

# **MonitorMouse<sup>®</sup>**

**FOR WINDOWS NT**  
**Version 2.0**

Manual Version 2.0

# **MonitorMouse<sup>®</sup> for Windows NT**

**Version 2.0  
User's Guide**

Manual Version 2.0

## Trademark Acknowledgements

IntelliTouch, AccuTouch, and MonitorMouse are registered trademarks, and ELODEV is a trademark of Elo TouchSystems, Inc. All other trademarks are the property of their respective holders.

## Copyright

Copyright © 1997 by Elo TouchSystems, Inc. All rights reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under copyright laws. Printed in USA. H55268.

## Software License Agreement

BY OPENING THE ACCOMPANYING DISK ENVELOPE, YOU ARE AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, INCLUDING THIS SOFTWARE LICENSE AND LIMITED WARRANTY.

### *Software License*

This software is protected by both the United States copyright law and international treaty provisions. Therefore, except as noted below, you should treat the software just like any other copyrighted material. Elo TouchSystems, Inc. (Elo) authorizes you to make archival copies of the software for the purposes of backing up your software and protecting your investment from loss, and to make additional copies for use within a single company or facility.

THIS SOFTWARE IS LICENSED FOR USE ONLY WITH ELO TOUCHSCREENS.

The enclosed software program object code (drivers, utilities, diagnostics, and/or demonstration programs) may be freely duplicated or distributed without charge, but may not be resold. You may not decompile, reverse assemble, reverse engineer, or patch any software program object code.

Any supplied software program source code is proprietary and may not be disclosed to third parties. Such source code may be modified and/or partially or completely incorporated into your own applications, together with any supplied object code, and the resulting programs may be used, given away or sold without additional licenses or fees.

You may not reproduce, distribute, or revise the program documentation without expressed written consent from Elo.

This software and accompanying written materials may contain reference to, or information about, Elo products (equipment or programs), that are not now available. Such references or information must not be construed to mean that Elo intends to provide such products, programming, or services.

### *Limited Warranty*

THIS SOFTWARE AND ACCOMPANYING WRITTEN MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. FURTHER, ELO DOES NOT GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR ACCOMPANYING WRITTEN MATERIALS IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY OR CURRENTNESS. IF THE INCLUDED SOFTWARE OR ACCOMPANYING WRITTEN MATERIALS ARE DEFECTIVE, YOU, AND NOT ELO OR ITS DEALERS, DISTRIBUTORS, AGENTS, OR EMPLOYEES, ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE AND ANY FURTHER PROGRAMS OR WRITTEN MATERIALS DEVELOPED UTILIZING THESE MATERIALS IS ASSUMED BY YOU.

Elo warrants only that the disk is free from defects in material and workmanship under normal use and service for a period of sixty (60) days after receipt.

Elo's entire liability and your exclusive remedy as to the disk shall be, at Elo's option, either return of the purchase price or replacement of the disk.

EXCEPT AS PROVIDED ABOVE, ELO DISCLAIMS ALL WARRANTIES, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE, WRITTEN MATERIALS OR DISK. IN NO EVENT SHALL ELO BE LIABLE FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND.

### *Governing Law*

This Agreement shall be governed by and construed in accordance with the laws of the State of Tennessee.

# CONTENTS

<b>Chapter 1 - Introduction .....</b>	<b>1</b>
About this Manual .....	1
About MonitorMouse for Windows NT .....	2
Touchscreen Controller Compatibility .....	2
Software Compatibility .....	2
Video Compatibility .....	3
Other Versions of MonitorMouse .....	3
Product Registration .....	3
Contacting Elo.....	3
<b>Chapter 2 - Installing MonitorMouse for Windows NT .....</b>	<b>5</b>
Before You Begin .....	5
Configuring the Controller .....	6
Installing the Software.....	6
MonitorMouse for Windows NT Files .....	9
Modifications to Windows NT Configuration .....	11
<b>Chapter 3 - Using MonitorMouse for Windows NT .....</b>	<b>13</b>
Touchscreen Control Panel .....	13
Running the Touchscreen Control Panel .....	14
Calibration .....	14
Mouse Button Emulation Mode .....	15
Double-Clicking .....	15
Click Sound on Touches .....	16
Tips on Using the Touchscreen in Windows NT .....	17
Removing the Mouse Pointer .....	17
Border Width .....	17
Scrolling.....	18
Maximizing Windows.....	18
Using the Overscan Area .....	18
Option Buttons and Check Boxes .....	18
Choosing the Control Buttons on Window Title Bars .....	18
Running DOS Programs from NT .....	19
Tips on Public Access Touchscreen Applications .....	19
<b>Appendix A - Troubleshooting .....</b>	<b>21</b>
General Information .....	21
Installation Problems.....	21
Display Problems .....	22
Touch Problems .....	22
Diagnostic Utilities.....	23
BUSSTAT .....	23
COMDUMP .....	24
SAWDUMP.....	25

**Appendix B - Programming Options.....27**  
Z-Axis Capabilities ..... 27  
Launching the Touchscreen Control Panel.....29  
Calibrating from Within Your Program .....29

---

# Introduction

- *About this Manual 1*
  - *About MonitorMouse for Windows NT 2*
    - *Product Registration 3*
    - *Contacting Elo 3*
- 

## ABOUT THIS MANUAL

This manual provides all of the information you need to install and use MonitorMouse® for Windows NT, Elo TouchSystems' touchscreen driver designed for Windows NT. This manual is organized as follows:

- |            |   |
|------------|---|
| Chapter 1  | Describes the software included, and the hardware requirements.           |
| Chapter 2  | Describes how to install the MonitorMouse for Windows NT software.        |
| Chapter 3  | Details the use of MonitorMouse for Windows NT and its options.           |
| Appendix A | Gives tips on troubleshooting a MonitorMouse for Windows NT installation. |
| Appendix B | Gives programming options for Windows NT applications.                    |

See the README.TXT file on the accompanying disk. It contains any additions or changes made after this manual was printed.

For details on available product configurations, touchscreen theory of operation, and general troubleshooting, see the *IntelliTouch Ultra Product Manual* or the *AccuTouch Product Manual* as appropriate.

## ABOUT MONITORMOUSE FOR WINDOWS NT

Included on the accompanying disk is MonitorMouse for Windows NT, a 32-bit touchscreen driver designed for Microsoft Windows NT. MonitorMouse for Windows NT allows you to run all Windows applications without modification, using an Elo touchscreen.

MonitorMouse for Windows NT and an Elo touchscreen are recommended for all Windows NT applications that require an intuitive user interface, such as a public kiosk application. Touchscreens are also popular in point-of-sale, process control, instrumentation, multimedia, gaming, education, simulation, and training applications. Elo touchscreens may be used in a harsh environment not appropriate for a mouse or keyboard.

With MonitorMouse for Windows NT, the touchscreen may be used alone, or in conjunction with any mouse. Various *mouse button emulation modes* provide options on how button clicks are simulated. You can choose the best mode for your application, and change between modes at any time.

Applications using only the left button are the easiest to use with the touchscreen.

## Touchscreen Controller Compatibility

Elo manufactures two types of touchscreen technologies. The IntelliTouch® surface wave technology provides excellent clarity and durability. The AccuTouch® resistive technology features pressure-activated screens with rapid response and high resolution. PC-Bus and serial touchscreen controllers are available.

MonitorMouse for Windows NT will work with all current Elo touchscreen controllers. A complete list is included in the README.TXT file on the accompanying disk.

## Software Compatibility

MonitorMouse for Windows NT is compatible with Windows NT and all Windows applications.

MonitorMouse for Windows NT also works with most DOS applications run in a window on the Windows NT desktop.

## Video Compatibility

The video function of the display and the Elo touchscreen installed on the display are entirely separate systems. Therefore, the touchscreen is compatible with all video cards and modes. Problems with the display should be treated as display problems, not touchscreen problems.

## Other Versions of MonitorMouse

MonitorMouse for Windows NT is only one in a family of MonitorMouse products for various operating systems available separately from Elo. Each has a similar "look and feel" for a consistent user interface across platforms.

MonitorMouse for DOS emulates the standard mouse driver MOUSE.COM on the IBM family of personal computers and 100% compatibles. Programs using MOUSE.COM, MOUSE.SYS, or compatible versions, will work with MonitorMouse. ELODEV™, the Elo touchscreen driver program for DOS, is also included.

MonitorMouse for Windows 95 works with all Windows applications and DOS applications run in a window on the Windows 95 desktop.

MonitorMouse for OS/2 works with all Presentation Manager and WIN-OS/2 mouse-driven applications. It also works with all OS/2 Full Screen and DOS applications run on the OS/2 desktop.

MonitorMouse for Macintosh works on the Apple Macintosh computer with System 6 or System 7.

## **PRODUCT REGISTRATION**

For the latest product announcements and software releases, be sure to fill out and return the product registration card packaged with this manual.

---

# Installing MonitorMouse for Windows NT

- *Before You Begin 5*
- *Configuring the Controller 6*
- *Installing the Software 6*
- *MonitorMouse for Windows NT Files 9*
- *Modifications to Windows NT Configuration 11*

---

## BEFORE YOU BEGIN

First, be sure Windows NT is installed and operating properly with your mouse.

You should make a backup copy of the MonitorMouse for Windows NT installation floppy.

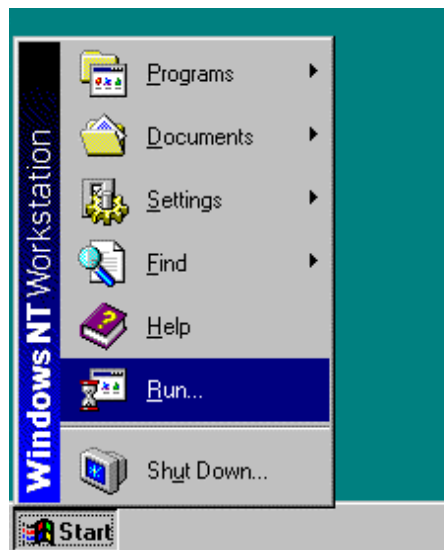
## CONFIGURING THE CONTROLLER

Elo typically ships touchscreen controllers preconfigured for use with Elo software, including MonitorMouse for Windows NT.

Install the Elo touchscreen hardware as described in the *AccuTouch Product Manual* or *IntelliTouch Ultra Product Manual*.

## INSTALLING THE SOFTWARE

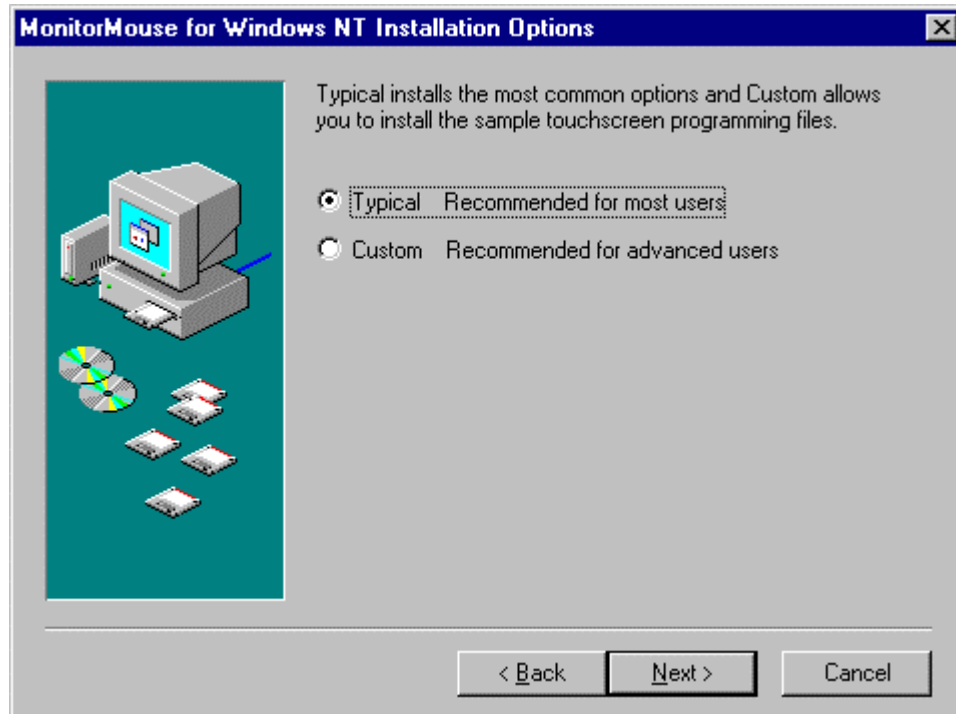
1. Start your computer.
2. Insert the MonitorMouse for Windows NT disk into drive A.
3. Click the Start button then click Run.



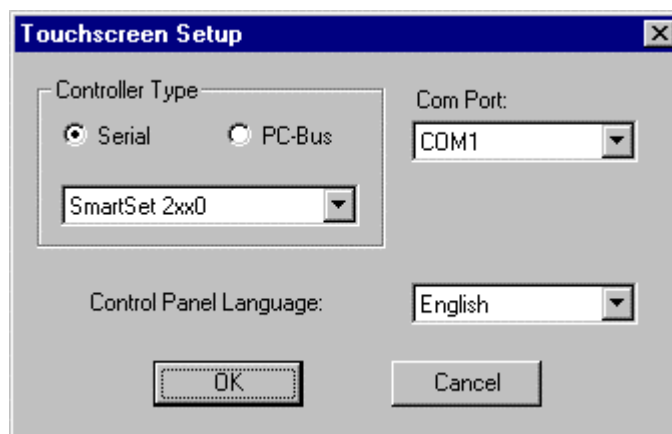
4. Type “a:\setup” in the space provided and press Enter.



5. Follow the directions on the screen.
6. MonitorMouse for Windows NT provides two Installation options. Most users should select Typical (the default) and click Next to continue. Custom allows you to install the sample touchscreen programming files.

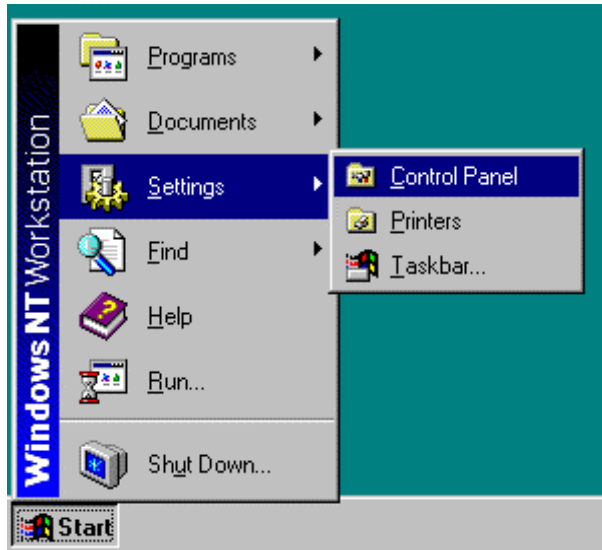


7. The Touchscreen Setup dialog box will appear. Specify the type of touchscreen controller you are using and how it is connected. Also specify the language you want to use in the Touchscreen Control Panel. Click OK to confirm that the Setup is correct.

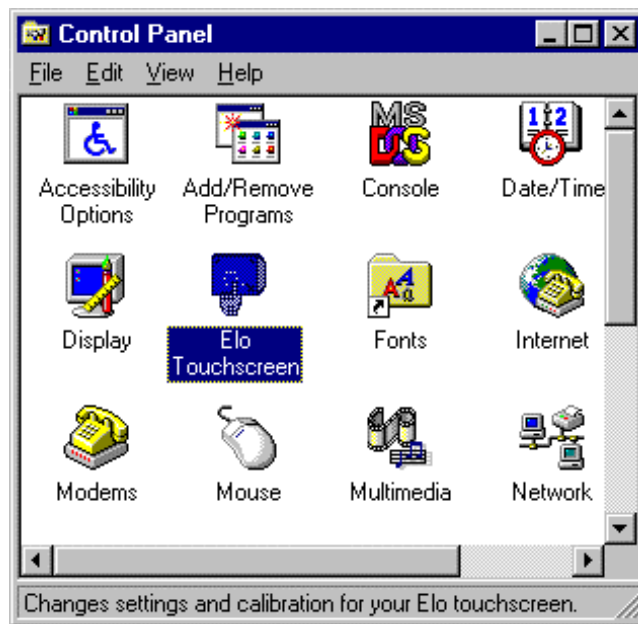


8. Complete the Setup program.
9. Shutdown and restart Windows NT. Continue the steps on the next page.

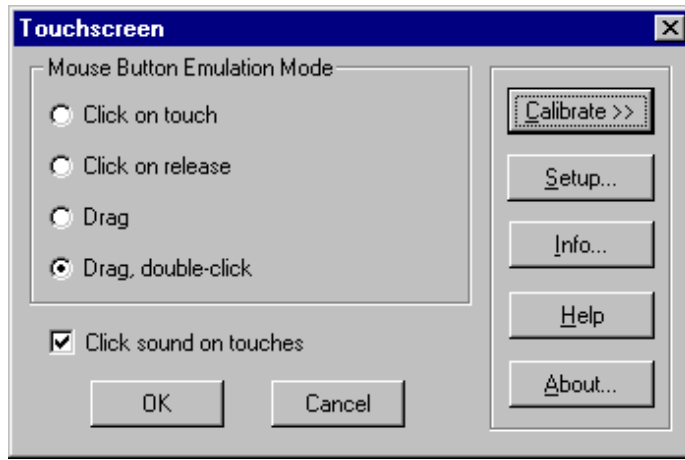
10. Click the Start button, then click Settings, then click Control Panel.



11. Double-click Elo Touchscreen to run the Touchscreen Control Panel.



12. Click the Calibrate button and touch each of the three targets as they appear on the screen. Touch Yes when the cursor lines up correctly with your finger. Touch OK to close the Touchscreen Control Panel.



## MONITORMOUSE FOR WINDOWS NT FILES

The following files are on the MonitorMouse for Windows NT disk:

README.TXT	Text file containing any additions or changes made after this manual was printed.
_SETUP.DLL	Setup program component.
_SETUP.LIB	Setup program component.
DATA.Z	File containing compressed driver files.
DISK1.ID	Disk ID file. Setup program component.
SETUP.EXE	Setup program.
SETUP.INS	Setup program component.
SETUP.PKG	Setup program component.
COMDUMP.EXE	DOS program for testing serial touchscreen controllers.
BUSSTAT.EXE	DOS program for testing PC-Bus touchscreen controllers.
SAWDUMP.EXE	DOS program for testing IntelliTouch touchscreen controllers.

The following files are installed by the MonitorMouse for Windows NT Setup program when using the Typical installation option:

README.TXT	Copied to \ELO directory.
MONMOUSE.SYS	Touchscreen driver designed for Windows NT. Copied to \WINNT\SYSTEM32\DRIVERS directory.
MONMOUSE.CPL	Touchscreen Control Panel program. Copied to \WINNT\SYSTEM32 directory.
MONMOUSE.HLP	Touchscreen Control Panel Help file. Copied to \WINNT\SYSTEM32 directory.
NULL.CUR	Invisible cursor file. Copied to \WINNT\CURSORS directory.
COMDUMP.EXE	Copied to \ELO directory.
BUSSTAT.EXE	Copied to \ELO directory.
SAWDUMP.EXE	Copied to \ELO directory.

## MODIFICATIONS TO WINDOWS NT CONFIGURATION

The MonitorMouse for Windows NT Setup program makes the following modifications to Windows NT:

1. Copies driver files to your hard disk and updates previous installations of Elo driver software.
2. Adds the following Registry keys. The default values of these keys are shown, but some values are determined by the type of touchscreen controller installed and calibration values.

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MonMouse
  \ErrorControl: REG_DWORD:0x1
  \Group:REG_SZ:PointerPort
  \Start:REG_DWORD:0x1
  \Tag:REG_DWORD:0x2
  \Type:REG_DWORD:0x1
```

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MonMouse\Parameters
  \BeepOnOff:REG_DWORD:0x1
  \ComPort:(dependent on controller)
  \ControllerAddress:(dependent on controller)
  \ControllerLength:(dependent on controller)
  \ControllerLevel:(dependent on controller)
  \ControllerModel:(dependent on controller)
  \ControllerType:(dependent on controller)
  \ControllerVector:(dependent on controller)
  \CplLanguage:REG_DWORD:0
  \MouseDataQueueSize:REG_DWORD:0x64
  \MouseMode:REG_DWORD:0x6
  \NumberOfButtons:REG_DWORD:0x2
  \OverrideHardware Present:REG_DWORD:0
  \PointerDeviceBase Name:REG_SZ:PointerPort
  \SampleRate:REG_DWORD:0x28
  \SwapButton:REG_DWORD:0
  \SwapFlag:REG_DWORD:0
  \XHigh:(dependent on calibration)
  \XLow:(dependent on calibration)
  \YHigh:(dependent on calibration)
  \YLow:(dependent on calibration)
  \ZHigh:(dependent on calibration)
  \ZLow:(dependent on calibration)
```

```
HKEY_CURRENT_USER\Control Panel\Mouse
  \DoubleClickWidth:REG_SZ:25
  \DoubleClickHeight:REG_SZ:25
```

```
HKEY_LOCAL_MACHINE\SOFTWARE\Elo TouchSystems\MonitorMouse for Windows NT
```



---

# Using MonitorMouse for Windows NT

- *Touchscreen Control Panel 13*
- *Tips on Using the Touchscreen in Windows NT 17*
  - *Running DOS Programs from NT 19*
- *Tips on Using the Touchscreen in Windows NT 19*

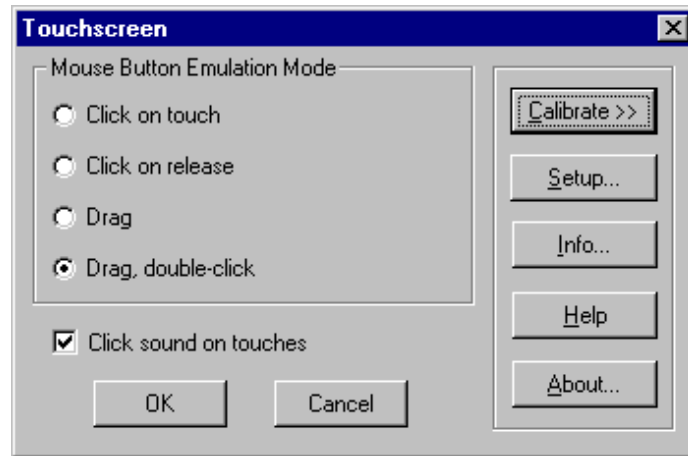
---

## TOUCHSCREEN CONTROL PANEL

The MonitorMouse for Windows NT Touchscreen Control Panel allows you to customize the way the touchscreen works with Windows NT. The Touchscreen Control Panel is detailed in the following sections.

## Running the Touchscreen Control Panel

1. Touch the Start button, then touch Settings, then touch Control Panel.
2. Double-touch Elo Touchscreen to run the Touchscreen Control Panel.



## Calibration

Select Calibrate to initiate the calibration sequence. Follow the on-screen instructions. You will be given the opportunity to confirm the calibration points.

The need for calibration is unique to the touchscreen. Unlike mouse or keyboard applications where the cursor is part of the image, a touchscreen is a physical overlay with an independent coordinate system. Only by knowing the position of the image can the driver software convert touchscreen coordinates into image coordinates.

Besides the differences in touchscreens and controllers, calibration also compensates for variations in the video image among displays. Calibration is also affected by horizontal and vertical adjustments on the monitor and by the physical mounting of the touchscreen.

Additional calibration complications include image blooming, where brightly-colored images expand, and the "pincushion" effect, which causes the corners of the display to be stretched. Poor display linearity can cause similarly sized boxes to be larger at the edges of the screen than they are in the middle, or vice-versa. The displayed image can also be tilted. Even changing video modes can affect the screen size.

The three-point calibration sequence used by the Touchscreen Control Panel automatically corrects inverted touchscreen installations and backwards cable connections.

Once calibrated, MonitorMouse for Windows NT will be ready to run automatically each time the system is restarted. Recalibration should only be necessary after moving or resizing the video image, or after changing either the touchscreen, controller, or monitor.

## Mouse Button Emulation Mode

Four mouse button emulation modes are supported, equivalent to the corresponding mode numbers in the other versions of the MonitorMouse family of products. The current mode may be changed with the Touchscreen Control Panel. “Drag, double-click” is the default.

The modes vary in the button actions they support (clicking, dragging, double clicking, etc.) You should choose the *simplest* mode that will work for your application.

**Click on touch**      Left button clicking only. No dragging or right button is supported. Works with large targets. Recommended for public kiosk applications.

Moves the mouse pointer to your point of touch and “clicks” the left button. A beep will confirm the click. The mouse pointer will not move, nor will any other clicks be simulated until your finger is lifted and the screen retouched.

**Click on release**      Left button clicking only. No dragging or right button is supported. Works with large and small targets.

The mouse pointer follows your sliding finger and the left button is clicked at the point of release (lift-off).

**Drag**      Left button clicking and dragging. No right button is supported.

The left button is held down when you touch the screen, and it stays down until you release, dragging as you move.

**Drag, double-click**      Same as Drag, but supports double clicking by tapping twice. Recommended for general use.

If the second touch occurs immediately after the first and is nearby, a double-click will be simulated at the exact point of the first touch. If you do not hear two beeps, you have tapped too fast.

### Double-Clicking

Double-clicking is common in the Windows NT environment, so the Drag, double-click emulation mode is recommended. If you are running a dedicated

application that does not use double-clicking, you may use one of the other modes.

The double-click speed for the mouse and touchscreen can be adjusted with the mouse Control Panel. It may be helpful to adjust this threshold toward the slower end of the scale.

The mouse parameters `DoubleClickHeight` and `DoubleClickWidth` are modified by the Touchscreen Control Panel when you select a mouse button emulation mode. These parameters determine the horizontal and vertical range for a legitimate double-click. If you invoke `REGDT32.EXE`, these parameters are located in `HKEY_CURRENT_USER` under Control Panel, in the Mouse area.

If you do not find the values chosen by the Touchscreen Control Panel acceptable, simply modify them with `REGDT32.EXE`.

Note: For the `DoubleClickWidth` and `DoubleClickHeight` values to go into effect, you must restart your system.

## Click Sound on Touches

A check box is available in the Touchscreen Control Panel for enabling/disabling a "click" sound from the PC's internal speaker for each touch. Audible feedback is important because the touchscreen does not have any tactile feedback mechanism. The "click" sound is enabled by default.

## TIPS ON USING THE TOUCHSCREEN IN WINDOWS NT

The following are suggestions for using the touchscreen with Windows NT. The Drag, double-click mouse button emulation mode is assumed.

### Removing the Mouse Pointer

A mouse pointer is not required in most touchscreen applications and may actually be a distraction. The user's eyes will follow the pointer's movement instead of focusing on the application. With a pointer present, the user performs the indirect act of moving the arrow to an object instead of the direct, natural act of touching the object.

If you do wish to eliminate the mouse pointer, you may replace it with the invisible cursor file, NULL.CUR, installed by MonitorMouse for Windows NT. Install the invisible cursor file as follows:

1. Open the Control Panel and select Mouse.
2. Select the Pointers tab.
3. Highlight the Normal Select cursor then select Browse.
4. Type "null.cur" in the space provided and select Open.
5. Select OK to confirm the cursor change.
6. Exit from the Mouse Control Panel. You should notice that the arrow cursor has disappeared. Note that all other cursors will still function as before.

### Border Width

To make resizing windows easier with the touchscreen, increase the size of the border in the Windows NT Control Panel. Select Display, then the Appearance tab. Select the Active Window Border from the Item list and enter a new Size. (17 is recommended for 640x480 resolution, 26 for 1024x768). For other resolutions, pick a border size that visually matches the title bar size. Wide borders may look strange at first, but after a while they will seem normal.

## Scrolling

Scrolling lists can be performed by touching the scroll bars, as you would with the mouse. Some lists can be scrolled more easily by touching the data area, sliding your finger to an edge, then continuing to slide outside the area. This technique works in drop-down lists, for example. In fact, the farther away you slide from the list, the faster the data scrolls.

## Maximizing Windows

To maximize a window, double-tap on the window's title bar, rather than tapping the Maximize button. Likewise, to restore a maximized window to its previous size, double-tap its title bar instead of tapping the Restore button. You can also use the Control menu to maximize, minimize, and restore windows.

## Using the Overscan Area

MonitorMouse for Windows NT pushes all touches in the display's overscan area within the screen image. This effectively enlarges the active touch area of zones on the edges of the image. You may touch the edges and corners of the CRT face to easily click on items displayed at the extremes of the image. For example, when a window is maximized, you may touch the upper-right corner to restore the window, touch the top to click on the window title, and double-tap the upper-left corner to close the window.

## Option Buttons and Check Boxes

It is easier to select an option by touching the option name rather than touching the option button. The same is true with check boxes.

## Choosing the Control Buttons on Window Title Bars

Windows can be maximized and restored easily by double-tapping the title bar as described above. The Minimize, Maximize, and Close buttons, and the Control-menu can be difficult to touch, as they are quite small on high-resolution screens. Since they all act on button release, you can slide among them before selecting as with the drop-down menus. You can also use the Control menu to easily maximize, minimize, restore, and close windows.

## **RUNNING DOS PROGRAMS FROM NT**

DOS mouse-driven programs run from Windows NT may also use the touchscreen when run in a window.

Limited support for Full-screen DOS mode is provided. The video mode in Windows NT and the video mode in the DOS application must be the same in order for the cursor to correctly track your finger. If the video modes are not the same the cursor will move as if out of calibration. A mouse can be used to access all programs run in full-screen DOS mode.

## **TIPS ON PUBLIC ACCESS TOUCHSCREEN APPLICATIONS**

If your touchscreen application is used in a public access setting such as an airport kiosk, the following tips may be helpful:

1. Run the application full screen. Do not have a title bar, menu bar, etc.
2. Use bright background colors (never black) to hide fingerprints and reduce glare. A patterned background, such as dithering, will also help the eye focus on the image rather than reflections, especially in background areas where there are no icons or menu choices. Fingerprints make a system look dirty and unattractive.
3. Use a simple point-and-click interface with large buttons. Do not use dragging, double clicks, scroll bars, drop-down menus, multiple windows, etc.
4. Turn off the cursor. Users will focus on the entire screen instead of the arrow. The presence of a cursor makes the user subconsciously think "how do I get the arrow to do what I want?" By removing the cursor, the user's thinking/action is direct instead of indirect, unlocking the true power of touchscreens.
5. Always give feedback to the user instantly on touch. A 3-D button effect works well (such as a Windows standard button). Output a click sound, etc. Clear the screen while searching for the next screen. Use an hourglass. Immediate feedback is necessary because the touchscreen has no tactile feedback mechanism.
6. Optimize the application for speed. A quick response keeps the user's attention. Users will walk away from a sluggish system. Speedy systems also reduce vandalism. Do not use a graphics mode with higher than necessary colors/resolution. It will only slow down the system. (Use of 256 colors is typically more important than resolution above 640x480.)

7. The application must be intuitive. Guide the user. Limit choices. Test the application on focus groups. If anyone pauses for even a moment, think how to improve the application so it is totally obvious.
8. Consider using a sound card with digitized speech to prompt the user through the application. For example, "Touch the first letter of the company you are looking for." Click. "Now touch OK." Click. There is something almost magical about a user interface with voice prompting and touch response. Your brain can simultaneously process voice while absorbing an image.
9. The main screen must attract the user from a distance. Animation and large fonts are ways to catch the eye. An attractive kiosk design is also important.
10. Other kiosk design tips: If forced air ventilation is used, position the fan at the top near the monitor's vents. The intake should not be near the floor where airborne dust from footsteps can become a problem. Air should not enter around the CRT face either. The kiosk design should accommodate variation in monitor dimensions and bezels, as monitor models may come and go every few months. Speakers should point towards the user's face, not their knees. Choose a finish that does not show fingerprints. Avoid polished stainless steel, chrome, and glossy black paint.

All of the above suggestions may seem obvious, but it is surprising how many developers miss some of these points. Sometimes a simple change to the application can mean the difference between a project's success or failure.

# Appendix A

---

## Troubleshooting

- *General Information 21*
  - *Diagnostic Utilities 23*
- 

### GENERAL INFORMATION

Should you experience operational difficulties with the touchscreen system either during or after installation, this appendix should help you determine the source of the problem.

The first step in troubleshooting a touchscreen system is to determine whether the problem is due to the hardware, driver software, or application software.

#### Installation Problems

##### **The Setup program fails to run.**

Most likely another program is causing a conflict with the MonitorMouse for Windows NT Setup program. Remove all programs from your Startup group and restart your system. Close all running programs. Retry running the Setup program.

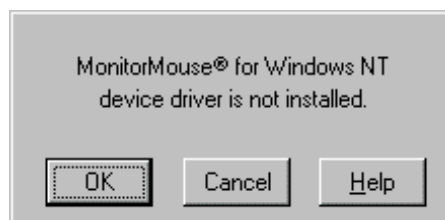
## Display Problems

If you are experiencing a display problem (such as no video, horizontal and vertical synchronization, raster non-linearities, etc.), realize that the video function of the display and the Elo touchscreen installed on the display are entirely separate systems. Therefore, problems with the display should be treated as display problems, not touchscreen problems. Diagnostic procedures and possible corrections for display problems should be performed using the troubleshooting procedures outlined in your display manual.

The touchscreen is powered from its controller. Sometimes the display and touchscreen controller have separate power supplies and operate independently. Thus it is possible for the touchscreen to interact with the computer even if the display is powered off. Internal serial controllers are usually powered by a supply that is separate from the display's power supply, but controlled through the display's power switch. In this case, even if display problems exist, the touchscreen system will probably function if the display power switch is on.

## Touch Problems

### **The touchscreen doesn't work and I get the following message when I run the Elo Touchscreen Control Panel:**



This message means that the driver failed to load because the touchscreen controller you specified in Setup could not be found.

- Click OK, then the Setup button to verify that the touchscreen setup parameters match your touchscreen hardware configuration. Restart your system.
- Make sure the touchscreen controller is powered up before powering the PC.
- Check all the cables and restart your system.
- Check your controller's switch and/or jumper settings.
- Boot the system with DOS. Use the DOS-based diagnostic utilities to help diagnose hardware problems. These programs are copied to the \ELO

directory by the Setup program. See COMDUMP and BUSSTAT later in this appendix.

**The cursor moves but does not line up with my finger.**

You need to run the calibration routine in the Control Panel.

- Open the Elo Touchscreen Control Panel and select the Calibrate button.

**Errors appear in the Event Log but the touchscreen still works.**

This is normal. Non-fatal errors are logged in the Event Log when the touchscreen is correctly installed. These errors are a warning that there is a normal resource overlap in the system. Ignore these non-fatal errors.

## DIAGNOSTIC UTILITIES

The following pages detail the use of the DOS-based diagnostic utilities (typically found in the \ELO directory).

### BUSSTAT

BUSSTAT.EXE allows you to test your Elo PC-Bus or Micro Channel controller from DOS. The utility displays the controller's switch or jumper settings, then displays the touch coordinates as they are received. BUSSTAT polls the controller directly rather than being interrupt-driven. This can help determine if a problem is interrupt related.

If on startup, Windows NT reports an error message that the controller cannot be found, or the driver did not load, run BUSSTAT from the DOS prompt as follows:

```
BUSSTAT [Enter]
```

The program will prompt you for the controller type and Base I/O Port. Initially, it should not be necessary to remove the Interrupt (IRQ) jumper.

Compare the displayed switch or jumper settings to those listed in the *IntelliTouch Product Manual* or the *AccuTouch Product Manual*. A common problem results from inverting all the switch settings.

Next, touch the touchscreen and see the raw (untranslated) coordinates. Normally, data is displayed at a constant rate only when the touchscreen is touched. If no data is displayed on touch, or if data is displayed when the touchscreen is not touched (either continuously or intermittently), you may have a hardware problem.

It is possible that non-linearity or other problems may exist even if data is being reported to the computer. From the coordinate display, you can determine which corner is the origin and watch both X and Y increase steadily as you slide your finger to the diagonally opposite corner. Check the other two corners to make sure their coordinates look correct and the touchscreen appears linear in each axis. If the coordinates appear to be meaningless, try changing the Base I/O Port address of the controller. Another device may be contending for the same I/O address.

See the *IntelliTouch Product Manual* or the *AccuTouch Product Manual* for hardware troubleshooting information. Contact Elo Technical Support if you believe you have a hardware problem.

To check for interrupt contention with another device, turn off the computer and remove the Interrupt (IRQ) jumper from the controller (the E271-2201 should not have an interrupt jumpered). Power on and run BUSSTAT from DOS. If the controller works, you probably have interrupt contention with another device. If you are using the E281-4035 controller, it is also possible that the interrupt specified during software setup did not match your jumper placement. Select another interrupt and double-check your jumper placement.

With the E271-2201 controller, you may simply change the interrupt specified in the Setup dialog of the Control Panel—it is not necessary to change jumpers (except when in E271-141 emulation mode). See the *AccuTouch Product Manual* for more information.

## COMDUMP

COMDUMP.EXE allows you to test Elo serial controllers (E281-2310, E281{A}-4002, E271-2210) from DOS. COMDUMP directly reads the system's serial ports and displays any received data. This can determine if a problem is hardware or software related.

COMDUMP can be used to determine the controller type, data mode and baud rate of a controller. COMDUMP gives you a way to verify switch and jumper settings if the controller is installed inside a cabinet and not easily accessible. Also use COMDUMP to check that a controller/touchscreen/cable combination is working correctly, which indicates that software driver setup is the problem.

To run COMDUMP, type the following at the DOS prompt:

```
COMDUMP <port 1 or 2> <baud rate>
```

The baud rate is optional. The default is 9600.

For example:

```
COMDUMP 2 2400
```

For instructions on how to format the command line and display the version number of COMDUMP, just type:

COMDUMP [Enter]

Once in COMDUMP, the raw (untranslated) data received from the controller is displayed. Normally, data is displayed at a constant rate only when the touchscreen is touched.

The baud rate can be changed by pressing [SPACE]. This cycles through all available baud rates. Change the baud rate until *something* displays. You can press the [M] key to toggle Hex and ASCII display modes. Try different combinations of [SPACE] and [M] until the data is recognizable. If you are not familiar with your controller, Elo Technical Support can help you interpret the displays. The *IntelliTouch Product Manual*, *AccuTouch Product Manual*, and the controller technical reference manuals also detail the data formats.

If no data is transmitted on touch, or if data is transmitted when the touchscreen is not touched (either continuously or intermittently), you may have a hardware problem. See the *IntelliTouch Product Manual* or the *AccuTouch Product Manual* for hardware troubleshooting information. Contact Elo Technical Support if you believe you have a hardware problem.

## SAWDUMP

SAWDUMP.EXE displays the firmware revision level, results of power-on diagnostics, signal gain values, and can capture further signal information into a .SWV file for analysis by Elo Technical Support. SAWDUMP works only with IntelliTouch controllers with 2.2 firmware or later (E281-2310, E281A-4002, or E281-4035).

To run SAWDUMP, type the following at the DOS prompt:

SAWDUMP [Enter]

and follow the on-screen instructions.



# Appendix B

---

## Programming Options

- *Z-Axis Capabilities 27*
  - *Launching the Touchscreen Control Panel 29*
  - *Calibrating from Within Your Program 29*
- 

### Z-AXIS CAPABILITIES

Windows NT programmers can take advantage of IntelliTouch Z-axis capabilities. For a demo of Z-Axis run ZAXIS.EXE from the \ELO directory. This program is installed by using the Custom option in Setup. Run Setup again to install the programming tools.

To access the Z reporting capabilities of the driver in Win32, the device is accessed via its Win32 name "MMDev". The CreateFile function opens the MMDev device and returns a handle to it. This handle is subsequently used in the DeviceIoControl function call which is the mechanism used to obtain Z data from the driver. The I/O control code IOCTL\_MM\_ZAXIS\_SUPPORT is the parameter used to query the driver to see if the controller supports Z reporting capabilities (IntelliTouch). The I/O control code IOCTL\_MM\_ZAXIS is the function parameter passed to the DeviceIoControl function to obtain Z data.

The following example code demonstrates how to obtain Z data. For a complete example, see the included source code for ZAXIS.EXE.

```
#include "mmioctl.h"

HANDLE hndFile;                // Win32 handle of "MMDev" device
LONG   IoctlCode;              // Function to perform
ULONG  DataLength;             // Output buffer data length
DWORD  ReturnedLength;         // Actual number of bytes returned
LONG   IOBuffer;               // Input/output system buffer
BOOL   IoctlResult;           // Result of operation

//
//                               // Open Win32 touchscreen device(MMDev)
//

hndFile = CreateFile(          // Open the device "MMDev"
    "\\.\MMDev",
    GENERIC_READ,
    FILE_SHARE_READ,
    NULL,
    OPEN_EXISTING,
    0,
    NULL
);

if (hndFile == INVALID_HANDLE_VALUE) { // Was the device opened?
    MessageBox(NULL, "Unable to open touchscreen",
        "Driver Loaded?", MB_OK);
return(0);
}

// See if controller supports Z before going any further

IOBuffer = 0;
IoctlCode = IOCTL_MM_ZAXIS_SUPPORT;
IoctlResult = 0;
DataLength = 4;

IoctlResult = DeviceIoControl(
    hndFile,                // Handle to device
    IoctlCode,              // IO Control code for Read
    NULL,                   // We don't need input data
    0,                      // No input buffer required, NULL bytes
    &IOBuffer,              // Buffer from driver.
    DataLength,             // Length of buffer in bytes.
    &ReturnedLength,        // Bytes placed in DataBuffer.
    NULL                    // NULL means wait till op. completes.
);

if (!IoctlResult)          // No z
    MessageBox(NULL, "This controller does not support Z reporting
capabilities",
        "NO Z", MB_OK | MB_ICONEXCLAMATION);

//
// Call the driver to get Z values
//

IOBuffer = 0;
IoctlResult = 0;
DataLength = sizeof(LONG);
```

```

IoctlResult = DeviceIoControl(
    hndFile,                // Handle to device
    IoctlCode,             // IO Control code
    NULL,                  // We don't need input data
    0,                      // No input buffer required, 0 bytes
    &IOBuffer,             // Buffer from driver.
                            // This buffer will contain Z data
    DataLength,            // Length of buffer in bytes.
    &ReturnedLength,      // Bytes placed in DataBuffer.
    NULL                    // NULL means wait I/O completes.
);
if (!IoctlResult)         // We had a real mouse event.
    ;                     // The DeviceIoControl call will
                            // return 0 if a mouse(not touch)
                            // event occurred.

```

## LAUNCHING THE TOUCHSCREEN CONTROL PANEL

It is possible to launch the Touchscreen Control Panel directly from a Windows NT application using the following code:

```
handle = WinExec("control monmouse.cpl", SW_SHOWNORMAL);
```

## CALIBRATING FROM WITHIN YOUR PROGRAM

You may also call the calibration procedure of the Touchscreen Control Panel directly from your program with the exported function Calibrate-Screen(). It takes two arguments-your window handle and a timeout value in seconds (0 = no timeout).

The following code fragment demonstrates how to call this function. For a complete example, see the included source code, CALIB.C.

```

WORD Timeout;
HWND hWnd;                // window handle
FARPROC lpfnCalibrateScreen; // pointer to Calibrate function
HANDLE hLibrary;          // MONMOUSE.CPL library handle
// Load the Control Panel applet
hLibrary = LoadLibrary("MONMOUSE.CPL");
if (hLibrary < (HANDLE)32) {
    MessageBox(NULL, "Could not load Calibration Library.",
        "NoLoad", MB_OK);
    break;
}
// Get address of CalibrateScreen function
lpfnCalibrateScreen = GetProcAddress(hLibrary, MAKEINTRESOURCE(4));
// timeout in 5 minutes
Timeout = 300;
// Call CalibrateScreen function
(*lpfnCalibrateScreen)(hWnd, Timeout);
FreeLibrary(hLibrary);

```

