
UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte MIROSLAV TRAJKOVIC,
VASANTH PHILOMIN,
and SRINIVAS GUTTA

Appeal 2007-0145
Application 10/183,797[1]
Technology Center 2100

Decided: April 12, 2007

Before LEE E. BARRETT, JOSEPH F. RUGGIERO, and
JOSEPH L. DIXON, *Administrative Patent Judges*.

BARRETT, *Administrative Patent Judge*.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) from the final rejection of claims 1-27. We have jurisdiction pursuant to 35 U.S.C. § 6(b).

We affirm-in-part.

---

[1] Application for patent filed June 27, 2002, entitled "Active Window Switcher."

BACKGROUND

The claims are related to inactivating a window if a time period of inactivity is detected to be above a predetermined time threshold. Such a window is called a "protected window." The inactivating may include minimizing the protected window, closing the protected window, or changing the protected window to a default window.

Claim 1 is illustrative:

> 1. A method for automatically switching a protected window displayed on a monitor, the monitor being connected to a processor which receives external input, the method comprising:
>
> detecting a time period of inactivity of the external input; and
>
> inactivating the protected window if the time period of inactivity detected is above a predetermined threshold.

THE REFERENCES

The Examiner relies on the following prior art references:

Hale	US 5,355,414	Oct. 11, 1994
Maddalozzo	US 6,445,400 B1	Sep. 3, 2002 (filed Sep. 23, 1999)
Shinya	JP 2002-091418	Mar. 27, 2002

THE REJECTIONS

Claims 1, 2, 5-11, and 14-19 stand rejected under 35 U.S.C § 103(a) as unpatentable over Hale and Maddalozzo.

Claims 3, 4, 12, 13, and 20-27 stand rejected under 35 U.S.C § 103(a) as unpatentable over Hale and Maddalozzo, further in view of Shinya.

DISCUSSION

Claim interpretation

We interpret a "window" to be a viewing area on a screen that contains an application or part of an application. Windows can be displayed to take up the full screen or tiled as in Appellants' Figure 3. A "window" environment requires an operating system (OS), such as Microsoft Windows OS or the Apple MacIntosh OS, to provide for multiple windows; however, the term "window" does not imply Microsoft Windows. Only one window at a time is "active" (i.e., being worked in). We find that ordinary computer users were familiar with windows in, at least, the Microsoft Windows OS environment at the time that this application was filed (in 2002).

A "protected window" refers to an active window that is to be inactivated after a predetermined period of time in which there is no detected activity (Specification ¶ 0032). Although the Specification discloses that a window may be protected because it displays secret or sensitive information, this is not claimed. The claims do not require that the "protected window" is one of plural windows displayed on the monitor screen.

The limitation "inactivating the protected window" means that an "active" window (i.e., one that is being worked in) is made "inactive." Although the Specification describes that "inactivating" involves concealing the contents of the protected window in some manner, such as by minimizing the protected window, closing the protected window, or changing the protected window to a default window (Specification ¶ 0014),

the term "inactivating" does not suggest anything about visibility of the window contents (compare claims 3-5). For example, a window is "inactive" if the window is not being worked in even if it is visible. Moreover, "inactivating the protected window" is so broad that it includes turning off the computer or putting the computer into a sleep mode.

The limitation "inactivating the protected window" is open-ended and includes inactivating other windows in addition to the protected window, i.e., it is not limited to inactivating *only* the protected window. Therefore, a system that inactivates the computer with a screen saver will inactivate the protected window (as well as unprotected windows). The claims do not require that the "protected window" is one of plural windows displayed as overlapping windows on the monitor at the same time as Appellants' Figure 3. In fact, the desktop (the graphical user interface that appears when a windows environment is started) is itself considered a window.

It is noted that "inactivating the protected window" does not say anything about how the protected window is "reactivated." Claim 6, for example, recites "reactivating the inactivated window." The protected window could be reactivated by the movement of a mouse or typing a key on the keyboard as with a conventional screen saver. Claim 7 recites that reactivating comprises entering a unique password as with a conventional password protected screen saver discussed in the next section.

The "computer readable program code means for inactivating the protected window" in claim 8, the program storage device for "inactivating

the protected window" in claim 11, and the "means for providing instructions to the processor for automatically inactivating the protected window" in claim 16, do not require that the program code is executed by the main computer processor as opposed to the keyboard processor in Hale. It is noted that the monitor in Hale is under control of the keyboard processor.

Many claims read on the admitted prior art screen saver

Before discussing the Examiner's rejection, we note that most of the claims are so broad that they read on the admitted prior art.

Appellants describe the prior art as follows (Specification ¶ 0004):

> The use of screen saver's [sic] is well known in the art. Upon detecting the inactivity of a computer monitor for some predetermined period of time, settable by a user, the screen saver automatically locks the screen. The screen saver generally displays a moving image to preserve the integrity of the monitor, but can also be a blank display. When activity is again detected, e.g., by movement of a mouse or depressing of a key on a keyboard, the screen saver is unlocked and the monitor reverts to the display at the time the screen saver was activated. However, upon unlocking the screen saver, the last used window is still active. Therefore, secret or sensitive information displayed in an active window may be revealed to an unintended viewer.

Appellants describe a screen saver where the screen saver disappears (i.e., the windows are "reactivated") when you touch a key or move a mouse. Appellants do not describe a screen saver with password protection, although this was notoriously well known at the time of the invention. In a

screen saver with password protection, a password is activated with the screen saver and the user must enter a password before he or she can reactivate and access the computer. The user sets the predetermined period of time before the monitor is inactivated by the screen saver. Screen savers with password protection were well known in Microsoft Windows systems years before the filing date of this application. *See* Mark Reed, *The word on passwords*, http://www.microsoft.com/ Windows98/ usingwindows/work/ articles/002Feb/password.asp (last updated Feb. 3, 2000) (downloaded 3/29/07) (copy attached).

The conventional screen saver clearly anticipates claims 1, 2, 5-11, and 14-19, as interpreted. As noted in the above claim interpretation, "inactivating the protected window" broadly means that the window is not being worked in and does not preclude inactivating all the windows. The Examiner should consider the admitted prior art in any further prosecution.

Rejection and arguments

The Examiner finds that Hale discloses a security system that is automatically inactivated after a predetermined time interval activity, and can include blanking the screen or displaying other data on the screen, but fails to expressly disclose that a protected window is inactivated (Final Rejection 3). The Examiner finds that Maddalozzo teaches a display system where access periods are tracked within multiple windows having different access periods (Final Rejection 3). The Examiner finds that "Maddalozzo and Hale are similar in that they both activate a mechanism to deactivate the

workspace for the user after a period of inactivity" (Final Rejection 3-4). The Examiner concludes that it would have been obvious to one of ordinary skill in the art "to modify the system of Hale to incorporate the individual session or window control as taught by Maddalozzo, in order to obtain a system that is able to automatically detect for a given window when the inactivity period has expired" (Final Rejection 4).

Appellants argue that Hale relates to a computer system that automatically disables peripheral device access to the computer system after the peripheral input devices remain inactive for a predetermined period (Br. 9). It is argued that there is no disclosure in Hale of a windows-based operating system and Hale teaches away from a windows-based operating system because it addresses problems with DOS-based TSR (terminate and stay resident) programs (Br. 10). It is argued that although Hale discloses blanking the display or displaying a unique pattern on the screen, Hale does not teach that a protected window may be inactivated because Hale does not teach a window and because the keyboard controller in Hale operates independently of the host computer (Br. 10). Appellants find that Maddalozzo tracks timeouts of accesses to secured databases or other variable parameters related to the date in the windows and changes the color of the frames or borders of the windows to provide a visual warning or alarm that a particular window will require renewal (Br. 11). It is argued that Maddalozzo also fails to inactivate a protected window, because the only

consequence of the time expiring for a window is that the data session in the window will be cut off, not that the window will become inactive (Br. 13).

The Examiner responds that the claims do not require a specific operating system (Answer 19). It is argued that Hale teaches an IBM PS/2 compatible computer and does not limit its teachings to DOS-based systems (Answer 19). The Examiner contends that Hale enhances operating system security regardless of type (Answer 20). The Examiner restates that the claims do not require a specific operating system and Maddalozzo teaches that window systems can be implemented on IBM based workstations, such as the IBM PS/2 (Answer 21).

Appellants reply that the Examiner errs in finding that the claims are not limited to windows-based systems because the claims recite windows and require an operating system that includes windows (Reply Br. 3). It is argued that it is necessary that Hale pertain to operating systems with windows for the teachings or suggestions of Hale to be pertinent to the claimed subject matter and Hale does not teach a windows-based operating system (Reply Br. 3).

Analysis

The Examiner's statement that he "cannot find a specific limitation or notation in any of the independent claims that claim or require a specific operating system to perform the method claim 1, to operate in the device of claim 11 or in the apparatus of claim 16" (Answer 19) implies that there is no requirement for a "windows" operating system. We agree with

Appellants that the term "window" in the claims requires a windows operating system, although not a specific windows operating system such as Microsoft Windows. Hale does not disclose a windows operating system and does not disclose "inactivating the protected window," as found by the Examiner. Nevertheless, we agree with the Examiner's finding (Answer 20) that Hale provides security no matter what operating system is used.

The Examiner applies Maddalozzo for its teaching of individual windows with different access periods and controlling the appearance of the windows separately. Maddalozzo discloses that the "[o]perating system 41 may be one of the commercially available windows type of operating systems" (col. 3, ll. 14-16). Maddalozzo discloses that the windows can represent access to secured databases which grant access for limited time periods (col. 4, ll. 18-23) and the borders of the windows change colors to indicate that the time period of access is expiring (col. 4, l. 61, to col. 5, l. 16). It is the time period of access to the secured database that is monitored, not the period of inactivity of the window, and it is the access to the database that expires, or becomes inactivated, not the windows themselves. Thus, Maddalozzo also does not disclose "inactivating the protected window."

Neither Hale nor Maddalozzo discloses "inactivating the protected window." The Examiner concludes that it would have been obvious "to modify the system of Hale to incorporate the individual session or window control as taught by Maddalozzo, in order to obtain a system that is able to

automatically detect for a given window when the inactivity period has expired" (Final Rejection 4). It is not clear how the Examiner proposes to modify Hale in view of Maddalozzo. We agree that it would have been obvious to incorporate the window control of Maddalozzo into Hale because there is no reason why Maddalozzo's method could not be used in Hale's computer; i.e., the window control and the timed screen blanking would be independent of one another. However, the Examiner evidently proposes to somehow adapt the teaching of inactivating the display in Hale by a keyboard controller to inactivating individual windows in Maddalozzo in a windows environment. The specific way that Hale and/or Maddalozzo would have to be modified and the source of motivation for that modification has not been explained and is not clear, especially since Hale is not a windows system. It appears that the Examiner must be trying to meet limitations that are not claimed, such as inactivating individual windows, but we cannot tell without a more detailed explanation of the modifications. If Hale was a windows environment, then extending the inactivation of one window to several individual windows would have been much simpler.

Nevertheless, the rejection clearly proposes combining a windows environment as taught by Maddalozzo with the computer in Hale and this combination is all that is required for most claims. Hale is not limited to any particular operating system. Therefore, one of ordinary skill in the art would have been motivated to use any conventional operating system, such as the windows operating system as taught by Maddalozzo, in the computer of

Hale. The window control method in Maddalozzo is not necessary to the rejection. When Hale disables the peripheral input to the computer, it is necessarily "inactivating the protected window" as well as any other windows which happen to be open. Similarly, when Hale deactivates the display by blanking the screen or displaying a unique pattern (col. 13, ll. 28-34), it is necessarily "inactivating the protected window" as well as any other windows which happen to be open. Hale discloses reactivating the display using a password. This combination meets the limitations of claims 1, 2, 5-11, and 14-19, as broadly interpreted. Therefore, the rejection of claims 1, 2, 5-11, and 14-19 is affirmed.

The combination of Hale and Maddalozzo discloses deactivating the display by blanking the screen or displaying a unique pattern, and thus inactivating whatever windows are open, but it does not disclose that "the inactivating comprises minimizing the protected window," as recited in claims 3 and 12, or "the inactivating comprises closing the protected window," as recited in claims 4 and 13. The Examiner applies Shinya. Shinya discloses minimizing an inactive window after a first period of time and then closing the window after a second period of time (cover page under "Solution"). Shinya does not disclose "inactivating" a window; the window is already inactive. However, the combination of Hale and Maddalozzo only suggests blanking the whole screen or displaying a unique pattern on the whole screen, which inactivates all windows, not an individual window. That is, the claim language is met because it is very broad. In these

dependent claims, "inactivating" by "minimizing" or "closing" a protected window requires that a specific window is inactivated, which is not shown by any of Hale, Maddalozzo, or Shinya. Thus, while Shinya is a very good reference that discloses minimizing and closing an inactive window, it does not disclose inactivating a specific window after a time period. For these reasons, the rejection of claims 3, 4, 12, and 13 is reversed. The rejection of claims 20-22 is also reversed because these claims depend on claim 3.

Dependent claims 23, 24, and 27 recite "reactivating the inactivated protected window by activating a status indicator" of the inactivated window. The "status indicators" refer to the status buttons 114 and 116 on the status bar 110 (also known as a taskbar) corresponding to the windows 105a and 105b in Figures 2A-2D. When a window 105a is minimized, the status button 114 appears on the status bar 110 (Fig. 2C) with a certain appearance, and when window 105a is closed there is no status button on the status bar (Fig. 2D) (Specification ¶ 0034). An inactive window is reactivated by activating the status button. The Examiner took Official Notice that it was well known in the art that windows are represented on the taskbar by icons and that windows were reactivated by activating these icons (Final Rejection 17; Answer 17). Appellants argue that the Examiner was required to submit a reference or an affidavit to establish facts within his personal knowledge (Reply Br. 2-3).

The use of status buttons on the taskbar (or status bar) in a windows environment to represent windows was so notoriously well known that it

cannot be seriously in dispute (although it always would be better for an Examiner to provide a reference where it would be easily available). Such status buttons change appearance to indicate the active window and it was well known to reactivate an inactive window by activating a status button. Anyone who used a computer with Microsoft Windows before 2002 would know these facts. Nevertheless, claims 23, 24, and 27 require reactivating an individual window and we have concluded that the combination of Hale, Maddalozzo, and Shinya does not disclose inactivating a specific window after a certain time period. Therefore, there is no motivation for reactivating an inactive window using a status indicator. Accordingly, the rejection of claims 23, 24, and 27, and dependent claims 25 and 26, is reversed.

CONCLUSION

The rejection of claims 1, 2, 5-11, and 14-19 is affirmed.

The rejection of claims 3, 4, 12, 13, and 20-27 is reversed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv) (2006).

AFFIRMED-IN-PART

tdl/ce

PHILIPS INTELLECTUAL PROPERTY & STANDARDS
P.O. BOX 3001
BRIARCLIFF MANOR, NY 10510

Notice of References Cited	Application/Control No. 10/183,797	Applicant(s)/Patent Under Reexamination TRAJKOVIC et al.	
	Examiner	Art Unit BPAI	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification	
	A	US-				
	B	US-				
	C	US-				
	D	US-				
	E	US-				
	F	US-				
	G	US-				
	H	US-				
	I	US-				
	J	US-				
	K	US-				
	L	US-				
	M	US-				

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification	
	N						
	O						
	P						
	Q						
	R						
	S						
	T						

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Mark Reed, The word on passwords, http://www.microsoft.com/ Windows98/usingwindows/work/ articles/002Feb/password.asp (last updated Feb. 3, 2000)
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Microsoft Windows 98

Windows Home

Search for
Search

Using Windows 98 | Download | Support

Windows 98 Home
Getting Your Work Done
Communicating with Others
Exploring the Internet
Maintaining Your Computer
Having Fun

Using Windows 98
Getting Your Work Done

The word on **passwords**

by Mark Reed

Turn on your computer, and you are asked to enter a password; then you need another one to access your Internet service; and yet another to look up information on your favorite Web site. These days, it's easy to be inundated with passwords. Unlike computers where you can add as much memory as you like, ours is limited and often overtaxed.

The key is knowing when using a password is really necessary. To help simplify things, here is a quick explanation of commonly-used passwords in Windows 98, so you can make educated decisions.

Your pass to personalization
When several people in a home use the same computer, Windows 98 has a useful feature called user profiles that allows different users to maintain individual preferences for things like screen savers, wallpaper, sound effects, and Web favorites. Each time the computer is turned on, a password box asks for a name and password, so Windows 98 can apply the correct settings and to log you into your personal identity in Outlook Express.

But if you are the only person using your computer, or if your family does not use profiles, you may find you're still prompted for your name and password each time you log on to Windows 98. If this is the case, entering a password to start Windows each time you turn on your computer may be an unnecessary step.

To simplify your logon procedure, you can eliminate the Windows password. Here's how to prevent Windows 98 from prompting you for a password at startup:

1. In **Control Panel**, double-click **Passwords**.

2. Click the **Change Passwords** tab, click **Change Windows Password**, and then click **OK**.

3. In the Change Windows Password dialog box, type your current Windows password in the **Old Password** box. Leave the **New Password** and **Confirm New Password** boxes blank, click **OK**, and then click **OK**.

4. Click the **User Profiles** tab and verify that the "**All users of this PC use the same preferences and desktop settings**" option is selected. Click **Close**.

5. **Shut down** and then **restart** Windows 98.

Increase security with a screen saver password
Whether or not you use a Windows password, you may want to have your computer secure if you take a break and walk away from it. Windows 98 allows you to assign a password that is activated with your screen saver to prevent anyone else from seeing what is on your

Want to change your Windows Password?

Here's how:

1. Click **Start**, point to **Settings**, and select **Control Panel**.

2. Click the **Passwords** icon.

3. In **Password Properties**, click **Change Windows Password** button, and then click **OK**.

4. Type your old password.

5. Type your new password, then type it again in **Confirm New Password**.
Note: If you are using Windows 98 on a network, you need to be logged on to the network to change your Windows password.

computer, or using it while you are away.

Normally, when you touch a key or move your mouse, the screen saver will disappear. However when password protection is turned on, you are asked to enter a password before you can access the computer. If the wrong password is entered, the screen saver will continue to cover the screen, and the keyboard can only be used to enter the correct password.

Here's how to set up a screen saver password:

1. Open the **Display Properties** dialog box, and select the **Screen Saver** tab.

2. Select the **Password protected** box, and then click the **Change** Button.

3. Type your **password**, and then confirm the password by typing it again.

Be sure to check the amount of time before the screen saver activates. The length of time should be short enough to protect your computer when you are away, but not so short it turns on every time you pause to look at your notes or take a sip of coffee.

Passwords are very important any time security is an issue, but if you are the only person in your house using your computer, simplify your passwords and you may be able to get rid of a few of those sticky-notes with cryptic words scribbled on them stuck to the edges of your monitor. ⊞

Mark Reed can never remember all of his passwords.

Last updated: Thursday, February 03, 2000

Top of Page