog. They acknowledge that Tab can be used for that, but argue that Enter is a far more established convention than Tab. I made the point that dialog boxes are frequently not a procedural arrangement, but a flat set of options, any of which the user may wish to deal with and then get out of the dialog. They agree and are comfortable with retaining Tab for navigation; they just want Enter for sequencing as well. They would support the concept of a master key that would submit the dialog, but just do not want it to be Enter.

Recommendation:

It would be an easy change to the style guide to allow an dialog to be displayed without a default button. However, to allow the Enter key to be a sequencing key is a significant design difference. In the PM shell and in most Windows/PM apps, there will always be a default button and Enter will press it. I am not completely sold on the idea that Enter should sequence. The following responses are possible:

- reject the request and risk alienating WP and Lotus
- incorporate the request as a replacement of standard dialog behavior and break the model for existing GUI apps
- recommend their design as an alternate model for dialog interaction
- integrate the behavior as a extension or option of the standard model.

The last two options make the most sense; however, we risk both complicating and watering down our guidelines in doing so. These are really two different models of interaction.

Regardless of the style issue, there is also the implementation issue. PM is flexible enough allow a developer to look for the Enter key and set focus to a particular control. However, it requires additional overhead. If we agreed that the proposed behavior should be standard, it should be better integrated in the system code.

X 193793
CONFIDENTIAL

However, there is still an issue as to what key to assign as the master submit dialog key as well as the visual for it. Alt + Enter would provide a reasonable choice. However, if the bold border is an indication of which button is mapped to Enter, then in dialog where Enter is used for sequencing you would not expect any push button to have the bold border. Therefore we would have to come up with another visual indicator to provide the cue to the user which button Alt + Enter would press.

3. Style guide should provide more flexibility for the meaning of certain navigation keys. For example, they use Home as a navigation modifier key and Page Up/Dn for moving a physical page. They agree that there is still need for paging a screenful. At one point, Ctrl + Page Up/Dn was suggested, but it was not a unanimous recommendation.

Recommendation:

After some discussion, they understood that we must provide some general definition for developers who do not care about WP compatibility. They are not proposing that we adopt their conventions. However, they would like not to be considered in violation of the style guide by using the definitions differently. They feel they must map these keys consistent with their DOS based products or else bringing their existing user base over to PM will be too difficult.

There are four alternative responses:

- reject WordPerfect's request
- remove any key mapping recommendations
- maintain the style conventions, but state that apps that support DOS compatibility can differ
- tell them to support standard style mapping, but provide alternate DOS key mapping (their choice on which is default?).

I recommend the last alternative.

4. Style guide should not recommend auto selection of radio buttons with arrow keys. They prefer radio buttons should operate as navigate and then select (with Enter). This way the user does not lose his default choice while moving through the field. (It is curious that they are still willing to see auto-select supported for list boxes.)

Recommendation:

I just got CUA to change radio buttons auto-select. The alternatives here are:

- reject WordPerfect's requested change
- change the convention
- try to come up with some rules for when to auto-select and when not to (allowing auto-select as a developer option).

The last option is not a good choice. It will only lead to long term inconsistency. We should accept their recommendation. I have even had feedback from Windows ISVs that they would like to see radio buttons consistent with check boxes (navigate and select). Fortunately, PM permits radio buttons to be implemented either way.

5. They reject the IBM proposed mouse button 2 functionality for the following reasons:

- it is potentially confusing to users
- it provides a way to destroy selected objects (potential accidental loss of data)
- it is not a proven or accepted technique
- it is not as useful as other alternatives

X 193794
CONFIDENTIAL

Recommendation:

I am not totally in favor of the IBM proposal either. I am waiting to see the results of our usability test on a prototype of PM built this way. It may work okay for a shell type of app, like the File Manager or IBM Office, but it raises serious questions for other situations. When I questioned Mondello and Hanrahan on this issue, they basically stated that they were pushing it forward regardless. However, I have gotten Dick Berry and Mary Williams to agree to review the situation.

PM 1.2 has already implemented the support for this in the File Manager where only button 2 can be used to move files. It has also been implemented in the system so that windows/icons can be moved or sized with button 2. This latter functionality is probably okay, but it is still appropriate to escalate it as a style issue.

Other feedback

Menus

- Esc key should back a user out one level at a time in the menu hierarchy instead of all the way out (including moving back from a dialog to the menu dropdown).

Recommendation:

They would accept having one key to back up one level and another key to do a major cancel (removes entire hierarchy in one key press). We almost have this now. Cancel in PM today goes back from the dropdown to the menu bar, then back to the body of the application. The menu key always goes back to the body of the application. (In Windows, where this functionality first existed, they are reversed.) However, this breaks down if we now include dialogs in the hierarchy since Alt is used as a mnemonic access key. (This is not yet a CUA convention.) Reversing them does not help. We would have to come up with some other key. They proposed Shift + Esc. This is presently assigned to accessing the system menu, though most people believe Alt + Space to be that key. (Ellen, what is documented in IBM's PM docs and Help?)

This is a significant difference from our current interface design for dialog boxes. We have the following options:

- reject the request
- incorporate the request as the standard dialog interface
- document as an alternate dialog interface
- integrate the proposed interface in as an extension or option of the standard model.

Unless we are willing to stand our ground on this issue, the last alternative is the only realistic one. This weakens the overall impact of the style guide, but keeps WP happy. It is the most risk free. If long term, WP and Lotus are correct, then it will be easier to transition to this as the standard style.

- Cascading menus should not be restricted to one level.

Recommendation:

They understand the intent is to avoid deep hierarchies, but they want us to soften this more so that if they occasionally feel the need to include multi-layers they can. (There is no restriction in the PM code.)

X 193795
CONFIDENTIAL

I will review what we have written. Rather than a specific restriction, I will make certain that the principle and tradeoffs are emphasized.

- Style guide should not restrict the addition of commands to the control (system) menu.

Recommendation:

I explained the intent and the importance of the menu representing the object that is the window. They still wanted to be able to add their own items. It is hard to enforce a rigid restriction here, since we already break this rule in our apps. Best I can do is to soften the wording and try to stress the principle of menu consistency. That way, the guideline will remain in force in some measure.

- The contents of the File Open, File Save and File Print dialogs should not be fixed by the style guide.

Recommendation:

They agreed to provide some guideline here, but they would like it to be more general. More importantly, they want to ensure that whatever code is provided in the system be flexible enough for them to adapt to their own design. This should not be hard, since the dialogs are part of sample code now, but if we plan to add to the system APIs in later releases, the implementer should be aware of this.

- The style guide should not restrict the use of key terminology and functionality familiar to the DOS user base. For example, Edit Clear would be more appropriately called Edit Delete. Also, File Save should bring up a dialog prompting the user for the name of the file just as most DOS applications do today. Why must we force our users to adopt the Mac style in areas where DOS already has a de facto standard?

Recommendation:

I explained to them that Delete and Clear were different operations. Clear removes data only (= Blank), while Delete removes data and structure. The more frequent operation would be Delete, but for apps that require a differentiation, Clear might still be appropriate. They seemed satisfied with this approach (as is CUA, who also raised this as an issue).

The Save issue is not one of missing functionality. They simply felt that their user base would expect Save to prompt. I do not believe that we should change the style guide here. Long term as we move toward object orientation, the system should always maintain integrity between the object in memory and on disk. There should be no concept of saving to a different name as this breaks the model that the user is operating on the actual object.

Having Save operate as Save As does today encourages inconsistency. I proposed a possible compromise. They would include Save As, but not Save, in their File menu. This would keep the design consistency with the style definitions. They indicated that they could agree to that if in that case, I allowed them to remap the "S".

This is an aberration that should be avoided, but the compromise is more acceptable than their request.

X 193796
CONFIDENTIAL

- * Numbers should be acceptable as menu mnemonics.

Recommendation:

I suggested would be acceptable in menu dropdowns or as an integral part of the title of a dropdown (e.g. C3PO). They want nothing more than that. It is CUA that actually makes the stronger restriction here. I can try to get CUA to change this.

Dialogs

- * Style guide should not require the use of Alt key for mnemonic selection within a dialog.

Recommendation:

The system supports letter and Alt+letter. I clarified why Alt+letter (to get out of list boxes and edit fields) and said that they could document either one. However, if an application wishes to avoid having Alt+letter do dialog navigation, then it must provide code to intercept Alt+letter combinations.

- * Style guide, and PM itself, should provide for the user of a mnemonic hierarchy within dialog boxes to support greater functionality for mnemonic (keyboard) manipulation.

Recommendation:

Since our applications will generally attempt to remain unique within a dialog there is no philosophical problem here. However, it definitely would require a change to the system. This is on the list proposed for 1.2, but has low priority so it is not likely to make the release. It may be still possible to do this without a change in the system. Lotus (Richard Wolf) has offered sample code to do this.

- * Style guide should not imply that group boxes are the only proper way to logically and visually group controls.

Recommendation:

Agree. I will change the style guide to suggest it as an appropriate, but not exclusive way to communicate grouping. I am not certain IBM CUA would necessarily agree.

- * Style guide should allow modal dialog boxes to be movable and have title bars.

Recommendation:

Agree. I am changing the style guide to reflect this. There will still be a visual indicator for modality, the thick, inner dialog frame. The presence of a title bar though will imply that the window/dialog is always movable.

- * Arrow keys in dialog boxes should not move in two dimensional space. This is only useful if controls happen to line up nicely, but often they do not. This feature would most likely confuse the user.

Recommendation:

Agree. This refutes the CUA proposal for 1.2 that was escalated and decided in favor of keeping 1.1 dialog behavior.

X 193797
CONFIDENTIAL

- * Style guide should not require that dialog boxes be positioned relative to the active window.

Recommendation:

We do not currently have such a definition. However, CUA does. They are willing to accept simple rules like keeping the dialog fully visible or ensuring that it is not too large, but not rules requiring fixing the position. This is no change for our style guide, but does require changing CUA.

- * Style guide should allow radio buttons to be tri-state just as tri-state check boxes are permitted.

Recommendation:

I agree, though this should not be a common occurrence.

We talked about what the visual should be. They were satisfied with having no button set, but would accept a gray fill pattern, like we use in check boxes (though, the fill pattern would require system change).

- * Style guide should not dictate a particular ordering for standard pushbuttons. Currently, only the CUA document has this guideline.

Recommendation:

- * I agree within limits. They were satisfied with my general principle of stacked on right or aligned along the bottom and even with the fact that OK should precede Cancel. Therefore, this is purely a CUA change.

Selection

- * Use of space bar for selection does not make sense. It should not be a requirement in either CUA or the MS style guide.

Recommendation:

We discussed briefly. They understand that for certain object types you need a select key. They did not have a specific recommendation for its assignment and wanted to review the latest selection spec which I promised to send them.

Navigation and scrolling

- * The style guide should not attempt to specify a particular implementation for splitting windows.

Recommendation:

They seemed to understand that I had to provide some guidelines for developers that did not have a "technique" for splitting windows. I invited them to provide me with other techniques that I could include, but they did not seem interested in that. They just wanted me to avoid making the split description in the style guide exclusive.

X 193798
CONFIDENTIAL

* The addition of the double arrows in a scroll bar is a nice concept, but a better icon should be used. PM already has too many arrow icons in it.

Recommendation:

They echo my sentiments exactly. I believe that the double arrow scroll bar went up for escalation and was killed for the 1.2 plan.

* The use of a double click in a scroll bar to split a window is not at all consistent with the meaning of a single click within a scroll bar and should not be a guideline.

Recommendation:

Agree. I clarified that this was an error in the document.

Function keys

* They are not in favor of IBM's current proposal to specify F3 as the accelerator for File Exit.

Recommendation:

Agree. I am not in favor of this either. I have tried to get CUA to change this without success. Perhaps with ISV support, I can accomplish this.

Other feedback

- They would support having Restore icons in the menu bar for a child window when it is maximized such that it takes over the parent window.
- They support the idea that Tab/Enter should navigate in a dialog box.
- They agree that pop-up menus should come up on button down, not just click.
- They agree that the cursor should be moved from the insertion point after a left/right arrow key is pressed.
- They prefer Search and Replace from the Find/Replace alternatives.

X 193799
CONFIDENTIAL