Touch Time II
Installation & Operation Guide

Model 3007

November 2017

3007-UG-001

Contents

FCC Compliance

EN 55022 Class A Warning Requirements

EN 55022 does not restrict the marketing of Class A information technology equipment, but does require it to include the following warning in the instructions for use.

> **Warning**
>
> *This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.*

FCC Required labeling for Verified Devices 47 CFR Part 15.19

Verified devices must have the following label permanently affixed in a location accessible to the user:

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

No distinction is made between Class A and Class B devices on the label.

When the device is so small or for such use that it is not practicable to place label on it, the information shall be placed in a prominent location in the instruction manual supplied to the user or, alternatively, shall be placed on the container in which the device is marketed.

Where a device is constructed in two or more sections connected by wires and marketed together, the label is only required to be affixed to the main control unit.

FCC Required labeling for Class B Personal Computers and Peripherals Devices 47 CFR Part 15.19 subject to Declaration of Conformity

Personal computers and peripherals subject to authorization under a Declaration of Conformity shall be labeled as follows:

(1) The label shall be located in a conspicuous location on the device and shall contain the unique identification described in Section 2.1074 and the following logo:
   (i) If the product is authorized based on testing of the product or system:
   (ii) If the product is authorized based on assembly using separately authorized components and the resulting product is not separately tested:

(2) When the device is so small or for such use that it is not practicable to place the statement specified under paragraph (b)(1) of this section on it, such as for a CPU board or a plug-in circuit board peripheral device, the text associated with the logo may be placed in a prominent location in the instruction manual or pamphlet supplied to the user. However, the unique identification (trade name and model number) and the logo must be displayed on the device.

(3) The label shall not be a stick-on, paper label. The label on these products shall be permanently affixed to the product and shall be readily visible to the purchaser at the time of purchase, as described in Section 2.925(d). "Permanently affixed" means that the label is etched, engraved, stamped, silk-screened, indelibly printed, or otherwise permanently marked on a permanently attached part of the equipment or on a nameplate of metal, plastic, or other material fastened to the equipment by welding, riveting, or a permanent adhesive. The label must be designed to last the expected lifetime of the equipment in the environment in which the equipment may be operated and must not be readily detachable.

FCC Required Instruction Manual Inserts CFR 47 Part 15.21 and 15.105

The user's manual must caution the user that changes or modifications not expressly approved by the manufacturer could void the user's FCC granted authority to operate the equipment. In addition the following information should be inserted:

(a) For a Class A digital device or peripheral, the instructions furnished the user shall include the following or similar statement, placed in a prominent location in the text of the manual:

***Note**: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.*

(b) For a Class B digital device or peripheral, the instructions furnished the user shall include the following or similar statement, placed in a prominent location in the text of the manual:

***Note**: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:*

- *Reorient or relocate the receiving antenna.*
- *Increase the separation between the equipment and receiver.*
- *Connect the equipment into an outlet on a circuit different from the receiver's.*

(c) The provisions of paragraphs (a) and (b) of this section do not apply to digital devices exempted from the technical standards under the provisions of § 15.103.

(d) For systems incorporating several digital devices, the statement shown in paragraph (a) or (b) of this section needs to be contained only in the instruction manual for the main control unit.

Disclaimer

No part of this publication may be reproduced, or transmitted in any form or by any means without the written permission of Control Module, Inc. The information described by this publication is specifically intended for use by customers and resellers of Control Module Data Collection products and data collection systems, and only for use with Control Module Data Collection products and data collection systems.

The information in this manual is subject to change without notice.

Important Notice

One of the most important features of CMI's TouchTime terminals is that they operate on the Windows operating system. For this reason, they are compatible with other Windows-based electronic systems that you may employ in your organization. In this way, they are capable of being integrated into your overall computer network and managed by your same Information Technology ("IT") staff. By using the Windows operating system, we have also made it possible for your IT staff to manage internet security, as it would for all of your company's Windows-based technology.

Since the TouchTime terminal is designed for compatibility and the potential for integration, the configuration of any firewall protecting the TouchTime terminal is the responsibility of your IT staff to manage and adjust based on your organization's specific requirements. Virus scanning software can be installed on TouchTime terminals as with any other Windows-based workstation. All decisions with respect to internet security, protection of computer systems and data, the installation and configuration of virus protection and firewall software, the configuration of the TouchTime terminal within your organization's network, the scheduling and execution of operating system and software updates, any backup procedures, and all matters having to do with security in general, including the development and training on security measures and remedial steps to be taken, are the sole responsibility of you the customer.

TOUCHTIME TERMINALS ARE SUBJECT TO CMI'S STANDARD WARRANTY ATTACHED HERETO AND INCORPORATED HEREIN. ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, ARE HEREBY DISCLAIMED, INCLUDING BUT NOT LIMITED TO, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR USE OR PURPOSE. IN NO EVENT SHALL CMI BE LIABLE FOR ANY LOSSES CAUSED BY COMPUTER VIRUS, RANSOMWARE, MALWARE, COMMUNICATION LINE FAILURE, OR DELAYS IN TRANSMISSION. IN

NO EVENT SHALL CMI BE RESPONSIBLE FOR ANY UNAUTHORIZED ACCESS, ANY LOST, DELETED, OR INACCESSIBLE DATA, OR ANY LOSS OR INJURY TO EARNINGS, PROFITS OR GOODWILL OR FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES EVEN IF CMI IS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. WITH RESPECT TO ANY DISCREET COMPONENT WHICH IS NOT MANUFACTURED BY CMI, THE WARRANTY OF THE MANUFACTURER THEREOF SHALL APPLY AND BE EXCLUSIVE. THE MAXIMUM LIABILITY OF CMI FOR CLAIMS ARISING FROM OR IN CONNECTION WITH THIS AGREEMENT SHALL BE THE FEES PAID TO CMI FOR THE TWELVE MONTH PERIOD PRECEDING THE CLAIM, EXCEPT FOR ANY CLAIMS FOR PERSONAL INJURY OR DEATH (FOR WHICH NO LIMIT APPLIES).

Copyright and Trademarks

Overview

The Touch Time II is a Windows 8.1-based touch screen data-collection terminal with integrated peripherals designed to provide a kiosk-style user experience, along with built-in employee auto-identification peripherals. It is ideally suited for workforce management, shop floor data-collection, or self-service kiosk-based applications. The terminal features a 10" capacitive touch screen interface with integrated peripheral options for fingerprint biometrics, barcode slot reader, magnetic slot reader, multi-class proximity/RFID cards reader, and 2D barcode scanner. When mounted to the wall with any or all of these peripheral options, it meets or exceeds ADA requirements. The Touch Time II also provides integrated WiFi, Ethernet, built-in PoE, and an external USB port, providing ease of connectivity to virtually any type of network.

The Windows 8.1 operating system provides a familiar interface for users and system administrators alike and is capable of running Java, .Net, browser-based, (such as Chrome or Internet Explorer) or any other Windows-based application. The Windows eco-system allows for the adoption or integration of a wide range of peripherals and components.

Application development and deployment is made even simpler with CMI's application loading and monitoring system, which facilitates deployment and updates of Java-based applications. CMI's terminal emulators provide a migration path forward by allowing applications developed for CMI's Genus 2 and SaveTime terminals to run on the Touch Time II, with very little development. In addition, CMI provides a lightweight API based on the Genus 2 API so that Java applications can make use of the terminal's peripherals and functions tailored towards data-collection, yet allow for UI or system interfaces based on the latest Java JDKs and frameworks.

Application servers such as CMI's System Manager terminal management platform can be leveraged to provide seamless data-collection and network management support of very large terminal deployments.

Frameworks:

- Legacy SaveTime command set emulator
- Legacy Genus 2 terminal application emulator
- Java-based application development with application launcher and auto-update features
- .NET-based applications
- Browser-based applications

Supporting tools:

- VNC Remote Management
- CMI System Manager terminal management application
- CMI Terminal Manager Application for SaveTime

Specifications

OS / Memory & Processor

- Windows 8.1
- 32 GB Internal Storage
- 1.33 GHz Intel ATOM Z3735f Quad Core

User and Communication Interface

- 10.1” WXGP 5-Point Multi-Touch HD Capacitive Display, 16:9 Aspect Ratio, Gorilla Glass
- WiFi (802.11a/b/g/n)
- Ethernet 10/100

IO Ports

- 2 Internal Reader Ports
- 1 External Reader Port
- FullSize USB 2.0 /
- Optional Access Relay

Physical

- 10.1" - 12.5 X 8.0 X 3.5 Inches w/o Wallmount
- Approximate Weight: 3.00 lbs

Sensors

- Ambient Light Sensor

Cameras and A/V

- HD Life Cam, FrontFacing
- One Microphone
- Stereo Speakers

Auto ID

- Barcode
- Magnetic
- Proximity (HID, MOTOROLA)
- 1D – 2D Scanner (MIFARE®, iCLASS®)
- Biometrics (Fingerprints)

Environmental

- Operational Temp: 5°C to 35°C (32° to 122°F)
- Storage: 20° to 65°C (4° to 158°F)
- Humidity: 0 to 80% Non-condensing
- Electrostatic Discharge: Min. 8 KV

Power

- 18W Power Supply (Output Voltage 12 Volts, 1.5A)
- 5000 aAh LI Polymer Battery
- Optional PoE 802.3at (PoE+)
- Configurable Power Cord

Installing the TouchTime II Unit

Site Preparation

- Refer to the *Environmental* section of **Specifications** on the previous page for operating requirements.
- The unit must be mounted at least four feet from the floor.
- To run TouchTime II (TT2) over Wi-Fi, use an AC power pack. **Note**: *Do Not* use a Genus G2 18-volt power pack.
- To run TT2 over Ethernet:
    - Provide one Cat5E Ethernet cable up to 100 meters in length.
    - For PoE, the Ethernet switch ports that are connected to TT2 terminals must be configured for 30W static power, OR you can use the CMI PoE Injector. The injector requires an additional Cat5E cable and can be used with standard, PoE or PoE+ switches.
    - See *Connecting Ethernet, Power and Other Devices* on page 12 for more information.

Note: PoE auto-negotiation will not work properly with most switches as they usually require an additional protocol to negotiate power above 15.4W, which is not provided in the TT2.

Box Contents

- TouchTime II unit with one of the following attached:
    - Magnetic Track II reader
    - Barcode reader
    - Proximity reader
    - Biometric reader
    - 1D to 2D Scanner
    - Biometric and Proximity readers
    - Biometric and Barcode readers
    - Biometric and Magnetic Track II readers
- Universal Power Supply (1)
- Screen Protector (Optional) (2)
- PoE Injector (Optional) (3)

1

2

3

Wall Mounting

Figure 1

1. Use the dimensions in **Figure 4** to mark and drill pilot holes if necessary, as well as the center access hole, for any cabling entering the enclosure from the rear. You can also use the opening in the bottom of the unit under the foam (**Figure 3**) to snake cabling up the wall. To accommodate ADA standards, the bottom of the mounting panel must be 45.5 inches from the floor (**Figure 4A**).
2. Remove the key taped to the back of the unit, insert the key on the right side and unlock it. (**Figure 1**)
3. Pull down and lift off the mounting panel from the back of the unit.
4. Orient the unit with the key-lock mechanism on the left so the retaining hooks (**Figure 2**) on which the TouchTime II mounts are facing upwards.
5. Pass cabling entering through the rear of the unit through the pre-cut access hole in the center of the base. See *Connecting Ethernet, Power and Other Devices* on the following page for specific connection instructions.
6. Ensuring the base is level, use four screws to mount the base to either a junction box (four inside mounting holes) or directly to the wall (four outside mounting holes). Screws supplied by Installer based on wall type.
7. The screw heads must lie below the foam gasket of the mounting base so they do not interfere with the TouchTime II sliding onto the retaining hooks.
8. With the key-lock mechanism in the unlocked position, slide the TouchTime II (**Figure 2**) onto the base's retaining hooks, ensuring it is seated securely.
9. Turn the key-lock mechanism to the locked position.
10. Apply the optional screen protector to the display glass.

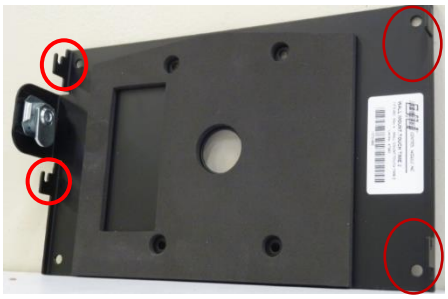


Figure 2

Figure 3

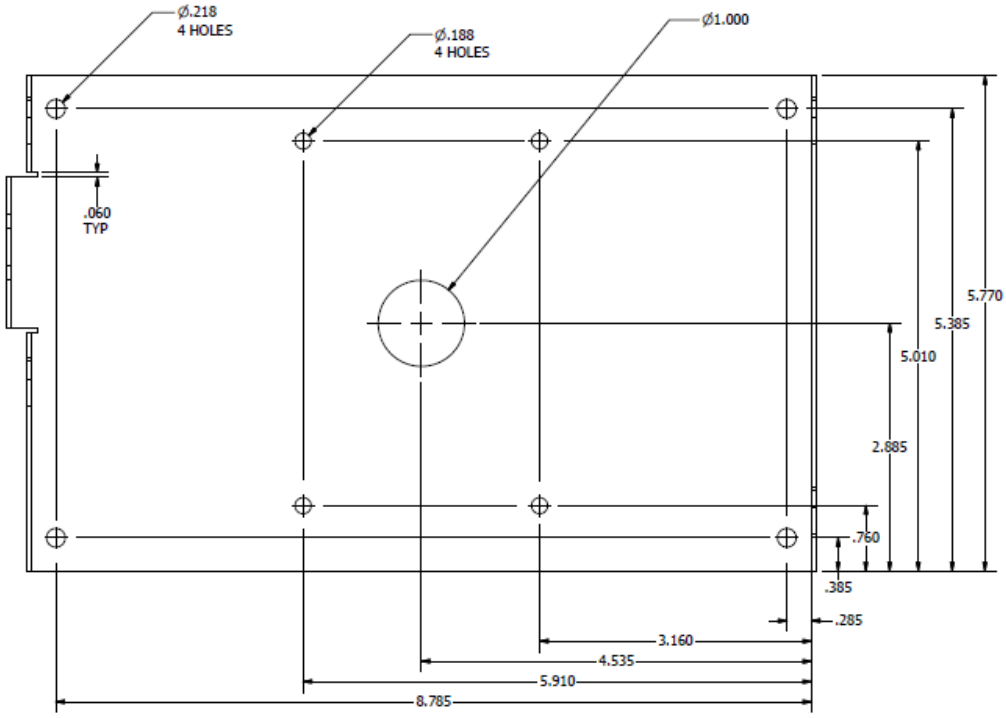


Figure 4

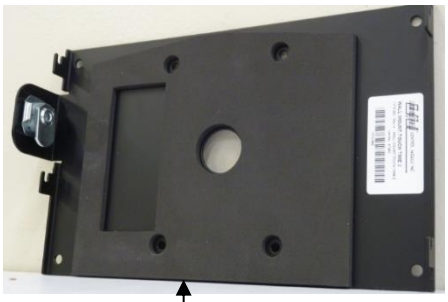


Figure 4A

Connecting Ethernet, Power and Other Devices

The TT2 unit provides three ports (**Figure 5**): USB, Ethernet PoE+ and +12VDC. You can connect external devices to the USB port as you would on any PC tablet. There are three possible sources of power:

- PoE+ Switch
- PoE+ Adapter
- Power Pack

Figure 5

PoE+ Considerations

- The TT2 requires the use of a PoE+ switch (IEEE standard 802.3at)
- Switch ports **MUST** be configured for static power and set for maximum watts; depending on the switch, the maximum output wattage can vary between 30.0W and 34.2W
- Must use Cat5E cable or better
- Maximum network cable length = 100m (328 feet)

Notes:

- The TT2 provides the ability for the switch to auto-detect the TT2 as a standard PoE 15.4W capable device, IEEE standard 802.3af
- Operating the TT2 at 15.4W will result in the switch resetting or shutting down the port since the power required is greater than 15.4 W when the battery is charging
- The TT2 does not implement the power-negotiation protocols necessary to identify itself to switch as a PoE+ 34.2W enabled device, standard IEEE 802.3af. The PoE+ standard increases the maximum power that can be drawn by a powered device from 15.4 W per port to a minimum of 30W per port.
- Switch ports set as static pre-allocates or reserves power to the port (even when no powered device is connected) and guarantees power will be available for the port at all times
- Any PoE or non-PoE powered device that draws less power can use a static port without damaging the device
- Standard PoE, IEEE standard 802.3af, provides a maximum of 15.4W which is insufficient for the TT2
- Powering the TT2 with a standard PoE switch will affect communication and cause the switch port to bounce and possibly send the port into an err-disable state
- Powering the TT2 with a standard PoE switch will eventually deplete the battery causing a critically low battery condition

PoE+ Injector (2070-007)

- Output: 56VDC, 535mA, 30W
- Must use Cat5E cable or better
- Maximum network cable length = 100m (328 feet)

Notes:

- Can be used with non-PoE, PoE or PoE+ switches
- Requires an additional Cat5E patch cable

Power Pack (2061-019)

- Output voltage: 12 volts, 1.5A
- Maximum network cable length = 100m (328 feet)

Note:

- If using the 2061-019 power pack, the TT2 cannot be plugged into a switch port supplying standard PoE power of 15.4W, or PoE+ power
- If the TT2 has the PoE option and is plugged into a standard PoE or PoE+ switch, the TT2 port must be set to *Never*. This will disable power to the TT2 switch port, making the port a data-only port
- Powering the TT2 with the power pack and a standard PoE switch will affect communication and cause the switch port to bounce, and possibly send the port into an err-disable state
- Powering the TT2 with a standard PoE switch will eventually deplete the battery causing a critically low battery condition

PoE Injector

Figure 6

If your Switch does not include PoE+ and you would like to obtain power via the Switch, install the optional PoE+ Injector (**Figure 6**). Connect the Injector to a power source. The **Status** LED turns green. Connect a Cat5E Ethernet cable from the Switch to the PoE Injector's **PoE/IN** port, and connect a Cat5E Ethernet cable from the PoE Injector's **LAN/OUT** port to the TouchTime II's **Ethernet** port (**Figure 5**). The TouchTime II's **Power** LED turns red, indicating power is coming into the TouchTime II.

Note: If you use the onscreen power button to turn off the system, it is only off temporarily. The system then reboots. To power down, unplug the terminal from its power source. The LEDs on the reader, if present, will turn off.

Powering On

When the terminal is in sleep mode, you can restore power by briefly pressing the silver button on the right side of the terminal (**Figure 7**) until it clicks. While facing the terminal, reach around to the right side until you feel the button, then click it. If the display does not return, hold the button for 15 seconds.

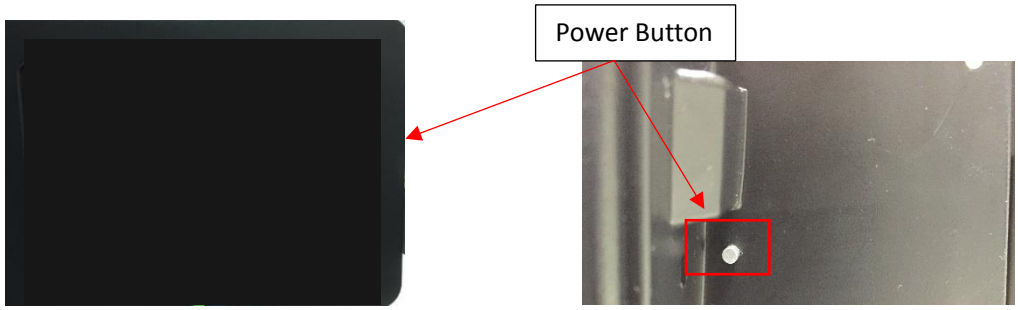


Figure 7

Resetting the Terminal

Use the Windows **Reboot** option. To permanently turn power off, unplug the terminal from its power source and apply pressure to the switch on the right side of the terminal for approximately 15 seconds until the LEDs on the Reader turn off (**Figure 7**). To turn the power back on, plug the terminal back into its power source.

Resetting the Terminal (Older Models)

If you use the onscreen power button to turn off the system, it is only off temporarily. The system then reboots. To permanently turn power off, unplug the terminal from its power source, remove the terminal from the wall if mounted, insert a paperclip into the *Reset* switch on the back, left side of the terminal (**Figures 8** and **9**) and apply pressure to the switch for approximately 15 seconds until the terminal powers down. To turn the power back on, plug the terminal back into its power source.

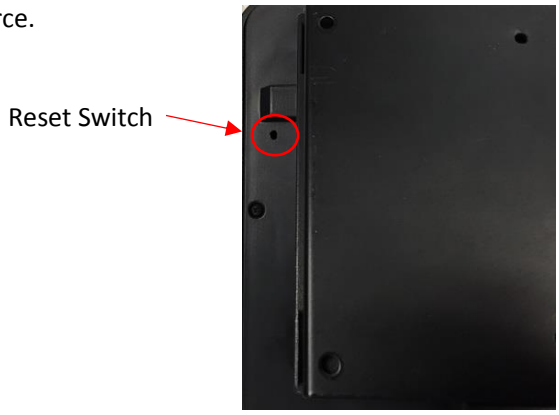


Figure 8

Figure 9

Configuring Windows Options

Depending on the application you are using, you will access the Microsoft configuration screens one of the following three ways:

- Remotely via a VNC remote desktop tool
- Built-in via the Genus 2 Emulator for Touch Time II
- Via your own application and your own interface method; proceed to ***Configuring the Network Adapter*** on page 16.

Remote Management with VNC

The TouchTime II can be remotely controlled using a remote desktop tool such as TightVNC. Use the following passwords:

- Standard Password: cmi$tt2$
- View Only Password: cmi^tt2^
- Control Password: cmi@tt2@

To access the Microsoft Control Panel:

1. Enter the following in the Login screen: your IP address and the `cmi$tt2$` password.
2. Bring up the Task Manager by clicking **CTL-ALT-DEL**.
3. Select **Task Manager -> Control Panel**.

You are now ready to configure Window options. Proceed to ***Configuring the Network Adapter*** on page 16.

Using the Local Windows Pro 8.1 Account

When initializing the TouchTime II, you will see the main Windows 8.1 desktop screen. This is because TouchTime II comes preprogrammed with a Kiosk user. This user account provides all of the privileges that are required to run the CMI Emulator Service/Emulator software. If, for some reason, you delete this user, you can restore it. From this desktop screen, you can now use this unit as you would any PC touchscreen tablet.

Swipe your finger to the left to reveal the menu bar. From here you can double-tap **Settings** to access the usual PC Control Panel options or double-tap the **Start** icon to start up the Kiosk (**Figure 1**). **Figure 2** displays. To return to the Desktop, double-tap Desktop. To put the unit to sleep, double-tap the Power icon.

Figure 2

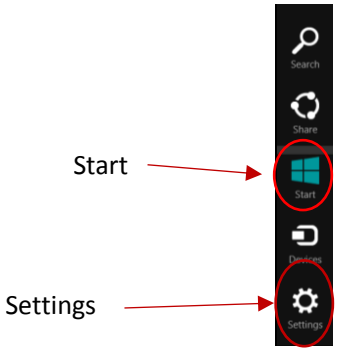


Figure 1

Using the Virtual Keyboard

A virtual keyboard is used for entering data.

1. To display the keyboard, from the Desktop, tap the Keyboard icon in the bottom right of the system tray (**Figure 3**).
2. To close the keyboard, tap the **X** in the keyboard's top-right corner (**Figure 4**). To hide the keyboard, double-tap the keyboard icon in the bottom left corner (**Figure 4**).

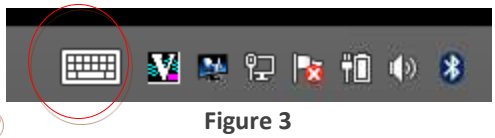


Figure 3

Figure 4

Configuring the Network Adapter

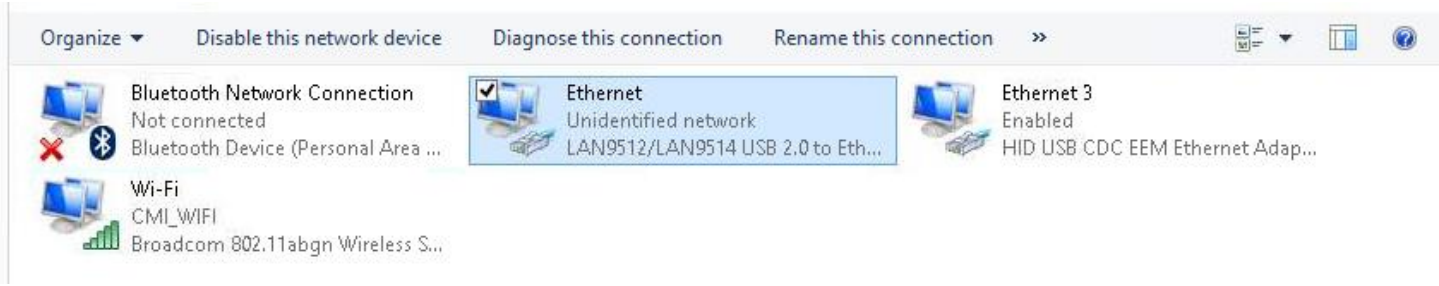
Once you have connected the unit to the switch via the Cat5E Ethernet cable, it is automatically added to your network. To proceed, you'll need the following network parameter information for your TouchTime II:

- DHCP
- IP Address
- Gateway
- Subnet Mask
- DNS Server

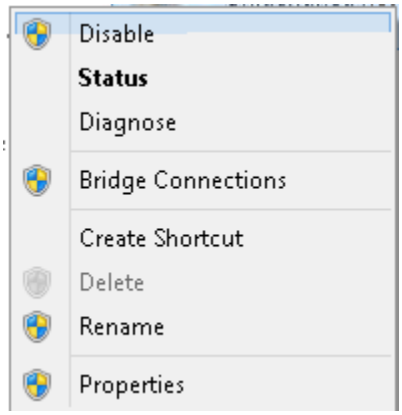
Ethernet Connections

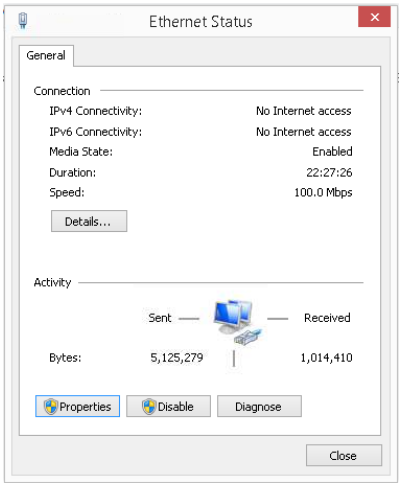
1. From the Control Panel select **Network and Sharing Center|Change Adapter Settings|Ethernet**.

   Note: DO NOT select Ethernet 3.

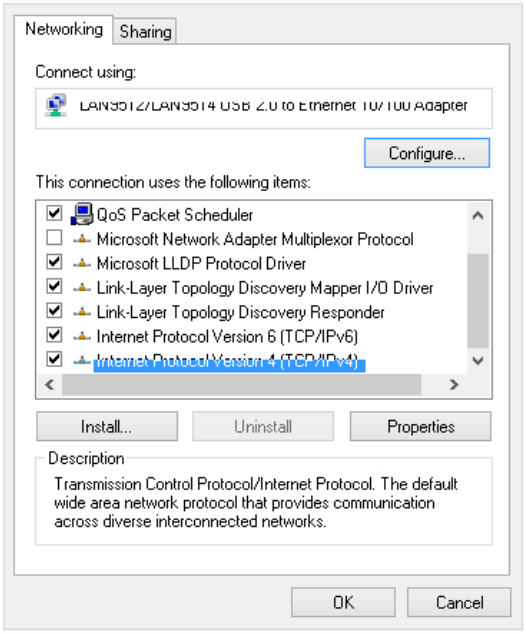


2. Double-tap the screen or right-click the mouse and select **Properties**.

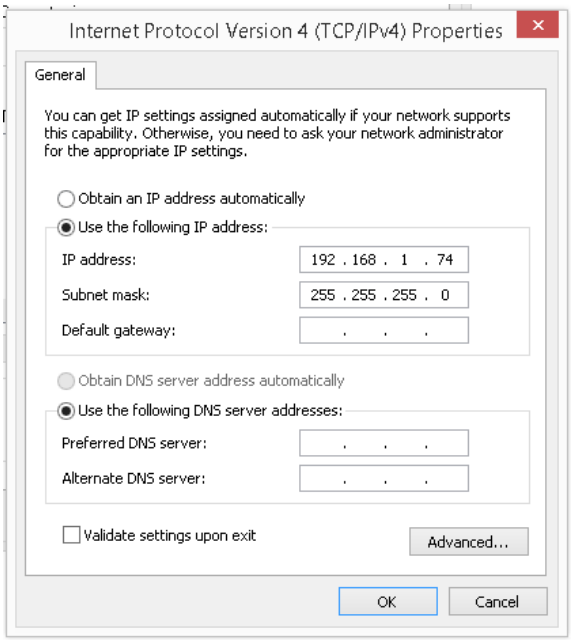




3. Highlight your TCPIP parameter and select **Properties**.

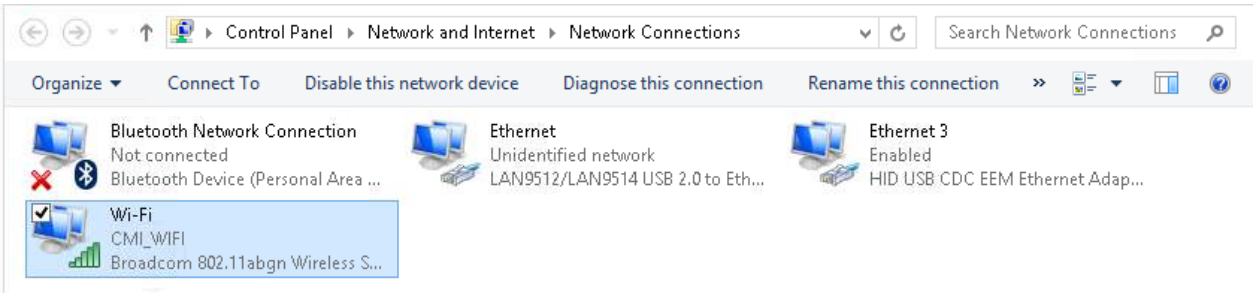


4. Fill in your network parameters and select **OK**.

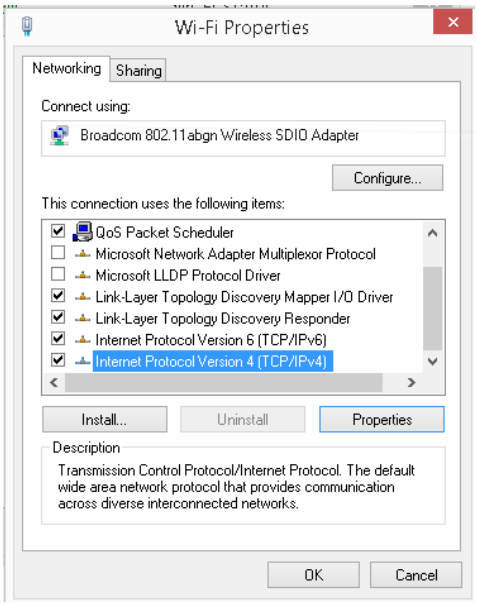


Wireless Connection

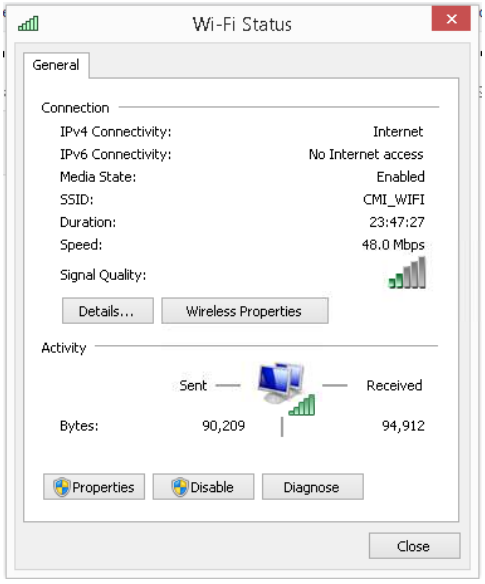
1. From the Control Panel select **Network and Sharing Center|Change Adapter Settings|Wi-Fi**.



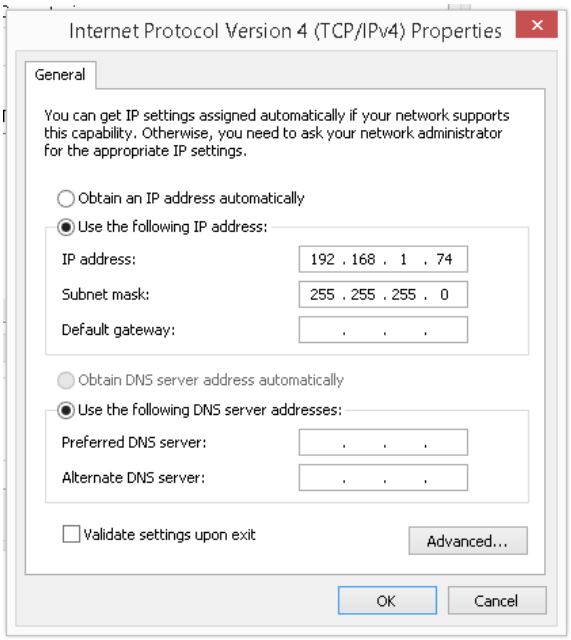
2. Double-tap the screen or right-click the mouse and select **Properties**.



3. Highlight your TCPIP parameter and select **Properties**.



4. Fill in your network parameters and select **OK**.



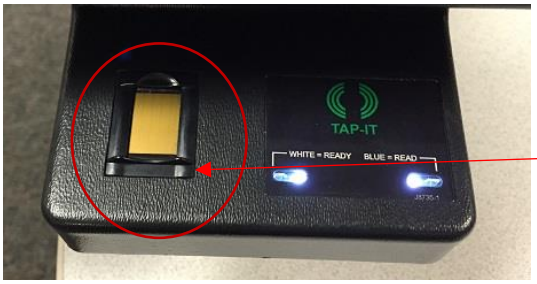
Using Readers and Peripherals for Identification

The TouchTime II provides multiple user auto-identification readers and peripherals for the following scanning options:

- Fingerprint (Biometric)
- Magnetic Swipe
- 1D-2D Barcode Reader
- Proximity Reader

Using the Biometric Scan

Using fingerprints for identification involves the proper scanning of a user's fingerprint from several angels to ensure a well-defined print. The Biometric scanner is located on the bottom left of the TouchTime II's pedestal.



Biometric Definitions

Enrollment is the operation of scanning a fingerprint, determining the quality of the fingerprint scan, and storing a good template with associated data within the memory of the scanner.

Verification is the operation of requesting the user to place their finger on the scanner, scanning the finger, comparing the current scan against stored fingerprint templates for that user and then notification of a successful validation or a failure.

Fingerprint Template is the term used to describe the data stored on the scanner that mathematically represents the ridge pattern of an enrolled fingerprint. This data is not the raw image of the fingerprint, but the result of processing a raw image through our unique algorithmic process, preparing the data for later comparisons, and compressing the data for maximum storage. An image of the uncompressed template data does resemble the raw image, but whereas a raw image is 90K bytes, the compressed template is only 350 bytes (1 to 1) verification and 2000 bytes in a (1 to N) Identification.

Fingerprint Core is the term used to describe distinguishing print characteristics usually found in the area of the print where the topography shows the tightest curvature. Although the entire fingerprint has significant data, the "core" is the most data-intensive area and therefore very important.

Scanning an Image

When the scanner properly reads a fingerprint, it looks for image *quality* and fingerprint *content*. When a raw image is collected from the sensor, the scanner searches for the **fingerprint core.**

Quality scores are based on how well the ridge pattern is defined within the image. For best image *quality*, be sure that the sensor window is clear of dirt, residue, or other material that can block the scanner view of the fingerprint.

Once the image is scanned, the scanner then creates and stores the resulting fingerprint template.

Storing User Templates on the Biometric Scanner

Verification

The BioScan II recognizes users by matching current images to stored templates of previously enrolled fingerprints. During *VERIFICATION,* a user enters their ID # and places their finger on the fingerprint capture device. The scanner will then scan the current fingerprint and compare it against the enrolled template for that specific ID. The initial finger scan takes ~0.5 seconds

and the each comparison takes ~0.5 seconds. So if the template results in a successful verification, the total time is ~1.0 seconds.

Identification

The scanner recognizes users by matching current images to stored templates of previously enrolled fingerprints. During *IDENTIFICATION,* a user places their finger on the fingerprint capture device. The scanner will then scan the current fingerprint and compare it against the enrolled templates for that specific ID. The initial finger scan takes ~20 seconds and the each comparison takes ~1.0 seconds. So if the template results in a successful verification, the total time is ~1.0 to 2.0 seconds.

Proper Finger Placement

The basics for successful operation of the scanner are simple but important. System performance improves dramatically with ***consistent finger placement.*** It is important to make sure that the position of the finger allows the scanner to record the unique features of the print. Here are the steps to follow for trouble-free fingerprint recognition.

- Use the Ridge-Lock and the Finger Guide to create "simple user instruction" and finger position. With the fingertip raised, position the finger so that the Ridge- comfortably within the first indentation of the finger. Next, lower the finger onto apply moderate pressure. This figure illustrates proper finger placement and the of the scanned fingerprint.

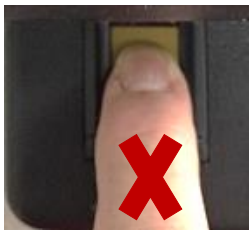
Common Mistakes

Correct finger placement is a significant component for reliable fingerprint imaging. The following figures illustrate some common mistakes to avoid.

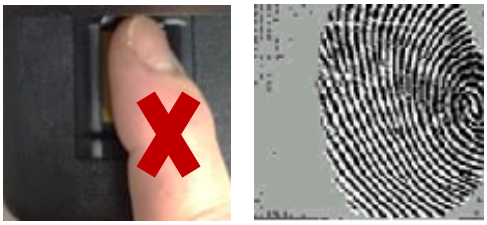
- Sliding the fingertip into place instead of lowering it onto the sensor will cause distortion of the fingerprint and will degrade image quality. Keep the fingertip raised while locating the Ridge-Lock, then lower the fingertip.

- Rotating the finger into position will also cause distortion of the fingerprint, subsequently making verification less reliable.

- Placing your finger as if punching a button will not provide adequate information and will degrade system performance. Proper sensor height and angle along with consistent use of the Ridge-Lock deters this behavior.



- Positioning the finger to one side and leaving a portion of the sensor exposed will degrade image quality. This figure demonstrates how poor finger placement degrades the image of the fingerprint. Notice how the core is well off-center and the sensor is not fully covered.



- Placing the finger at an angle to the finger guide, as shown below, is another common mistake. Rotation of the fingertip will not provide a reliable image of the fingerprint.

- Not using the Ridge-Lock will prevent the scanner from imaging the main features of interest. This figure illustrates the user neglecting the Ridge-Lock and resulting fingerprint image. Notice how the core is well below center and the sensor is not fully covered.

Image quality

Dry skin is another factor that can contribute to an unreliable image of a fingerprint. A normal amount of moisture on the skin makes the ridges and valleys of the fingerprint stand out to the sensor. Too little moisture makes the image “noisy” and will “cause BioScan II to reject the image during processing. Lightly moisturizing the finger will enhance the contrast of the print and provide more reliable verification. The increased sensitivity of the silicon sensor is dramatically reducing problems in this area.

Image consistency

Once a user’s fingerprint template has been enrolled, the best performance in the candidate matching process depends on consistency. Obviously, the user must use the same finger for ID verification as was used to form the original template. It is also important to position the finger correctly for each verification, as was done when the template was enrolled, so the scanner “sees” approximately the same information each time. **Consistent use of the Ridge-Lock system and Finger Guide ensures consistent finger placement.**

Reasons for Low Scores

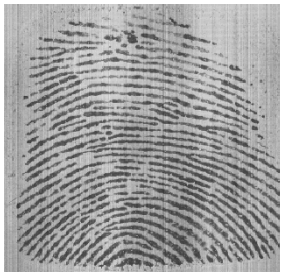
Some reasons for poor sampling results are listed below:

Possible Reason	Correction
Finger movement while sampling	Instruct the user to remain still while scanner is sampling.
Finger not positioned properly	With the fingertip raised, position the finger so that the Ridge-Lock rests comfortably within the first indentation of the finger. Next, lower the finger onto the sensor and apply very moderate pressure.
User might be pressing too hard	Too much pressure on the sensor will blur the fingerprint ridges. Allow the user to apply gentle pressure while sampling.
User might not be pressing hard enough	You must apply gentle pressure when enrolling. The fingerprint should lay flat upon the sensor surface.
Finger too moist or wet	If the user washed their hands, but failed to completely dry the finger that is sampled, excessive moisture may cause the sample to be more difficult to obtain. Dry wet or moist fingers before sampling.
Finger too dry	Depending upon the geographical area, the season, and the skin type of the user, their fingerprint might be excessively rough or dry. Excessively dry skin may affect the sample quality. Try applying skin moisturizer a few minutes before enrolling to improve image quality.

Examples of Good and Bad Fingerprint Images

This is an example of a **GOOD** print. Notice that the core is well centered, the ridges are well defined and the sensor is covered properly.

This is an image of the Pinky (little finger). The Pinky is a BAD choice to use as a fingerprint since the print is small compared to other fingers. As you can see, very little of the sensor area is covered.



This is an image of the Thumb. The Thumb is also a BAD choice for fingerprint enrollment. Although it presents a very large data area, you can see that the core is very low or even non-existent. Do not use a Thumb for enrollment or verification

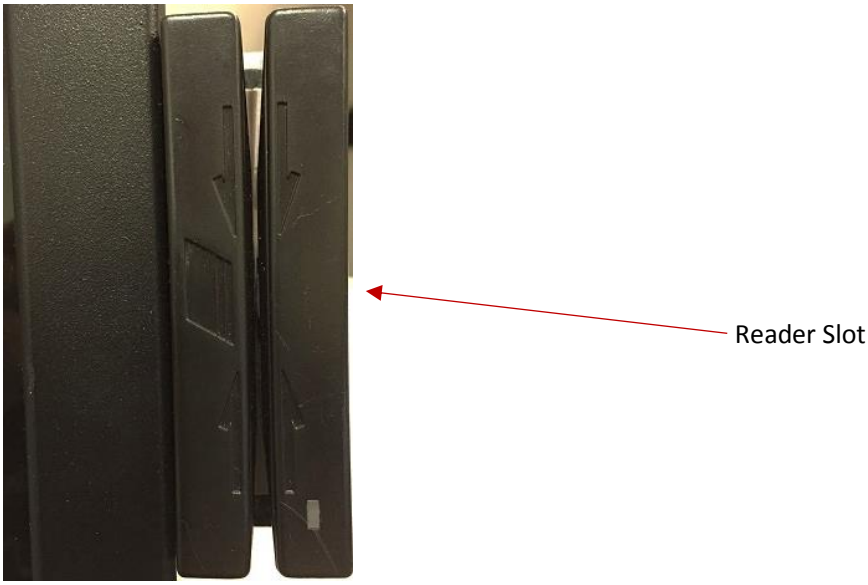
This is an example of a user not applying enough pressure. Although the core is centered, you can see that the image coverage is very poor.

This is an example of a **DRY** fingerprint. Notice how the ridge pattern is very light and not well distinguishable.

This is an example of a WET fingerprint. Notice how the ridge pattern blurs into surrounding ridges and the causes problems in the imaging. In this case, dry the finger of the user and retry.

Using The Magnetic Swipe Reader

The magnetic swipe is located on the right side of the TouchTime II's display. It supports Track 1 and Track 2.



A card can be swiped through the Reader slot when the LED is green. When using a USB Swipe Reader (with a single head) in either HID or Keyboard mode, the magnetic stripe must face toward the front (the side with the LED) and may be swiped in either direction. When using a USB SureSwipe reader (with dual heads) in either HID or Keyboard mode, the magnetic stripe can face toward the front or the back, and may be swiped in either direction. If there is data encoded on the card, the reader will attempt to decode the data and then send the results to the host via a USB input report. After the results are sent to the host, the device will be ready to read the next card.

Using The Barcode Swipe Reader

The IR barcode slot reader, mounted on the right side of the TouchTime II display, is capable of decoding a wide range of barcode symbologies. The barcode slot reader is registered with the system as a keyboard wedge interface device so data read by the reader is presented to the system as if someone entered it using a keyboard. This means testing the reader can be achieved by simply opening a program such as notepad.exe and then swiping a barcode card through slot reader.

Supported barcode formats:

Barcode Format options include:
Code 39 Settings
Code 128 Settings
Codabar Settings
MSI/Plessey Settings
Telepen Settings
Interleaved 2 of 5 Settings
Industrial 2 of 5 Settings
FEBRABAN Conversion

To use the reader:

1. Slide the card, in either direction, through the reader slot, with the bar code facing the optical head (LED side) or the magnetic stripe facing the magnetic head (opposite side).
2. Once the entire bar code or magnetic stripe has been read, the LED indicator will light up as green to signal a "good read." If a good read is not obtained, the LED indicator will light up as red.
3. A beep will also sound to indicate a good read on the bar code or each magnetic track, as appropriate. If all three tracks have been read successfully, the reader will beep three times.
4. The decoded data will be transmitted to the host application.

Using The 1-D/2-D Barcode Scanner

This reader replaces the Biometric/Proximity reader in the pedestal of the TouchTime II. It is a fixed-position 2D imager scanner that enables high speed scanning of standard linear (1D) and 2D symbologies.

To use the scanner, read the barcode from a distance of 3"/2" (80mm/45mm) from the front of the device using the green aiming light as a guide.

Using The Multi-class RFID/Proximity Reader

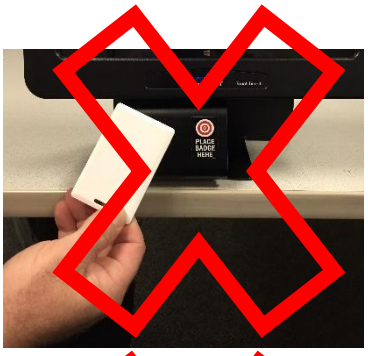
The proximity reader, located on the bottom of the TouchTime II's pedestal, reads 125KHz or 13.56MHz RFID cards. These cards are usually made of PVC, and contain an antenna coil and integrated electronic chip. Clean with a soft, non-scratching cloth and water. A Web browser interface allows you to enable/disable the reader configuration features (http://192.168.63.99).

Using the Reader

To use the reader, place the card over the bullseye area of the reader and wait for the white LED to flash blue on the right side of the reader. A tone acknowledgement sounds after the input has been accepted.

You can only place the badge vertically or horizontally over the bullseye.






Applications For Data Collection

Applications can be developed for the TouchTime II in a number of ways. To simplify the development process, CMI provides a number of applications and SDKs so that developers can choose the technology or method that best suites them.

For legacy terminal integration, where SaveTime or Genus SaveTime Emulation is used, the ***Genus 2 Emulator for Touch Time I***I can be used with a ***Touch Time II SaveTime Emulator*** application. This will allow the TouchTime II to co-exist or be used alongside terminals within a SaveTime environment with minor modifications to the SaveTime download script file.

For legacy terminal integration alongside Genus 2 terminals running Java-based applications, users can run just the ***Genus 2 Emulator*** on the TouchTime II. This will allow the terminal to run Genus 2 terminal applications.

For applications that are being developed solely for the TouchTime II terminal using Java, developers can chose to build their applications using the ***TouchTime II Light API***. This API provides a Java-based interface to the peripherals on the terminal and has built-in hooks to allow interaction with the terminal's *Emulator Service*. The same *Emulator Service* is responsible for enabling application loading/reloading and kiosk mode settings to facilitate rapid development.

Should developers wish to interface with the terminal using other programming languages, such as Microsoft's .NET platform, other APIs provided by CMI can be used, such as DLL's providing biometric reader interfaces, etc.

For more information regarding SDKs or technical services, contact your CMI sales representative at 1-800-722-6654.

Troubleshooting

The following table contains possible errors you might encounter. Instructions for the resolutions requiring actions follow the table.

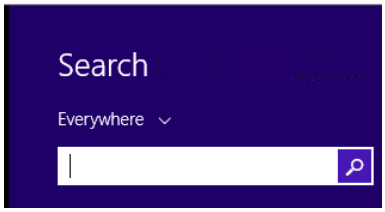
Terminal Issue	Possible Causes	Resolution
Terminal does not power on.	No power to the terminal.	If a power adapter is used, verify power is connected to the adapter and the power cable from the adapter is connected to the unit's **+12VDC** port. If a PoE Injector is used, verify power is connected to it, and a CAT5E Ethernet cable is connected from the Injector's **LAN/OUT** port to the unit's **Ethernet PoE+** port.
Password Expires	The terminal ships with a 90-day window not requiring passwords to be changed. After 90 days, the password expires and you must supply a new one.	Set password to never expire.

Setting Password to Never Expire

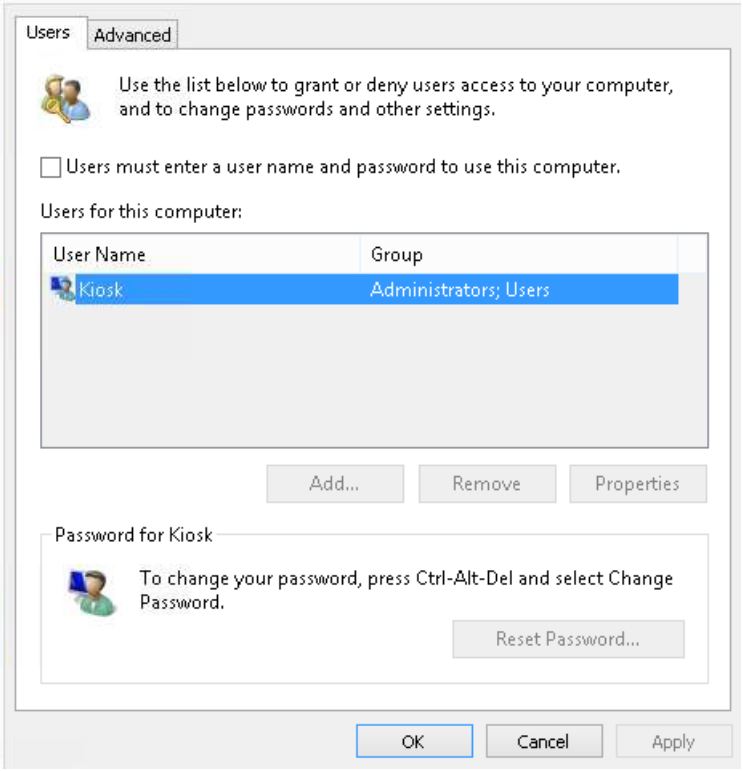
1. Open the cmd window as an administrator
2. Enter `net accounts /maxpwage:unlimited`
3. You should receive: `The command completed successfully`
4. Enter `net user kiosk`
5. Verify `Password expires` should be set to **Never**.
6. In order to verify the command sticks, reboot the terminal
7. Open a cmd window as an administrator
8. Type: `net user kiosk`
9. Verify `Password expires` is set to **Never.**

Restoring the Kiosk User

1. Side-swipe the touchscreen to the left and select **Search**.



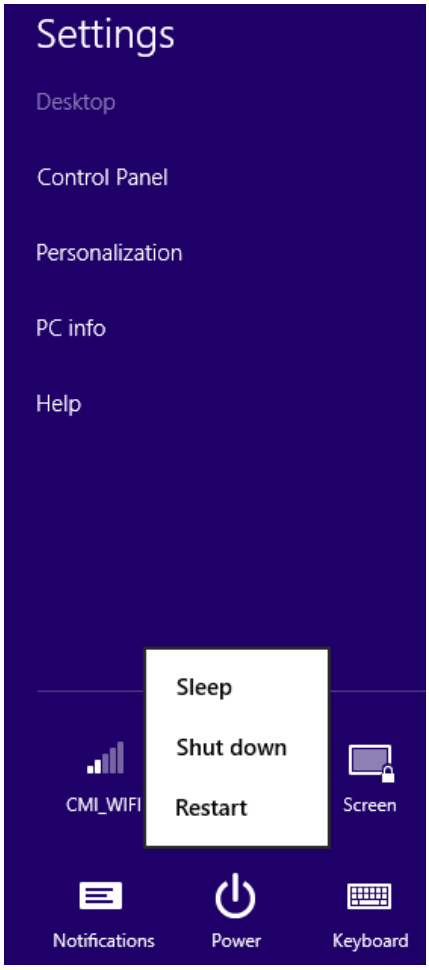
2. Enter **netplwiz** on the virtual keyboard and click **Enter**.



3. Uncheck the checkbox and highlight the **Kiosk** user name. Select **OK**.

   Note: Be sure to set up the user account and password to never expire.

4. Swipe the touchscreen to the left and select **Settings|Power|Restart**.



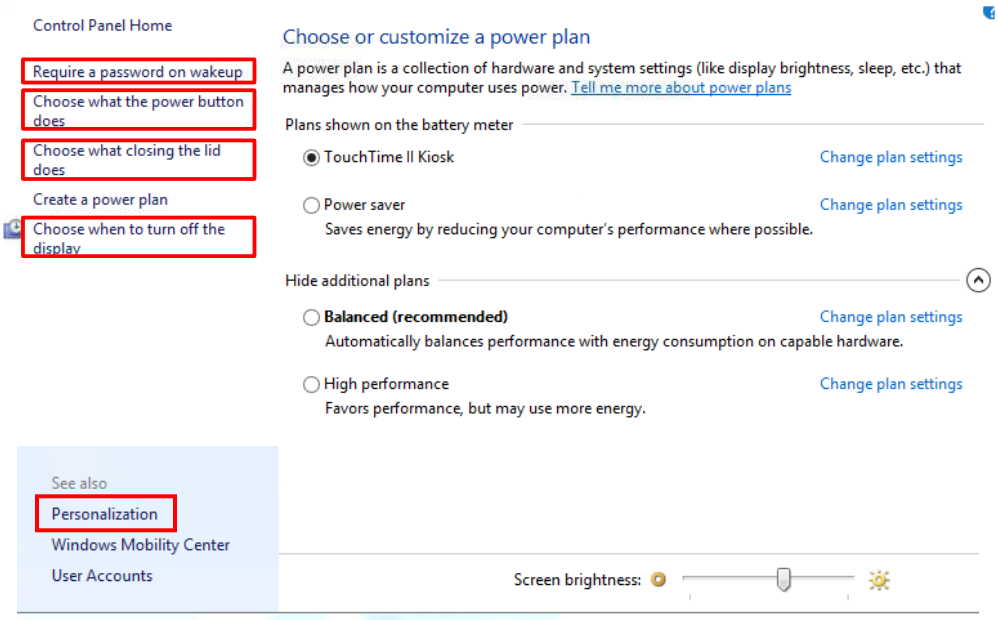
Appendix A: Factory Settings

NOTE: These settings are set at CMI. This Appendix is only a reference and should not be changed.

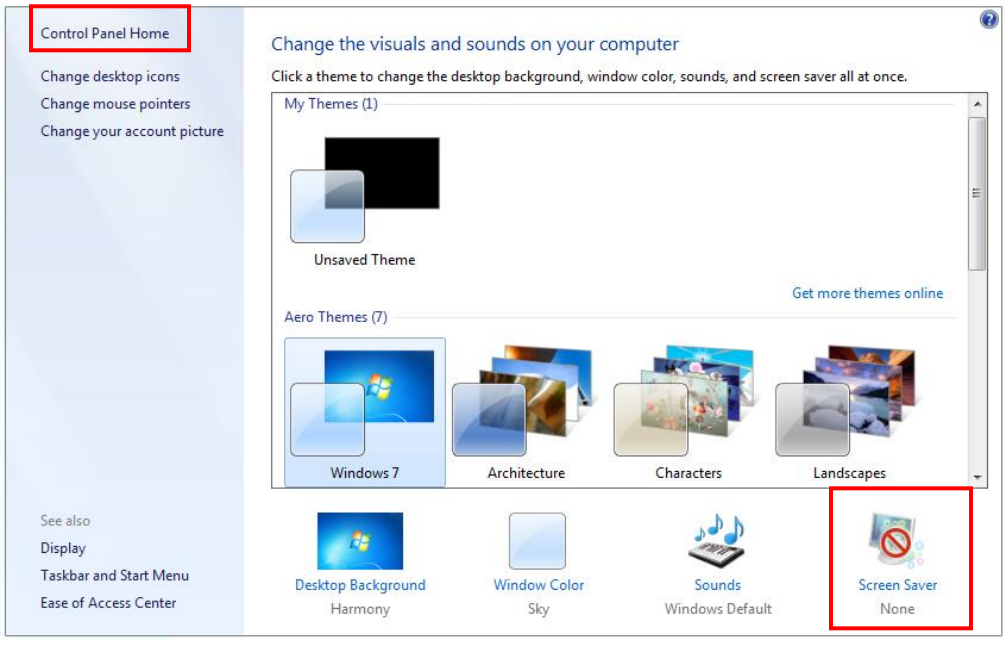
Power Plan Settings

You can manage all of your power plan settings by using **Power Options** in Control Panel. You can further optimize the computer's power consumption and system performance by changing advanced power settings. For optimum performance, we require all sleep functions to be disabled.

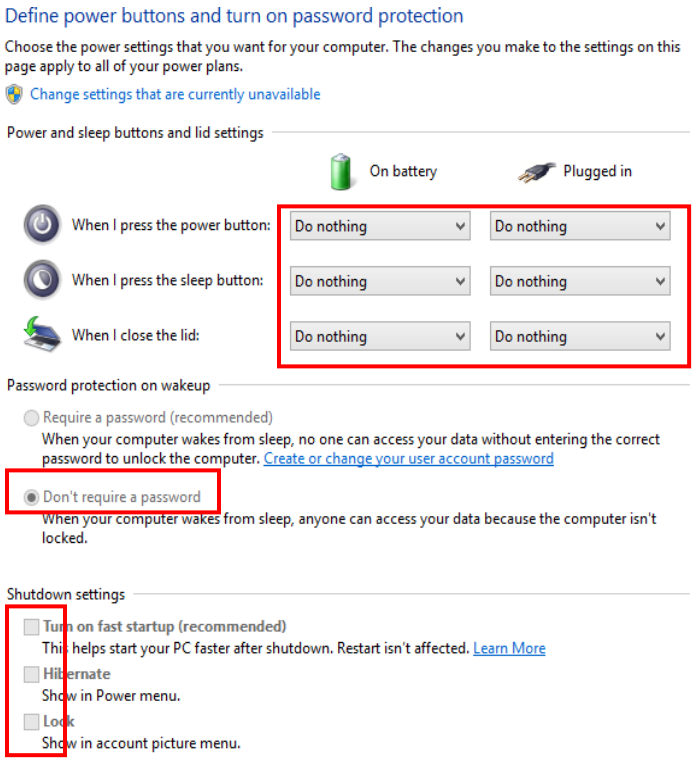
1. Swipe the touchscreen from right to left and select **Control Panel|All Control Panel Items|Power Options**.



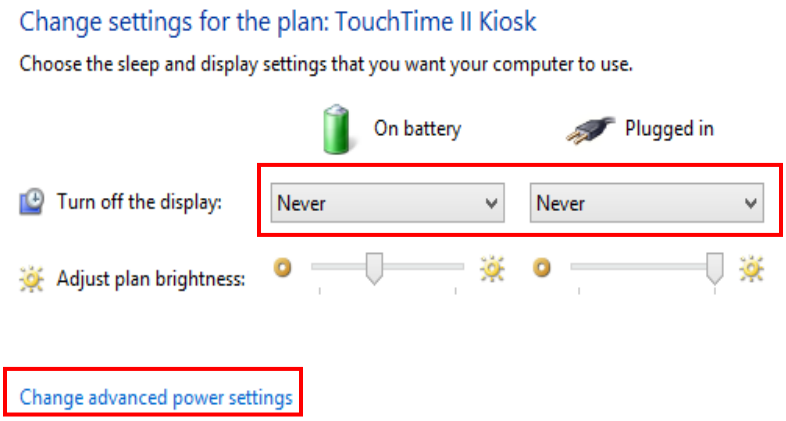
2. Click **Personalization** and select **Screen Saver None**.



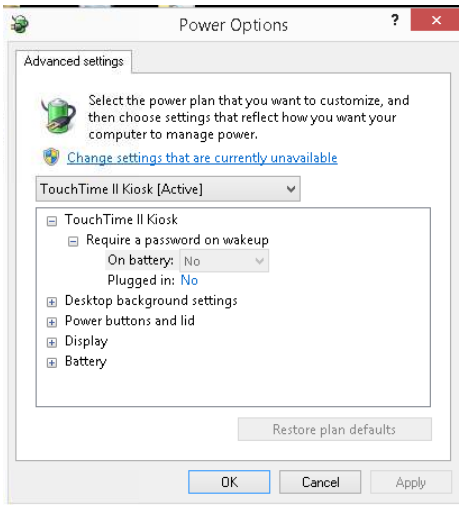
3. Click **Control Panel Home -> Power Options** to return to the previous *Power Options* screen.
4. Click **Require a password on wakeup.**



5. Make sure all power, sleep button and lid settings actions are set to **Do nothing**.
6. Select **Don't require a password**.
7. Make sure no items are checked under **Shutdown settings**.
8. Click **Save Changes** to return to the main *Power Plan* screen.
9. Leave the default TouchTime II Kiosk. To change these settings, select **Change plan settings** for this default.
10. Make sure the display settings are set to **Never**.

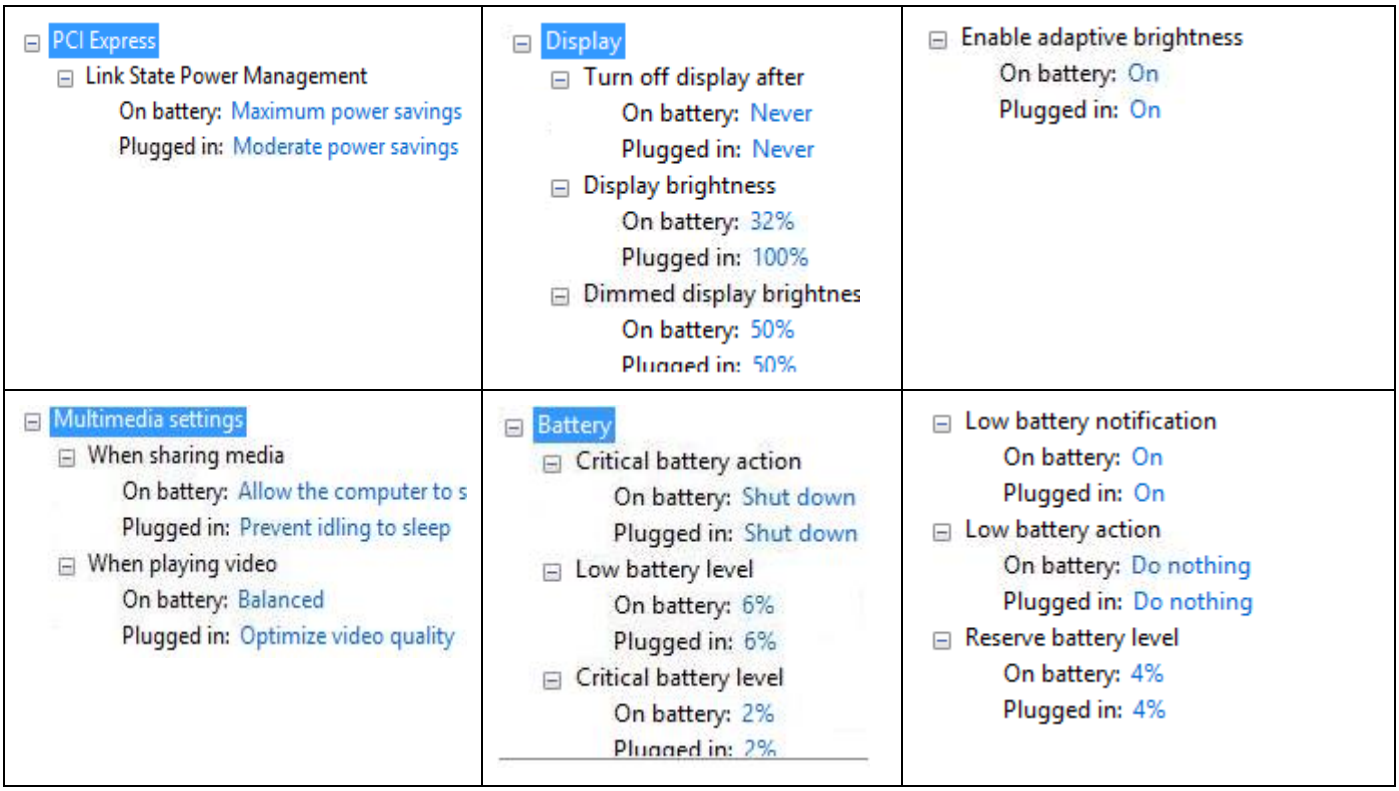


11. Select **Change advanced power settings**.

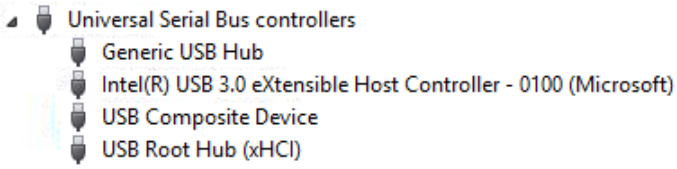


12. Click the **+** sign next to each item and make sure they match the following settings:

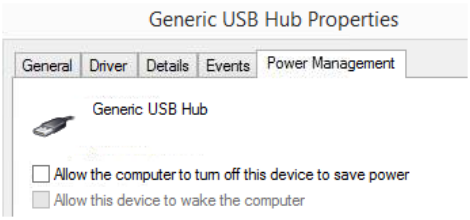
TouchTime II Kiosk Require a password on wakeup On battery: No Plugged in: No	Hard disk Turn off hard disk after On battery: 1 Minute Plugged in: 1 Minute	Internet Explorer JavaScript Timer Frequency On battery: Maximum Power Savings Plugged in: Maximum Performance
Desktop background settings Slide show On battery: Paused Plugged in: Available	Wireless Adapter Settings Power Saving Mode On battery: Medium Plugged in: Maximun	Sleep Hibernate after On battery: Never Plugged in: Never
Sleep Allow wake timers On battery: Enable Plugged in: Enable	USB settings USB selective suspend set On battery: Enabled Plugged in: Enabled	Intel(R) Dynamic Platform & Thermal Framewo Power Limit On battery: 3 Level Plugged in: 3 Level Acoustics Limit On battery: 3 Level Plugged in: 3 Level Low Power Mode Setting On battery: Disable Plugged in: Disable
Power buttons and lid Lid close action On battery: Do nothing Plugged in: Do nothing Power button action On battery: Do nothing Plugged in: Do nothing Sleep button action On battery: Do nothing Plugged in: Do nothing	Processor power managemer Minimum processor state On battery: 5% Plugged in: 5% System cooling policy On battery: Passive Plugged in: Active Maximum processor state On battery: 100% Plugged in: 100%	Intel(R) Graphics Settings Intel(R) Graphics Power Plan On battery: Balanced Plugged in: Balanced



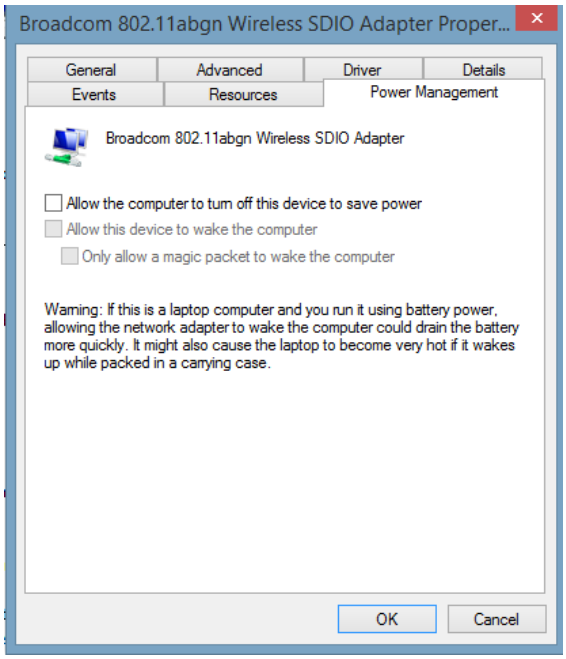
13. Click **Apply** and **OK**.
14. Return to the main *Control Panel* screen and click **Device Manager**.
15. Scroll down to **Universal Serial Bus controllers** and click the arrow to expand the list.



16. Double-click **Generic USB Hub** to display its Properties screen.



17. Click the **Power Management** tab, make sure both boxes are unchecked and click **OK**. Repeat this step for each item in the expanded list.
18. Return to the Device Manager list and expand Network Adapters.
19. Double-click **Broadband 802.11abgn Wireless SDIO Adapter** and select **Properties**.

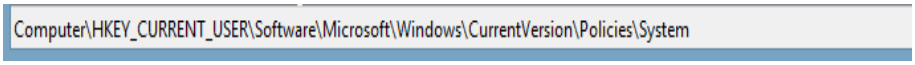


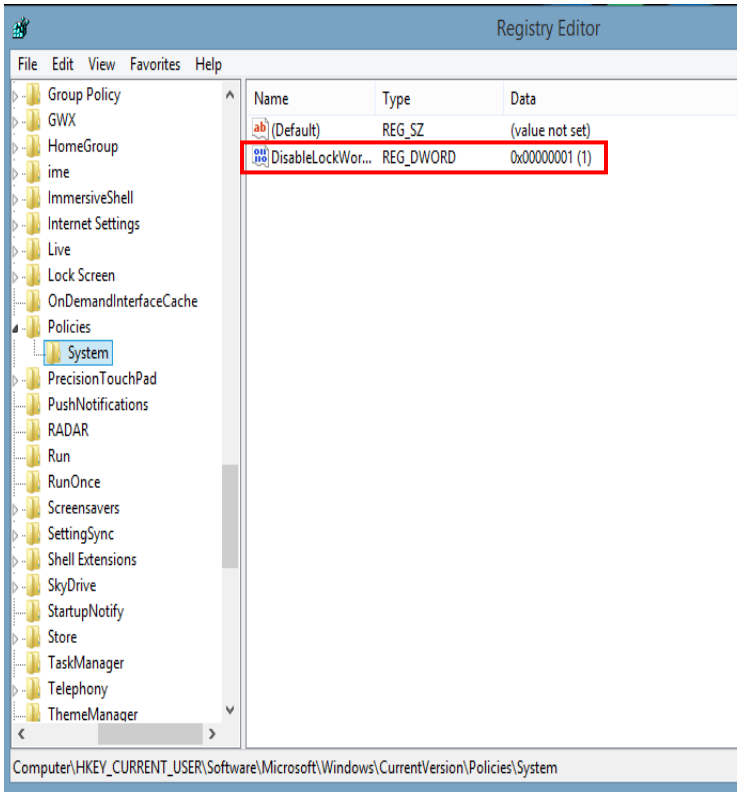
20. Click the **Power Management** tab, make sure both boxes are unchecked and click **OK**.

Tablet Settings

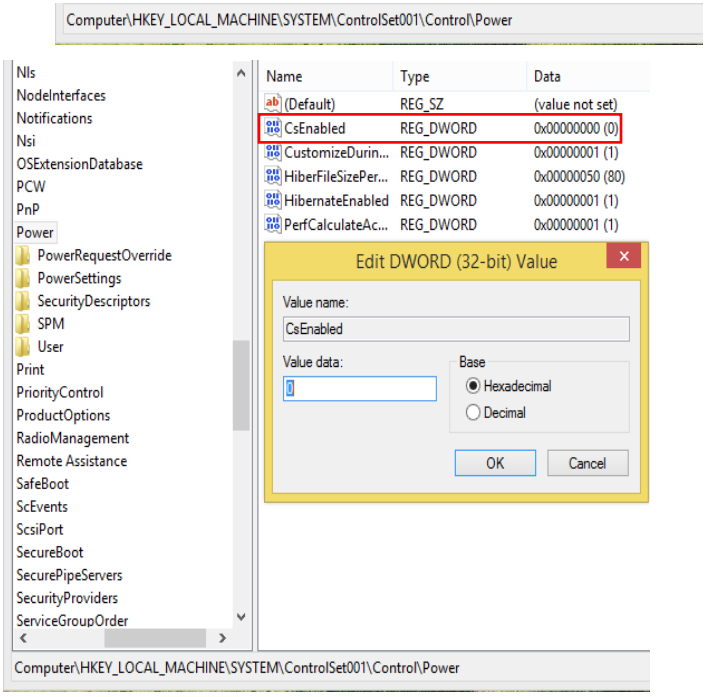
In the event you deviate from the tablet's CMI factory settings, these are the required settings.

1. From Microsoft's main program menu, click Computer and enter the following path:

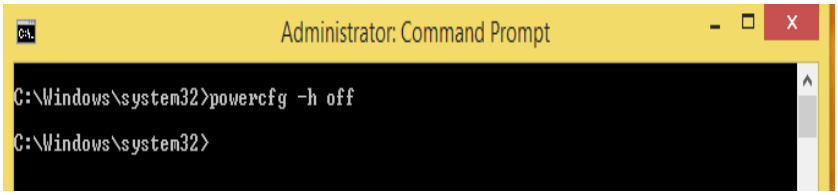




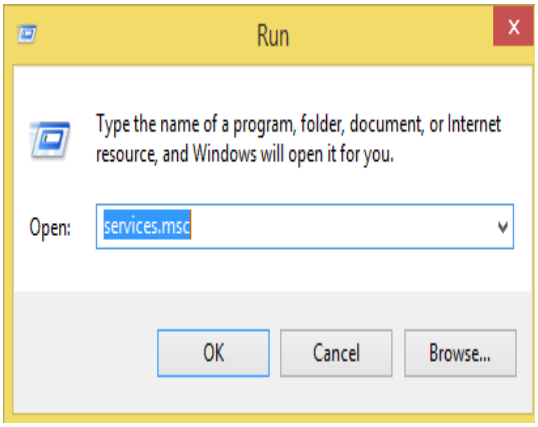
2. Make sure the **DisableLockWorkstation** data value is set to `0x00000001 (1)`.
3. From Microsoft's main program menu, click Computer and enter the following path:



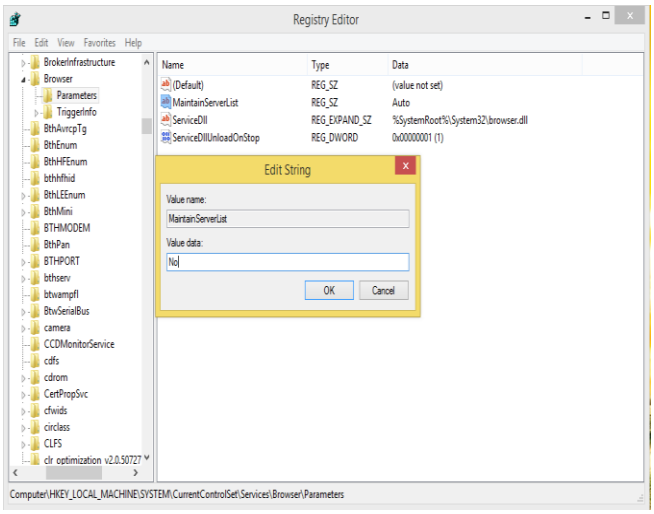
4. Disable Connected Standby by entering a **0** value and click **OK**.
5. Open a shell and disable Hibernation:



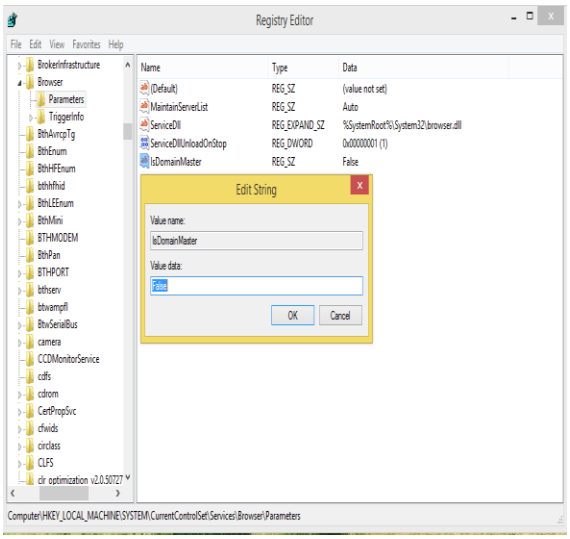
6. Disable Windows Browser Service:



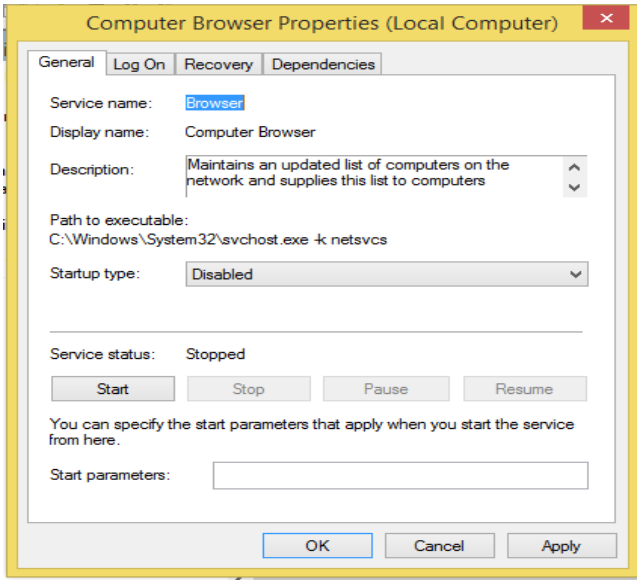
7. Run the Registry program and disable List Machines:



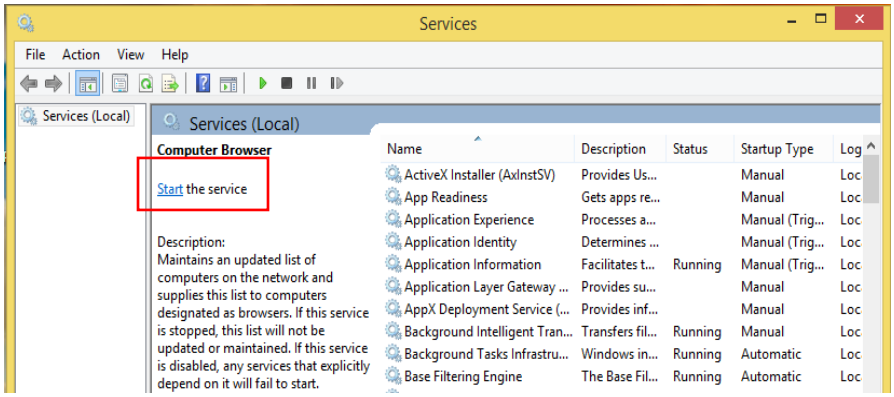
8. Create the IsDomainMaster string value:



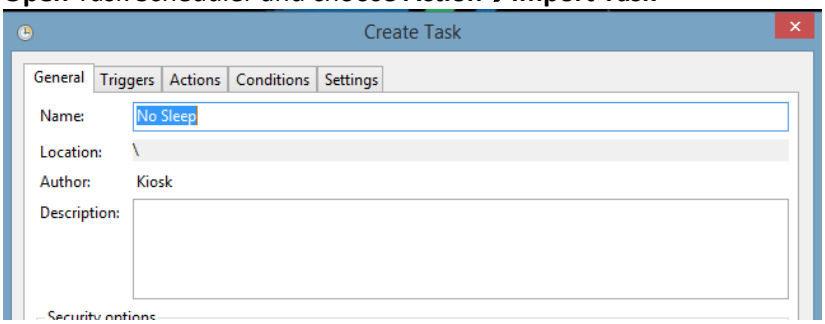
9. Set computer browser properties:

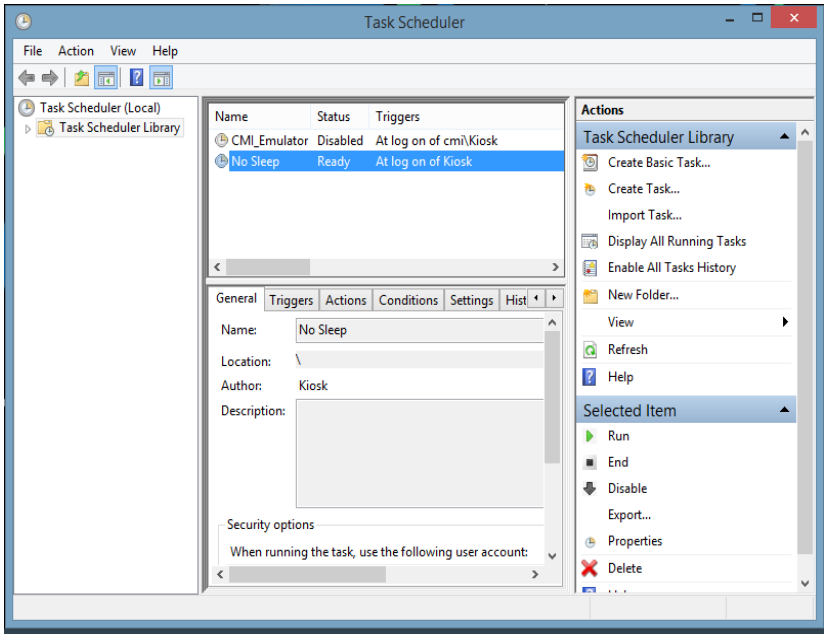


10. Create a **No Sleep** task:
    a. TFTP `nosleep.exe` and `nosleep.xml` to `\flashdisk` on the tablet.
    b. Right-click the **Start** button and choose **Control Panel→Administrative Tools.**



    c. **Open** Task Scheduler and choose **Action→Import Task**





Appendix B: Genus Emulator on TT2

The CMI Genus Emulator program for the Touch Time II provides a framework for running applications developed for CMI's Genus 2 data-collection terminal. The Emulator in conjunction with the Emulator Service provide application supervision and a container for applications providing a migration path for applications developed for CMI's Genus 2 and SaveTime terminals to run on the Touch Time II with minimal development. CMI also provides a light weight API based on the Genus 2 API so that Java applications can make use of the terminal's peripherals and functions tailored towards data-collection, yet allow for UI or system interfaces based on the latest Java JDKs and frameworks.

Application servers such as CMI's System Manager terminal management platform can be leveraged to provide seamless data-collection and network management support for very large terminal deployments.

Frameworks:

- Legacy SaveTime command set emulator
- Legacy Genus 2 terminal application emulator
- Java based application development with application launcher and auto-update features

Supporting tools:

- VNC Remote Management
- CMI System Manager terminal management application
- CMI Terminal Manager Application for SaveTime

TT2 Emulator Components

Emulator Service (TouchTime 2-EmulatorService)

- TFTP Server (Java-TFTPClientServer)
- Emulator Service Updater (TouchTime 2-EmulatorServiceUpdater)
- RMI Client/Server (TouchTime 2-RMI Base)
- Logging Framework (SLF4J frontend, Log4J 2 backend)

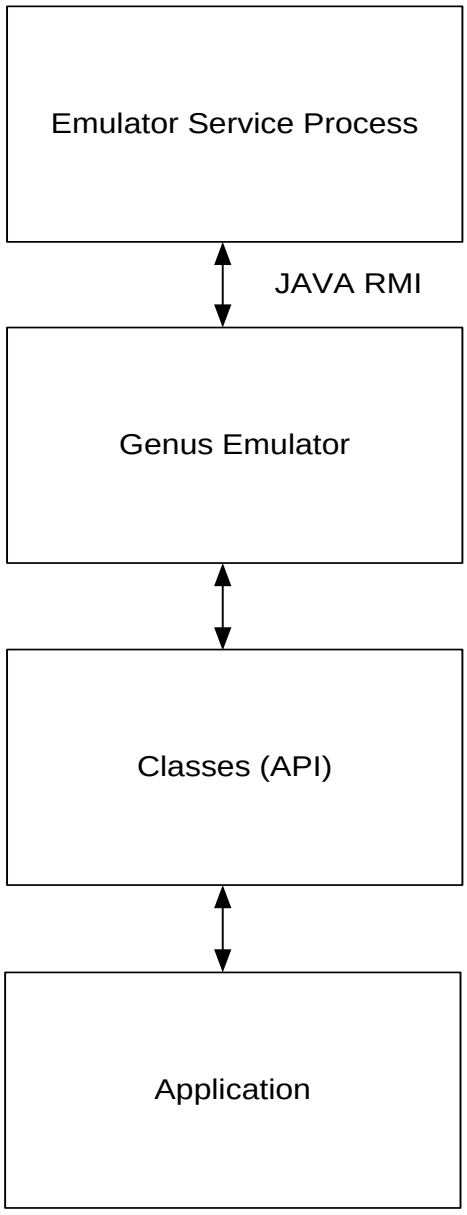
Emulator (TouchTime 2-Emulator)

- CMHID library (CMHID)
- CMMagtek library (not used anymore - CMMagtek)
- RMI Client/Server (TouchTime 2-RMI Base)
- Classes (TouchTime 2-Classes)
- Application

Utilities

- Remote Emulator Service Management Utility (TouchTime 2-RMI-EmulatorServiceManager)
- Remote Emulator Management Utility (TouchTime 2-RMI-EmulatorManager)

Software Components



EmulatorService.jar
- tftp server
- Monitors \flashdisk
- Emulator star/stop control
- Resides in \ramdisk\emulator
- If no app loaded defaults to picture Default.jpg

Emulator.jar
- Runs \flashdisk\App.jar in the \run directory
- Communicates with Emulator Service Process
- Resides in \ramdisk\emulator
- Config files
  \config\Settings.xml\SettingsUI.xml
  \audio\audiofiles
- Run environment for Emulator, App, Classes is located in the \run directory and can be from where ever the application may be running from.

Classes.jar
- Standard Classes
- Slightly modified API, PC Support and Readers

App.jar
- Standard Genus App
- Customer Application may need small UI modifications

Emulator Service

The Emulator Service runs as its own process (VM) and manages the Emulator process (VM). These processes communicate with each other by utilizing RMI (Remote Method Invocation). The Emulator Service also includes the TFTP server so that it can always receive file updates and restart the Emulator process accordingly. The Emulator Service monitors the `flashdisk` directory for file changes using the `FileMonitorService` class. The `FileUpdateService` combines the functionality that the `FileMonitorService` class provides with the functionality that the TFTP server library provides to be efficient. The TFTP server library dispatches events when a TFTP server transfer starts, completes, and fails. The `FileMonitorService` class then calls the appropriate method based on whether or not the file was transferred successfully. The files can also be manually copied in the `flashdisk` directory, or received via HTTP or any other method. When this occurs, a `FileMonitor` is started that monitors the size of the file being received. After 1 minute, if the file size does not change, the `FileMonitor` assumes that the transfer is complete and attempts to update that file. Once a transfer is complete, the Emulator Service then checks the file (if it is a JAR file), to see if it is valid. Note that only specific JAR's are checked, and they are listed below.

Checks for Application (App.jar)

1. The Emulator Service will attempt to extract the main class from the JAR file. It checks to see if a `Main-Class` attribute is defined inside the JAR file at `META-INF/MANIFEST.MF`, and if so, it assumes the JAR file is valid.
2. The Emulator Service will check if the class `CMIApp.class` exists in the root of the JAR file, and if so it assumes the JAR file is valid.
3. Otherwise the Emulator Service will reject the JAR file and will restart using the original JAR file. **Note**: The *bad* JAR file will still be in `flashdisk`.

Checks for Classes API (Classes.jar)

1. The Emulator Service will attempt to extract the manifest version from the JAR file. It checks to see if an `Application-Name` and `Application-Version` attribute is defined inside the JAR file at `META-INF/MANIFEST.MF`, validates them, and if so, it assumes the JAR file is valid.
2. Otherwise, the Emulator Service will reject the JAR file and will restart using the original JAR file. **Note**: The *bad* JAR file will still be in `flashdisk`.

Checks for Emulator (Emulator.jar)

1. The Emulator Service will attempt to extract the manifest version from the JAR file. It checks to see if an `Application-Name` and `Application-Version` attribute is defined inside the JAR file at `META-INF/MANIFEST.MF`, validates them, and if so, it assumes the JAR file is valid.
2. Otherwise, the Emulator Service will reject the JAR file and will restart using the original JAR file. (Note: The *bad* JAR file will still be in flashdisk)
3. Regardless of whether or not the file is accepted and copied to the run directory, if the file is a JAR file, the Emulator will restart once the operation finishes. In this way, updates to files that aren't JAR files are transparent to the end user as they can continue to use the terminal normally.

The Emulator Service launches the Emulator using the JAR files in the `Run` directory, and not the `flashdisk` directory. In a sense, the `Run` directory can be viewed as the `scratchdisk` on a Genus terminal. When the Emulator Service process starts the Emulator Process, it adds all JARs in the `Run` directory to the Emulator VM's `classpath`.

When the Emulator process is started, *ALL* files in the `Run` directory are added to the class path of the `Virtual Machine/Process` that the Emulator will run under. This Emulator also supports using native libraries (ie., DLL's). To do so, you would have to TFTP or place any native libraries into the `\libraries` folder so they can be called directly.

Some of the functionality provided through the Emulator Services RMI interface is outlined below:

- **Start**: The Emulator Service process will start the Emulator process. If the Emulator process is running, the request is ignored.
- **Stop**: The Emulator Service process will shut down the Emulator process. If the process isn't running, the request is

ignored.

- **Restart**: The Emulator Service process will restart the Emulator process. If the Emulator process is running, it is first shut down, and then started again. If the Emulator process isn't running, it is started.

The Emulator Service process also includes functionality to restart the Emulator process in the event an error occurred. If, for any reason, the Emulator process closes unexpectedly, the Emulator Service process will automatically restart the Emulator process. Also, if an unhandled exception ever occurs in the Emulator process, the Emulator Service process will automatically restart it.

Emulator

The Emulator runs as its own process (VM) and emulates the Genus application. For simplicity, the CMHID library is used to interface with all of the readers BUT the Suprema biometric reader, as well as the Opticon 2D barcode scanner and the IDTECH barcode reader. All the reader events then get routed through the `TouchTimeEmulator` class and then get injected as the appropriate reader event in the standard Classes API. For the Opticon 2D barcode scanner, and the IDTECH barcode reader, these readers act as Keyboard Wedges and their data is also handled by the `TouchTimeEmulator` class.

`EmulatorUtilities` in the TouchTime II Classes provides an interface between the Emulator and the Classes. For example:

Stop: The Emulator process will shut down. It will first send a message to the Emulator Service process to let it know that it is a clean shutdown (and not crashing).

Setup

Configuring and installing the TouchTime II Emulator is pretty straightforward. From the TouchTime II Emulator bundle, you just run the **Setup.cmd** file as an Administrator. The batch script is pretty self-explanatory, but it goes through copying the required files to their appropriate destination, checking for and installing Java, installing TightVNC Server, configuring the firewall for Java, creating directory symlinks from the root drive `ramdisk, flashdisk, novdisk`, and `scratchdisk` folders to the location of the TouchTime II Emulator bundle, configuring the desktop background, configuring the current power profile, installing Lenovo Access Connections Manager (for WiFi network management within the Emulator), and setting up and configuring the TouchTime II Emulator scheduled task.

Configuration Utility

The configuration utility can be run by browsing to the root directory and then running **Configure.cmd**. The files required to run the configuration utility are Emulator.jar and Classes.jar. The utility itself is pretty simple to use and it just provides a UI for customers to customize their emulation experience. This utility writes to two different configuration files: `\ramdisk\emulator\config\Settings.xml` and `\ramdisk\emulator\config\SettingsUI.xml`. The `Settings.xml file` contains all the emulator settings not related to the UI, and the `SettingsUI.xml` file contains those that do. You can remotely push and overwrite these files, then restart the emulator so the changes can take effect. When you need to get a customer's configuration, have them send the entire `ramdisk` folder so you can extract the customer-specific configuration files from the Emulator directory.

Note: Utilities are located in the **Utilities** folder; example/utility scripts are also there.

EmulatorServiceManager

Communicates via Java RMI with the Emulator Service.

Usage: `java -jar EmulatorServiceManager.jar -rmi-host <ip-address> [-rmi-port <port>] [-start|-restart|-shutdown|-force-shutdown|-check-mode|-set-restricted-mode|-set-unrestricted-mode|-get-tftp|-set-tftp [true|false]|-clear-memory [0|1]]`

Required Arguments:

```
-rmi-host <ip-address> [-start|-restart|-shutdown|-check-mode|-set-restricted-mode|-set-unrestricted-mode|-get-tftp|-set-tftp [true|false]|-clear-memory [0|1]]
```

`-rmi-host <ip-address>:` Specifies the IP address of the target Emulator Service to communicate with.

`-start:` Starts the application if it's not running.

`-restart:` Restarts the currently loaded application.

`-shutdown:` Shuts down the currently loaded application.

`-check-mode:` Checks the mode of the Emulator Service. Returns true if in "Kiosk Mode", false otherwise.

`-set-restricted-mode:` Checks the mode of the Emulator Service, and if it is not running in restricted mode ("Kiosk Mode"), it will switch the Emulator Service into that mode.

`-set-unrestricted-mode:` Checks the mode of the Emulator Service, and if it is not running in unrestricted mode (with Windows shell), it will switch the Emulator Service into that mode.

`-get-tftp:` Gets whether or not the TFTP server is currently running.

`-set-tftp [true|false]:` Starts or stops the TFTP server accordingly.

`-clear-memory [0|1]:` Clears either `flashdisk directory (0)` or `ramdisk directory (1)` and restarts the Emulator.

Optional Arguments :

`-rmi-port:` Specifies the port in which the Emulator Service is running its RMI server on. Only change this if you changed the default port of 1101.

Shutdown Example :

```
java -jar EmulatorServiceManager.jar -rmi-host 127.0.0.1 -shutdown
```

Start/Restart Example :

```
java -jar EmulatorServiceManager.jar -rmi-host 127.0.0.1 -restart
```

EmulatorManager

Communicates via Java RMI with the Emulator.

Usage: `java -jar EmulatorManager.jar -rmi-host <ip-address> [-rmi-port <port>] [-shutdown|-check-status]`

Required Arguments:

`-rmi-host <ip-address> [-shutdown|-check-status]`

`-rmi-host <ip-address>:` Specifies the IP address of the target Emulator Service to communicate with.

`-shutdown:` Requests the Emulator to cleanly shutdown, notifies the Emulator Service to not relaunch the Emulator.

`-check-status:` Requests the Emulators status flag, return false if the Emulator is in an error state and true otherwise.

Optional Arguments:

`-rmi-port:` Specifies the port in which the Emulator is running its RMI server on. Only change this if you changed the default port of 1100.

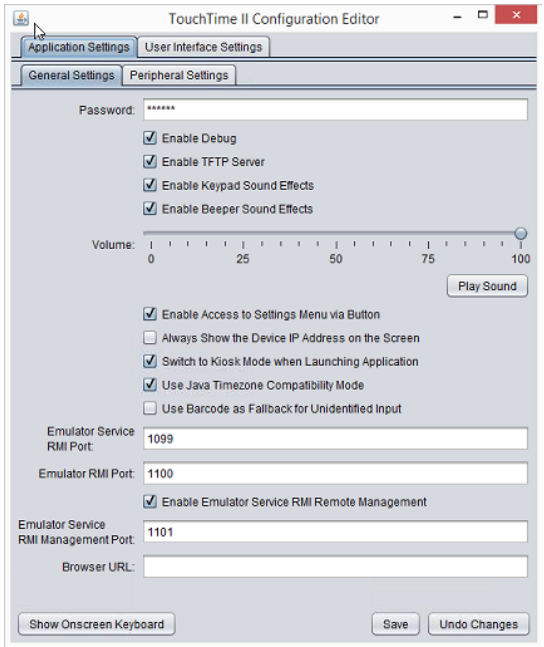
Shutdown Example:

```
java -jar EmulatorManager.jar -rmi-host 127.0.0.1 -shutdown
```

Appendix C: TouchTime II Configuration Command

To run the Touch Time II Configurator Editor:

1. Using the Settings button on the main screen, log on.
2. Launch the Control Panel.
3. Navigate to `C:\Users\Kiosk\Desktop\TcouhTimeIIEmulator\Root`
4. Run **config**. The TouchTime II Configuration Editor displays.



Note: Settings for both General and Peripheral are saved/stored within the **SettingsUI.xml** file located within `C:\ramdisk\emulator\config\SettingsUI.xml`

General Settings

- Password - Allows you to set a password in order to access the settings when the settings button is clicked.
- Enable or Disable the following:
    - **Debug** - Enable logging events to be posted to the **Errors.txt** file located within `\ramdisk\errors.txt`.
    - **TFTP** (Trivial File Transfer Protocol) - This is the method we use to transfer files, and update the Emulator. Leave this setting **Enabled**.
    - **Keyboard Sound Effects** – Use this to set the sound for when a user presses a key or button on the Emulator User Interface.
    - **Beeper Sound Effects** – Use this to set for when a user scans, swipes a badge, or presses a finger on the reader.
- **Volume** - Use the **Volume** slider to adjust the overall sound level and click **Play Sound** to hear the results.
- **Enable Access to Setting Menu via Button** - Allows you to enable or disable a user's to access the Settings Menu via the **Settings** button on the main Emulator screen.
- **Always Show the Device IP on the Screen** - Allows you to choose whether or not the IP address displays on the Emulator screen.
- **Switch to Kiosk Mode when Launching Application** - This mode prevents the Windows side menus (aka Charms) from being accessed. This setting is recommended.

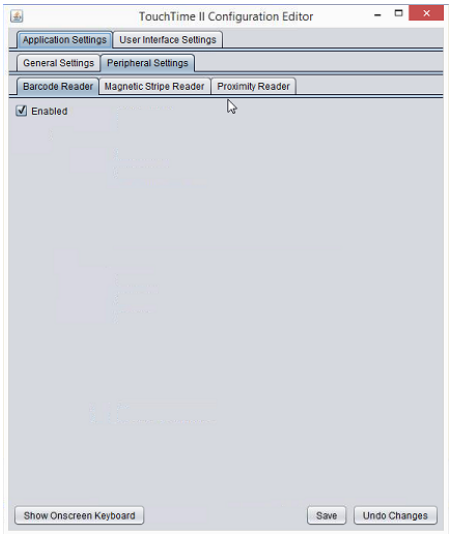
- **Use Java Timezone Compatability Mode** - Allows you to enable or disable the use of Java Timezone Compatability Mode.
- **Use Barcode as Fallback for Unidentified Input** - Makes the barcode reader the default reader if the reader type can't be identified.
- **Emulator Service RMI Port** - Enter the Remote Method Invocation port the Emulator Service is running the RMI server on. The RMI allows the Emulator Service and Emulator Process to communicate with each other using these ports. The Emulator Service process starts, stops, and restarts the Emulator process. It also can restart the Emulator process if an error or unhandled exception occurs, or the Emulator process closes unexpectedly. Default = 1099.
- **Emulator RMI Port** - Enter the port the RMI server is running on. Default = 1100.
- **Enable Emulator Service RMI Remote Management -** Allows you to enable or disable the remote management of the Emulator Service RMI.
- **Emulator Service RMI Remote Management Port -** Enter the port for remote management of the Emulator Service RMI Management. Default = 1101.
- **Browser URL** - Allows you to specify the browser URL that will open when you click the **Browser** button on the Emulator screen.
- **Show Onscreen Keyboard** - Click to show or hide the keyboard on the Emulator screen.

Click **Save**.

Peripheral Settings

Use this area to configure Barcode, Magnetic Stripe and Proximity Readers. Changes made in this section require restarting the Emulator. To restart the Emulator, click the **Setting** button, log in, click the **System** tab, select **Restart Emulator**, and click **Confirm**.

Barcode Readers

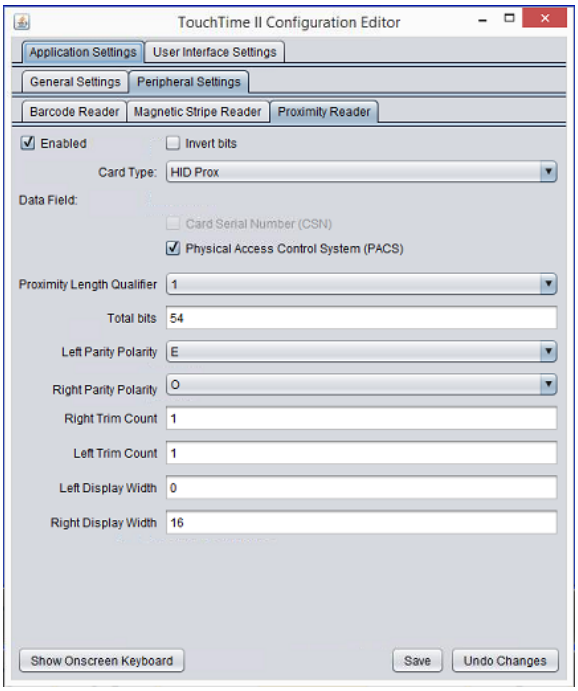


1. Click the checkbox to enable Barcode Readers associated with your system.
2. Click **Save**.
3. Restart the Emulator.

Magnetic Stripe Readers

1. Click the checkbox to enable Magnetic Stripe Readers with your system.
2. Use the **Data Track** dropdown menu to select the tracks this reader reads.
3. Enter the start and end numbers for the **Substring** index. **Start IDX specifies the index of the first character to read. End IDX specifies the index of the last character to read to. A zero indicates that you wish to read the entire track data.**
4. Click **Save**.

Proximity Readers

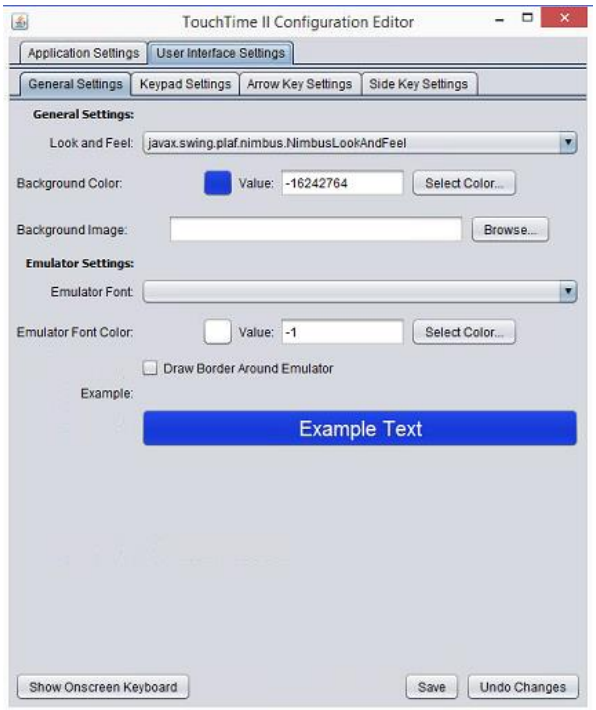


1. Click the checkbox to enable Proximity Readers with your system.
2. Click the **Inverting Bits** checkbox to read inverted bits from the card. Converts 0's to 1's and vice versa.

3. Use the **Card Type** dropdown menu to select the type of proximity card your system will be reading.
4. The **Data Field** is no longer applicable.
5. Select the **Proximity Length Qualifier** from the dropdown menu. DISABLE = 0, ENABLED_BITCNT_EQUAL_TB = 1, ENABLED_BITCNT_LESS_OR_EQUAL_TB = 2. This field will accept or reject cards based on whether the bits on the card matches the value of the **Total Bits** field. It will reject cards if the bit counts do not match or if they are larger than specified.
6. The **Total Bits** field sets the total number of bits that are expected to be read from the card.
7. Select the **Left** and **Right Parity Polarity** from the dropdown menu to optionally set the parity of the left-leading and left-leading parity bits.
8. Enter the **Left** and **Right Trim Count** to specify the number of bits to trim from the left and the right.
9. Enter the Left and Right Display Width to specify the number of bits to include in the left and right portions of data to read from the card, typically a facility or site code.
10. **Prox as Barcode** (Emulator Version 1.4.21 or later) allows for proximity data to be parsed, converted to decimal, and delivered to the application as if a barcode badge were swiped.
11. Click **Save**.

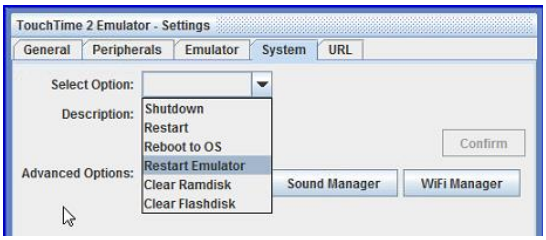
User Interface Settings

This tab allows you to edit the basic look and feel of the user interface, along with keypad, arrow key and side key settings.

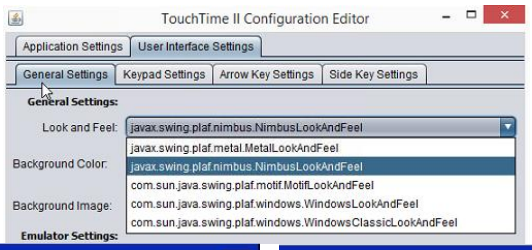


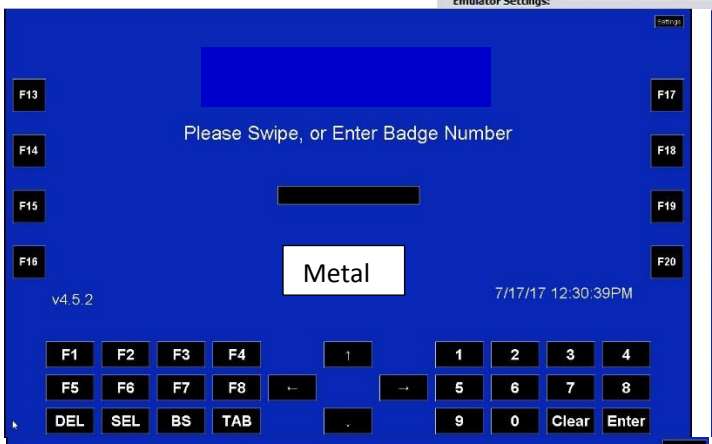
General Settings

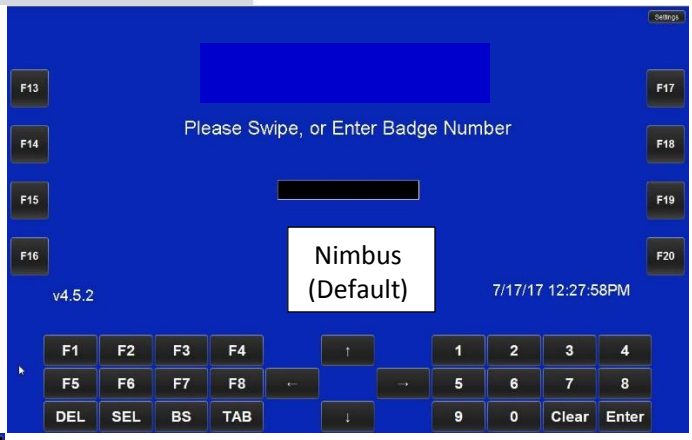
Changes made in this section require either restarting the Emulator or clicking **Save** to see the results of the change. **Undo Changes** only allows you to undo prior to saving. Once changes have been committed via **Save**, you need to edit back to the original settings if you want to restore the previous settings. To restart the Emulator, click the **Setting** button, log in, click the **System** tab, select **Restart Emulator**, and click **Confirm**.

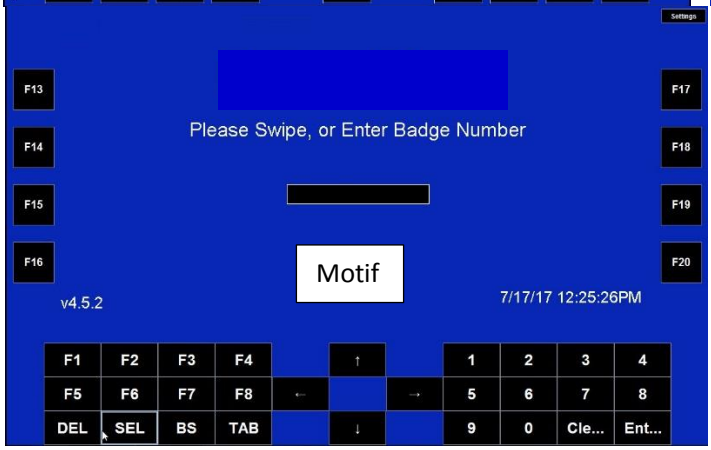


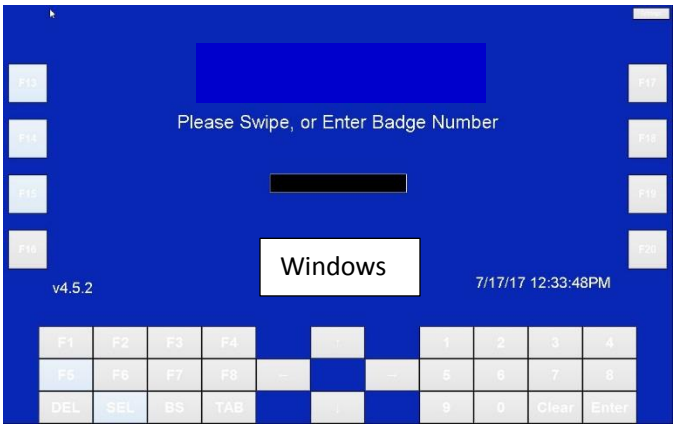
1. **Look and Feel** has five themes that can be used to change the look of the TouchTimeII user interface. The default is `javax.swing.plaf.nimbus.NimbusLookAndFeel`. Use the Look and Feel dropdown menu to select a different theme. We do not recommend Windows Classic. Click **Save** and restart the Emulator.





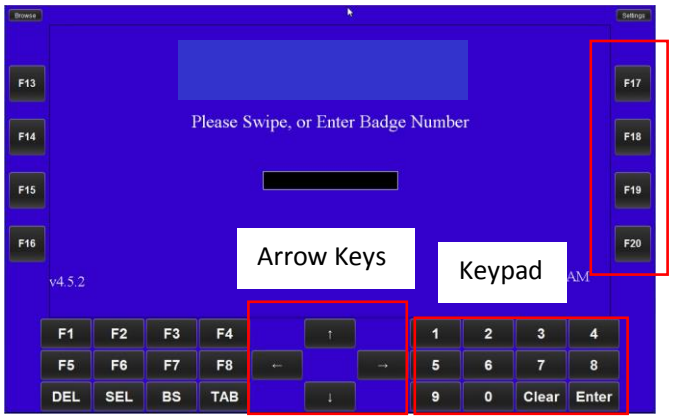




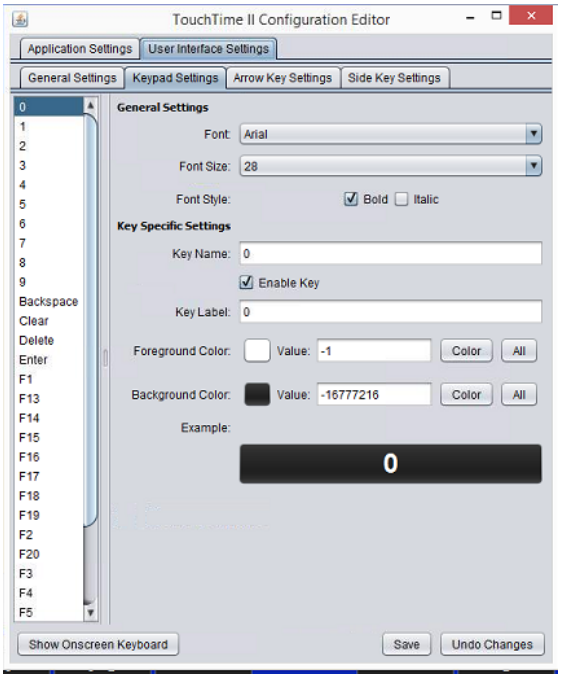


2. Click **Select Color** to choose a Background Color for your user interface. Click **Save** to view the change.
3. Click **Browse** select a Background Image for your user interface. Click **Save** and restart the Emulator.
4. Use the **Emulator Font** dropdown menu to select your user interface font. Click **Save** and restart the Emulator.
5. Click **Select Color** to choose a color for your Emulator Font.
6. Click the **Draw Border Around Emulator** checkbox to draw a border around the entire Emulator screen. Click **Save** and restart the Emulator.
7. Click **Save** if you have not done so already.

Keypad Settings

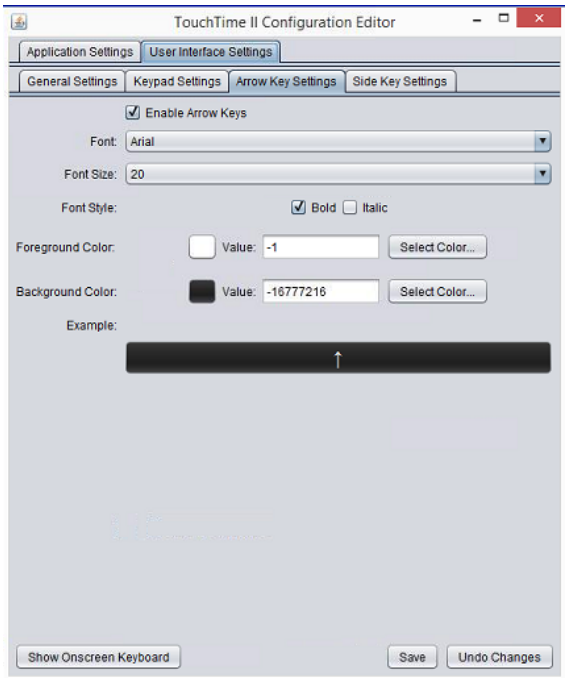


Use this screen to customize the Onscreen Keyboard that displays at the bottom of the Emulator screen.



1. To edit an existing key, scroll through the column on the left and highlight it. Click **Enable Key**.
2. Select the font type from the **Font** dropdown menu.
3. Select the font size from the **Font Size** dropdown menu.
4. Select **Bold** or **Italic** for the **Font Style**.
1. Enter the name of the key displayed on the emulator user interface in the **Key Name** field. The new name displays in the column on the left.
2. Click the **Enable Key** checkbox to activate this key. To remove the key from the display, uncheck this box.
3. Enter a label for the key in the **Key Label** field.
4. Click the **Color** button to select the key's Foreground Color. Clicking **All** applies this setting to all keys.
5. Click the **Color** button to select the key's Background Color. Clicking **All** applies this setting to all keys. The results of your choices display in the **Example** field.
6. Click **Save**. **Note**: Once you save, you cannot undo any changes. You will have to manually edit back to the previous settings.

Arrow Key Settings

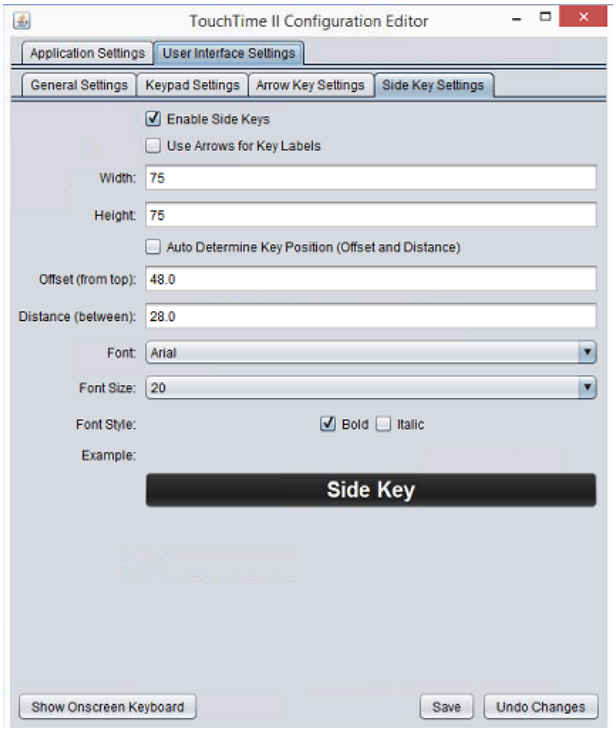


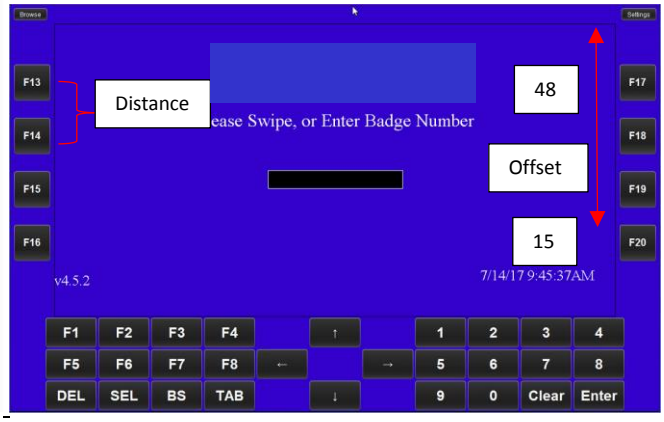
1. Click the checkbox to enable (Default setting) the use of arrow keys. Note how the keypad looks with arrow keys disabled:

2. Select the font type from the **Font** dropdown menu.
3. Select the font size from the **Font Size** dropdown menu.
4. Select **Bold** or **Italic** for the **Font Style**.
5. Click **Select Color** to choose a Foreground Color for your user interface.
6. Click **Select Color** to choose a Background Color for your user interface. The results of your choices display in the **Example** field.
7. Click **Save**. **Note**: Once you save, you cannot undo any changes. You will have to manually edit back to the previous settings.

Side Key Settings

If enabled in Keypad Settings, these are the function keys displayed on the side of the Emulator screen.





1. **Click the checkbox to enable** the use of side keys on the left and right of the Emulator screen.
2. Click the checkbox to enable the use of the arrows as key labels for the side keys.
3. Enter the width of the side keys in the **Width** field.
4. Enter the height of the side keys in the **Height** field.
5. Click the **Auto Determine Key Position** checkbox to automatically determine the key's position on the screen based on the offset from the top of the screen and distance of the key from other keys.
6. Enter the Offset (from the top) number if using Auto Determine. The default is 48. Changing to a higher number positions the Side keys further down the display. A value of 15 is the lowest setting allowed.
7. Enter the Distance number if using Auto Determine. The default is 28. Changing to a higher position places the Side keys farther apart and a lower number closest together on the display. A value of 15 is the lowest setting allowed.
8. Select the font type from the **Font** dropdown menu.
9. Select the font size from the **Font Size** dropdown menu.
10. Select **Bold** or **Italic** for the **Font Style**. The results of your choices display in the **Example** field.
11. Click **Save**.

Appendix D: Touch Time II Auto-Identification Options

This section is a reference to assist with the selection of optional card readers for the Touch Time II terminal. It provides a list of the supported card formats and an overview of the settings that can be used to adjust or format reader output.

Supported Card Formats and Reader Options

In order to support a wide range of employee auto-identification technology, the Touch Time II was designed to accept a number of optional card and fingerprint biometric readers. The Touch Time II can be equipped with a one side-mounted card swipe reader (magnetic stripe or barcode), a bottom-mounted fingerprint biometric scanner, a bottom-mounted proximity card reader, a bottom-mounted 2D barcode imager or a bottom-mounted combination proximity card reader and biometric fingerprint scanner. Typically, only a single reader is installed in either the side or bottom locations, with the exception being the biometric/proximity combination reader.

- Barcode Swipe Reader
- Magnetic Track II Stripe Reader
- Proximity – High & low frequency multi-class reader (13.56MHz & 125.6kHz combination card reader)
- 2D Barcode imager
- Biometric – (Can be purchased with 4MB or 8MB internal storage to hold 9000 or 18000 fingerprint templates)

Barcode Swipe Reader

The Touch Time II barcode reader accepts a wide range of barcode formats from a single reader. The physical interface is USB and data output is keyboard emulation.

IDTech Model#: WCR3237-700US

USB-Keyboard wedge interface

Media Formats:

Bar Code: UPC-A, UPC-E, EAN-8, EAN-13, Code 39, Telepen, Interleaved 2 of 5, Industrial 2 of 5, Code 128, MSI/Plessey, Codabar

Media Thickness: Bar Code: 0.005 inches (0.127mm) to 0.050 inches (1.27mm)

Slot Width: 0.055 (1.37mm)

Swipe Speed: Bar Code: 5 to 65 inches per second, bi-directional**

Bar Code Source Light: Infrared 930nm

Bar Code PCS: 60% minimum

Bar Code Centerline Length: 0.49 inches (12.50mm) from bottom of slot to reading window center

Bar Code Resolution Distance: 0.006 inches minimum

Barcode Swipe Card Data Manipulation

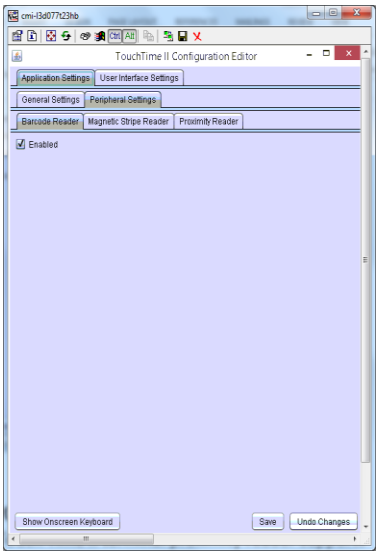
The Touch Time II Emulator Configuration Editor allows for enabling or disabling the generation of reader events for barcode swipe cards.

To launch the Configuration editor, you can run the following commands from the `c:\ramdisk\root` folder:

```
cd %~dp0
java -classpath run\* com.controlmod.touchtime.emulator.ui.UIConfigure
```

or

Navigate to the `c:\ramdisk\root` folder and run the `Configure.cmd`



- **Enabled** checkbox enables or disables barcode reader data events.
- Click **Save** then the **X** window close button to exit the configuration application.
- Next you will need to restart the emulator, which you can do from the **System** tab drop down.
- Once the emulator restarts, it will use the values that were updated from the `Settings.xml` file. (`c:\ramdisk\emulator\config\`)
- You can use TFTP to get the Settings file from one terminal and then transfer it to another terminal.

Magnetic Track II Stripe Reader

The Touch Time II magnetic swipe reader accepts a number of magnetic encoding standards for tracks 1 & 2 data. The physical interface is USB and data output is via Windows HID-formatted custom events.

Magtek Model#: 21040104

USB HID Swipe Reader

The USB (Universal Serial Bus) HID (Human Interface Device) Swipe Reader is a compact magnetic stripe card reader with a single read head that conforms to ISO standards. The Reader is compatible with any device with a host USB interface. A card is read by sliding it, stripe down and facing the LED side, through the slot either forward or backward. An LED (Light Emitting Diode) indicator on the Reader panel provides the operator with continuous status of the Reader operations.

The Reader conforms to the USB HID Class specification Version 1.1. This allows host applications designed for most versions of Windows to easily communicate with the device using standard Windows API calls that communicate with the device through the HID driver that comes with Windows. Unlike HID keyboard emulation readers, this device does NOT use keyboard emulation. It behaves like a vendor-defined HID device so that a direct communication path can be established between the Host application and the device without interference, such as keystrokes from other HID devices.

- Powered through the USB – no external power supply required
- Hardware-compatible with PC, or any computer or terminal with a USB interface
- Bi-directional card reading
- Reads encoded data that meets ANSI/ISO/AAMVA standards and others, such as ISO track 1 format on track 2
- Reads tracks 1 & 2 card data
- LED for status
- Compatible with USB specification Revision 1.1
- Compatible with HID specification Version 1.1
- Can use standard Windows HID driver for communications. No third-party device driver is required.

Magnetic Card Data Manipulation

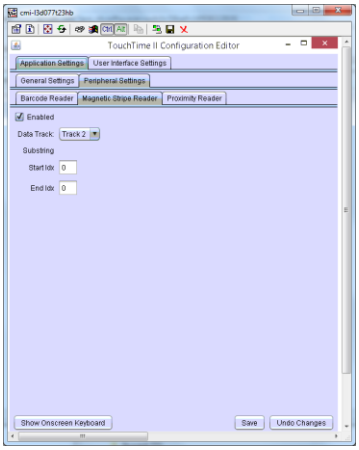
The Touch Time II Emulator Configuration Editor provides user-configurable parsing routines to specify the track and the start/end character positions used to extract data from the magnetic stripe, as well as enabling or disabling the generation of reader events for magnetic stripe cards.

To launch the Configuration editor you can run the following commands from the `c:\ramdisk\root` folder:

```
cd %~dp0
java -classpath run\* com.controlmod.touchtime.emulator.ui.UIConfigure
```

or

Navigate to `c:\ramdisk\root` folder and run the `Configure.cmd`



- Enabled checkbox enables or disables magnetic stripe reader data events.
- **Data Track** dropdown specifies the track 1 or 2 data to read.
- **Start IDX** specifies the index of the first character to read.
- **End IDX** specifies the index of the last character to read to. (A zero indicates that you want to read the entire track data)
- Click **Save** then the **X** window close button to exit the configuration application.
- Next you will need to restart the emulator, which you can do from the **System** tab dropdown.
- Once the emulator restarts, it will use the values that were updated from the `Settings.xml` file. (`c:\ramdisk\emulator\config\`)
- You can use TFTP to get the Settings file from one terminal and then transfer it to another terminal.

Supported Proximity Card Formats

The Touch Time II terminal proximity reader supports reading both 125kHz & 13.56MHz card technologies.

Model#: HID Omnikey 5427 CK

Supports the following card types:

- Proximity - High Frequency (13.56MHz)
  - iCLASS
  - iCLASS SE
  - iCLASS Seos
  - MIFARE Classic1K/4K
  - MIFARE Ultralight

- MIFARE Ultralight C
- MIFARE Plus
- MIFARE DESFire 0.6
- MIFARE DESFIre EV1
- ISO 14443 A with 848 Kbps transmission rate (depending on card)
- ISO 15693 with 26 Kbps transmission rate (depending on card)

- Proximity - Low Frequency (125KHz)
  - HID Prox
  - Indala Prox
  - EM4100
  - EM4102
  - EM4200
  - EM4305
  - EM4450 (Standard Mode Only)

Configuring the reader is a multi-step process where the card types and decode options are first set in the physical reader, and then binary bit conversion takes place within the application framework or through configuration options within the Emulator.

Setting up the reader to accept various card types is accomplished by uploading a configuration file to the reader. CMI preloads different configuration files on Touch Time II terminals based on the evaluation of customer cards and common card technologies. The following list provides an overview of the current set of configuration options that come pre-loaded on the Touch Time II:

- 6260-001: StandardProx_binary_PACS.cfg
- 6260-002: iCLASS_binary_PACS.cfg
- 6260-003: iCLASS_SE_binary_PACS.cfg
- 6260-004: iCLASS_Seos_binary_PACS.cfg
- 6260-005: Mifare_binary_CSN.cfg
- 6260-006: Indala_binary_PACS.cfg
- 6260-007: Mifare Ultralight_binary_CSN

The first set of numbers in this list, beginning with 6260, indicates the CMI number assigned to the configuration specified. If your card type falls within the list, pre-loaded configurations the 6260 number can be supplied with your order and CMI will configure the proximity reader with this configuration. If you have a different card type, then the card should be sent to CMI for evaluation and another card configuration can be created.

Creating Proximity Reader Configurations to Support Additional Proximity Card Types

CMI's technical support group can assist you with creating card reader configuration files for other supported card types.

Configuring the TT2 to Handle Proximity Cards with More Than 30 bits Of Badge Data

The application allows for proximity bit conversion to decimal badge numbers for badges less than or equal to 30 bits. This will accommodate most low frequency badge types but will not accommodate most types of high frequency card types. To provide a work around, CMI provided an option to allow the Emulator to be configured such that it will create barcode event types and convert the binary proximity data to a decimal number based on parameters set within the Emulator Configuration Editor. This work around is only available with **Classes.jar (v.1.0.187) & Emulator.jar (v1.4.21).**

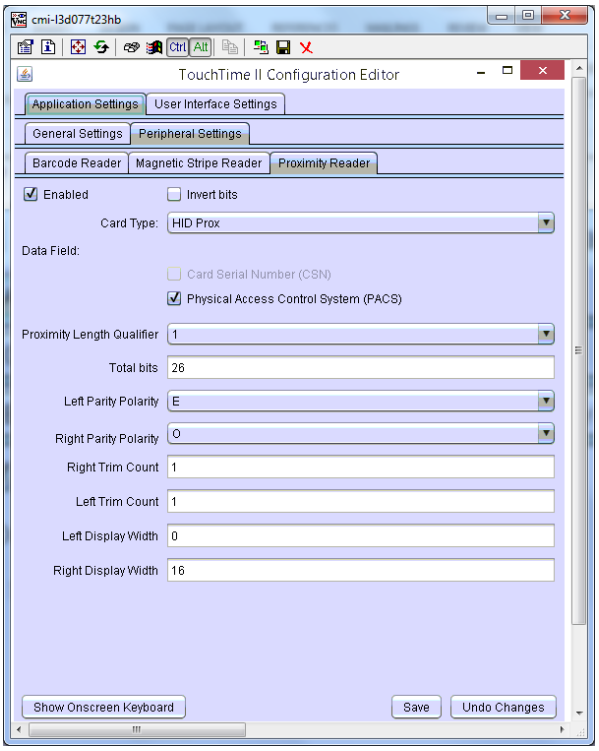
The Emulator proximity configuration will allow you to extract and convert all or only a partial number of bits within the proximity data based on bit offsets, trim, and bit widths specified.

To launch the Configuration editor you can run the following commands from the `c:\ramdisk\root` folder:

```
cd %~dp0
java -classpath run\* com.controlmod.touchtime.emulator.ui.UIConfigure
```

or

Navigate to `c:\ramdisk\root` folder and run the `Configure.cmd`



- **Enabled** checkbox – enables or disables proximity input.
- **Invert bits** checkbox – inverts bits read from the card if checked. (converts 0's to 1's and vice versa)
- **Card Type** dropdown – Obsolete – this field was used to differentiate between different card types. It used a data prefix specified in the reader configuration output string to specify the type of card being read. Currently all reader configurations use a prefix of 301 which equates to 'HID Prox' so this field should not be changed.
- **Data field** (obsolete – preset to PACS when using 'HID Prox' option reader configuration specifies the data output tag so this field should not be changed).
    - CSN – Card serial number
    - PACS – Personal access control system
- **Proximity Length Qualifier** (DISABLE = **0** ENABLED_BITCNT_EQUAL_TB = **1** ENABLED_BITCNT_LESS_OR_EQUAL_TB = **2**) *This field will accept or reject cards based on whether the bits on the card match the value of the **Total bits** field. It will reject cards if the bit counts do not match or if they are larger than specified.*
- **Total bits** – sets the total number of bits that are expected to be read from the card.
- **Left Parity Polarity** - optionally sets the parity of the left leading parity bit.
- **Right Parity Polarity** – optionally sets the parity of the right parity bit.
- **Right Trim Count** – specifies the number of bits to trim from the right.
- **Left Trim Count** – specifies the number of bits to trim from the left.
- **Left Display Width** – specifies the number of bits to include in the left portion of data to read from the card (typically a facility or site code).
- **Right Display Width** - specifies the number of bits to include in the right portion of data to read from the card (typically the badge number data).
- **Prox as Barcode** – Emulator Version 1.4.21 or later allows for proximity data to be parsed, converted to decimal, and delivered to the application as if a barcode badge were swiped.

- Click **Save** then the **X** window close button to exit the configuration application.
- Next you will need to restart the emulator, which you can do from the **System** tab drop down
- Once the emulator restarts, it will use the values that were updated from the `Settings.xml` file. (`c:\ramdisk\emulator\config\`)
- You can use TFTP to get the Settings file from one terminal and then transfer it to another terminal.

Barcode 2D Imager

The Touch Time II allows for an integrated 1D & 2D barcode imager. The Imager is installed in the place of the proximity or biometric reader location. It reads a wide range of barcode formats, including driver's licenses, etc.

The 2D CMOS fixed-mount imager, quickly and efficiently scans 1D and 2D barcodes. This scanner is ideal for retail or industrial barcode scanning applications. It features red LEDs for illuminating the viewable area and green LEDS for precisely targeting the barcode to be scanned.

Model#: Opticon NLV-3101

Media Formats:

BARCODE (1D): UPC -A, UPC -A Add-on, UPC -E, UPC -E Add-on, EAN-13, EAN-13 Addon, EAN-8, EAN-8 Add-on, JAN-8, JAN-13, Code 39, Tri-Optic, Codabar (NW-7), Industrial 2 of 5, Interleaved 2 of 5, S-Code, IATA , Code 93, Code 128, MSI/Plessey, UK/Plessey, TELEPEN, Matrix 2of5, Code 11, GS1 DataBar, GS1 DataBar Limited, GS1 DataBar Expanded, Composite GS1-DataBar, Composite GS1-128, Composite EAN, Composite UPC

BARCODE (2D): PDF417, MicroPDF417, Codablock F, QR Code, Micro QR Code, DataMatrix (ECC 0 - 140 / ECC 200), MaxiCode (Modes 2 to 5), Aztec Code, Chinese Sensible Code

BARCODE (POSTAL CODES): Chinese Post, Korean Postal Authority code

Biometric Fingerprint Reader

The Touch Time II provides biometric fingerprint enrollment, verification, identification, and fingerprint template management functions. Options include a standalone fingerprint reader or proximity/biometric reader combination.

FEATURES

- World's most reliable fingerprint algorithm
- Powerful 533MHz DSP
- High-speed fingerprint enrollment and authentication speed
- Compact size
- Low power consumption
- Fast power on time
- Various communication interfaces
- 256-bit AES fingerprint data encryption
- Supports various fingerprint output images, including RAW, GRAY(4-bit and 8-bit gray) and WSQ compressed image (certified by FBI)

Service and Technical Support

RMA Policy

Return Material Authorization (RMA) Procedure: The CMI Service Center assigns an RMA number for all products returned for service. If you have a product that requires service, please contact the CMI Service Center at 1-800-527-4998 or 860-253-4218.

The CMI Service Center provides various service options:

- Maintenance - Annual (five day in-house turnaround)
- Incident Maintenance - Flat Fee (five day in-house turnaround)
- Time & Material - Three options (two, five and 10 day turnaround)

The following information is required to process the return:

- Model and serial number of product
- Brief description of problem
- Name and telephone number of technical contact
- Customer's return address
- Customer's billing address

After an RMA # is issued, return the product to the address below in a shock-proof package to the CMI Service Center. Ensure the RMA# is clearly marked on the outside of the package and ship to:

CMI Service Center
89 Phoenix Avenue
Enfield, CT 06082-4439
Attn: RMA#

Note: Returned products cannot be processed without an RMA number.

Technical Support

CMI's Technical Support Number: 888-753-8222 can be reached during the following hours of operation:

M-F, 8:00 A.M. - 4:30 P.M. EST, excluding holidays

Standard Terms and Conditions of Sale

For Standard Terms and Conditions, and Warranty information:

http://controlmod.com/technical-support/order-terms/

CMI Time Management LLC

A Division of Control Module, Inc.

89 Phoenix Ave., Enfield, CT 06082

Local Phone: 860.745.2433

Toll Free Phone: 800.722.6654

www.cmitime.com