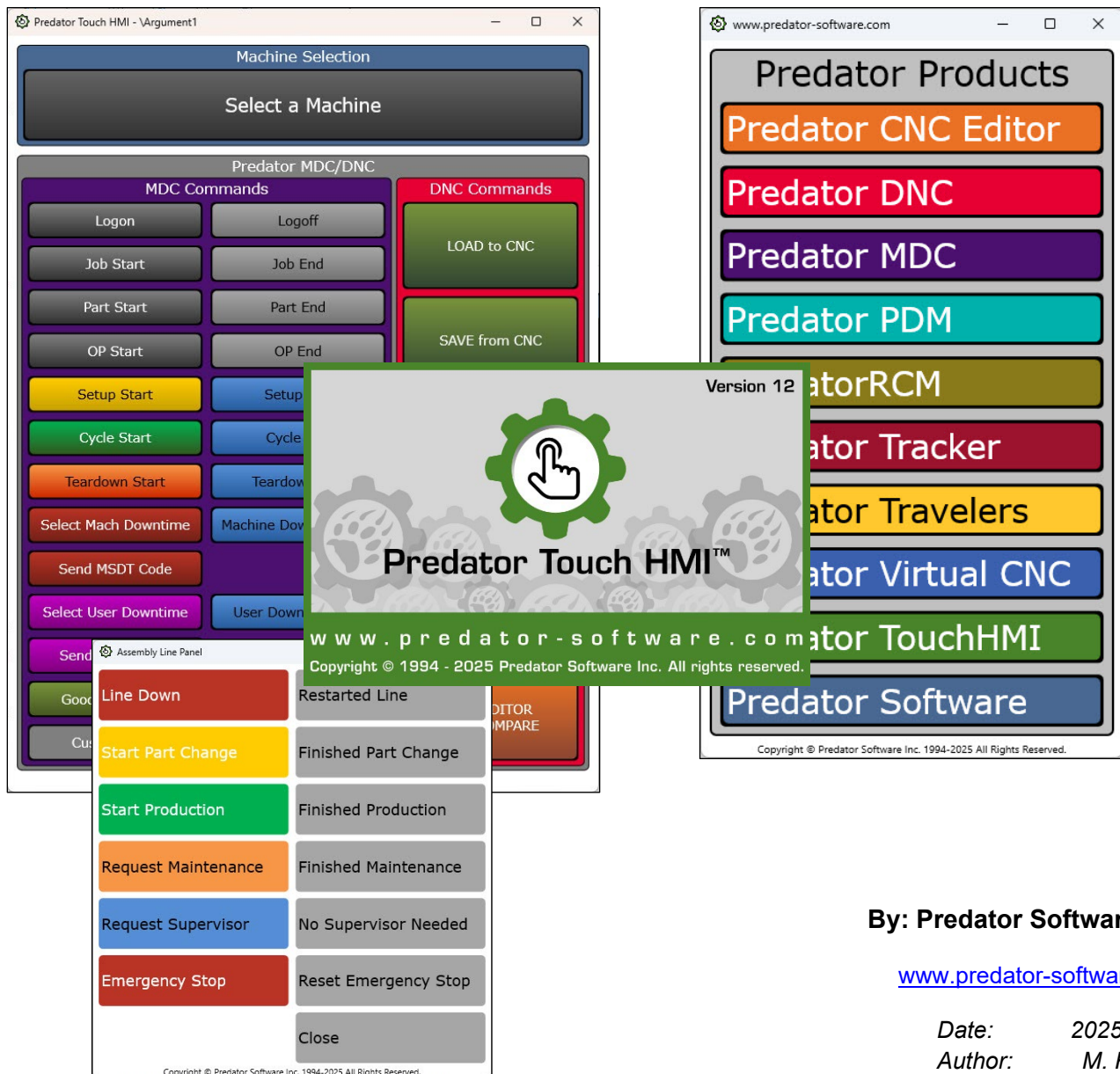




Predator Touch HMI v12 Reference Manual

Creating easy to use Manufacturing HMI panels for tablets and touch screen PCs



By: Predator Software Inc.

www.predator-software.com

Date: 2025-11-21

Author: M. Rogers

Version: v12.0.25.28

Predator Software Inc.

11092 SW Allen Blvd • Beaverton, OR 97005-4821 • Phone 1 503-292-7151 • www.predator-software.com

Copyright © Predator Software Inc. 1994-2025. All Rights Reserved.

Information in this document is subject to change without notice. No part of this manual may be reproduced or transmitted in any form or by any means electronic or mechanical, for any purpose, without the express written permission of Predator Software Inc. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

All trademarks and registered trademarks referenced in this manual are owned by their respective holders.

Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual is reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Printed in the United States of America.

Document Name: TECHNICAL_PREDATOR TOUCH HMI REFERENCE MANUAL (V12.0.25.28).DOCX

Predator Touch HMI v12 Manual

For Microsoft Windows O/S

Predator Software Inc.
11092 SW Allen Blvd South
Beaverton, OR 97005-4821 USA

Phone: 1 503-292-7151

E-Mail: support@predator-software.com
Website: <https://www.predator-software.com>

[This page has been intentionally left blank]

Predator Touch HMI



Dear Valued Customer,

Thank you for purchasing Predator Touch HMI. Everyone at Predator Software Inc. has worked very hard to make this a quality product, and we hope you agree.

I invite you to join us on our web site. We provide several services on-line that should prove quite valuable, including technical questions and answers, new feature requests, bug reports, product demo versions, and brochures.

If you have any concerns about Predator Touch HMI, please tell us so we can address them. We are available through any of the following methods:

Phone: 1 (503) 292-7151
E-Mail: support@predator-software.com
Website: www.predator-software.com

We would like to hear from you if Predator Touch HMI has improved your productivity, made you more competitive, or saved you money. Your success story could be featured in an upcoming article in a major manufacturing publication.

If you like Predator Touch HMI, please tell others. A high percentage of our sales are from referrals by satisfied customers, and we would appreciate your ongoing support.

Thank you again for selecting Predator Software Inc.

At your service,

A handwritten signature in black ink, appearing to read "Jim Abbassian".

Jim Abbassian
President
Predator Software Inc.

[This page has been intentionally left blank]

Contents

OVERVIEW	10
OPERATING SYSTEM SUPPORT	11
DELIVERABLES	11
PREDATOR TOUCH HMI INSTALLATION	12
INSTALLATION BEST PRACTICES	12
REMOVING A TOUCH HMI PANEL OR INSTALL.....	13
OPERATION	14
COMMAND LINE OPTIONS	14
<i>Command Line Option Descriptions.....</i>	<i>14</i>
THE TOUCH HMI CONFIGURATION	17
USE OF COMMENTS.....	17
THE [FORM] SECTION	17
THE [FRAME] SECTION	25
THE [BUTTON] SECTION	28
<i>Button Defaults Section.....</i>	<i>28</i>
<i>Button Section</i>	<i>30</i>
<i>Primary Button Settings.....</i>	<i>30</i>
<i>Alternate Button Settings.....</i>	<i>31</i>
<i>Down Button Settings.....</i>	<i>31</i>
<i>Touch HMI Button Configurations</i>	<i>32</i>
BUTTON COMMANDS	36
<i>Ethernet.....</i>	<i>36</i>
<i>Serial</i>	<i>36</i>
<i>RFID</i>	<i>37</i>
<i>Program.....</i>	<i>37</i>
<i>ProgramWait.....</i>	<i>38</i>
<i>Arguments</i>	<i>38</i>
<i>Copy</i>	<i>39</i>
<i>Rename</i>	<i>39</i>
<i>Delete</i>	<i>39</i>
<i>ReloadINI.....</i>	<i>39</i>
<i>Delay</i>	<i>39</i>
<i>Password.....</i>	<i>40</i>
<i>Exit.....</i>	<i>40</i>
PROMPTS	41
<i>Use of the “File” Prompts</i>	<i>41</i>
<i>Use of the “OSFile” Prompts</i>	<i>41</i>
<i>Use of the “Last” Prompts</i>	<i>44</i>
BUTTON EXAMPLES	45
TOUCH HMI AND THE PREDATOR DNC SERVER TERMINATOR.....	48
TOUCH HMI AND THE PREDATOR DNC COMPORT HSM TERMINATOR.....	48
DEFAULT PREDATOR DNC AND PREDATOR MDC COMMAND REFERENCE	50
DEFAULT PREDATOR DNC COMMANDS	50
DEFAULT PREDATOR MDC COMMANDS	50
MDC DOWNTIME COMMAND EXAMPLES	51

PREDATOR TOUCH HMI V12 INTERFACE	52
NEW TOUCH HMI INTERFACE FEATURES.....	52
THE TOUCH HMI PROPERTIES EDITOR.....	52
<i>The Main Form Configuration.....</i>	<i>53</i>
<i>Touch HMI Color Picker Tool.....</i>	<i>57</i>
<i>The Frame Configuration</i>	<i>58</i>
<i>The Button Configuration</i>	<i>60</i>
THE TOUCH HMI PROPERTIES EDITOR CONVENTIONS	68
TOUCH HMI POP-UP MENU TUTORIAL	69
<i>Step 1 - Create the Pop-Up Menu on a Button</i>	<i>70</i>
<i>Step 2 - Creating the Pop-Up menu Options</i>	<i>70</i>
<i>Step 3 - Configuration the new Pop-Up Buttons</i>	<i>71</i>
TOUCH HMI VARIABLES REFERENCE LIST.....	77
REPLACEMENT VARIABLES (REVISED AS OF V12.0.25.X).....	77
USING PATHS WITH PREDATOR TOUCH HMI COMMANDS.....	79
ABSOLUTE PATHS	79
RELATIVE PATHS.....	79
UNC PATHS	80
URL PATHS.....	80
FIREWALLS	81
FAQS	81
TROUBLESHOOTING.....	82
TIP 1 - RESTRICTED ACCESS WITH GROUP POLICIES	82
TIP 2 - NO INTERNET ACCESS	82
TIP 3 - ISSUE WITH TEXT IN AN INI FILE WITH "DOUBLE QUOTES"	82
CNC SUPPORT	83
CMM SUPPORT	86
COLOR REFERENCE	87
COMMON RGB REFERENCE COLORS.....	87
PREDATOR APPLICATION COLORS	91
PREDATOR MDC STATUS COLORS	92
WORKING WITH GRADIENT COLORS IN TOUCH HMI V12.....	93
TOUCH HMI V10 - V11 INTERFACE NOTES.....	95
UPGRADING TOUCH HMI INI FILE FROM VERSION 10 TO VERSION 11	95
THE V10 - V11 APPLICATION.INI FILE.....	95
<i>Use of Comments.....</i>	<i>95</i>
<i>The [Form] Section.....</i>	<i>96</i>
<i>The [Button#] Section.....</i>	<i>97</i>
TOUCH HMI V12 INTERFACE NOTES	98
CONVERTING V12 TOUCH HMI INI FILES TO VERSION 12.0.25.X.....	98
THE TOUCH HMI PROPERTIES EDITOR CONVENTIONS	99
TOUCH HMI V12 EXAMPLES	100
PTOUCHHMI.EXE	100
ASSEMBLY LINE EXAMPLE	106
PREDATOR DNC EXAMPLE	107
MDC DOWNTIME EXAMPLE	108



PREDATOR LIGHT TOWER EXAMPLE.....	109
PTOUCHLIGHT EXAMPLE	110
PTOUCHDARK.EXE.....	111
PTOUCH FILE OPTIONS EXAMPLE	112
SHOP FLOOR EXAMPLE	113
PTOUCHDEMO EXAMPLE	114
TOUCH HMI VERSION HISTORY	115
TOUCH HMI v12.0.2x.x	115
TOUCH HMI v11.0.22.7 (OCT. 12, 2022)	115
TOUCH HMI v10.0.5.27 (NOV. 23, 2015)	115

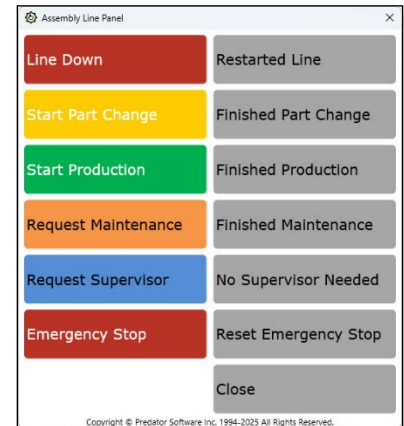
Touch HMI Reference Manual

Overview

Predator Touch HMI (from Predator Software Inc.) is an easy-to-use tool that allows the user to create “Big Button” Shop Floor interfaces for automating tasks for both the Predator Software Applications and other 3rd party Windows applications.

This document will focus on the general use of Touch HMI for a number of different applications. The Touch HMI Application can be installed on any shop floor PC or Tablet running the Windows Operating System. The device (PC or Tablet) should also have network access to the PC or Server running the Predator Applications that it will be communicating with.

Predator Touch HMI supports the creation of easy-to-use big button panels for data collection, process control, or shop floor automation. Clicking or touching a button can trigger one or more actions. Predator Touch HMI panels can be used to replace bar code readers and QR code readers on the shop floor.



This Windows based software is licensed per site, so an unlimited number of Touch HMI panels can be created and installed on Windows based PCs, CNCs, CMMs, test stands, and other shop floor equipment at one facility or location. Predator Touch HMI is designed for use with today's IT security requirements, with most functions running on the server via Predator DNC, MDC, or PDM. For example, E-Mail messages can be sent from a central Predator DNC Server triggered by a Touch HMI panel and not from the individual shop floor PC running the Predator Touch HMI application.

New Predator Touch HMI Panels are created by simply copying the source default executable file “**PTouchHMI.exe**” and its matching INI file with the same name (i.e. “**PTouchHMI.ini**”) from the following folder (the default install location):

C:\Predator Software\Touch HMI 12.0

and pasting them to become a new Touch HMI instance. For example:

- Make a copy of “**PTouchHMI.exe**” and rename it to “**LightTower.exe**”
- Make a copy of “**PTouchHMI.ini**” and rename it to “**LightTower.ini**”

You can then edit the **LightTower.ini** file to configure the number of rows, columns, buttons, colors, and other functions for the new Touch HMI Panel. Typically, a shortcut to the new **LightTower.exe** can be copied into the Windows startup folder if you want it to be launched when the PC starts up for day-to-day operation, or it can be run from a Desktop Icon. Another option is to create a shortcut in the Windows startup folder to a remote copy of a “PTouchHMI.exe” file which can be installed on another computer or server.

[NOTE: It is not necessary to copy the INI file with the EXE file if you are creating a completely new Touch HMI Panel from scratch. Version 12 of TouchHMI.exe includes a Touch HMI Editor Tool that will automatically launch if a mating INI file is not found when you double-click on a new TouchHMI.exe file. The Touch HMI Editor and its functions are described in detail in a later section of this manual.]

Operating System Support

Predator **Touch HMI v12.x** supports the following Microsoft Windows Client and Server operating systems:

Windows 11	Windows Server 2022
Windows 10	Windows Server 2019
Windows 8 and 8.1	Windows Server 2016
Windows 7	Windows Server 2012

[NOTE: Support for all Windows Editions and Service Packs is included.]

Predator **Touch HMI v11.x** supports the following Microsoft Client and Server operating systems as well:

Windows XP	Windows Server 2008
Windows 2000	Windows Server 2003
Windows ME	Windows Server 2000
Windows 98	Windows NT v4.0 Server
Windows 95	

[NOTE: Windows 95 PCs may have to install Winsock 2.0 support via "ws2setup.exe" for Networking support.]

It is common to find a wide range of Windows Operating System versions on shop floor PCs, CNCs, CMMs, test stands and other shop floor equipment. A list of supported Windows based CNCs and CMMs is provided at the end of this document.

Deliverables

Predator Touch HMI is delivered as:

- A single EXE called **PTouchHMI.exe** and its mating PTouchHMI.ini
- Along with a Series of Demo INI files that can be launched to show the full capabilities of Touch HMI
- And a copy of this Touch HMI Manual in PDF format

Predator Touch HMI does not require any additional files (such as DLLs, ActiveX components), registry entries, runtime libraries or .NET requirements, dependencies, or pre-requisites. Optional startup values for PTouchHMI.exe can be specified via command line arguments.

Predator Touch HMI Installation

Use the following procedure to install Predator Touch HMI on a PC or Server:

1. After you have downloaded the Install File (ptouch12.exe) from the Predator Software Website, copy it to a PC or Server where you want to install it.
*[NOTE: This Install download is not a standard Windows Install package, but rather a password protected ZIP file containing the latest PTouch.exe file, several Example *.ini files, and a copy of the latest Touch HMI v12 Manual.]*
2. Right-click on the “**ptouch12.exe**” install file and select “Run as administrator” to begin the install. You will be prompted for an Install Password, which should have been provided to you with your purchase of Predator Touch HMI. Enter the v12 install password to begin the installation.
3. The install procedure will extract all the files and place them in a new Folder with the default file location of:
C:\Predator Software\Touch HMI 12.0\.
4. Once the files have been extracted, you can go to the Install Folder location and start to review the examples and begin copying and editing your own Touch HMI Panels. Running the default **PTouchHMI.exe** will launch a Touch HMI Panel that will enable you to launch and view each of the included Demo INI files that are included with the Install. You can also launch the Manual from this Main Demo Panel as well.

Installation Best Practices

Predator Touch HMI v12 can be installed on a local drive at each CNC, robot cell, CMM, test stand, or other shop floor equipment running Microsoft Windows. However, it is best practice to run Predator Touch HMIs via a shortcut to a network share. To do this, step through the following procedure:

1. Create a shortcut to the appropriate “PTouchHMI.exe” in the startup folder for “All Users” (Refer to the following folder locations listed below).
2. Within the shortcut properties add a “-p” command line argument with the appropriate port number (Refer to the following command line arguments listed below).

[NOTE: Predator Touch HMI v12 does not include a typical Windows installation or MSI file since that would add additional requirements to the Windows based CNCs, robot cells, CMMs, test stands, and other shop floor equipment.]

The startup folder for “All Users” can vary depending on the version of Windows that is running on the device. Refer to the following defaults:

Windows 2022, 2019, 2016, 2012, 2008 & 2003 Server and Windows 11, 10, 8, 7 and Vista

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup

Windows 2000 Server, Windows XP, Windows 2000

C:\Documents and Settings\All Users\Start Menu\Programs\Startup

Windows NT

C:\WinNT\Profiles\All Users\Start Menu\Programs\Startup

Windows ME, 98 and 95

C:\Windows\Start Menu\Programs\StartUp

[NOTE: If Predator Touch HMI is installed in a locked down VLAN environment, the appropriate Ethernet Ports that you want to use must be opened up and allowed access over the network for the commands in the Predator Touch HMI Panels to function properly.]

Do not install any Predator Touch HMI applications in the local “C:\Program Files” or “C:\Program Files (x86)” folders or sub folders. These folders are reserved for Windows applications that require a proper install via a “*.MSI” or “setup.exe” process. When a local “installation” of the Predator Touch HMI application is desired, you should use one of the following two recommended folder locations:

- C:\Predator\
- C:\Predator Software\Touch HMI 12.0\ (the default installation location)

Removing a Touch HMI Panel or Install

Predator Touch HMI does not use a standard Windows Installer (via an .MSI or .EXE program). Therefore, it does not show up in the Windows Control Panel under “Programs and Features”. To remove a Touch HMI panel and its INI file you simply need to delete the two files from the local Hard Drive of the PC or Server where it was installed.

Operation

Predator Touch HMI Panels operate as standard 32-bit Windows applications. Support for standard Windows functions such as resize, minimize, maximize and multiple instances are all supported.

Clicking on or touching a button will start the defined automated actions. The Buttons on a Touch HMI Panel can graphically show a depressed condition when using a mouse to click on them, and an automatic tool tip will report when a button was last clicked on or touched by a user.

Command Line Options

Optionally Predator Touch HMIs can be run with one or more command line options from a Windows Shortcut. Predator Touch HMI supports the following command line arguments:

- i IP Address
- p Ethernet Port Number
- sp Serial Port
- t Title
- A1 Argument Variable One
- A2 Argument Variable Two
- A3 Argument Variable Three
- A4 Argument Variable Four
- A5 Argument Variable Five
- A6 Argument Variable Six
- A7 Argument Variable Seven
- A8 Argument Variable Eight
- A9 Argument Variable Nine
- convert Converts an INI file to the latest Version v12 replacement variables when opened with this command line option. *(This option was added in v12.0.25.23 and later)*
- properties This Command Line Option will launch the Touch HMI panel and open the Touch HMI Editor for that panel.
- trace This Command Line Option will save a Trace Log in the Touch HMI Log folder location.

Command Line Option Descriptions

The **-i** option allows the Ethernet IP address to be specified for use with the Touch HMI command, for example:

PTouchHMI.exe -i 10.2.0.104

The above IP address must match the IP address of the PC running the application that you want to control, i.e., Predator DNC. Typically, this is only necessary if multiple instances of an application are active, then the appropriate IP address must be specified in a command line argument. This will override the IP Setting in the Touch HMI INI file, if one exists.

The **-p** option allows the Ethernet Port Number to be specified for use, for example:

PTouchHMI.exe -p 11001

This will override the Port Setting in the Touch HMI INI file, if one exists.

Command line arguments can also be combined for specific applications. For example:

PTouchHMI.exe -i 10.2.0.104 -p 11001

This will override both the IP Address and the Port Setting in the Touch HMI INI file if they exist.

The **-sp** option allows the Serial Port Number to be specified for use, for example:

PTouchHMI.exe -sp COM5

The **-t** option allows the Title displayed in the Touch HMI Panel to be specified, for example:

PTouchHMI.exe -t Okuma1-PC

PTouchHMI.exe -t "Fadal VMC 4020"

*[NOTE: If a title or any text string has spaces in it, then double quotes "xxx xx" are needed to surround the title text. The double quotes character for Touch HMI INI File use **MUST BE** the ACSII Code DEC 34 Character (") or Straight quotes character, and not the Curly quotes characters "xxx xx", which are often referred to as smart quotes. That's because they curve toward the text you're quoting. These characters can be swapped out automatically in most word processors, like MS Word. Straight quotes are what they sound like – a set of straight vertical marks, without any indication of which word or phrase it's surrounding. See the Troubleshooting Section of this manual for more information on this issue.]*

The **-A1** though **-A9** command line arguments are optional and provide a method to pass additional run-time values into a Predator TouchHMI.INI configuration. For example:

PTouchHMI.exe -A1 "Jim Smith" -A2 "jims"

Launching the above application would provide two replacement variables **\Argument1** and **\Argument2** for use within the PTouchHMI.ini file. By running the above application, **Jim Smith** and **jims** will be available when running the application. For example:

[Form]

Title = Current User Login is: **\Argument1**

[Button]

Text = Check In

ProgramWait=c:\program files (x86)\predator software\tracker 12.0\ptracker.exe

Arguments=-user **\Argument2** -checkin

[Button]

Text = Check Out

ProgramWait=c:\program files (x86)\predator software\tracker 12.0\ptracker.exe

Arguments=-user **\Argument2** -checkout

[Button]

Text = Log off **\Argument1**

Program=. \Logon.exe

Exit=Silent

[NOTE: The -A1 through -A9 Command Line Options refer to and can set the values for the \Argument1 through -\Argument9 Replacement Variables found within a Touch HMI INI File. The Replacement Variable Names were changed from simply \A to \Argument due to the possibility that a user could use a string that contained a "\A" in a command inside of the Touch HMI INI file and thereby confusing the operation of the Button on a panel.]

[NOTE: Within the .INI file, the \Argument1 through \Argument9 Replacement Variables can be used as many times as needed. The above examples use each argument variable twice. If a \Argument Replacement Variable is not defined when a Touch HMI application is launched, the variable will return a value of the "Variable Name" (i.e. \Argument1) when executed. This will assist the user in determining where the error is in the INI File.]

The **-convert** command line option was introduced in v12.0.25.23. It was added to give the user a way to automatically update an existing v12 Touch INI file with the new Replacement Variable names when the Touch HMI Panel is opened. The following syntax examples show the format of this command line:

PTouchHMI.exe -convert

MyTouchHMIPanel.exe -convert

This **-convert** option will do the following:

- Add a new "**Version=**" setting to the top of the Form Section of the INI file. This shows the Version of the EXE that was used when the INI file was last run with the **-convert** option. An example would be:
 [Form]
 Version=12.0.25.28
- All of the older replacement variable **\A1** through **\A9** names will be updated to the new **\Argumentx** variable names.
- All of the other older variable names such as **Port=**, **PC=**, **IP4=**, and **User=** will also be converted to the updated names as of v12.0.25.23.

The **-properties** option allows a Touch HMI Panel to be opened along with the new v12 Touch HMI Properties Editor. The following syntax examples show the format of this command line:

PTouchHMI.exe -properties

MyTouchHMIPanel.exe -properties

This topic will be discussed in detail later in this manual.

Whenever possible try to use the default IP Address, Ethernet Port, Serial Port, and Title properties that are specified within the header of the .INI file. Command line arguments and multiple shortcuts are used to override these values within the header of the .INI file for special applications or use cases.

[NOTE: All command line options should be specified in the Windows shortcut's properties.]

The Touch HMI Configuration

Predator Touch HMIs are configured by setting different values within their **application.ini** file. The Touch HMI application.ini file can be organized into four sections:

[Form] - supported in Touch HMI v10, v11, and v12.

[Frame] - only supported in Touch HMI v12 and later.

[Button] - supported in Touch HMI v10, v11, and v12.

[Option] - only supported in Touch HMI v12 and later. This is an Option to a Button to create drop-down lists.

***[NOTE:** This is the format for Touch HMI v12 application.ini files. Touch HMI v10 and v11 had earlier formats that are discussed later in this document. Most of the information in the following sections applies to all Versions of Touch HMI (v12, v11, and v10) unless specifically noted.]*

Use of Comments

Within the .INI file it is often desired to comment out a line when testing a new .INI file configuration. Typing a semi colon character (;) at the beginning of a line will comment out that line. In other words that line will be ignored by Predator Touch HMI when it executes the INI file.

For example:

```
;Color = 255,255,255
```

```
Color = 0,0,0
```

In the above example the first Color line will be ignored, and the second Color line will be honored.

The [Form] Section

The **[Form]** section of the TouchHMI.ini file has been part of all past versions of Touch HMI (Touch HMI v10, v11, and v12). The Touch HMI.ini file should always begin with a **[Form]** header and always end with a **[EndForm]** end statement as the last line of the INI File.

This section of the INI File can include any of the following property variables:

Version (new as of v12.0.25.23)

Top (changed in v12.0.25.13)

Left (changed in v12.0.25.13)

Width (changed in v12.0.25.13)

Height (changed in v12.0.25.13)

Title

Color

Rows

Columns

Auto Center

Stay On Top

Minimize

Maximize

Exit

IP Address

Port

Serial Port
Baud Rate
Parity
Data Bits
Stop Bits
Monitor
Scaling
Timeout
RFID Protocol
RFID IP Address
RFID Port
[Log \(new in v12.0.24.12\)](#)
[Log Folder \(new in v12.0.24.12\)](#)

The **Version** property variable is new to the Form Section of an INI file as of Touch HMI version 12.0.25.23. It is automatically entered and saved at the top of the Form Section in the INI file when a Touch HMI Panel is opened, edited and saved using the TouchHMI Properties Editor or by using the **-convert** command line option. It is used to indicate what version of the Touch HMI EXE was used to edit or update the INI file.

The **Top** and **Left** property variables describe the position of the Touch HMI Panel when it opens on a Window Desktop from the upper left-hand corner of the current screen. The **Width** and **Height** variables describe the overall size of the Touch HMI Panel on a Window Desktop. These values can be edited manually in the INI file when the panel is not running to manually move or resize the Panel when it opens again.

[NOTE: The four Panel Size and Location variables, Left, Top, Width, Height, were changed as of Version 12.0.25.13. These four variables are now set as a Percentage (%) of the screen size and NOT a PIXEL value. This change was made to better support a wider variety of screen resolutions, including WFHD (2560x1080), UWQHD (3840x1600), and 4K (3840x2160) screens. If you open an older Touch HMI INI file with a newer v12 EXE (v12.0.25.13 and higher), you may need to resize and re-position your Panel and then save the changes.]

The **Left**, **Top**, **Width**, and **Height** variables are automatically updated whenever the Predator Touch HMI window is resized, maximized, minimized, or moved on the Windows Desktop. Although they can be changed manually within the .INI file, these values will be overwritten whenever the Panel is moved or resized on the Windows Desktop. The new values will be saved to the INI file when the Touch HMI Panel is closed.

*[NOTE: If the "application.ini" file is set to **read only** then Predator Touch HMI will not be able to update the Left, Top, Width, and Height variables automatically. Often this is desired to force the application to always start in a known location, size, and condition on the Windows Desktop.]*

[NOTE: If an "application.ini" file is copied from one PC to another with a differing overall screen size or resolution, it is possible that the Touch HMI Panel may not be visible when first run on the second PC. This is due to one or more of the Left, Top, Width, and Height variables being set to a value that falls outside of the screen resolution of the new PC. This is one of the times when editing these values manually in the INI file is warranted to correct this issue. This issue should not be a problem with the new Screen Percentage values for these settings.]

Title

The Title property is optionally used to further describe the HMI's purpose to end users within the title bar of the Main Panel. The Title Text entered in the Form Section of the INI File will be displayed at the top of the panel when it opens (as shown on the right).

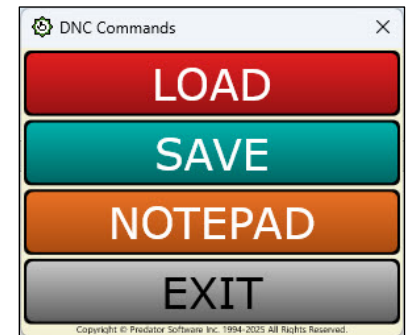
The title property is optional and is not required. Some examples are:

Title = DNC Commands (as shown in the example on the right)

Title = Machine 101 - Okuma

Title = Haas 2 Shop PC

[NOTE: The Title Property can be overridden with a -t command line argument. If the -t command line argument is used and spaces are required in the Title, then the title text must be encased in double quotes. However, within the INI file the Title property can be defined with spaces as shown above and the double quotes are not needed.]



Color

The color property in the **[Form]** Section is used to define the background color of the application. The default value is 255, 255, 255 for white. The color property is defined by three RGB (or Red-Green-Blue) values. For example:

Color = 255, 128, 0

With the above definition the Red value is 255, the Green value is 128, and the Blue value is 0. The minimum RGB value is 0 and the maximum RGB value is 255. At the end of this document there is a comprehensive list of common colors and their RGB values. A RGB Value of 0,0,0 is Solid Black, while a RGB Value of 255,255,255 is White.

The Touch HMI Editor has a built in Color Selection Tool to help the user to select colors for their Touch HMI panels and buttons.

Rows

The Rows property is used to define the number of rows to be used for Buttons and/or Frames within the HMI Panel or within a Frame in the Panel. The minimum value is 1 and the maximum value depends on the screen resolution.

Columns

The Columns property is used to define the number of columns to be used for Buttons and/or Frames within the HMI Panel or within a Frame in the Panel. The minimum value is 1 and the maximum value depends on the screen resolution.

Padding

The Padding property is used to define the amount of space between the edge of the Touch HMI Form and any Frames or Buttons that are displayed inside of the Form. The value for Padding is in Pixels. For example:

Padding = 5

[NOTE: By default, this property is set to "0". The example above would have a 5-pixel padding around the Form.]

Auto Center

The Auto Center property is used to enable or disable the ability for the Touch HMI Panel to be auto centered in the middle of the current Monitor that you are running the Touch HMI Panel on, or honor the Top and Left values described above. Auto Center is OFF by Default. For example:

Auto Center = True

Auto Center = False

11092 SW Allen Blvd • Beaverton, OR 97005-4821 • Phone 1 503-292-7151 • www.predator-software.com

Stay On Top

The Stay on Top property is used to define the HMI Window's behavior on the Windows Desktop. If it is set to **True**, then the Touch HMI Panel will stay on top of all other windows on the Desktop. If it is set to **False**, then it will behave like standard Windows applications and can be covered up by other windows on the Desktop. For example:

Stay On Top = True

Stay On Top = False

[NOTE: By default, this property is false.]

Minimize

The Minimize property is used to define the display and use of the Windows Minimize button in the HMI Panel's upper right-hand corner. If it is set to **True**, then the HMI Panel will include a Windows style Minimize Button. If it is set to **False**, then it will not have a Minimize Button. For example:

Minimize = True

Minimize = False

[NOTE: By default, this property is true.]

Maximize

The Maximize property is used to define the display and use of the Windows Maximize Button in the HMI Panel's upper right-hand corner. If it is set to **True**, then the HMI Panel will include a Windows style Maximize Button. If it is set to **False**, then it will not have a Maximize Button. For example:

Maximize = True

Maximize = False

[NOTE: By default, this property is true.]

Exit

The Exit property is used to define the display and use of the Windows Exit Button in the HMI Panel's upper right-hand corner. If it is set to **True**, then the HMI Panel will include a Windows style Exit Button. If it is set to **False**, then it will disable the Exit Button. For example:

Exit = True

Exit = False

[NOTE: By default, this property is true. A Disabled Exit Button is still visible but will not function if clicked. As of v12.0.25.23 the Touch HMI Editor has been updated to add an Exit Button by default to any new Touch HMI Panel that may be started from scratch with the Editor so it can be closed if no changes are made to its configuration.]

IP Address

The IP Address property is used to define the default IP Address of the Predator Server/ PC that the Touch HMI Panel will be communicating with. In addition to an IP Address, you can also specify the Server name if DNS is running on the network. For example:

IP Address = 127.0.0.1

IP Address = PServer1

NOTE: The above property is in the **[Form]** Section of the INI file but can be overridden by specifying the **-i** Command Line argument when Touch HMI is run, or in a Button's configuration with a **IPv4** Replacement Variable in the INI file.

Port

The Port property is used to define the Ethernet Port used to connect to the Application (i.e., the Predator DNC Server). The port number must match what is in the Predator DNC Server Terminator, for successful operation. For example:

Port = 11001

NOTE: The above property in the **[Form]** Section of the INI file but can be overridden by specifying the **-p** Command Line argument when Touch HMI is run, or in a Button's configuration with a **MachinePort** Replacement Variable in the INI file.

Serial Port

The Serial Port property is used to define the default serial port used to connect to the Predator DNC Server. Support for up to 34 COM Ports is supported by Predator Touch HMI. For example:

Serial Port = COM1

Serial Port = COM2

Serial Port = COM3

Serial Port = COM4

*[NOTE: By default, this property is set to **COM1**.]*

Baud Rate

The Baud Rate property is used to define the serial port's baud rate used to connect to the Predator DNC Server. Support for a wide range of baud rates is supported by Predator Touch HMI. For example:

Baud Rate = 110

Baud Rate = 300

Baud Rate = 600

Baud Rate = 1200

Baud Rate = 2400

Baud Rate = 4800

Baud Rate = 9600

Baud Rate = 14400

Baud Rate = 19200

Baud Rate = 38400

Baud Rate = 56000

Baud Rate = 57600

Baud Rate = 115200

Baud Rate = 128000

Baud Rate = 256000

*[NOTE: By default, this property is set at **9600** baud.]*

Parity

The Parity property is used to define the serial port's parity used to connect to the Predator DNC server. Support for the following parity settings is supported by Predator Touch HMI. For example:

Parity = None
Parity = Even
Parity = Odd
Parity = Mark
Parity = Space

*[NOTE: By default, this property is set to **None**.]*

Data Bits

The Data Bits property is used to define the serial port's number of data bits used to connect to the Predator DNC server. Support for the following data bits is supported by Predator Touch HMI. For example:

Data Bits = 8
Data Bits = 7
Data Bits = 6
Data Bits = 5

*[NOTE: By default, this property is set to **8 Data Bits**.]*

Stop Bits

The Stop Bits property is used to define the serial port's number of data bits used to connect to the Predator DNC server. Support for the following data bits is supported by Predator Touch HMI. For example:

Data Bits = 1
Data Bits = 1.5
Data Bits = 2

*[NOTE: By default, this property is set to **1 Data Bit**.]*

Monitor

The Monitor property is used to define which Monitor the Touch HMI Panel should appear on in a multi Monitor System Configuration. Touch HMI supports any number of Monitors that may be available on a PC system. For example:

Monitor = 1
Monitor = 2
Monitor = 3

Once selected, this will become the default Monitor that the Touch HMI Panel will appear on when restarted. If you move a panel to another Monitor and close it, the Default Monitor setting will be updated.

Scaling (eliminated as of v12.0.25.13)

The Scaling property is used to adjust the size of a Touch HMI Panel based on the current Monitor's screen resolution. This is used to address the use of newer 4K Screens which have a much higher resolution than other monitors. This setting **should not** be adjusted manually as the value will be set by the Touch HMI application automatically when a Panel is opened or displayed on a PC system with different resolution monitors. For example:

Scaling = 1

If you move a Touch HMI panel to another Monitor and close it, the Scaling setting will be updated if that Monitor has a different screen resolution to maintain the Touch HMI panel size.

[NOTE: As of Touch HMI v12.0.25.13, the need for the Scaling Property has been eliminated. It is no longer necessary as the size of a Touch HMI Panel has been converted from Pixel values to Percentages of the current Screen size. The Scaling property can be deleted from the INI file or left as a value of "1".]

Timeout

The Timeout property is used to define an optional timeout of the HMI, which will automatically close the HMI. This option can be handy when one Touch HMI panel runs another HMI. The default value is zero which means the HMI does not timeout. The timeout property is defined in seconds. For example:

Timeout = 20

With the above definition the form will automatically timeout or exit after 20 seconds. Every button push will extend the timeout another 20 seconds. If a button has an exit function it will override the timeout property when the Exit Button is pressed.

RFID Protocol

The RFID Protocol allows for different RFID manufacturers to be supported. Currently only Balluff Ethernet style controllers are supported. For example:

RFID Protocol = Balluff

[NOTE: By default, this property is set to Balluff. Since this is the only supported RFID protocol at the moment, this setting is currently optional. If you have another brand of RFID System in your shop, contact Predator Software Technical Support about supporting your RFID system.]

RFID IP Address

The RFID IP Address property is used to define the IP Address of the RFID Controller. For example:

RFID IP Address = 127.0.0.1 (default)

RFID IP Address = 172.16.30.31

RFID Port

The RFID Port property is used to define the Ethernet Port used to connect to the RFID controller. For example:

RFID Port = 10001 (default)

[NOTE: The above property value of 10001 is the default, and for Balluff RFID readers this would be optional.]

Log

The Log property is used to define the status of a Log File for a Touch HMI Panel. It is a True/False option. If True, a Log file will be created in the same location as the parent Touch HMI EXE with the same name, by default. The Log file will record button activity, including the name of the button, and the date and time that an operator clicked on it. If False, no log file will be created.

Log = True

Log = False

[NOTE: By default, this property is false.]

Log Folder

The Log Folder property is used to define the folder location for Touch HMI Log files. By default, if no Log Folder location is defined, the location will be the same folder as the parent Touch HMI exe file. For example:

Log Folder = C:\Predator Software\Touch HMI 12.0\ Logs

*[NOTE: The new **Trace Log** feature that can only be activated via a Command Line Option (-trace) from a Windows Shortcut, can provide a much more detailed Log File output for the operation of a Touch HMI Panel.]*

The [Frame] Section

The **[Frame]** section of the PTouchHMI.ini file **is new with the release of Predator Touch HMI v12** and allows the user to create Frames and nested Frames (Frames within a Frame) on a Touch HMI Panel. A Frame in the PTouchHMI.ini file should always begin with a **[Frame]** header and always end with an **[EndFrame]** statement as the last line of the Frame definition. This structure allows for multiple Frames in a Panel, and Frames within a Frame can be defined.

This section of the INI File can include any of the following property variables:

Parent Row
Parent Column
Height
Width
Text
Image
Color
Outline
Line Width
Rows (in the Frame)
Columns (in the Frame)
Padding

***[NOTE:** Frames are optional, and if a Touch HMI v12 Panel does not have any Frames defined in its INI File then it will look and act like a v10 or v11 Touch HMI Panel, which only had the Main Form and Buttons defined. This enables existing users to open and run their current Touch HMI v10 and v11 Panels (ini files) with the Touch HMI v12 .EXE. They just need to replace the current older TouchHMI.EXE file with a copy of the new Touch HMI v12 EXE file.]*

***[NOTE:** The introduction of Frames with Touch HMI v12 enables the user to eliminate what was the need for a large number of Touch HMI EXEs and INI files to support a large number of machines and applications on the Shop Floor. We can now use the concept of Frames to nest a number of functions all in one Touch HMI Panel with one INI file.]*

Parent Row

The Parent Row property defines the Row position of the Frame in either the Main Form, or the parent Frame if this Frame is a nested Frame. For example:

Parent Row = 1 (for the 1st row of the Form or Frame)

Parent Row = 2 (for the 2nd row of the Form or Frame)

Parent Column

The Parent Column property defines the Column position of the Frame in either the Main Form, or the parent Frame if this Frame is a nested Frame. For example:

Parent Column = 1 (for the 1st row of the Form or Frame)

Parent Column = 2 (for the 2nd row of the Form or Frame)

***[NOTE:** For a Frame to appear properly in the Main Form or a Parent Frame, the parent **MUST HAVE** the proper number of Rows and Columns defined in its definition first to enable the new Frame to be properly displayed.]*

Predator Touch HMI

Height

The Frame Height property defines the height of the Frame in its Parent (Form or Frame). The value for Height can be in screen pixels (as an integer) or as a Percentage (as a %) of the Parent height. For example:

Height = 600 *[this would give you a fixed Frame Height of 600 Pixels high]*

Height = 50% *[this would give you a % Frame Height of 50% of the Parent's available space]*

Width

The Frame Width property defines the width of the Frame in its Parent (Form or Frame). The value for Width can be in screen pixels (as an integer or whole number) or as a Percentage (as a %) of the Parent width. For example:

Width = 950 *[this would give you a fixed Frame Width of 950 Pixels wide]*

Width = 75% *[this would give you a % Frame Width of 75% of the Parent's available space]*

Text

The Frame Text property defines the Text to be displayed as the Title that appears at the top of the Frame in the Touch HMI Panel. For example:

Text = DNC Commands

Text = Predator MDC

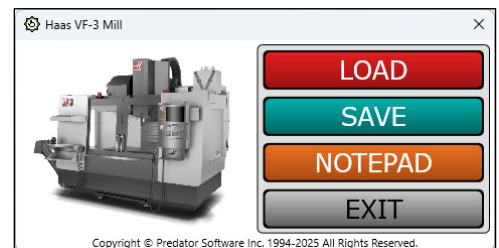
Image

The Image property enables a user to add a background image to a Frame. The Image property can have a path defined or no path. For example:

Image=HaasVF-3.bmp

Image=C:\Predator Software\Touch HMI 12.0\ HaasVF-3.bmp

The image should be a .BMP Format file and should be in the same directory as the Touch HMI.exe and .ini files for best performance and reliability. In the example panel shown on the right there are no buttons defined for the Frame on the left side of the panel so that the image shows through.



[NOTE: Predator Touch HMI also supports having a Background Image as the background of the overall FORM of a Touch HMI panel. In this case you do not need to add anything to the INI File, simply have a BMP Image in the same folder location as the TouchHMI.exe and TouchHMI.ini files, and the image must have the same exact filename as those two files.]

Color

The Frame Color property defines the color to be used as a background color for the Frame. The default value is 255, 255, 255 for white. The color property is defined by three RGB (or Red-Green-Blue) values. For example:

Color = 255, 128, 0

With the above definition the Red value is 255, the Green value is 128, and the Blue value is 0. The minimum RGB value is 0 and the maximum RGB value is 255. At the end of this document there is a comprehensive list of common colors and their RGB values. A RGB Value of 0,0,0 is Solid Black, while a RGB Value of 255,255,255 is White.

The Touch HMI Editor has a built in Color Selection Tool to help the user to select colors for their Touch HMI panels.

Outline

The Frame Outline (color) property defines the color to be used for the outline of the Frame. The color property is defined by three RGB values. For example:

Color = 0,0,0

With the above definition the Red value is 0, the Green value is 0, and the Blue value is 0, resulting in a Black Outline Color. The minimum RGB value is 0 (black) and the maximum RGB value is 255 (white). At the end of this document is a comprehensive list of common colors and their RGB values.

Line Width

The Line Width property defines the width of the Frame Outline in pixels to be used for the outline of the Frame. For example:

Line Width=1 (would give you a 1-pixel outline)

Line Width=5 (would give you a 5-pixel outline)

Line Width=0 (would give you a 0-pixel outline, or no line at all)

Rows (in the Frame)

The Frame Rows property defines the number of Rows that are to be used in the Frame. For example:

Rows = 1 (gives the Frame only 1 Row for Buttons)

Rows = 3 (gives the Frame 3 Rows for Buttons)

Columns (in the Frame)

The Frame Columns property defines the number of Columns that are to be used in the Frame. For example:

Columns = 1 (gives the Frame only 1 Column for Buttons)

Columns = 3 (gives the Frame 3 Columns for Buttons)

The total number of Buttons that can be added to an existing Frame is defined by the multiplication of the values for Rows times Columns. For example, if a Frame has values of Rows=3 and Columns=2, then the Frame can have a maximum of 6 Buttons defined in it. You can have less than the maximum number of Buttons defined in a Frame, and these extra areas will show up as blank areas, which can be used to show the background color or an image in the Frame.

Padding

The Frame Padding property defines the width of the padding or spacing between the Frame outline and the Button outlines inside of it. The value for Padding is in screen pixels (as an integer). For example:

Padding = 2

Padding = 0

[NOTE: The value for Padding can be set to "0", but all the Buttons will be positioned tight to the edge of the Frame and may not look as good as a configuration with some value for the Frame Padding.]

The [Button] Section

The **[Button]** sections of the TouchHMI.ini file have been part of all past versions of Touch HMI (Touch HMI v10, v11, and v12). Buttons have three major sections of the INI file that can affect them:

- **[Button Defaults]** Section
- **[Button]** Definition Sections
- **[Option]** Definition Sections (new with Touch HMI v12)

Button Defaults Section

The **[Button Defaults]** Section of an INI file is used to define the default settings for all Buttons in the INI file if these properties are not specifically called out in any of the Button Definition sections. Button Defaults can include the following property variables:

Text Position
Font Color
Indicator Position
Indicator On Color
Indicator Off Color

Text Position

The Text Position property in this section can define the default position that text will appear on a Button if the Text Position is not specifically defined in the actual Button Definition itself. The options for this include None, TopLeft, TopCenter, TopRight, CenterLeft, Center, CenterRight, BottomLeft, BottomCenter, and BottomRight.

Some of the syntax examples for this Section are:

Text Position = None

Text Position = TopLeft

Text Position = Center

Text Position = BottomRight

Font Color

The Font Color property in this section can define the default font color for the Button text that will appear on a Button if the color is not specifically defined in the actual Button Definition itself. The Font color property is defined by three choices:

Font Color = Auto

Font Color = Black

Font Color = White

With the above syntax options, **Auto** will determine whether to use Black or White Text by default based on the Button Background color (Black for lighter colors and White for darker colors), or you can select a Default value of **Black** or **White**.

Indicator On Color

The Indicator On Color property in this section can define the default color for an Indicator Light that can be displayed on a Button.

Indicator On = 0,255,0

With the above definition the Red value is 0, the Green value is 255, and the Blue value is 0, resulting in a bright green Indicator Color when it is supposed to be ON. The minimum RGB value is 0 (color off) and the maximum RGB value is 255 (color full on). At the end of this document is a comprehensive list of common colors and their RGB values.

Indicator Off Color

The Indicator Off Color property in this section can also define the default color for an Indicator Light that can be displayed on a Button.

Indicator Off = 0,128,0

With the above definition the Red value is 0, the Green value is 128, and the Blue value is 0, resulting in a darker green Indicator Color when it is supposed to be OFF. The minimum RGB value is 0 (color off) and the maximum RGB value is 255 (color full on). At the end of this document is a comprehensive list of common colors and their RGB values.

Button Section

The individual **[Button]** Sections are the most complicated sections to configure in a Touch HMI INI file. They can contain a large list of property variables, and v12 Buttons now support a new set of options:

- **Primary Button Settings** - The main settings for a button. These follow the Touch HMI v10 and v11 format options.
- **Alternate Button Settings** - Allows a user to define a 2nd State Condition for a Button when it has been pressed. You toggle between the Primary and Alternate States by clicking on the button. You will need to fill in data for both conditions for this function to work.
- **Down Button Settings** - Allows the user to define the Look of the Button when the Button has been pressed and stays down. This can occur when you have a Button that has a definition that uses a **ProgramWait** Command or a **Pop-Up** Option.
- **Button Pop-Up Menu Options** - The use of Pop-Up Options can only be enabled from the Primary State Menu for a Button. This will enable the user to add one or more Option Buttons to the Main Button's Function, creating a drop-down list of options to choose from. This is a complex topic and will be described in more detail later in this document.

Each of these adds new property variables to the INI File so that each one can be configured differently.

Primary Button Settings

The main **Primary Button** Property variables are:

- Row in Parent
- Column in Parent
- Button Style (Rectangle or Circle)

The rest of the Primary Button Configuration Properties are:

Text

- Button Text
- Text Position
- Font Color
- Font Size Adjust

Color

- Button Color
- Gradient Color
- Outline Color
- Line Width

Indicator

- Indicator Position
- Indicator On Color
- Indicator Off Color
- Indicator State

Commands

There are 11 different commands that can be added to a Button. Multiple Command Options can be added to a single Button Definition. Since these Commands can be complex, they will be discussed in more detail later in this document.

Pop Up

Button Pop-Up Options are a new feature in Touch HMI v12 that enables the user to create a Drop-down List to select from on a Button. This new function can set a list of variables to make a single Button have a variety of functions based on the variables selected.

The Pop-Up Configuration Properties are:

- Pop-Up Title
- Pop-Up Color
- Pop-Up Width
- Pop-Up Height
- Pop-Up Count (new in v12.0.25.7, to enable long Pop-Up menus to have a scroll bar)
- An **[Add Option]** Button that is used to add Pop-up Option Buttons to a Panel Button.
- A Checkbox for: Show selection on button

*[NOTE: Pop-Up Buttons **CANNOT** be used to execute commands. Their sole purpose is to set a series of Arguments so that another Button can execute a command with that new set of variables. For example, a Pop-up Button can be used to set the Name of a Machine, the Path to the DNC Log for that machine, and the Ethernet Port number for that machine, before a DNC Command is executed.]*

Alternate Button Settings

The **Alternate Button** Property variables are:

- Alt.Text
- Alt.Text Position
- Alt.Color
- Alt.Gradient
- Alt.Outline
- Alt.Line Width
- Alt.Indicator Position
- Alt.Indicator State

Down Button Settings

The **Down Button** Property variables are:

- Down.Text
- Down.Position
- Down.Font Size
- Down.Color
- Down.Gradient
- Down.Outline
- Down.Line Width

Touch HMI Button Configurations

Simple buttons on a Touch HMI Panel may perform only one action. Advanced buttons can perform multiple actions with one mouse click on a button. Some examples of advanced button functions could be the following:

1. Log off the previous Part Number and Operation within Predator MDC
2. Log on the new Part Number, Operation Number, start Setup Time within Predator MDC
3. Run a CAM system's post-processor
4. Read a RFID Tag and trigger a task based on the contents of the tag
5. Send the CNC program and offsets via Predator DNC to a CNC machine
6. Display the work instructions for a Part within PDM
7. Send an E-Mail message
8. Update a Microsoft Excel spreadsheet to record a transaction
9. Turn on a light on the CNC Machine's Light Tower

All of the above actions could occur in the background via the different Predator applications (Predator DNC, MDC, and PDM) and all triggered by a single button click within Predator Touch HMI. Review the following simple Button examples:

[Button]

Row = 1
Column = 1
Text = Line Down
Color = 185,51,37
Ethernet = MSLD

[Button]

Row = 1
Column = 2
Text = Restart Line
Color = 166,166,166
Ethernet = ME

[Button]

Row = 2
Column = 1
Text = Send Queued File
Color = 231, 0, 51
Serial = LOAD

[Button]

Row = 2
Column = 2
Text = Send Tool Offsets\13\10to CNC
Color = 231, 0, 51
RFID=1
Ethernet = LOAD \RFID1.offset

[NOTE: The [Button] Header Name format has changed from the format used for Touch HMI v10 and v11 [Button#], to what we now use in Touch HMI v12 [Button]. All Button Names in Touch HMI v10 and v11 had a Number in them, such as [Button3], to determine the order of the Buttons in the Panel. With the new format for Touch HMI v12, which added Frames, this was discontinued. The examples shown above are all Touch HMI v12 examples.]

The simple MDC and DNC Button examples shown above are common applications for Predator Touch HMI. All of the button configuration is done within the appropriate INI file. Here is a review of the following button properties:

Row in Parent

The Row property is used to define the button's row position in its parent (Form or Frame). The minimum value is 1 and the maximum value depends on the number of rows defined in the Parent. If the button's row value is greater than the number of rows defined in the Parent, then the button will not be displayed. For example:

Row = 1

Column in Parent

The Column property is used to define the button's column position in its parent (Form or Frame). The minimum value is 1 and the maximum value depends on the number of columns defined in the Parent. If the button's column value is greater than the number of columns defined in the Parent, then the button will not be displayed. For example:

Column = 1

Button Style

The Style property is used to define the type of button to be used. The options are Rectangle or Circle. For example:

Style = Rectangle

Style = Circle

Text

The Text property is used to define the text that appears on the button. Multiple lines of text can be specified by using \13\10 (Carriage Return, Line Feed) variables within the text. For example:

Text = Line Down

Text = Tool Change\13\10or Tool Adjustments

Text = Manufacturing && Quality Hold

NOTE: If the "&" character is desired to be displayed within the button text, then enter it twice within the .INI file as shown above.

Text Position

The Text Position property is used to define the position of the text that appears on the button. The options are None, TopLeft, TopCenter, TopRight, CenterLeft, Center, CenterRight, BottomLeft, BottomCenter, and BottomRight.

For example:

Text Position = TopLeft

Text Position = BottomRight

Text Position = Center

Font Size

The Font Size property is used to define the size of the Font used for the button. The Font Size of each Button will be automatically set by the Touch HMI Application as a Panel is enlarged or made smaller by dragging the panel around on the screen. The Font Size setting helps to keep the Button Text readable by adjusting the Font Size up or down from its default size. The values for Font Size can be either a positive integer or a negative integer depending on whether you want the displayed Font Size to be larger or smaller than the calculated Font Size. For example:

Font Size = +4

Font Size = -2

Color

The Color property is used to define the foreground color of the button. The color property is defined by three RGB values. For example:

Color = 255, 128, 0

With the above definition the Red value is 255. The Green value is 128. The Blue value is 0. The minimum RGB value is 0 and the maximum RGB value is 255. At the end of this document is a comprehensive list of common colors and their RGB values.

Often it is a good idea to make the color of the buttons match a Predator application or our Predator corporate blue color. Refer to the Predator Application Color List later in this document.

It can also be a good idea to make the color of the buttons match Predator MDC's default colors. Refer to the Predator MDC Default Color List later in this document.

Gradient

The Gradient property is used to define a gradient color for a button. The Gradient property defines a 2nd color to be used to create a gradual color change from the top of the button to the bottom. The Button Color will start at the top of the button and the Gradient Color will be at the bottom of the button. The Gradient property is also defined by three RGB values. For example, the LOAD Button on the right has the following colors configured:



Color = 227,31,32 [which is a light RED]

Gradient = 255, 128, 0 [which is a darker shade of RED]

With the above definition, the Button Color Setting is 227,31,32, which has a RED value of 227, a GREEN value of 31, and a BLUE value of 32. The Gradient Color Setting has a RED value of 255, a GREEN value of 128, and a BLUE value of 0. The Gradient Color is a darker shade of Red, so the Button has a Light to Dark color shading from top to bottom. There are also some tips on how to select good looking gradient colors for buttons in the [Color Reference Section](#) of this manual.

Outline

The button Outline (color) property defines the color to be used for the outline of the Button. The color property is defined by three RGB values. For example:

Color = 0,0,0

With the above definition the Red value is 0, the Green value is 0, and the Blue value is 0, resulting in a Black Outline Color. The minimum RGB value is 0 (black) and the maximum RGB value is 255 (white). At the end of this document is a comprehensive list of common colors and their RGB values.

Line Width

The button's Line Width property defines the width of the border outline around the Button. The Line Width property is defined in pixels. For example:

Line Width = 2

Indicator Position

The Indicator Position property defines the position to be used for the location of an Indicator Light on the Button. The options are None, UnderText, UnderTextLeft, UnderTextRight, TopLeft, TopRight, BottomLeft, BottomRight. For example:

Indicator Position = UnderText

Indicator Position = BottomRight

Indicator On Color

The Indicator On Color property defines the color of the Indicator to be used when the Indicator Light is turned ON on a Button. For example:

Indicator On Color = 14,255,144

With the above definition the Red value is 14, the Green value is 255, and the Blue value is 144, resulting in a bright green color. The minimum RGB value is 0 (black) and the maximum RGB value is 255 (white). At the end of this document is a comprehensive list of common colors and their RGB values.

Indicator Off Color

The Indicator Off Color property defines the color of the Indicator to be used when the Indicator Light is turned OFF on a Button. For example:

Indicator Off Color = 0,60,0

With the above definition the Red value is 0, the Green value is 60, and the Blue value is 0, resulting in a dark green color. The minimum RGB value is 0 (black) and the maximum RGB value is 255 (white). At the end of this document is a comprehensive list of common colors and their RGB values.

Indicator State

The Indicator State property defines the condition of the Indicator Light when the Button is in its default condition. For example:

Indicator State = Default

Indicator State = Off

Indicator State = On

The Indicator State = On condition indicates that the Indicator Light on the Button will be showing the Indicator On Color by default. Pressing the Button so that it is in a "Down" or "Alternate" condition will display the opposite color on the Button (Indicator Off).

The Indicator State = Off condition indicates that the Indicator Light on the Button will be showing the Indicator Off Color by default. Pressing the Button so that it is in a "Down" or "Alternate" condition will display the opposite color on the Button (Indicator On).

[NOTE: Both the *PTouchDark* and *PTouchLight* example INI files included with Touch HMI have examples of the use of Indicator Lights. You can review these INI files for a better understanding on how they work.]

Button Commands

The main function of a Button in a Touch HMI Panel is to execute a command or set of commands.

There are eleven main Commands that can be placed on a Button. They include:

- Ethernet
- Serial
- RFID
- Program
- Copy
- Rename
- Delete
- ReloadINI
- Delay
- Password
- Exit

Ethernet

The Ethernet command is used to define an action to send an Ethernet text message to an IP Address and Port Number. An Ethernet text message is typically a Predator DNC or MDC command. For a list of default Predator DNC and MDC commands review the Default Command section of this document. Multiple lines can be specified by using the **\13\10** (Carriage Return, Line Feed) variables within the Ethernet property. Here are some examples of the Ethernet Command:

Ethernet = SS

Ethernet = CE\13\10SE

Ethernet=LOAD \Prompt("Type or bar code scan a file to LOAD into the CNC",)

Ethernet=PS \LastPrompt

In the first example the MDC command **SS** is sent to an IP Address and Port Number for **Setup Start**. In the second example, the MDC commands **CE** and **SE** are sent to an IP Address and Port Number with CR and LF characters sent in between. This will process the **Cycle End** and **Setup End** as two separate commands. The third example is used to send a LOAD command to Predator DNC and prompt for a file selection. The last example sends a **Part Start** command to MDC and uses the data from the last Prompt for the Part Number.

[NOTE: Multiple Ethernet actions per button is supported. Combinations of Ethernet, Program, Serial, and Exit actions per button are also supported.]

Serial

The Serial command is used to define an action to send a Serial or RS-232 text message via the COM Port. The serial text message is typically a Predator DNC or MDC command. For a list of default Predator DNC and MDC commands review the Default Command section of this document. Multiple lines can be specified by using **\13\10** (Carriage Return, Line Feed codes) within the Serial property. For example:

Serial = SS

Serial = CE\13\10SE

11092 SW Allen Blvd • Beaverton, OR 97005-4821 • Phone 1 503-292-7151 • www.predator-software.com

In the first example the MDC command **SS** is sent via the PC's Serial Port for **Setup Start**. In the second example, the MDC commands **CE** and **SE** are sent via the PC's Serial Port with CR and LF characters sent in between, for **Cycle End** and **Setup End**.

[NOTE: Multiple Serial actions per button is supported. Combinations of Ethernet, Program, Serial and Exit actions per button are also supported. Serial support is included when it is preferred or when Ethernet is not available. In general, most Touch HMI applications should use Ethernet commands and not Serial actions if possible.]

RFID

The RFID command is used to define an action to read the contents of an RFID chip when the button is pushed. The value indicates which reader head to use. For most applications there is only one reader head. For example:

RFID = 1

RFID = 2

After an RFID tag has been successfully read, the following variables will contain the contents of the RFID tag.

\RFID1

\RFID2

\RFID3

\RFID4

\RFID5

\RFID6

The above variables are line based. So **\RFID1** is the contents of the first line within the tag contents and **\RFID2** is the contents of the second line. RFID tags can be blank and can also contain less than six lines. When this is the case the **\RFID1** through **\RFID6** variables will return an empty string. Should there be an error reading a RFID tag the **\RFID1** through **\RFID6** variables will return an empty string.

*[NOTE: If Predator Touch HMI can't find the RFID controller or a tag is not in range of the reader, then an error dialog will be displayed and the button will stop processing, and any commands after **RFID = x** will be ignored. **\RFID1** through **\RFID6** variables are typically used within an Ethernet, Serial, Program, or ProgramWait action.]*

Program

The program command is used to define an action that launches an application when the button is pushed. For example:

Program = c:\program files (x86)\predator software\mdc 12.0\pmdc.exe

The Program command can be used in combination with the Arguments property.

Program = \OSFileOpen("Select *.EXE to Open","C:\Predator Software\Touch HMI 12.0\")

Arguments=-properties

The example above is used in the default **PTouchHMI.ini** file shipped with Touch HMI v12 to enable a user to open a dialog box and select any other existing PTouchHMI.EXE file and launch it with the "-properties" command line argument which opens the new Touch HMI Properties Editor so you can review and edit the INI file.

[NOTE: If Predator Touch HMI can't find the program listed in the command, then a warning "Program Not Found" will be displayed on the button instead of the button text.]

The Program command can also be used to call other Predator Touch HMI .EXE files. For example, the main Predator Touch HMI panel could have a button for each machine. In addition, machine specific HMIs could be used for a series of functions per machine. When this is desired, we recommend that all the Touch HMI .EXE and .INI files be in the same folder and the Program property would look like one of the following:

Program = .\machine1.exe

Program = .\machine2.exe

[NOTE: The need for one Touch HMI Panel to call a 2nd one has been greatly reduced by the introduction of the new Frame and Button Options functions in Touch HMI v12. This topic will be discussed in detail later in this manual]

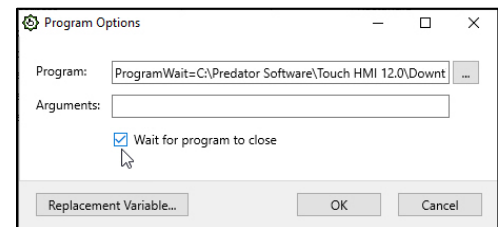
ProgramWait

The ProgramWait command also is used to define an action that launches an application when a button is pushed just like the Program command. The difference is that the Program Command will allow additional actions to be started after the initial program has been started. The ProgramWait command, however, will place the Touch HMI panel on hold until the initial program request has been completed and has been closed. For example:

ProgramWait = C:\Predator Software\Touch HMI 12.0\Downtime.exe

The ProgramWait command is also used in combination with the Arguments property, just like the Program property discussed above.

While the ProgramWait command is a separate command in the INI File, it does not have its own command option in the Touch HMI Editor. To enable a ProgramWait command, simply select a Program command from the drop-down list and then check the Checkbox for “Wait for program to close” to make the command a ProgramWait option. As show in the dialog on the right.



[NOTE: If Predator Touch HMI can't find the program listed in the command, then a warning “Program Not Found” will be displayed on the button instead of the button text.]

The ProgramWait command can be used to call other Predator Touch HMIs. For example, the main Predator Touch HMI could have a button for each machine. In addition, machine specific Touch HMI panels could be used for a series of functions per machine. When this is desired, we recommend that all the TouchHMI.EXE and .INI files be in the same folder and the ProgramWait property would look like one of the following:

ProgramWait = .\machine1.exe

ProgramWait = .\machine2.exe

[NOTE: The difference between the “Program” command and the “ProgramWait” command is that the Program command can run multiple instances of an application at the same time. The ProgramWait command limits the application to a single instance at a time. The current Program must end before another can be started.]

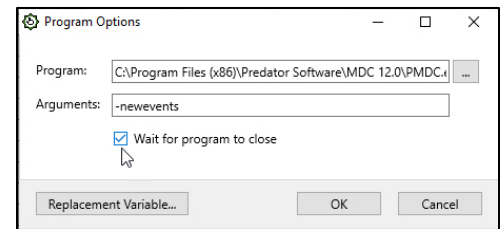
Arguments

The Arguments property is used to define command line arguments used when a program action is run. For example:

Arguments = -newevents

Arguments = -p 11005

The Arguments property is used in combination with the “Program” command and the “ProgramWait” command. Specific arguments may or may not be available depending on the program you are trying to launch. Refer to the target program’s online help or documentation for specific command line arguments that are available. In the example on the right, the PMDC.exe application has a **-newevents** command line argument that can be used. However, PMDC.exe does not support a **-p** command line argument for selecting an IP Address.



Sometimes an argument should be specified by the Touch HMI operator prior to launching the application. When this is the case Predator Touch HMI provides three different prompting options for that input. Review the **Prompts** section of this document for further details.

Copy

The Copy command can be used to copy a file from one location to another. An example of this would be:

Copy=PTouchHMI.exe,Assembly Line.exe

This example copies the existing PTouchHMI.exe file to another name so it can be used to run a 2nd Touch HMI Panel.

Rename

The Rename command can be used to rename an existing file and give it a new file name. An example of this would be:

Rename=PTouchHMI.exe,Assembly Line.exe

This example renames the existing PTouchHMI.exe file to Assembly Line.exe.

Delete

The Delete command can be used to delete an existing file. An example of this would be:

Delete=C:\Predator Software\Touch HMI 12.0\Assembly Line.exe

ReloadINI

The ReloadINI command is used to enable a Button to reload the current Touch HMI Panel (*.exe) with a new configuration (*.ini) file. This lets one initial TouchHMI.exe switch to any number of other preconfigured INI files.

*[NOTE: The **PTouchDark** and **PTouchLight** sample INI files show the use of this command. They both have a Button that swaps out the current INI file for the other, allowing the user to switch from PTouchLight To PTouchDark with a click of a button.]*

Delay

The Delay command is used to enter the length of time (in seconds) that you want to delay the action of a Button before executing any further commands on that Button.

Delay=20

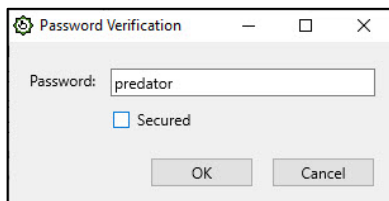
Password

The Password command is used to enable the user to require a Password that will need to be entered by the operator before any further commands can be executed on a multi command Button. This would normally be the first command listed in a Button Definition.

Password=predator

SecurePassword=z0Db7haDITQ

The examples above show an unsecured password (where the required password is displayed and can be read in the INI file, and a secured password where the value is encrypted when the password is entered in the Touch HMI Editor dialog box shown below.



The difference between these two examples in the Touch HMI Editor is that the Secured Checkbox is checked in the Dialog Box when the Password is entered in the command.

Exit

The Exit command is used to define the way the Predator Touch HMI Panel will exit or close. This can be done via a dedicated "Close" button, or at the end of an advanced button configuration. Predator Touch HMI supports two different styles of exiting. For example:

Exit = Silent

Exit = Confirm

The Silent style closes the Predator Touch HMI Panel immediately without any other user intervention. The Confirm style prompts the user for confirmation dialog box prior to closing the Predator Touch HMI Panel.

Prompts

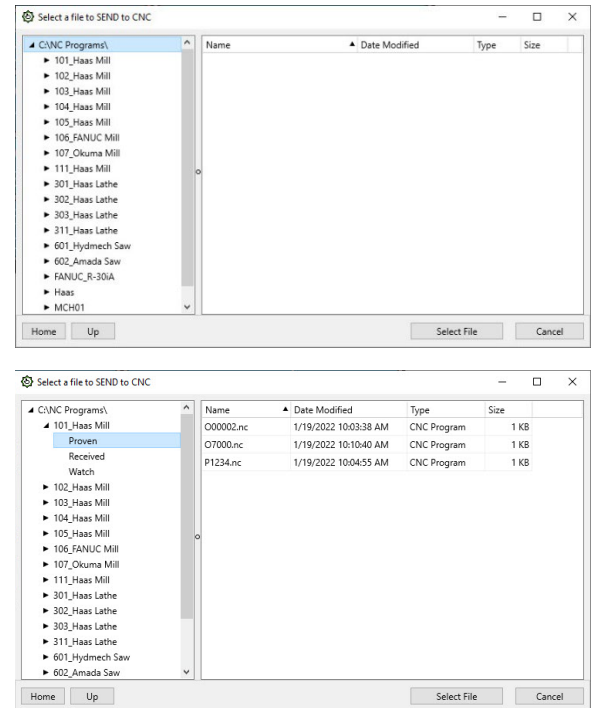
Predator Touch HMI Panels can optionally prompt for simple input such as machine numbers, tool numbers, and filenames prior to executing the button's commands. This advanced feature can reduce the number of buttons by relying on the HMI operator for simple input. There are two different types of Prompts that can be used.

Use of the “\File” Prompts

Predator Touch HMI supports the following case-sensitive prompt functions. The first six examples below use the **internal Predator File Dialog** (shown on the right) built into Touch HMI. These Predator File Dialog Boxes do not allow the user to browse above the “optional default folder” location specified in the command.

- **\Prompt**("prompt message", "optional default value")
- **\FileNameOpen**("titlebar message", "optional default folder")
- **\FileOpen**("titlebar message", "optional default folder", optional extension filters)
- **\FileNameSave**("titlebar message", "optional default folder")
- **\FileSave**("titlebar message", "optional default folder", optional extension filters)
- **\Folder**("titlebar message", "optional default folder")

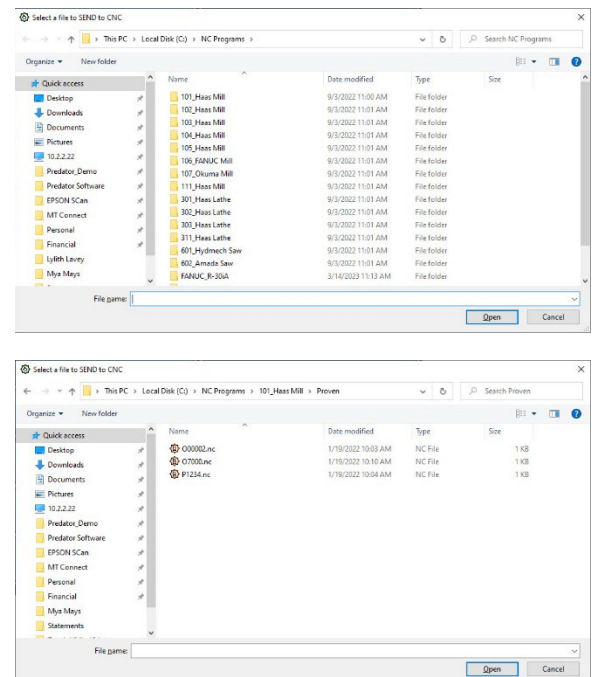
Once the Dialog Box opens, you can move around in the tree on the left side to select any root or sub-folder displayed. After you make your folder selection, you will see a listing of the files in that folder on the right side. This dialog box will allow you to sort the listings based on the File Name, Date Modified, File Type and Size.



Use of the “\OSFile” Prompts

These additional five prompt options will use the **default Windows File Dialog** built into the Windows Operating System that is currently running on the PC where the Touch HMI Panel is installed. These Windows File Dialog Boxes will allow the user to use the full features allowed by the Windows O/S to browse and search for a filename above and below the starting folder named as the “optional default folder” in the command. This means that the user will have full access to the full range of files, folders, and network access that the user has access to on his or her PC.

- **\OSFileNameOpen**("titlebar message", "optional default folder")
- **\OSFileOpen**("titlebar message", "optional default folder", optional extension filters)
- **\OSFileNameSave**("titlebar message", "optional default folder")



- **\OSFileSave**("titlebar message",optional default folder",optional extension filters)
- **\OSFolder**("titlebar message", "optional default folder")

\FileNameOpen (**\OSFileNameOpen**) and **\FileNameSave** (**\OSFileNameSave**) only return the filename.

\FileOpen (**\OSFileOpen**) and **\FileSave** (**\OSFileSave**) all return the filename with the complete path.

\Folder and **\OSFolder** returns the selected folder name.

One or more extension filters are supported by using a semi colon as a separator. For example:

\OSFileOpen("Select a CNC program",c:\my CNC programs\fanuc",.NC;.CNC)

In the above example, files with the extensions .NC and .CNC are displayed within the defined folder and sub folders.

[NOTE: Due to the complexity that can be created with these File Options, Predator Software has included a Demo INI file that shows the functionality for all of these options. That file is the **PTouchFileOptions.ini**.

All of these prompts can be used within an INI file and multiple prompts can be added to a single button. Refer to the following example:

[Button]

Row=1

Column=1

Text=LOAD

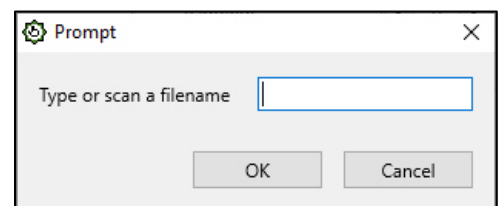
Color=210,105,30

Program=c:\program files (x86)\predator software\dnc 12.0\premove.exe

Arguments=-i 127.0.0.1 -p 11001 -s "LOAD **\Prompt**("Type or scan a filename",)"

So, prior to executing the Predator "premove.exe" application, Predator Touch HMI will display a simple prompt dialog box. Entering data and clicking the **[OK]** button on the prompt dialog box will run the "premove.exe" application with the entered data as input. The data entered in the Prompt Dialog Box will be sent to the PC with the IP Address and Port Number shown in the Button Config. Along with a LOAD command.

Clicking the **[Cancel]** button on the prompt dialog box will cancel the current action and any remaining button actions.



Predator Touch HMI

In the following example, the **Predator File Save** dialog will be displayed.

[Button]

Row=1

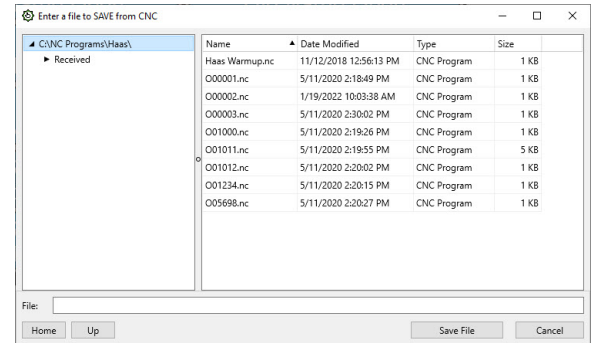
Column=2

Text=SAVE

Color=64,128,128

Ethernet=SAVE **FileSave**("Select a file to SAVE", " C:\NC Programs\Haas\")

You will notice in the image on the right that there is no way to browse above the default folder level of "C:\NC Programs\Haas" or anywhere else on the PC or Network with this Predator File Save Prompt.



In this example, the **Window's File Save** dialog will be displayed. This dialog will vary slightly depending on what version of Windows is installed on the PC.

[Button]

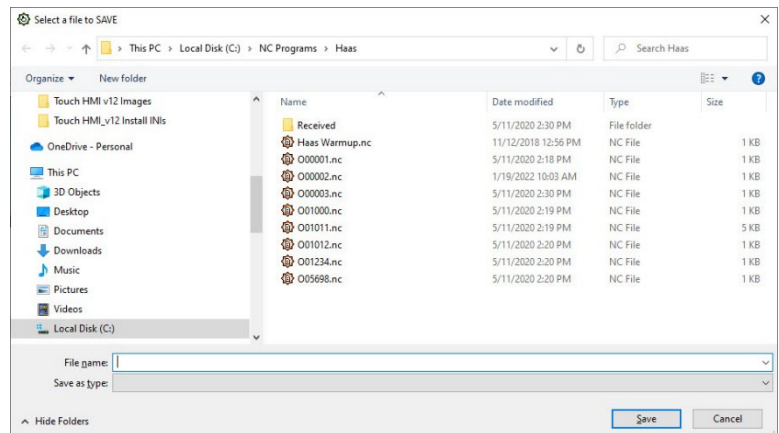
Row=1

Column=2

Text=SAVE

Color=64,128,128

Ethernet=SAVE **IOSFileSave**("Select a file to SAVE", " C:\NC Programs\Haas\")



With this example you can see that it is possible to browse anywhere on the PC or Network that your Windows User Account allows.

These two different styles of File Selection Dialog Boxes within Touch HMI give the user greater freedom to design the exact style and file accessibility that is desired for the Touch HMI Panel on the Shop Floor.

Predator Touch HMI



In this example, a Folder dialog picker is displayed:

[Button]

Row=1

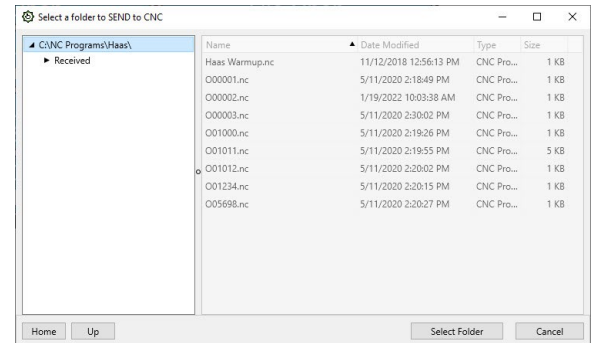
Column=3

Text=Send Folder to Machine

Color=64,128,64

Ethernet=LOAD \Folder("Select a folder to send","C:\NC Programs\Haas\")

This command would allow the user to select an entire Folder and its contents to be sent to a machine or another location, but you will notice that the user cannot select an individual file as they are all greyed out. Only a Folder can be selected with this Prompt.



Use of the “\Last” Prompts

Sometimes the results of one of these prompt variables needs to be used a second or third time. When this is the case use the following case-sensitive prompt functions:

\LastPrompt

\LastFileOpen

\LastFileSave

Refer to the following example:

[Button]

Row=2

Column=2

Text=SAVE

Color=64,128,128

Ethernet=LOAD \Prompt("Type or scan a part number",)

Ethernet=PS \LastPrompt

Ethernet=SS

All prompt functions can be used within the following button actions:

Arguments=

Ethernet=

Serial=

Prompting provides a lot of flexibility when creating Button actions. Common uses would be for selecting filenames for DNC, specifying an “O” number program on a CNC control, specifying a tool number for tool break restarts, or prompting for a bar code scan. When multiple prompts per command are specified, they are processed in order from left to right.

[NOTE: If the “\Last” style prompt commands are used prior to the use of any of the standard prompt commands, then the Prompt data will be empty. When multiple prompt variables are used the last style variable only contain the last prompt’s values.]

Button Examples

Predator Touch HMI Panels can contain as many buttons as you want. Refer to the following list of button examples:

[Button]

Row=1
Column=1
Style=Rectangle
Text=Setup Start
Font Color=0,0,0
Color=255,204,0
Outline=0,0,0
Line Width=2
Ethernet = SS
Pop-Up Button Text=False



Setup Start

[Button]

Row = 1
Column = 2
Style=Rectangle
Text=MDC Events
Font Color=255,255,255
Color=75,16,111
Outline=0,0,0
Line Width=2
Program=c:\program files (x86)\predator software\mdc 12.0\pmdc.exe
Arguments=-user Administrator -password predator -newevents
Pop-Up Button Text=False



MDC Events

[Button]

Row = 2
Column = 1
Style=Rectangle
Text=View OEE Status
Font Color=255,255,255
Color=0,0,255
Outline=0,0,0
Line Width=2
Program=c:\program files (x86)\predator software\mdc 12.0\pmdc.exe
Arguments=-user Administrator -password predator -oeestatus
Pop-Up Button Text=False



View OEE Status

[Button]

Row = 2
Column = 2
Text = View Machines
Font Color=255,255,255
Color=0,0,255
Outline=0,0,0
Line Width=2
Color = 185,51,37
Program = c:\program files (x86)\predator software\mdc 12.0\pmdc.exe
Arguments=-user Administrator -password predator -machinestatus
Pop-Up Button Text=False

View Machines

[Button]

Row = 3
Column = 1
Style=Rectangle
Text=Check in Tooling
Font Color=255,255,255
Color=156,19,46
Outline=0,0,0
Line Width=2
Program=c:\program files (x86)\predator software\tracker 12.0\ptracker.exe
Arguments=-user Administrator -password predator -checkin
Pop-Up Button Text=False

Check in Tooling

[Button]

Row = 3
Column = 2
Style=Rectangle
Text=Check Out Tooling
Font Color=255,255,255
Color=156,19,46
Outline=0,0,0
Line Width=2
Program=c:\program files (x86)\predator software\tracker 12.0\ptracker.exe
Arguments=-user Administrator -password predator -checkout
Pop-Up Button Text=False

Check Out Tooling

[Button]

Row = 4
Column = 1
Text = LOAD PART123
Color = 185,51,37
Ethernet = LOAD \\server1\c\predator software\cnc programs\machine1\part123.left
Ethernet = LOAD \\server1\c\predator software\cnc programs\machine1\part123.right
Exit = Silent

LOAD PART 123

[Button]

Row = 4

Column = 2

Style=Rectangle

Text = LOAD PART456 OP20

Color = 119,147,60

Outline=0,0,0

Line Width=2

Ethernet = Ethernet = PS PART456\13\10OS 20\13\10SS

Serial = LOAD \\server1\c\predator software\cnc programs\machine1\part456.nc

Serial = LOAD \\server1\c\predator software\cnc programs\machine1\part456.offsets

Exit = Silent



[Button]

Row = 5

Column = 1

Text = Read RFID

Color = 185,51,37

RFID = 1

Ethernet = PS \RFID1\13\10OS \RFID2\13\10SS

Ethernet = LOAD \RFID1_\RFID2.nc

Ethernet = LOAD \RFID1_\RFID2.offsets

Exit = Silent

[Button]

Row = 5

Column = 2

Text = Close

Color = 185,51,37

Exit = Confirm



Touch HMI and the Predator DNC Server Terminator

Predator Touch HMI Panels are often used to output Ethernet text messages to Predator DNC via the **Server Terminator** in PDNC. The Server Terminator is responsible for listening for any communication over the network from an Ethernet Port, which can come from Predator Touch HMI Panels, one or more CNCs, robots, CMMs, test stands, and other shop floor equipment.

For a Touch HMI application, the Predator Server Terminator is typically used within an “Other” style command in Predator DNC and can be combined with any appropriate predator objects. For example:

A typical **Remote Start Command** in Predator DNC would contain:

- Predator MDC for SQL Server Link
- Remote Request Link
- Server Terminator

The Ethernet Port Number specified within the Server Terminator must match the Ethernet Port Number specified within Predator Touch HMI's .INI file or command line argument.

Touch HMI and the Predator DNC ComPort HSM Terminator

Predator Touch HMIs can also output Serial text messages to Predator DNC via the **ComPort HSM Terminator** in PDNC. The ComPort HSM Terminator is responsible for listening for any RS-232 communication over a COM Port from Predator Touch HMI Panels, one or more CNCs, robots, CMMs, test stands, and other shop floor equipment.

For a Touch HMI application, the Predator ComPort HSM Terminator is typically used within an “Other” style command in Predator DNC and can be combined with any appropriate predator objects. For example:

A typical **Remote Start Command** would contain:

- Predator MDC for SQL Server Link
- Remote Request Link
- ComPort HSM Terminator

The COM Port Number specified within the ComPort HSM Terminator must be physically connected between the shop floor device and the PDNC Server via Predator Grizzly cable or a similar serial cable to the COM Port specified within Predator Touch HMI's .INI file or command line argument.

Predator Touch HMI's serial functions are compatible with the following software applications:

- Predator CNC Editor Express
- Predator CNC Editor
- Predator CNC Editor Enterprise
- Predator DNC Express/x
- Predator DNC/x
- Predator DNC Enterprise/x
- Predator MDC/x
- Predator MDC Enterprise/x
- Third party applications that support standard RS-232 operation

Predator Touch HMI's serial functions are compatible with the following serial hardware:

- Predator Express
- Predator Flex
- Predator Flex/2
- Predator Flex/N
- Predator Express/W
- Predator Flex/W
- Third party hardware that supports standard RS-232 operation

*[NOTE: Predator DNC's **ComPort STD Terminator**, **ComPort HSM Terminator**, and **ComPort EVT Terminator** can all be used with Predator Touch HMI.]*

Default Predator DNC and Predator MDC Command Reference

The following default Predator DNC and Predator MDC commands can be used with both the Serial and Ethernet button definitions within Predator Touch HMI.

Default Predator DNC Commands

LOAD filename

SAVE filename

LOAD1 filename

LOAD2 filename

SAVE1 filename

SAVE2 filename

[NOTE: Within Predator DNC, filenames can optionally include a drive letter, folders, partial folders, extensions, etc. Also, Predator DNC supports the use of DNC commands without a filename; the filename itself is optional.]

Default Predator MDC Commands

LS username	Logon Start
LE	Logoff
JS jobnumber	Job Start
JE	Job End
PS partnumber	Part Start
PE	Part End
OS operationnumber	Operation Start
OE	Operation End
CS	Cycle Start
CE	Cycle End
SS	Setup Start
SE	Setup End
TS	Teardown Start
TE	Teardown End
MS downtimenumbers	Machine Downtime Start
ME	Machine Downtime End
US downtimenumbers	User Downtime Start
UE	User Downtime End
GP quantity	Good Part
SP quantity.scrapreason	Scrap Part with optional scrap reason

These are the Standard Predator MDC Commands that are included in every installation of Predator MDC.

MDC Downtime Command Examples

MSBT	BT = Broken Tool
MSER	ER = Error Response
MSES	ES = Emergency Stop
MSFH	FH = Feed Hold
MSIT	IT = Machine Interrupted
MSMA	MA = Machine Alarm
MSMF	MF = Machine Failure
MSMR	MR = Machine Repair
MSNM	NM = No Response
MSNO	NO = No Operator
MSNR	NR = No Response
MSNT	NT = No Tooling
MSOS	OS = Optional Stop
MSUN	UN = Machine Unavailable
MSWC	WC = Waiting on Crane
MSWF	WF = Waiting on Fixtures
MSWG	WG = Waiting on Gages
MSWL	WL = Waiting on Forklift
MSWM	WM = Waiting on Material
MSWQ	WQ = Waiting on Quality
MSWT	WT = Waiting on Tooling
MSDV	DV = Development
MSNW	NW = No Scheduled Work
MSPM	PM = Preventative Maintenance
USSK	SK = Sick Time
USEP	EP = Employee PTO
USBR	BR = Breaks
USMT	MT = Meetings
USTR	TR = Training
USVC	VC = Vacation

The MDC Downtime Codes listed above are suggested Predator default commands. MDC Downtime Codes are configured at the discretion of the customer, and not all of these will be implemented at every customer. In addition to these lists, other custom Predator DNC and Predator MDC commands may be used, but these will vary per customer implementation. All Predator DNC and Predator MDC commands used within Predator Touch HMI are triggers and must have matching commands created within Predator DNC and Predator MDC to execute the actual tasks or functions.

For the latest information on Predator Touch HMI, check our web site at <http://www.predator-software.com>

Predator Touch HMI v12 Interface

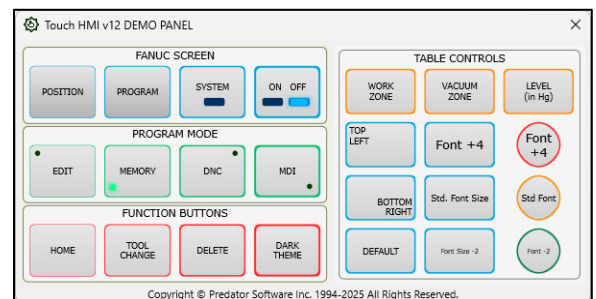
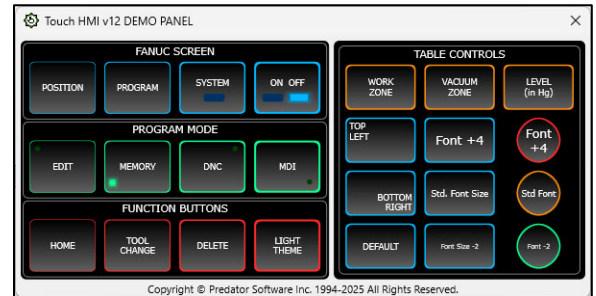
With the release of Predator Touch HMI v12, the product now has a new more modern looking user interface and a menu driven configuration utility, which enables the user to customize the look and feel of any Touch HMI Panel.

New Touch HMI Interface Features

The new Touch HMI v12 Interface introduces a few new features for configuring your Touch HMI Panels. A Touch HMI Panel can now have the following options for its configuration:

- A Touch HMI Panel can have a standard **Form** with one or more rows and columns of Buttons (similar to v10 and v11 versions).
- A Touch HMI Panel can now have one or more “**Frames**” defined in the Form that can be used to segregate a series of Buttons on the Panel. This allows one Touch HMI Panel to have several sections of buttons with different functions (as shown on the right).
- Each Frame and Button on the Panel can have its own shape, color, border sizing, and color gradient to help make the panels more user friendly. The text on each Frame and Button can also be adjusted for size, color, and orientation.

All of these new functions are still controlled by the Touch HMI Panel's associated *.INI file, which has made the structure and size of these INI files much larger and harder to edit manually. To make the configuration of the INI files easier to understand and build, we have created a new **Touch HMI Properties Editor**.



The Touch HMI Properties Editor

With the past versions of Touch HMI v10 and v11, a Touch HMI Panel could only be configured by manually editing the associated .INI file for each Touch HMI panel that you created with a simple Text Editor like Windows Notepad or the Predator CNC Editor. Now with Version 12 we can use the new **-properties** command line function to launch a Touch HMI Properties Editor that lets you configure and see the immediate results of your configuration changes on a live Touch HMI Panel view. You can access this utility in one of several ways:

- You can simply click on a new **TouchHMI.exe** file, and if no INI exists for it, the Properties Editor will open automatically, along with a Blank Touch HMI Panel.
- You can also create a Desktop Shortcut to an existing TouchHMI.exe file that you want to edit, and then edit the shortcut to add the argument **-properties** to the command line of the shortcut. This will always run the Touch HMI Properties Editor when you use this shortcut to open the Touch HMI Panel, as long as it is a v12 *.EXE.
- Predator Software has also provided a sample **PTouchHMI.ini** file with the deliverables that has buttons for each DEMO INI file that is part of the install, along with a Button configured to allow you to “**Edit Any PTouch.exe**”. To use these buttons, simply click on a Demo Button to launch that .INI file with the Properties Editor or select the **Edit Any PTouch.EXE** Button to open a dialog box to select an existing TouchHMI.EXE file and opening it with the Properties Editor, enabling you to edit and update the existing .INI file for that Panel.

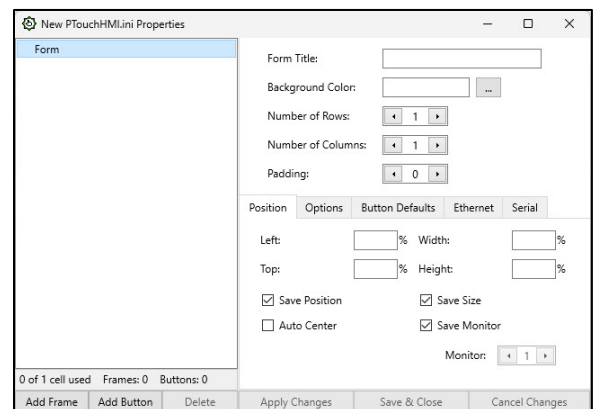
[NOTE: This new Touch HMI Properties Editor is embedded in the new v12 Touch.exe files, so it cannot be used with a v10 or v11 Touch.exe file, but you can use an older v10 or v11 *.INI file with a new v12 PTouch.exe, and then you can use the v12 Properties Editor to edit and update the older *.INI file to work with a newer v12 Touch HMI application.]

[NOTE: When the Touch HMI Properties Editor is open for an existing Touch HMI Panel, you can see the configuration and settings for the Frames and Buttons, but no actions or commands can be tested or executed while the Configuration Dialog Box is still open. To test you actions you will need to save your changes and close the Properties Editor Window first and then run your testing of the Touch HMI application.]

The initial Properties Editor Window for a new Touch HMI panel starts as a blank slate for creating your desired panel layout and functions. On the left side of the new Configuration Utility Window, you have the **Panel Layout Tree** that shows all the configured Form, Frames, and Buttons as you add them to your configuration. At the bottom of the window, you have a row of buttons to enable the user to:

- Add a Frame
- Add a Button
- Delete (a Frame or Button)
- Apply Changes
- Save & Close
- Cancel Changes

These buttons will become available for use or removed from use depending on the item you have selected or the changes you have made in the Panel Layout Tree area. For example, on a brand-new panel configuration the **[Add Frame]** and **[Add Button]** buttons are the only ones available (as shown on the right) as nothing has been added to the configuration yet. There is also an INI Info area at the Bottom of the Tree Pane to show you how many Frames and Buttons have been configured, and how many are in use.



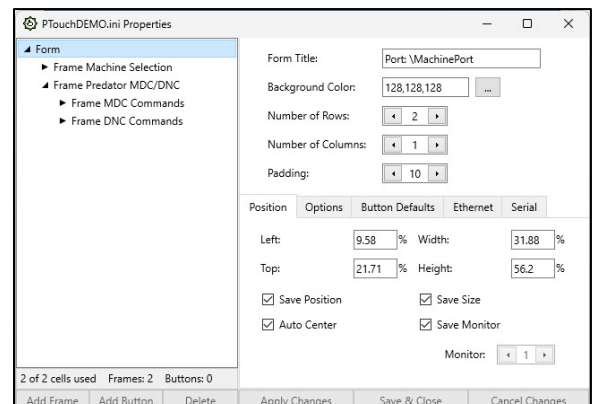
On the right side of the Properties Editor Window, you have settings and options for each item you have added in the Layout Tree. The right-side options will change depending on whether you have selected the **Main Form**, a **Frame**, or a **Button** in the Layout Tree.

The Main Form Configuration

The upper most level of a Touch HMI Panel Configuration is the **FORM**. The TouchHMI.ini file begins with the **[Form]** Header and ends the entire configuration of a TouchHMI.ini file with an **[EndForm]** Footer. All of the other configuration sections and details in the INI file fall between these two lines in the INI file.

The Configuration Screen options for the Form Section of the INI file are shown in the dialog box example on the right. It is broken into two sections. The top section includes fields for:

- Form Title
- Background Color (for the entire Panel)
- Number of Rows
- Number of Columns
- Padding



The **Form Title** sets Text that will be displayed at the top of the main Touch HMI Panel when it opens.

The **Background Color** field sets the background color of the Touch HMI Panel using the standard R,G,B format.

The **Number of Rows** field sets the total number of rows where Frames and Buttons can be created in the Panel.

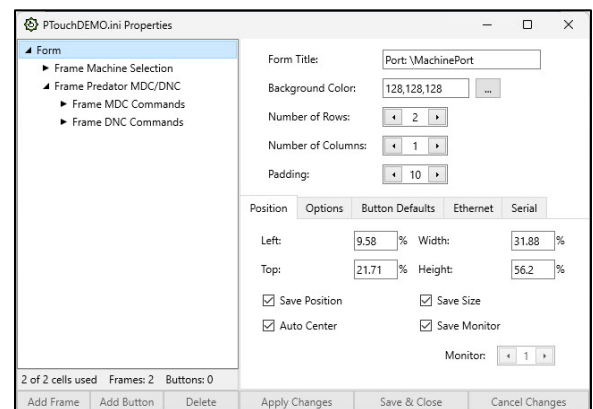
The **Number of Columns** field sets the total number of columns where Frames and Buttons can be created in the Panel.

The **Padding** field sets the size of the spacing between the Frames and Buttons in the Panel. This number is in screen pixels.

*[NOTE: If you open up the configuration of an existing TouchHMI.ini file with the Configuration Utility, you may find that your **[Add Frame]** and **[Add Button]** buttons are unavailable. This is due to the fact that you do not have any more room to add anything based on your current Row and Column Settings for the Form. If you want to add any Frames or Buttons to the current configuration you will have to increase the Row and/or Column settings in the Form Section for the Panel first.]*

The lower section of the Form Window includes Tabs for:

- **Position (this is new in v12.0.25.25)**
- Options (for the entire Panel)
- Button Defaults Section
- Ethernet Settings
- Serial (RS-232) Settings

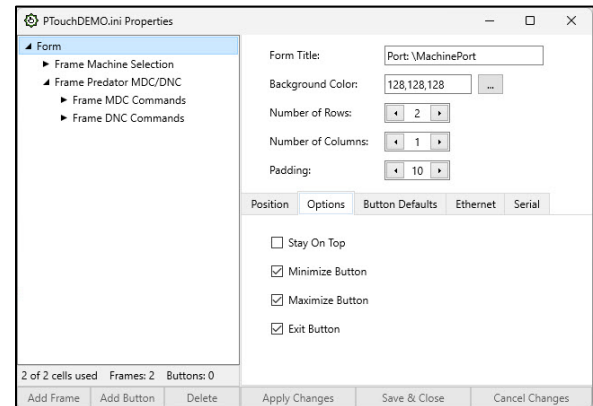


The **Position Tab** includes settings and check boxes for items such as:

- **Position for Left and Top** (position of the Touch HMI Panel as a % of screen size)
- **Size for Width and Height** (size of the Touch HMI Panel as a % of the screen size)
- **Save Position** (to save the new panel position when the dialog box is closed)
- **Save Size** (to save the new panel size when the dialog box is closed)
- **Auto Center** (turn on auto center to keep the panel centered in the current monitor. This will override the position settings)
- **Save Monitor** (saves the current Monitor Settings when the dialog box is closed)
- **Monitor** (allows the user to set the default Monitor location to be used. The Save Monitor Check Box must be unchecked to adjust this setting)

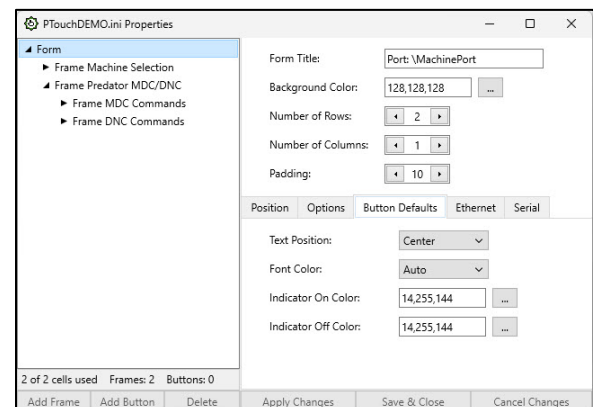
The **Options Tab** includes check boxes for items such as:

- **Stay on Top Option** (to keep the Touch HMI Panel on top of all other Windows on the Desktop)
- **Minimize Button** (enables the standard Windows Minimize Button in the upper right-hand corner of the Panel)
- **Maximize Button** (enables the standard Windows Maximize Button in the upper right-hand corner of the Panel)
- **Exit Button** (enables the standard Windows Exit Button in the upper right-hand corner of the Panel)



The **Button Defaults Tab** includes fields for:

- **Text Position** (to set the default text position for all buttons. This setting can be overridden for each button individually)
- **Font Color** (enables the user to set a default font color for the buttons. Options are Auto, Black, or White. Auto will select Black or White based on the background color.)
- **Indicator ON Color** (enables the user to set a default color for when the button is showing ON, for this example it is set to 14,255,144. You can either enter the RGB Value for the color if you know it or use the new **Touch HMI Color Picker** by clicking on the [...] Button to the right of the color fields.)
- **Indicator OFF Color** (enables the user to set a default color for when the button is showing OFF, for this example it is set to 0,60,0. You can either enter the RGB Value for the color if you know it or use the new Touch HMI Color Picker by clicking on the [...] Button to the right of the color fields.)

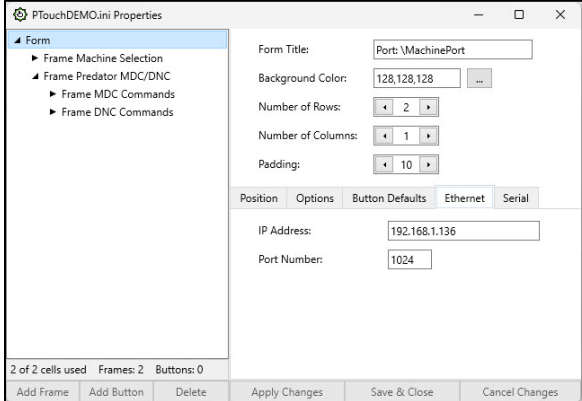


All of these Button Default settings can be overridden by other entries in the individual Frame and Button settings, but it allows for a Default Setting that will be used if these settings are not found in the individual button configurations.

Predator Touch HMI

The **Ethernet Tab** includes fields for Ethernet Settings such as:

- **IP Address** (this will be the default IP Address to be used for Ethernet commands in the Panel if a specific network IP Address is not specified for a Button command.)
- **Port Number** (this will be the default Ethernet Port Number to be used for Ethernet commands in the Panel if a specific Port Number is not specified for a Button command.)



PTouchDEMO.ini Properties

Form

- ▶ Frame Machine Selection
- ▶ Frame Predator MDC/DNC
- ▶ Frame MDC Commands
- ▶ Frame DNC Commands

Form Title: Port:\MachinePort

Background Color: 128,128,128

Number of Rows: 2

Number of Columns: 1

Padding: 10

Position Options Button Defaults Ethernet Serial

IP Address: 192.168.1.136

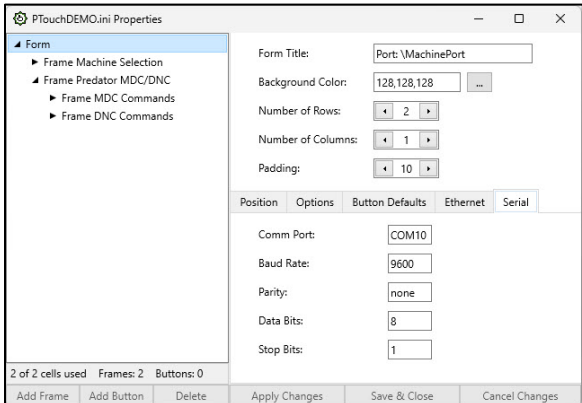
Port Number: 1024

2 of 2 cells used Frames: 2 Buttons: 0

Add Frame Add Button Delete Apply Changes Save & Close Cancel Changes

The **Serial Tab** includes fields for RS-232 Serial Settings such as:

- **Comm Port** (this will be the default Serial Comm Port to be used for Serial commands in the Panel if a specific Serial Comm Port Number is not specified for a Button command.)
- **Baud Rate** (this will be the default Serial Baud Rate to be used for Serial commands in the Panel if a specific Baud Rate is not specified for a Button command.)
- **Parity** (this will be the default Serial Parity setting to be used for Serial commands in the Panel if a specific Parity setting is not specified for a Button command.)
- **Data Bits** (this will be the default Serial setting for Data Bits to be used for Serial commands in the Panel if a specific Data Bits setting is not specified for a Button command.)
- **Stop Bits** (this will be the default Serial setting for Stop Bits to be used for Serial commands in the Panel if a specific Stop Bits setting is not specified for a Button command.)



PTouchDEMO.ini Properties

Form

- ▶ Frame Machine Selection
- ▶ Frame Predator MDC/DNC
- ▶ Frame MDC Commands
- ▶ Frame DNC Commands

Form Title: Port:\MachinePort

Background Color: 128,128,128

Number of Rows: 2

Number of Columns: 1

Padding: 10

Position Options Button Defaults Ethernet Serial

Comm Port: COM10

Baud Rate: 9600

Parity: none

Data Bits: 8

Stop Bits: 1

2 of 2 cells used Frames: 2 Buttons: 0

Add Frame Add Button Delete Apply Changes Save & Close Cancel Changes

All of these settings can be overridden by other entries in the individual button settings, but it allows for Default Settings that will be used if these settings are not found in the individual button configurations.

All of the entries described above will end up in either the **[Form]** or **[Button Defaults]** sections at the top of the INI file after the changes have been saved.

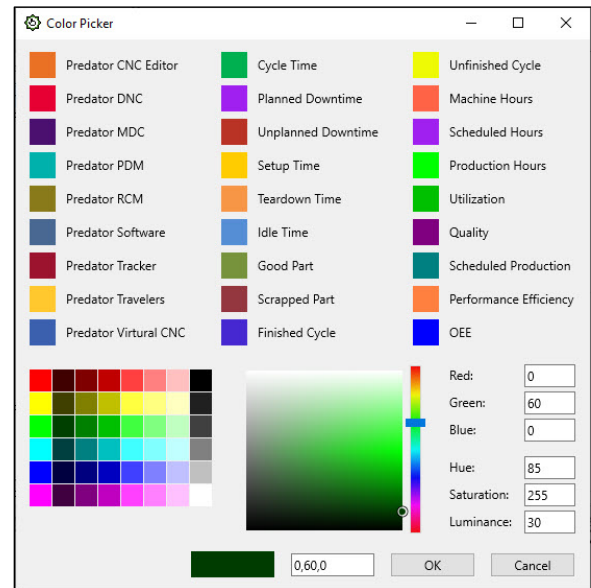
There is no specified order for the entries in these two sections within a Touch HMI INI file as they are read and processed together when the Touch HMI Panel first opens.

Touch HMI Color Picker Tool

The new **Touch HMI Color Picker Tool** described above is a new feature that allows the user to easily see and select a button color from one of several different methods:

- You can pick a Button color by selecting from one of the many Predator Default colors shown on the Tool's pallet.
- You can enter the R,B,G values directly if you know them.
- You can also use the alternate color definition method by selecting Hue, Saturation, Luminescence values to select a color.
- Or you can use the Color Selector Screen and Slider Bar to select any possible color that you may want to use.

This new tool gives the user a much better idea of the color selections available, with real-time visual feedback at the bottom of the window for your selection. There is more information on the use of this tool in the **Color Reference Section** of this manual.



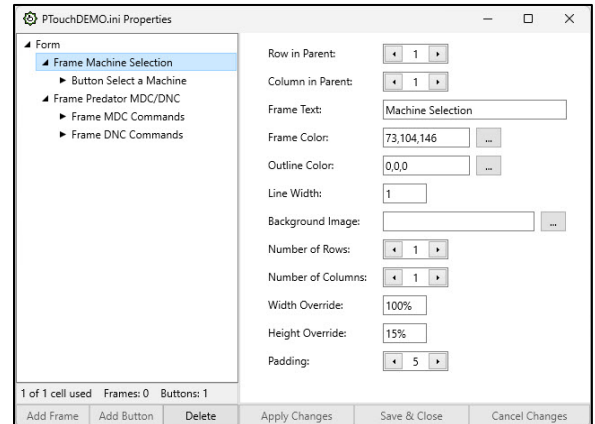
Predator Touch HMI

The Frame Configuration

The second level of configuration for a Touch HMI Panel is the **FRAME**. A Frame can be used to organize a series of Buttons together in the same manner that you may find them on a CNC Control Panel. **This is a new feature in Touch HMI v12.**

The **[Frame]** Header starts the definition of a Frame in the Touch HMI INI file, and it ends with a **[EndFrame]** Footer. All of the other configuration settings and details for this Frame fall between these two entries in the INI file.

Touch HMI v12 also supports the use of Frames within a Frame, so it may be possible to have one Frame start but not end, until other Frames have been started and ended in the INI file.



A nested Frame structure in the INI file would look similar to the following:

[Form]

[Frame] (Main Frame)

xxxx

[Frame] (FANUC Buttons)

xxxx

xxxx

[EndFrame]

[Frame] (TABLE CONTROLS Buttons)

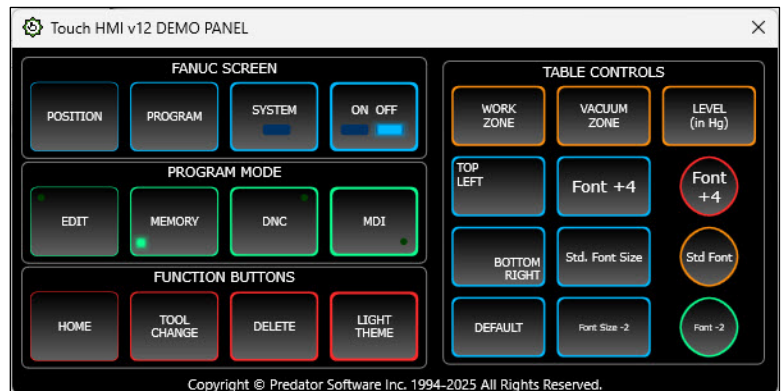
xxxx

xxxx

[EndFrame]

[EndFrame]

[EndForm]



(This new Frame feature allows for a configuration like the example shown above. Here we have a total of 6 Frames, with two Main Frames (a left and a right) in the Panel, and three smaller stacked Frames (FANUC Screen, Program Mode, and Function Buttons) contained inside of the left Main Frame and a single Frame (Table Controls) in the right Main Frame)

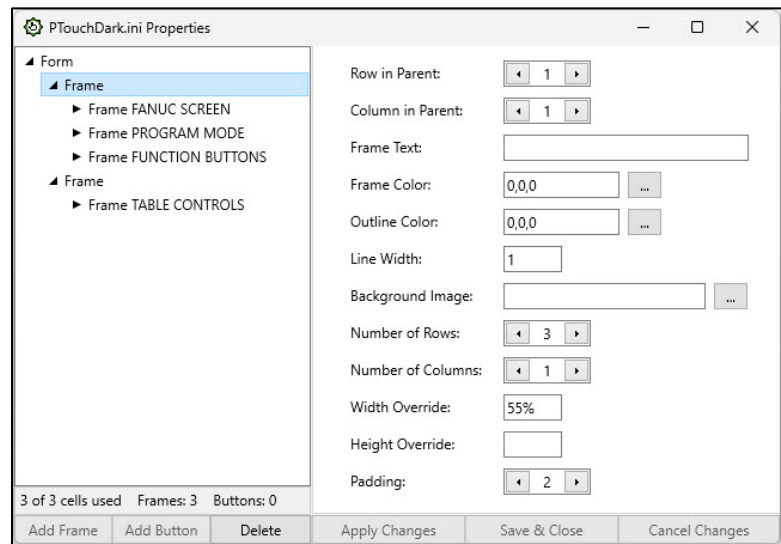
This new feature can enable the user to create some unique Touch HMI Panel designs.

The Configuration Options for the Frame Sections of the INI File are shown in the dialog box example below. Notice in this example we have the INI file configuration for the Touch HMI Panel image shown above, where we have the 6 Frames nested together.

The upper most Frame Definition Section (highlighted in blue below) is for the left-hand Main Frame, with the three smaller Frames nested below it. These are followed by the Definition Section for the right-hand Main Screen and its smaller Frame below it. The Tree Layout area helps to better visualize the final layout of the Frames in your Panel.

The Frame section includes fields for:

- Row in Parent
- Column in Parent
- Frame Text
- Frame Color
- Outline Color
- Line Width
- Background Image
- Number of Rows
- Number of Columns
- Width Override
- Height Override
- Padding



The **Row in Parent** field sets the row position of the Frame in its Parent (either the main Form or a Parent Frame)

The **Column in Parent** field sets the column position of the Frame in its Parent (either the main Form or a Parent Frame)

The **Frame Text** allows the user to give the Frame a name that will be displayed at the top of the Frame in the Panel

The **Frame Color** field allows the user to set the color of the Frame in the standard R,G,B format

The **Outline Color** field allows the user to set the color of the Frame's Outline in the standard R,G,B format

The **Line Width** field allows the user to set the width of the Frame's Outline in Pixels

The **Background Image** field allows the user to add a background image to the Frame in a **.bmp** format

The **Number of Rows** field sets the value for how many rows will be allowed inside of the Frame

The **Number of Columns** field sets the value for how many columns will be allowed inside of the Frame

The **Width Override** field sets the Percentage (%) of the Parent Width to be used by this Frame

The **Height Override** field sets the Percentage (%) of the Parent Height to be used by this Frame

The **Padding** field sets the amount of spacing between Buttons (or other Frames) contained inside of the Frame in pixels.

Predator Software includes a number of examples of TouchHMI.ini files (including the ones shown in this section) with the deliverables for the Touch HMI Product. You can review these examples and see how they are used in an actual Touch HMI Panel configuration.

Predator Touch HMI

The Button Configuration

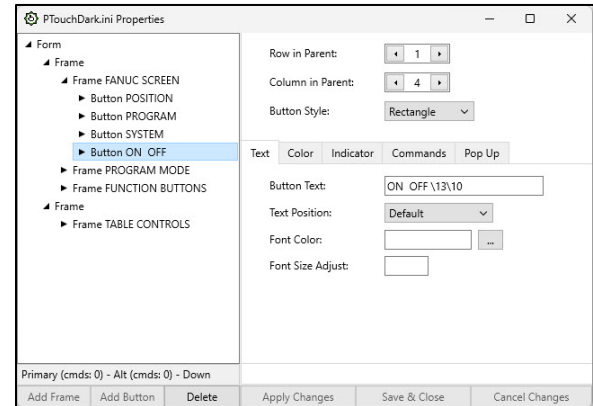
The Configuration Screen options for the Button Section of the INI File have had some major improvements in Version 12 and are shown in the dialog box example on the right.

Notice in this example we have the INI file configuration for the Touch HMI Panel image shown earlier where we have the nested Frames. This is the configuration for one of the Buttons **[ON OFF]** in the 1st Nested Frame, labeled “FANUC SCREEN”.

Like the Form Definition Section, the Button Definition Section is also split into an upper and lower area. The upper area of the Button Definition Section sets the Button Position in the Parent (Form or Frame) and the Button Style.

These entries are:

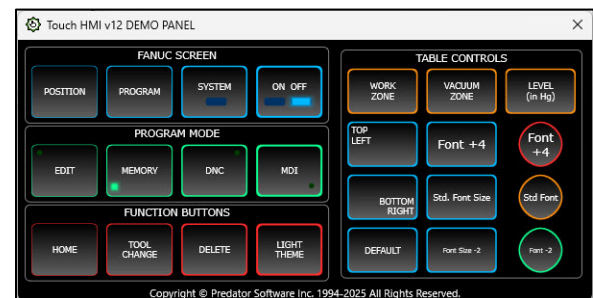
- Row in Parent (field)
- Column in Parent (field)
- Button Style (drop-down list)



The **Row in Parent** field sets the row position of the Button in its Parent (either the main Form or a Parent Frame)

The **Column in Parent** field sets the column position of the Button in its Parent (either the main Form or a Parent Frame)

The **Button Style** drop-down list allows the user to select a Button style from either **Rectangle** or **Circle**



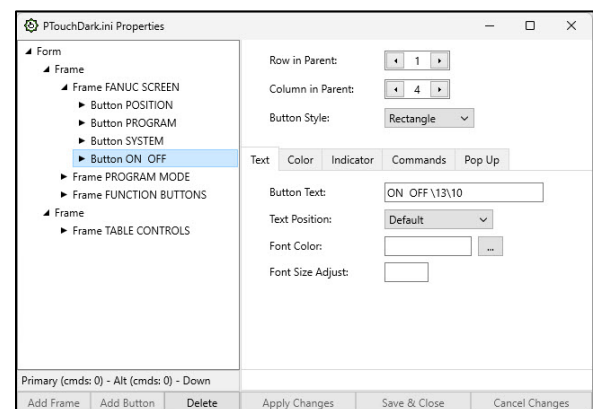
(The Demo Touch HMI Panel shown above illustrates how these settings can be used to change the “Look and Feel” of the Buttons on a Panel)

The lower area of the Button Definition Section has a number of Tabs that help to set the configuration of the Button.

The Tabs include:

- Text
- Color
- Indicator
- Commands
- Pop Up

Each Tab has its own unique configuration options for the Button.



Another new feature in Touch HMI Version 12 is that Buttons now have 3 New **Optional States** which are shown in the Tree Pane and can be configured. These three states are:

- **Primary** - Allows the user to configure the Initial State of a Button. The display for Primary is the same as the display if you click on the Button in the tree.

[NOTE: When selecting a Button Node in the tree you will automatically see the Primary State Options.]

- **Alternate** - Allows a user to define a 2nd State Condition for a Button when it has been pressed. You toggle between the Primary and Alternate States by clicking on the button. You will need to fill in data for both conditions for this function to work.
- **Down** - Allows the user to define the Look of the Button when the Button has been pressed and stays down. This can occur when you have a Button that has a definition that uses a **ProgramWait** Command or a **Pop-Up** Option.

There is also an optional 4th Button State:

- **Pop Up** (optional State only for Primary) - The Pop-Up Option can only be enabled from the Primary State Menu for a Button. This will enable the user to add one or more Option Buttons to the Main Button's Functions. This can be used to create a Sub-Menu of options so that one Button can have multiple options to choose from. Once an Option has been selected, the results will be displayed on the original Button. This is a complex topic and will be described in detail later in this document.

Now that these Button States have been defined, we can explore the configuration screens for each one and view the results they can offer.

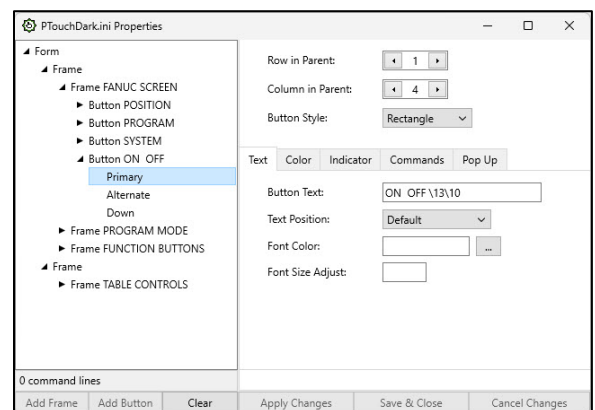
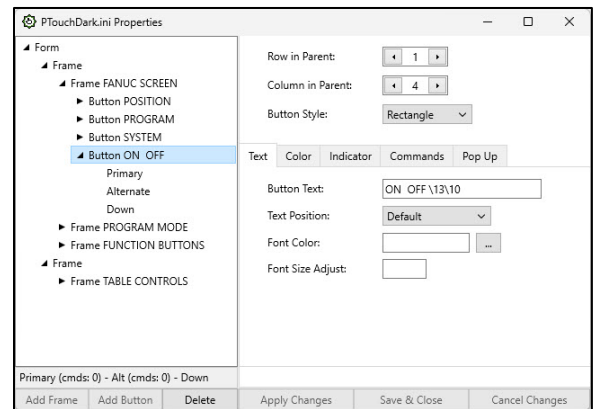
Primary Button State

The Primary Button State has 5 Tabs for its configuration. These include:

- Text
- Color
- Indicator
- Commands
- Pop Up (optional)

The **Text** Tab has options for:

- **Button Text** - Allows the user to enter the text to be displayed on a Button. You can also use the **\13\10** (CR/LF) string in this field to enter a 2nd line of text on a Button (i.e. TOOL\13\10CHANGE would put the word TOOL over CHANGE).
- **Text Position** - This is a Drop-down List that allows the user to select the text position on the Button from **Default**, **TopLeft**, **TopCenter**, **TopRight**, **CenterLeft**, **Center**, **CenterRight**, **BottomLeft**, **BottomCenter**, or **BottomRight**.

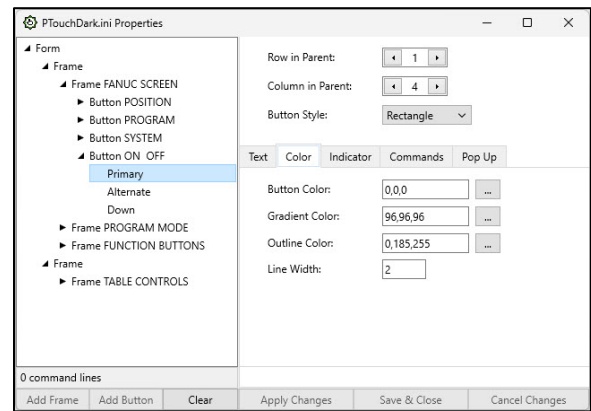


- **Font Color** - This field lets the user select a color for the Text Font on the Button. This field can use the **Touch HMI Color Picker Tool** by selecting the [...] Button to the right of the field.
- **Font Size Adjust** - This field allows the user to adjust the Font Size on a Button using a +n or -n number, which will adjust the Button's auto-calculated Font Size either up or down by that amount. For example, if the Button's current Font Size is calculated to be a 16pt font, then a value of +2 or -2 would make the new Font Size on the Button to be either 18pt or 14pt, respectively.

[NOTE: The Text Font Size for Buttons and Labels are automatically calculated by the Touch HMI Program so that they fit cleanly on the Buttons and Frames and still allow them to change if the Touch HMI Panel is dragged up or down in size.]

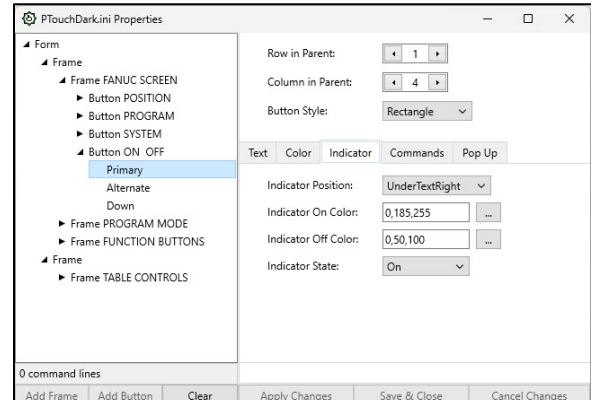
The 2nd Tab is the **Color** Tab, which has the following options:

- **Button Color** - Allows the user to set the color of the Button using the standard R,G,B color format. This field can use the **Touch HMI Color Picker Tool** by selecting the [...] Button to the right of the field.
- **Gradient Color** - This field allows the user to select a Gradient color to be used on a Button. The Gradient Color will start at the top of the Button and be gradually faded into the Button Color entered above at the bottom of the Button. This field can use the **Touch HMI Color Picker Tool** by selecting the [...] Button to the right of the field.
- **Outline Color** - This field allows the user to select a color for the outline of the Button using the standard R,G,B color format. It also can use the **Touch HMI Color Picker Tool** by selecting the [...] Button to the right of the field.
- **Line Width** - This field sets the width of the Button's outline in pixels, using the outline color selected above.



The 3rd Tab is the **Indicator** Tab, which is used to add an Indicator Light to a Button. It has the following options:

- **Indicator Position** - This field is a drop-down list of positions for the Light on the Button, including **Default**, **UnderText**, **UnderTextLeft**, **UnderTextRight**, **TopLeft**, **TopRight**, **BottomLeft**, and **Bottom Right**.
- **Indicator On Color** - This field allows the user to select a color to be used when the Indicator Light is ON on a Button. This field can use the **Touch HMI Color Picker Tool** by selecting the [...] Button to the right of the field.
- **Indicator Off Color** - This field allows the user to select a color to be used when the Indicator Light is OFF on a Button. This field can use the **Touch HMI Color Picker Tool** by selecting the [...] Button to the right of the field.
- **Indicator State** - This field allows the user to select what state the Indicator Light will be in when the Button is in the selected condition. The options are Default, OFF or ON.



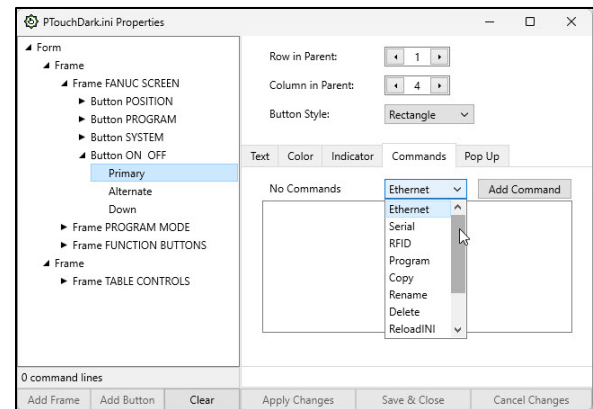
Predator Touch HMI

The 4th Tab is the **Commands** Tab. This is where we will configure one or more commands to be used on the Button.

This Tab does not have a set of default options, but rather an Entry Box with a Drop-down List of Command Types to select from, and then a Button to “Add a Command” from the selected type with configuration screens for each. This helps the user fill in the proper syntax to enable the command to work.

There are eleven Command types. They are listed and described below:

- Ethernet
- Serial
- RFID
- Program
- Copy
- Rename
- Delete
- ReloadINI
- Delay
- Password
- Exit



The **Ethernet** Command Dialog Box has an entry field where you can enter the Text String that you want the Ethernet Command to send out over the configured IP Address and Ethernet Port.

It also has a **[Replacement Variable]** Button that will allow you to select from the list of supported variables in Touch HMI and allow you to copy and paste them into your command (these will be discussed in more detail at the end of this section).



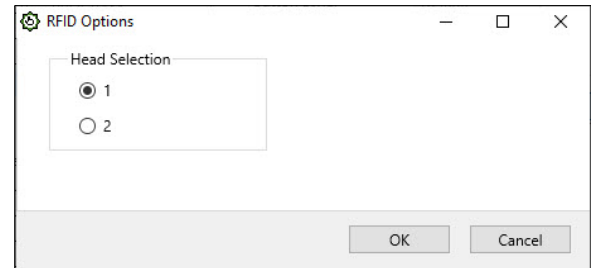
The **Serial** Command Dialog Box also has an entry field where you can enter the Text String that you want the Serial Command to send out over the configured RS-232 Serial COM Port.

It also has a **[Replacement Variable]** Button that will allow you to select from the list of supported variables in Touch HMI and allow you to copy and paste them into your command (these will be discussed in more detail at the end of this section).



Predator Touch HMI

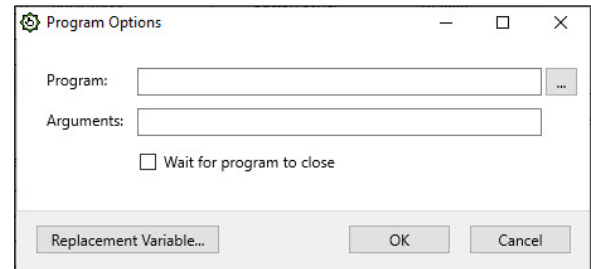
The **RFID** Command Dialog Box is used to select which Reader Head to use on a RFID Reader Controller. The options are radio buttons to select between Head 1 or Head 2.



The **Program** Command Dialog Box has two entry fields and a check box, where you can configure the program that you want the command to launch.

The first field is the Program field where you can either type in or select a program and its path using the [...] Button.

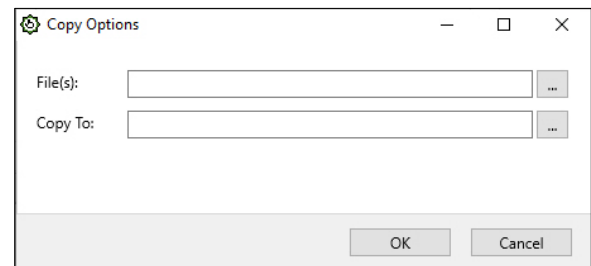
The 2nd field is for entering arguments to be added to the program before launching it. There is also a Check Box that will change the command from a standard "Program" command to a "ProgramWait" command.



This dialog box also has the **[Replacement Variable]** Button that will allow you to select from the list of supported variables in Touch HMI (these will be discussed in more detail at the end of this section).

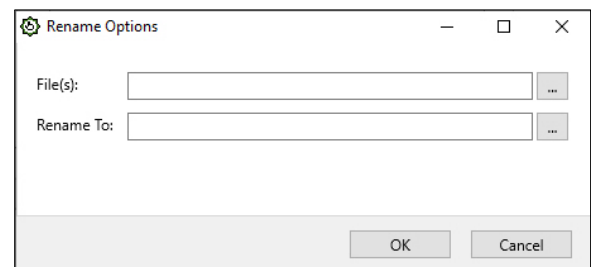
The **Copy** Command Dialog Box is used to select a file to copy from one location to another. The dialog box has an entry field for the original **File(s)** and the **Copy To** file names and location.

If you use the [...] Buttons to select the file or files, you will get the full path entered for your Command.



The **Rename** Command Dialog Box is used to select an original file and its new name for the Rename Command. The dialog box has an entry field for the original file and a 2nd one for the new file name.

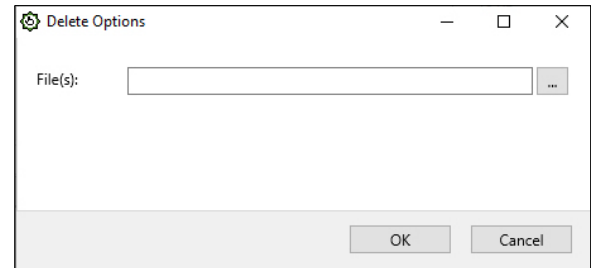
If you use the [...] Buttons to select the files, you will get the full path entered for your Command.



Predator Touch HMI

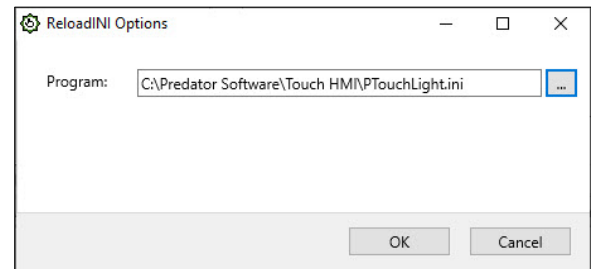
The **Delete** Command Dialog Box is used to select a file to be deleted. The dialog box has an entry field for the file(s) to be deleted.

If you use the [...] Buttons to select the files, you will get the full path entered for your Command.

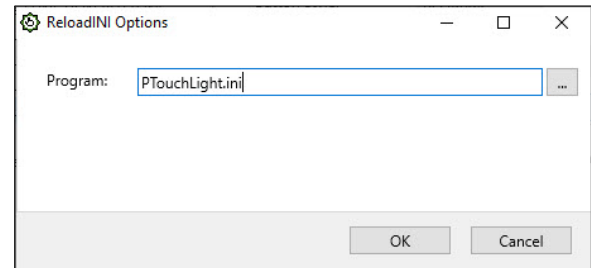


The **ReloadINI** Command Dialog Box is used to select another Touch HMI INI file to reload into the current Touch HMI panel. In this case when the button is selected the Touch HMI Panel (*.exe) will reload with the new configuration from the selected INI File.

If you use the [...] Button at the end of the command line and select a file, the result will include the entire path to the new INI file.



You can also simply enter the file name of the new INI file without the path information, and in this case the new INI file must be in the same path location as the original EXE and INI files.

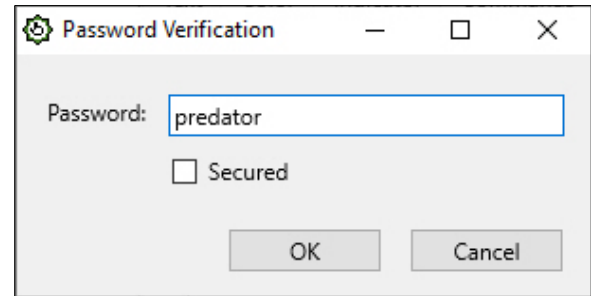


The **Delay** Options Command Dialog Box has a single-entry field where you can enter the length of time (in seconds) that you want to delay before executing any further commands on a Button.

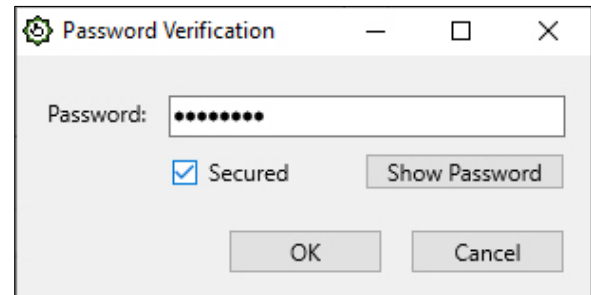


Predator Touch HMI

The **Password** Verification Command Dialog Box has a single-entry field where you can enter a Password that will be required to continue the execution of the command or commands on a Button. It also has a Checkbox for “Secured”, which will encrypt the password in the .INI file so that it is not visible.



If you click on the Secured Checkbox, then your password will be encrypted in the INI file and will no longer be displayed in the dialog box. If you wish to see the password again you can click on the [Show Password] Button to display the encrypted password before you save it to the INI file.



This is what the two different options would look like in the Command Editor Dialog box after you accept each of the two dialog options shown above.

... and this is what the INI would look like with both options in it. You would only use one or the other on a Button, not both.

[Button]

Row=1

Column=1

Style=Rectangle

Text=Password

Color=96,96,96

Gradient=200,200,200

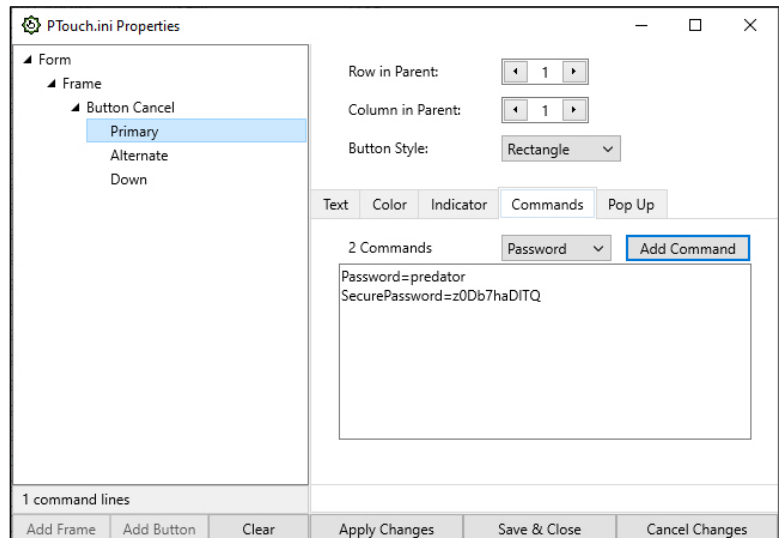
Outline=0,0,0

Line Width=2

Password=predator

SecurePassword=z0Db7haDITQ

Pop-Up Button Text=False



The **Exit** Options Command Dialog Box has two Radio Buttons to choose from, to select between a Confirm or Silent Exit. The Confirm choice will prompt the user for confirmation before closing the Panel, while the Silent choice will simply close the Panel immediately.



The Touch HMI Properties Editor Conventions

As of this version of the Touch HMI Properties Editor (v12.0.25.28), we have made some changes to the auto-correct features of this tool. Here are the new features and format changes to the Touch HMI .INI file structure:

As of Touch HMI v12.0.25.28, the Touch HMI Properties Editor will support the following:

1. The **TouchHMI.ini** file can have any text or comments in the file as long as they appear either above the **[Form]** heading in the file, or below the **[EndForm]** heading in the file. No special formatting is required in these two areas of the .INI file for comments.
2. If you want to add comments within a **[Form]**, **[Frame]**, or **[Button]** section of a TouchHMI.ini file, they must begin with a semicolon character (;) to indicate that this is a comment, and not part of the INI file to be processed. When a TouchHMI file is edited with this latest version of the Touch HMI Properties Editor, any comments found in the body of an INI file will be moved to the top of the section where they were found.
3. When a TouchHMI file is edited with this latest version of the Touch HMI Properties Editor, any lines in an INI file that do not match the current syntax for TouchHMI v12 will be commented out using a colon character (:) and moved to the bottom of the section where the text was found. These lines will not be processed when the Panel runs and can be deleted by the user if they want to clean up the .INI file. This allows a user to update their old TouchHMI.ini files to the latest version and syntax. As described above, lines beginning with a “;” will be ignored and left as comments in the file.

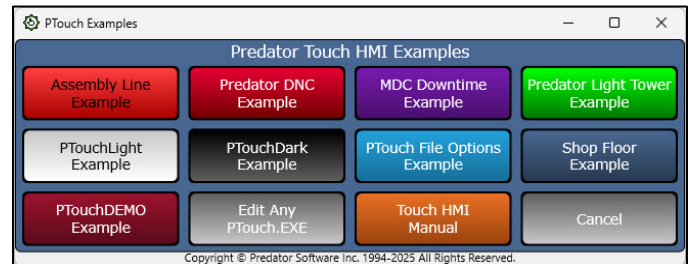
[NOTE: *The actions of the Touch HMI Properties Editor listed above will only take place if a TouchHMI Panel is opened, edited, and then saved with the Properties Editor. If a file is opened and then closed with no changes made, then none of these changes to the .INI file will be processed.]*

Touch HMI Pop-Up Menu Tutorial

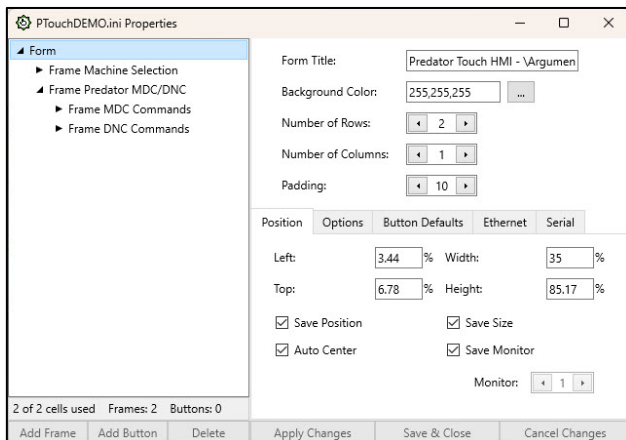
Working with Pop-Up Menus is a new and complicated topic for Touch HMI Users. The following section of this manual will walk through the creation and configuration of the Pop-Up Menu that is included in one of the example INI files that is part of the Touch HMI Install. The INI file that we will be reviewing is from the **PTouchDEMO** example that is included with the Touch HMI Installation.

To follow along, you can open the **PTouchDEMO.ini** file in the Configuration Utility. The easiest method to open any of the example TouchHMI INI files in the Properties Editor is to use the default **PTouchHMI.exe** which has buttons for launching all of the included DEMO INI Files, including:

- The 8 other Demo Touch HMI Examples
- The **PTouchDEMO** Example we want to review
- The Button to **Edit Any Touch HMI File**
- The Button to open the Touch HMI v12 Manual
- And a CANCEL Button to close the Panel



Select the PTouchDEMO Example Button in the lower left-hand corner. This will open the Touch HMI Properties Editor and a Sample Display Screen for the PTouchDEMO example as shown below.



[NOTE: while the Touch HMI Properties Editor is open, none of the functions on the Touch HMI Display Screen side will be active. You will have to close the Properties Window for any of the Buttons or Commands to function, with one Exception.

The Pop-Up Menu option will work in this condition so that the user can see what the Pop-Up Menu looks like while in development, but you will not be able to continue with changes to the Pop-Up Menu while it is open on the Display Screen. You must close the Pop-Up Menu on the Display side if you want to continue to make changes or save your changes in the Properties Editor.]

Now that you have the PTouchDEMO example open, let's continue with the Pop-Up Menu Tutorial.

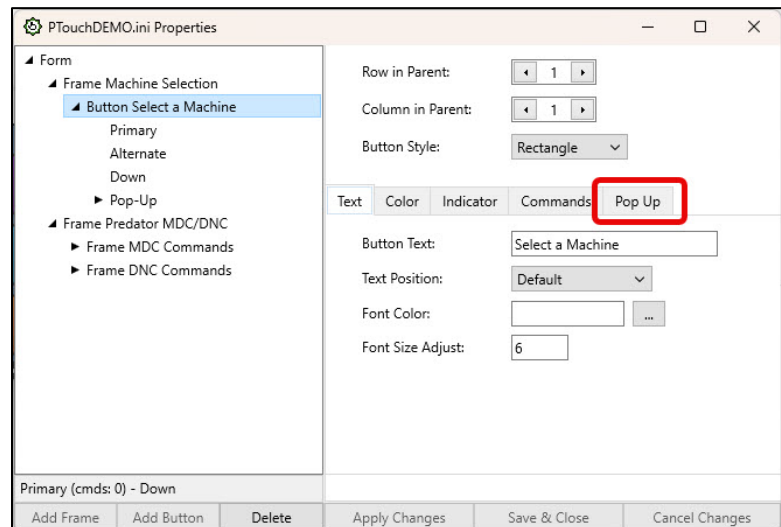
Step 1 - Create the Pop-Up Menu on a Button

As you will notice, the option to add a Pop-Up Menu only exists on the configuration side of an existing Button in a Touch HMI INI File. So, we must start by selecting an existing Button first. In this example we are using the existing Button “**Select a Machine**” in the Machine Selection Frame.

For this Button, we will only configure the Text and Color options. The Indicator and Commands Tabs can be left as default.

You should set the text on the Button to be your default button text that you want the user to see at first. In this case we set it to “**Select a Machine**” as shown on the right.

You can also pick a color and any gradient color settings that you may want for this Button on the Color Tab. In this example the color was a light grey (100,100,100) fading into a darker grey (50,50,50) gradient color at the bottom of the button.



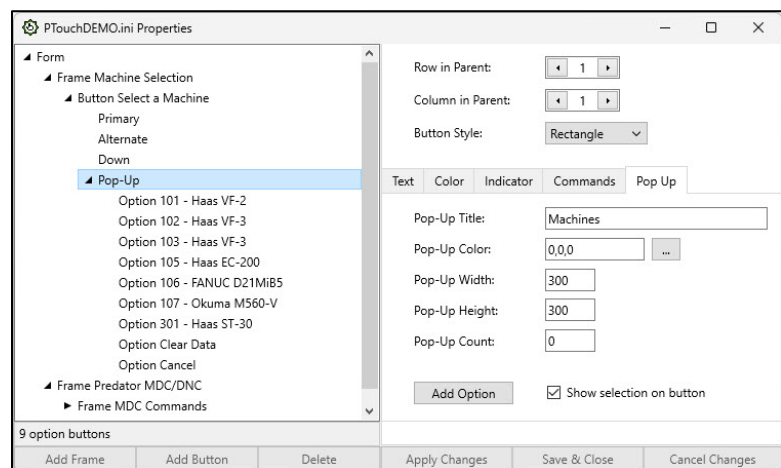
Now we can move onto the Pop-Up Tab.

Step 2 - Creating the Pop-Up menu Options

The first thing we need to decide is how many options (or buttons) do we want to have on the Pop-Up Menu. That will determine how many times to click on the **[Add Option]** Button to create what we want.

You will notice in our example that this Pop-Up menu has 9 Option Buttons:

- 101 - Haas VF-2
- 102 - Haas VF-3
- 103 - Haas VF-3
- 105 - Haas EC-200
- 106 - FANUC D21MiB5
- 107 - Okuma M560-V
- 301 - Haas ST-30
- Clear Data
- Cancel



We would need to click on the **[Add Option]** Button 9 times to create these Menu Buttons under the Pop-Up Heading in the Tree, as shown on the right. **You will also need to set a Pop-Up Width and Height as Pop Up menus do not auto size.** These values will change depending on how many buttons you have and the text width on each one.

The Pop-up Width can be set to a number of Pixels by entering a number (i.e. 200), or to a percentage of the overall Screen width (i.e. 20%). Whatever you choose the size should allow all of your Pop-up Button Text names to fit within the Button Width.

The Pop-up Height can also be set to a number of Pixels by entering a number (i.e. 300), or to a percentage of the overall Screen height (i.e. 50%). You can come up with a good value for Pop-up Height by selecting a number of pixels per button (i.e. 30 pixels) and then multiplying that by the number of Buttons you will have in your Drop-down List. So, a Pop-up List with 10 Buttons at 30 pixels per Button will have an overall height of 300 pixels.

Each Option Button will be used in this example to set a series of Predator “**Variables**” for each Machine Tool in the list so that the Buttons in the other Frames labeled “MDC Commands” and “DNC Commands” will use the selected set of variables for their commands. This ability to create sub-menus greatly improves the functionality that we can do in a single Panel with Touch HMI v12. These same functions would have required multiple Touch HMI panels to do the same functions in Touch HMI v10 or v11.

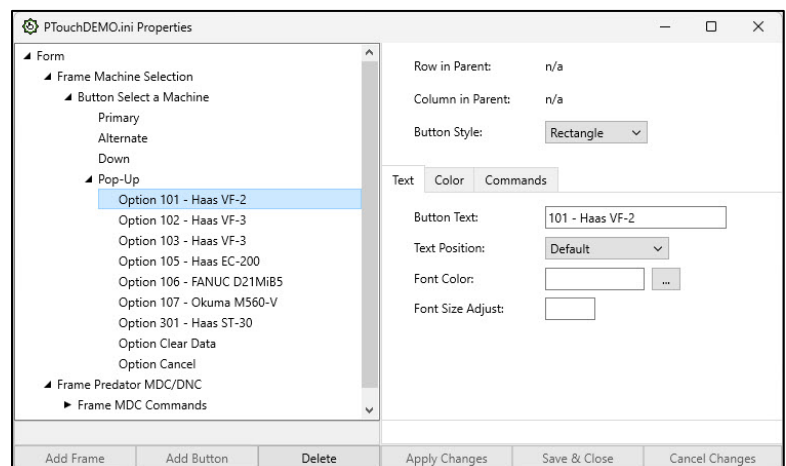
Step 3 - Configuration the new Pop-Up Buttons

Now that we have our Pop-Up Buttons created, we need to configure each one with the functionality that we want each Option Button to execute.

Let’s start with the first button in the menu. This button will set the variables for sending files to our Machine 101 - Haas VF-2 Milling Center.

Click on the **Text Tab** to configure the visible text on the Menu Button. Enter the following:

- **Button Text** - This is the actual text you will see on the Option Menu Button. In this case we want to use the name of the CNC Machine that this button controls so enter the machine name “101 - Haas VF-2”.

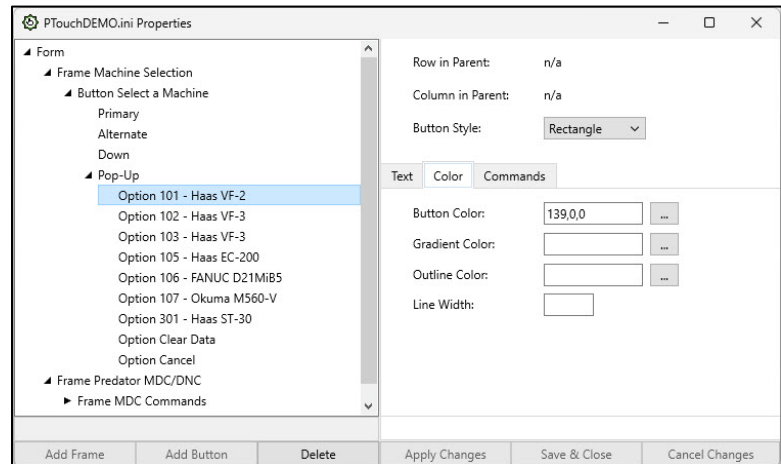


[NOTE: since the “Show Selection on Button” option was checked for this Menu, the text 101- Haas VF-2 will replace the text on the main parent Button in the Frame after this button is pressed.]

- **Text Position** - This is so you can control the position of the text on the button. For this example, leave it as Default.
- **Font Color** - This is so you can control the color of the text font on the button. For this example, leave it blank for the default, which will automatically set the Font Color to either black or white depending on the Button Color selected.
- **Font Size Adjust** - This is so you can control the size of the text on the button. For this example, leave it blank for the default, which will automatically scale the font size based on the size of the Touch HMI Panel if you resize it on the Windows Desktop.

Next, click on the **Color Tab** to configure the color for the Menu Button. Enter the following:

- **Button Color** - This is the color you will see for this button on the Pop-Up Menu. In this case we chose a dark red so enter the RGB Values of 139,0,0.
- You can also change the settings for **Gradient Color**, **Outline Color**, and the button's **Line Width**. In this case we are leaving these all set to default or off so you can leave these fields blank.



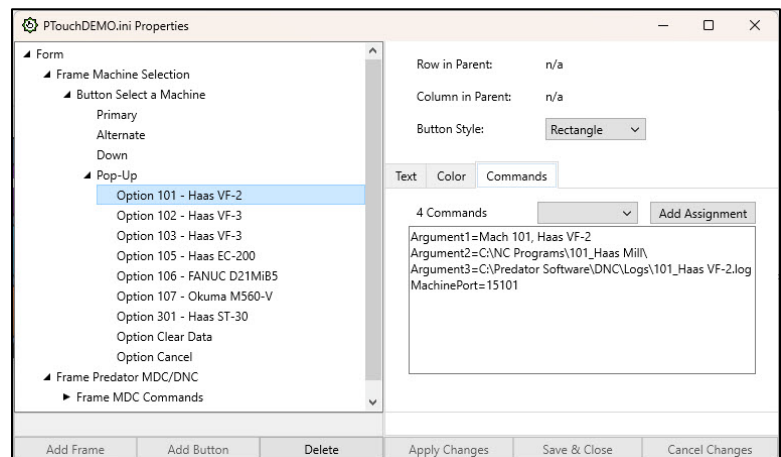
Now that brings us to the **Commands Tab**. This is where we will add all of the variable commands needed to set the data values that this button will control.

In this example, these buttons for the different Machine Tools in the list will be setting the values for 4 different replacement variables that will be used by the DNC and MDC Command Buttons in the other Frames. These commands are:

- **Setting the \Argument1 Variable** - In this case the **\Argument1** replacement variable is being used to enter the "Machine Name" at the top of the Main Panel when a Machine has been selected. The syntax for this command is:
Argument1=Mach 101, Haas VF-2
- **Setting the \Argument2 Variable** - In this case the **\Argument2** variable is being used to enter the Default Folder Location for the NC programs on the DNC Server that will be sent to the CNC Control via Predator DNC.

The syntax for this command is: **Argument2=C:\NC Programs\101_Haas Mill**

- **Setting the \Argument3 Variable** - In this case the **\Argument3** variable is being used to enter the path and file name for the Predator DNC Log File for this machine. This information can be used to execute a View DNC Log Button in the DNC Commands Touch HMI Menu. The syntax for this command is:
Argument3=C:\Predator Software\DNC\Log\101_Haas VF-2.log
- **Setting the Ethernet Port Variable** - In this command we are setting the Ethernet Port Number that will be used for this machine to trigger the Remote Request Command in Predator DNC. The syntax for this command is:
MachinePort=15101

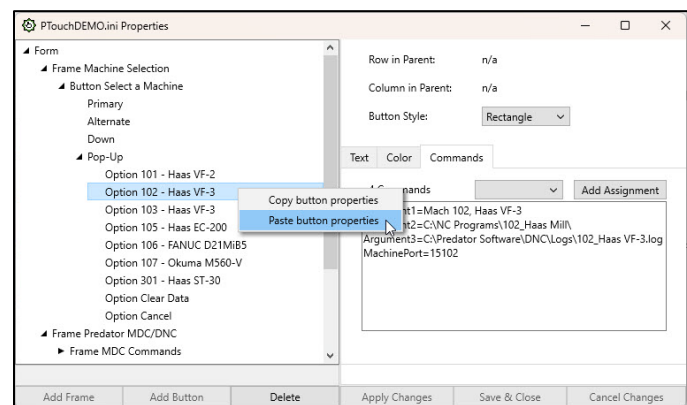
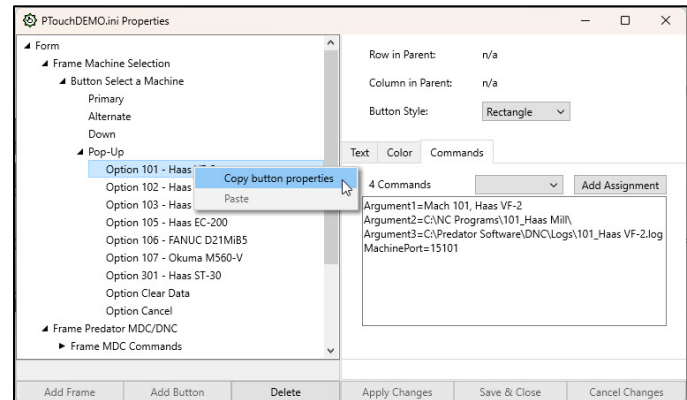


Once all of the commands have been entered for the Button, it is a good idea to save your work. The Buttons at the bottom of the Touch HMI Configuration Utility can be used to **[Apply Changes]** or **[Save and Close]** the Utility. The **[Apply Changes]** Button will save the .INI configuration and temporarily close both the Display Panel and Config Panel and then reopen them with the refreshed .INI data. The **[Save and Close]** Button will save the .INI configuration and close both the Display Panel and Config Panel.

Once you have the first machine's button configured, you can enter the new values for the other Machine buttons in this Pop-Up Menu example as they will be using the same commands.

You can also use the Copy and Paste option for a new Button. To do this:

1. Start by selecting your finished Button that you want to copy on the Tree side of the Configuration Utility and right-clicking on it to show the Copy and Paste menu.
2. Click on **Copy button properties** to copy the current button.
3. Create a new Button in the Pop-Up Menu by clicking on the **[Add Option]** Button on the Main Pop-Up Item in the Tree or select another blank Option Button that has already been created.
4. Now you can move your cursor in the Tree to the new Button you want to create and select **Paste button properties** from the Right-click menu. Touch HMI will paste the settings from the first button for all of the Tabs **except the Text Tab** into the new Button's configuration. You will have to add your own values for the Text Tab and then edit the values for the Color and Commands Tabs for the new Button's functions.
5. You can also use the **[Delete]** Button at the bottom of the Dialog Box to remove any Frames or Buttons in the configuration that you don't want to keep.



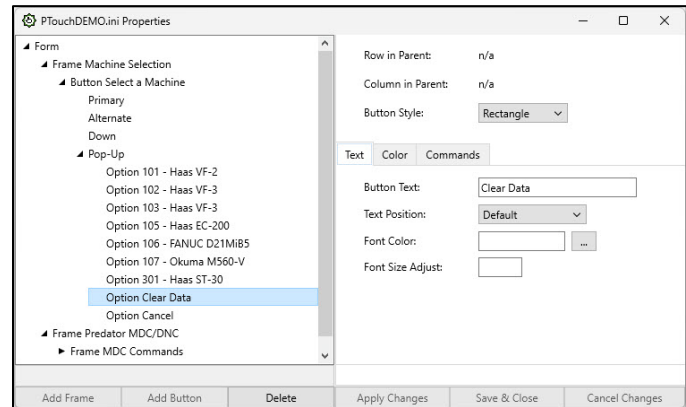
[NOTE: you cannot delete a Pop-Up Menu from a Button in the Tree without first deleting all of the Pop-Up Option Buttons below it in the Menu's configuration.]

Predator Touch HMI

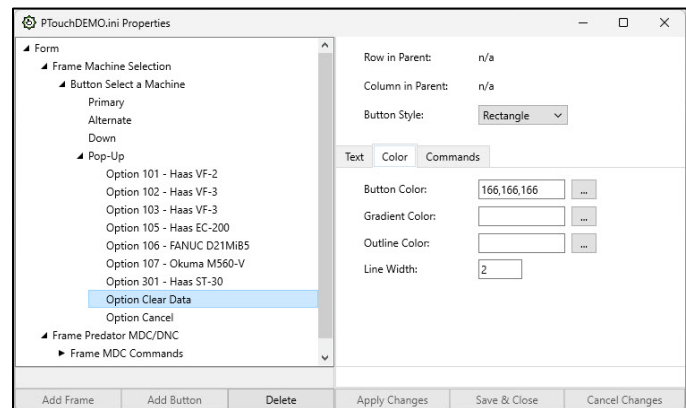


Now that all of the Buttons have been created for the different CNC Machines in the Pop-Up Menu List, we have two more buttons to configure. The first is a **Clear Data Button** and the second is the **Cancel Button**.

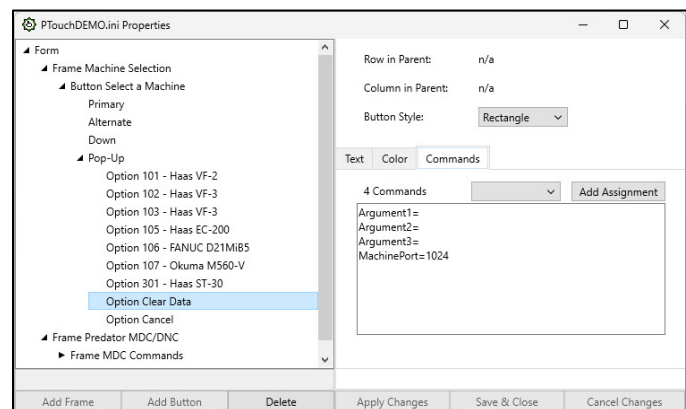
Add a new Button to the Menu as described above and change the text for this Button to be **“Clear Data”**. You can leave all of the other options set to Default or blank on this Tab.



Next click on the Color Tab and change this button's color to your desired color. In this case we used a color of 166,166,166, which is a light grey. We also have a Line Width of 2 pixels. You can leave all of the other color options blank on this Tab.



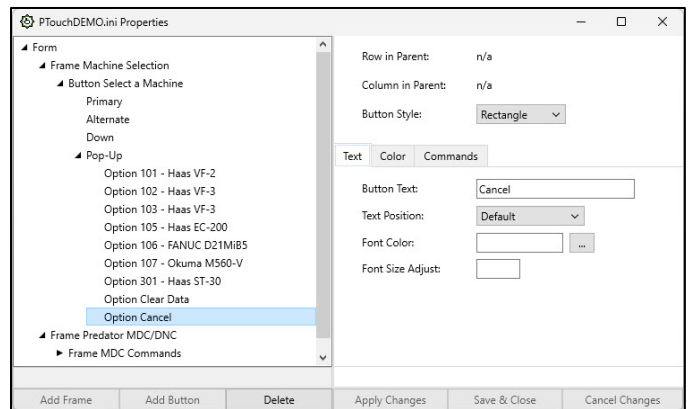
Now click on the Commands Tab to set the commands for this Button. In this case we are going to clear all of the data for each command to set the panel back to no selected data, so you will see that each of the Argument1, Argument2, and Argument3 variables have no data, and the MachinePort value is set back to the default setting of 1024. This will eliminate any data from being sent if a user presses a Button in any of the other Frames. Closing and opening the Touch HMI panel does the same thing.



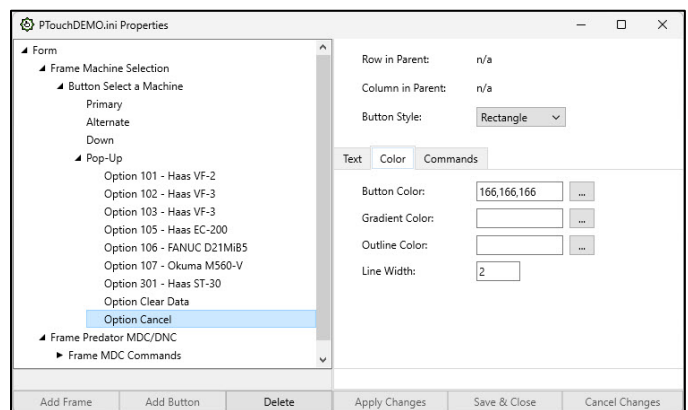
Predator Touch HMI

The last Button to add will be the **Cancel Button** and will allow the user to exit the Pop-Up Menu without making any selections or changes.

Add a new Option Button to the Menu as described above and change the text for this Button to be **“Cancel”**. You can leave all of the other options set to Default or blank on this Tab.

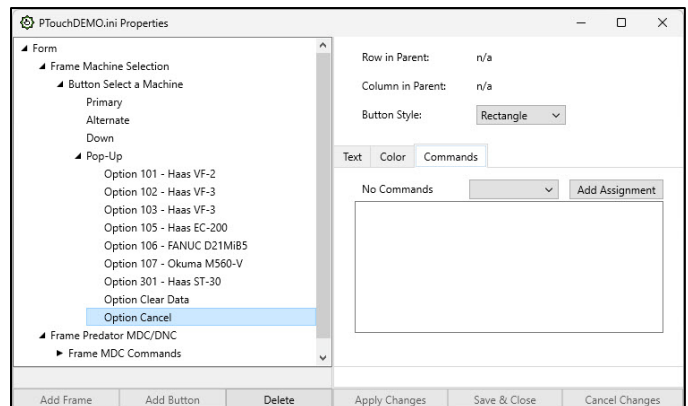


Next click on the Color Tab and change this button's color to your desired color. In this case we used a color of 166,166,166 again, which is a light grey. We also have a Line Width of 2 pixels. You can leave all of the other color options blank on this Tab.



Now move on to the Commands Tab. Here you will see that we do not have any commands selected for the Cancel Button. That is because we do not want this button to do anything to the configuration of the parent Button other than to close the Pop-Up menu.

The function of any Option Button in a Pop-Up menu is to execute the commands attached to it and then close the Pop-Up menu, so this button will only do the latter, and close the Pop-Up menu so we can continue with other functions on the Touch HMI Panel.

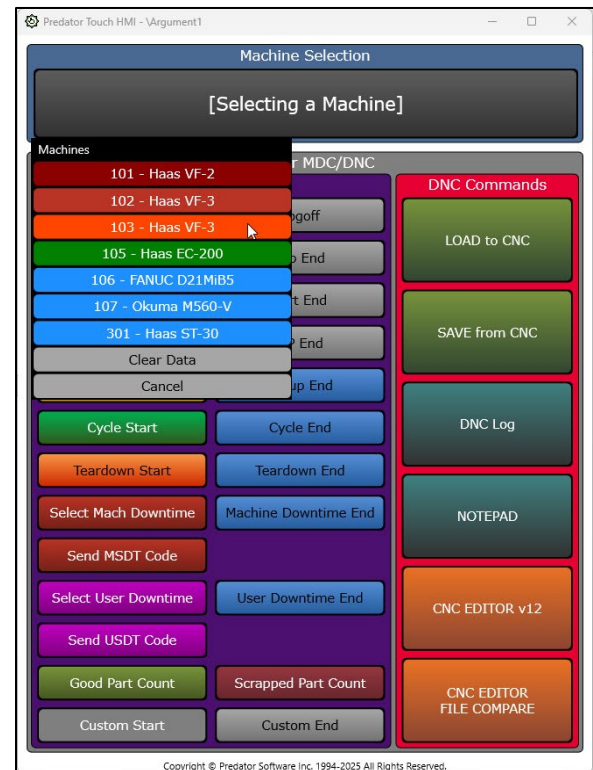


Predator Touch HMI



You have now finished configuring a Pop-Up Menu to give you a list of machines to select from and set their individual variables for the other commands to use on the MDC and DNC Command Frames.

You can use the included **PTouchDEMO.exe** and **PTouchDEMO.ini** to explore the configuration of this Panel in more detail.



Touch HMI Variables Reference List

The following is the list of Variables that can be added to commands on a Touch HMI Button. **Highlighted Variables** are new to Touch HMI v12 ONLY.

Replacement Variables (Revised as of v12.0.25.x)

Variable	Description	Use
\13\10	Carriage Return Line Feed	Used to add a Carriage Return/Line Feed between text lines on a Button. It can also be used to separate MDC Commands and place them on separate lines.
\A1 to \A9	Changed in v12.0.25.14	All of the 9 “\Ax” Variables in v12 were changed as of v12.0.25.14 to now be “\Argumentx” Variables. This was due to the possibility that some users may have “\A” string in their argument data (i.e. Folder Names) and that would confuse some commands.
\Argument1	Variable A1	Variable A1 to be used to store data for use in a command.
\Argument2	Variable A2	Variable A2 to be used to store data for use in a command.
\Argument3	Variable A3	Variable A3 to be used to store data for use in a command.
\Argument4	Variable A4	Variable A4 to be used to store data for use in a command.
\Argument5	Variable A5	Variable A5 to be used to store data for use in a command.
\Argument6	Variable A6	Variable A6 to be used to store data for use in a command.
\Argument7	Variable A7	Variable A7 to be used to store data for use in a command.
\Argument8	Variable A8	Variable A8 to be used to store data for use in a command.
\Argument9	Variable A9	Variable A9 to be used to store data for use in a command.
\ComputerName	Windows Computer Name	Sends out the current PC's Computer Name (Windows)
\FileOpen	File Open Prompt	Prompt for File Open using the Predator File dialog format. This prompt returns the filename with full path.
\FileSave	File Save Prompt	Prompt for File Save using the Predator File dialog format. . This prompt returns the filename with full path.
\FileNameOpen	Filename Open Prompt	Prompt for FileName Open using the Predator File dialog format. This prompt returns just the filename.
\FileNameSave	Filename Save Prompt	Prompts for FileName Save using the Predator File dialog format. This prompt returns just the filename.
\Folder	Folder Prompt	Prompts for a Folder using the Predator File dialog format. This prompt returns the selected folder name.
\IP4	Deprecated in v12.0.25.5	Use \IPv4 (shown below) instead.
\IPv4	IP Address (IPv4)	Sends out the current IPv4 IP Address set in the Touch HMI INI File (for Windows)
\LastFileOpen	Last Filename opened	Reloads the filename for the last file that was opened.
\LastFileSave	Last Filename saved	Reloads the filename for the last file that was saved.
\LastPrompt	Last Prompt	Recalls data from the Last Prompt to be used again on a following command.

\MachinePort	Ethernet Port Number	Sends out the current active Port Number set in the Touch HMI INI File (for Windows)
\Port	Deprecated in v12.0.25.5	Use \MachinePort (shown above) instead.
\Prompt	Simple Prompt	Simple prompt for data to be entered by the user.
\OSFileOpen	O/S File Open Prompt	Prompts for File Open using the Windows O/S File dialog format. This prompt returns the filename with full path.
\OSFileSave	O/S File Save Prompt	Prompts for File Save using the Windows O/S File dialog format. This prompt returns the filename with full path.
\OSFileNameOpen	O/S Filename Open Prompt	Prompts for FileName Open using the Windows O/S File dialog format. This prompt returns just the filename.
\OSFileNameSave	O/S Filename Save Prompt	Prompts for FileName Save using the Windows O/S File dialog format. This prompt returns just the filename.
\OSFolder	O/S Folder Prompt	Prompts for a Folder using the Windows O/S File dialog format. This prompt returns the selected folder name.
\PC	Deprecated in v12.0.25.5	Use \ComputerName (shown above) instead.
\RFID1	Stored Data from RFID Line 1	Variable to store the data from Line 1 of a RFID Scan.
\RFID2	Stored Data from RFID Line 2	Variable to store the data from Line 2 of a RFID Scan.
\RFID3	Stored Data from RFID Line 3	Variable to store the data from Line 3 of a RFID Scan.
\RFID4	Stored Data from RFID Line 4	Variable to store the data from Line 4 of a RFID Scan.
\RFID5	Stored Data from RFID Line 5	Variable to store the data from Line 5 of a RFID Scan.
\RFID6	Stored Data from RFID Line 6	Variable to store the data from Line 6 of a RFID Scan.
\TouchHMITitle	Touch HMI Form Title	Sends out the Title set in the Form Section of the active Touch HMI INI configuration.
\Title	Deprecated in v12.0.25.5	Use \TouchHMITitle (shown above) instead.
\User	Deprecated in v12.0.25.5	Use \WindowsUser (shown below) instead.
\WindowsUser	Current Windows User	Sends out the current Windows Username (Windows)
\000 to \255	Support for ASCII Codes	Enables user to send ASCII characters in Commands. This is mainly used for Serial Communications.
\x00 to \xFF	Support for Hex Codes Changed in v12.0.25.21	Enables user to send Hex characters in both Serial and Ethernet Commands.
\xHex00 to \xHexFF	Support for Hex Codes Updated in v12.0.25.21	Enables user to send Hex characters in both Serial and Ethernet Commands.

[NOTE: Remember all variables are case sensitive and must start with a backslash “\” character.]

[NOTE: Both the ASCII and HEX Codes listed above are not available in the Replacement Variable Drop-down List in the Touch HMI Properties Editor. These can be added manually to you commands in the INI File, but you will have to know the values for the ASCII or HEX characters that you want to send (i.e. \13\10 for ASCII and \0d\0a for HEX, for the Carriage Return / Line Feed characters).]

Using Paths with Predator Touch HMI Commands

There are a number of ways that a filename and its path can be entered within a command in Predator Touch HMI. These can include:

- An Absolute Path
- A Relative Path
- A UNC Path
- or A URL Path

Absolute Paths

An absolute, or full, path begins with a drive letter followed by a colon, such as “C:” and then the full path information with the filename such as:

- C:\Predator Software\Touch HMI 12.0\PTouchDEMO.exe
- D:\NC Programs\Haas Mills\O12345.nc

Relative Paths

A relative path refers to a location that is relative to a current directory. Relative paths use two special symbols, a dot (.) and a double dot (..). These translate into the current directory and the parent directory respectively. Double dots are used to move up in the directory hierarchy while the single dot represents the current directory. For example:

- .\PTouchDEMO.exe
- ..\O12345.nc

The first example would look for the file “PTouchDEMO.exe” in the same directory as the Predator Touch HMI exe that is executing the command.

The 2nd example would look for the file “O12345.nc” in the directory above the current location, which might be \NC Programs\O12345.nc.

The Double Dot path can also be stacked to move farther back up a path chain. For example:

If we had a local Path of: **D:\Predator\Stored Files\NC Programs\Haas Mills** where the files were stored and the PTouchHMI.exe file was running from the **D:\Predator\Stored Files\NC Programs** folder, we could have a Path statement in a Command such as:

- **.\O12345.nc**, which would look in the current folder location of D:\Predator\Stored Files\NC Programs\ for the file O12345.nc.
- **.\Haas Mills\O12345.nc**, which would look in the sub-directory \Haas Mills in the current folder for the file O12345.nc.
- **..\O12345.nc**, which would look in the folder “D:\Predator\Stored Files\” for the file O12345.nc
- **..\..\O12345.nc**, which would look in the folder “D:\Predator\Predator\” for the file O12345.nc.
- **..\..\..\O12345.nc**, which would look in the root “D:\” folder for the file O12345.nc.

All of these relative Paths are relative to the current working folder where your TouchHMI.exe is running from.

[NOTE: In Predator Touch HMI any command in an INI file where the path is omitted (i.e. Program=TouchHMI.exe) is considered to be the same as a “.” local path of the current working directory.]

UNC Paths

UNC stands for Universal (or Uniform or Unified) Naming Convention and is a syntax for accessing folders and files on a network of computers. The syntax is as follows:

\\<computer name>\<shared directory>\

This path structure can be followed by any number of directories and terminated with a directory or file name, for example:

- \\Predator Server\NC Programs\Haas Mills\O12345.nc
- \\192.168.10.125\NC Programs\Haas Lathes\O13446.nc

The computer name is always preceded by a double backward slash (\\). In UNC, the computer name is also known as the Host Name. The Host Name can also be an IP Address in place of a DNS computer name.

URL Paths

URL stands for Uniform Resource Locator and uniquely specifies the address of any document on the internet. The components of a URL are:

- The protocol used to access the resource, such as HTTP (Hypertext Transfer Protocol) or FTP (File Transfer Protocol)
- The host (server) with which to communicate
- The path to the file on the host

For example:

https://predator-software.com/predator_support.htm

Firewalls

All Windows and corporate firewalls need to open the appropriate Ethernet Ports for TCP traffic on the network to enable Predator Touch HMI to operate via Ethernet. Predator Touch HMI will display error messages when the ports are blocked.

If Ethernet based serial devices will be used with Predator Touch HMI, then the appropriate Ethernet Ports that are required by the serial device drivers need to be opened for TCP traffic on the network to enable Predator Touch HMI to communicate with the COM Ports.

FAQs

1. Can Predator Touch HMI Panels be installed on a WAN or slow network? **Answer: Yes.**
2. Can Predator Touch HMI Panels be used in a mix of network domains, work groups, and home groups?
Answer: Yes.
3. Does Predator Touch HMI work with older versions of Windows? **Answer: v12 works with Windows 7 and newer. Support for older O/S versions (Windows 95 and later) is available with Touch HMI v11.**
4. Where can I download ws2setup.exe to add Ethernet Support for Windows 95? **Answer: Our Support Folder on our Website has a download link.**
5. Can Predator Touch HMIs be used with older serial hardware? **Answer: Yes. We only require that they have a working and reliable Windows device driver.**

Troubleshooting

The following is a list of troubleshooting tips to help solve issues with Touch HMI applications and implementations.

Tip 1 - Restricted Access with Group Policies

Often CNCs, CMMs, Test stands, and other shop floor PCs run with restricted access privileges typically controlled by a Windows Group policy. If this is the case one might experience the following error:

This program is blocked by group policy. For more information contact your system administrator.

The names of the group policies that can affect operation of Predator Touch HMIs are the following:

Don't run specified Windows applications

Run only allowed Windows applications

Even if the Windows CNC, CMM, Test stand, or shop floor PC is no longer on the domain the current user may still be operating with previously inherited policies. This can be double checked within the Windows Registry. Review the following Windows Registry entries:

hklm\software\microsoft\windows\currentversion\policies\explorer\restrictuser

hkcu\software\microsoft\windows\currentversion\policies\explorer\restrictuser

hklm\software\microsoft\windows\currentversion\policies\explorer\disallowrun

hkcu\software\microsoft\windows\currentversion\policies\explorer\disallowrun

In the case of a RestrictRun policy, the **PTouch.exe(s)** need to be added to the list of registry entries and in the case of DisAllowRun the PTouch.exe(s) needs to not be listed. Basically, RestrictRun is a "whitelist" of applications that are allowed to run and DisAllowRun is a "blacklist" of applications that cannot be run under Windows.

Tip 2 - No Internet Access

Often CNCs, CMMs, Test stands, and other shop floor PCs run without access to the internet and may not be able to verify the digital certificate assigned to Predator Touch HMI. This often happens when a shortcut is used to run the HMI from a Server. When this happens, either resolve the ability to verify digital certificates or run the HMI directly from the local hard drive of the CNC, CMM, test stand, or other shop floor PC.

Tip 3 - Issue with Text in an INI File with "Double Quotes"

If a Touch HMI Command requires the use of a Text String that has spaces in it, then double quotes "xxx xx" are needed to surround the text for it to work in the Command. The double quotes character for Touch HMI INI File use **MUST BE** the **ASCII Code DEC 34 Character** (") or Straight quotes character, and not the Curly quotes characters "xxx xx", found in most Word Processors or Editors. The Curly Quotes Characters are not part of the standard ASCII 128-character set.

Curly quotes are often referred to as Smart quotes because they curve toward the text you're quoting. These characters are used and inserted automatically in most word processors, like Microsoft Word. Straight quotes are what they sound like - straight, vertical marks without any indication of which word or phrase it's surrounding. While we all may use these quotes characters interchangeably in our day-to-day lives, a computer program such as Touch HMI sees them as different characters and the text in your Touch HMI Panels will not display properly if the Straight quotes are not used.

All of the Predator products, such as the Touch HMI Properties Editor and the Predator CNC Editor, will only use the proper Straight Double Quotes Character (") when used to edit a Touch HMI INI File or a NC Program.

CNC Support

The following list of CNC controls that are supported with Predator Touch HMI.

Anca AMC5 CNC
Anilam 6000i CNC
Beckoff TwinCAT CNC TX1270
BendPro CNC
Bosch Rexroth IndraMotion MTX CNC
CamSoft CNC Professional
CamSoft CNC Lite
CamSoft CNC Plus
Centroid M-39 CNC
Centroid M-400 CNC
Cincinnati 2100
CNC2000
CNC 82.00
CNC 84.00
CNCPlusG
CNCPlusH
CNCPlusM
CNCPlusT
CODESYS SoftMotion CNC Current EDM
D.Electron CNC Z32
Datron HSCpro Datron Next
Delta Tau Advantage 900
EaziCNC
Eding CNC
Fagor 8060
Fagor 8065
Fagor 8070
Fanuc Open CNC 30i
Fanuc Open CNC 31i
Fanuc Open CNC 32i
Fanuc Open CNC 35i
Ficep CNC
Fidia C10
Fidia C20
Fidia C40
Fidia nC 12R

Fidia nC 15

FlashCut CNC Foba Laser Gerber C200MT Gerber Paragon

Hurco Winmax

Hypertherm EDGE Pro

Indigen CNC

Isel KAY

Laserdyne System 94P

Laserdyne System 94W

Linatrol Profiler

Linatrol Profiler Plus

Linatrol Infinity CNC

Mach 3

MachineMate eCNC

MachineMate L2

MachineMate LW

MAXNC

Mazak Mazatrol 640M

Mazak Mazatrol 640M Nexus

Mazak Mazatrol 640T

Mazak Mazatrol 640T Nexus

Mazak Mazatrol 640MT Pro

Mazak Mazatrol Fusion

Mazak Mazatrol Matrix

Mazak Mazatrol Matrix 2

Mazak Mazatrol Matrix Nexus

Mazak Mazatrol Matrix Nexus 2

Mazak Mazatrol Nexus

Mazak Smooth Series

MDSI OpenCNC

Minitex CNC

Mitsubishi M700VW

Mitsubishi M800W

NEXCOM NControl CNC

NUM Flexium+

NUM Flexium

NUM 1020 CNC

NUM 1040 CNC

NUM 1060 CNC

NUM 1080 CNC

NXGEN CNC
Okuma P100
Okuma P200
Okuma P300
Parpas CNC Z32
Planet CNC USB Controller
ProtoTRAK SMX
ProtoTRAK SLX
SaleCNC
Selca S4000
ServoWorks S-100M
ServoWorks S-120M
ServoWorks S-140M
ServoWorks S-100T
ServoWorks S-120T
ShopBot CNC
Siemens 2100
Siemens 840D
Simplex CNC
Sodick LN2
Sodick LQ1W
Sodick LN1W
Star CAM CNC
Techno CNC
TechMoon Waterjet CNC
Thermwood QCore
UTE Navigator CNC
Vickers 2100
Vital CNC WARDJet Classic
WARDJet XL Winbro
WinCNC
WinPC-NC
WinPCNC Light
WinPCNC Economy
WinPCNC USB
WinPCNC Professional
Zinser CNC 2030
Zinser CNC 2050
Zinser CNC 4010, and more ...

CMM Support

The following list of CMM controls that are supported with Predator Touch HMI.

Aberlink 3D
Cognex VisionPro
Cognex VisionPro ViDi
Delcam PowerInspect
Eley True Measure 4
GEOMET JUNIOR
GEOMET 101
GEOMET 301 HandyPROBE Next PCMM
Hexagon PCDMIS
Hexagon QUINDOS
Keyence XM Series
LK Metrology Altera
Lynx AR Inspection
MetLogix M2
MetLogix M3
MetLogix M3 DC/DXF
Metrosoft QUARTIS
MicroVu InSpec
Mitutoyo MCOSMOS CMM
Nikon CAMIO
Nikon CMM-Manager
Nikon I++/DME Server
OpenDMIS 3D
Perception TouchDMIS
Polyworks Inspector
Renishaw MODUS
SIPCON SipMeas
SIPCON SipMeas 3D
SIPCON IK 5000
Zeiss Calypso
and more...

Color Reference

The following chart lists the common reference colors which can be used for Button Colors with Predator Touch HMI.

[NOTE: If color matching to the Predator applications is required, refer to the additional color lists following this one.]

Common RGB Reference Colors

Color	Color Name	RGB Values
	Maroon	128,0,0
	Dark Red	139,0,0
	Brown	165,42,42
	Firebrick	178,34,34
	Crimson	220,20,60
	Red	255,0,0
	Tomato	255,99,71
	Coral	255,127,80
	Indian Red	205,92,92
	Light Coral	240,128,128
	Dark Salmon	233,150,122
	Salmon	250,128,114
	Light Salmon	255,160,122
	Orange Red	255,69,0
	Dark Orange	255,140,0
	Orange	255,165,0
	Gold	255,215,0
	Dark Golden Rod	184,134,11
	Golden Rod	218,165,32
	Pale Golden Rod	238,232,170
	Dark Khaki	189,183,107
	Khaki	240,230,140
	Olive	128,128,0
	Yellow	255,255,0
	Yellow Green	154,205,50
	Dark Olive Green	85,107,47
	Olive Drab	107,142,35

Predator Touch HMI

	Lawn Green	124,252,0
	Chartreuse	127,255,0
	Green Yellow	173,255,47
	Dark Green	0,100,0
	Green	0,128,0
	Forest Green	34,139,34
	Lime	0,255,0
	Lime Green	50,205,50
	Light Green	144,238,144
	Pale Green	152,251,152
	Dark Sea Green	143,188,143
	Medium Spring Green	0,250,154
	Spring Green	0,255,127
	Sea Green	46,139,87
	Medium Aquamarine	102,205,170
	Medium Sea Green	60,179,113
	Light Sea Green	32,178,170
	Dark Slate Gray	47,79,79
	Teal	0,128,128
	Dark Cyan	0,139,139
	Aqua	0,255,255
	Cyan	0,255,255
	Light Cyan	224,255,255
	Dark Turquoise	0,206,209
	Turquoise	64,224,208
	Medium Turquoise	72,209,204
	Pale Turquoise	175,238,238
	Aquamarine	127,255,212
	Powder Blue	176,224,230
	Cadet Blue	95,158,160
	Steel Blue	70,130,180
	Corn Flower Blue	100,149,237
	Deep Sky Blue	0,191,255
	Dodger Blue	30,144,255

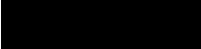
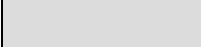
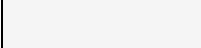
Predator Touch HMI

	Light Blue	173,216,230
	Sky Blue	135,206,235
	Light Sky Blue	135,206,250
	Midnight Blue	25,25,112
	Navy	0,0,128
	Dark Blue	0,0,139
	Medium Blue	0,0,205
	Blue	0,0,255
	Royal Blue	65,105,225
	Blue Violet	138,43,226
	Indigo	75,0,130
	Dark Slate Blue	72,61,139
	Slate Blue	106,90,205
	Medium Slate Blue	123,104,238
	Medium Purple	147,112,219
	Dark Magenta	139,0,139
	Dark Violet	148,0,211
	Dark Orchid	153,50,204
	Medium Orchid	186,85,211
	Purple	128,0,128
	Thistle	216,191,216
	Plum	221,160,221
	Violet	238,130,238
	Magenta / Fuchsia	255,0,255
	Orchid	218,112,214
	Medium Violet Red	199,21,133
	Pale Violet Red	219,112,147
	Deep Pink	255,20,147
	Hot Pink	255,105,180
	Light Pink	255,182,193
	Pink	255,192,203
	Antique White	250,235,215
	Beige	245,245,220

Predator Touch HMI

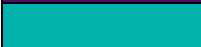
	Bisque	255,228,196
	Blanched Almond	255,235,205
	Wheat	245,222,179
	Corn Silk	255,248,220
	Lemon Chiffon	255,250,205
	Light Golden Rod Yellow	250,250,210
	Light Yellow	255,255,224
	Saddle Brown	139,69,19
	Sienna	160,82,45
	Chocolate	210,105,30
	Peru	205,133,63
	Sandy Brown	244,164,96
	Burly Wood	222,184,135
	Tan	210,180,140
	Rosy Brown	188,143,143
	Moccasin	255,228,181
	Navajo White	255,222,173
	Peach Puff	255,218,185
	Misty Rose	255,228,225
	Lavender Blush	255,240,245
	Linen	250,240,230
	Old Lace	253,245,230
	Papaya Whip	255,239,213
	Sea Shell	255,245,238
	Mint Cream	245,255,250
	Slate Gray	112,128,144
	Light Slate Gray	119,136,153
	Light Steel Blue	176,196,222
	Lavender	230,230,250
	Floral White	255,250,240
	Alice Blue	240,248,255
	Ghost White	248,248,255
	Honeydew	240,255,240
	Ivory	255,255,240

Predator Touch HMI

	Azure	240,255,255
	Snow	255,250,250
	Black	0,0,0
	Dim Grey	105,105,105
	Grey	128,128,128
	Dark Grey	169,169,169
	Silver	192,192,192
	Light Grey	211,211,211
	Gainsboro	220,220,220
	White Smoke	245,245,245
	White	255,255,255

Predator Application Colors

It is often a good idea to make the color of a button match a Predator application or our corporate blue color. Refer to the Predator Color List below for a list of these colors and their RGB values:

Color	Color Name	RGB Values
	Predator CNC Editor	234,113,37
	Predator DNC	231,0,51
	Predator MDC	75,16,111
	Predator PDM	0,177,172
	Predator RCM	137,122,26
	Predator Tracker	156,19,46
	Predator Travelers	255,200,46
	Predator Virtual CNC	0,48,140
	Predator Touch HMI	74,133,43
	Predator Software Blue	73,104,146

Predator MDC Status Colors

It can also be beneficial to make the color of a button match Predator MDC's default status colors. Refer to the following list for these colors and their RGB values:

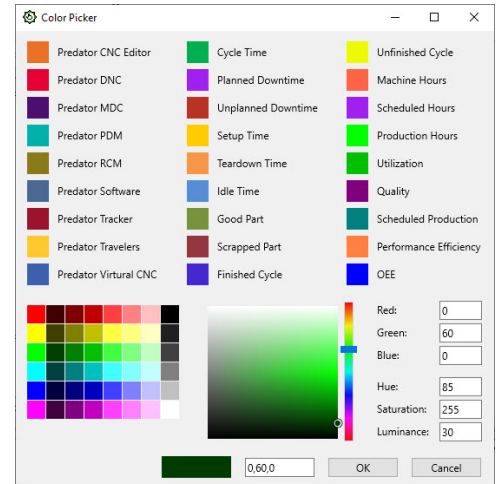
Color	Color Name	RGB Values
	MDC Cycle Time	0,176,80
	MDC Planned Downtime	160,32,240
	MDC Unplanned Downtime	185,51,37
	MDC Setup Time	255,204,0
	MDC Teardown Time	247,150,70
	MDC Idle Time	85,142,213
	MDC Good Parts	119,147,60
	MDC Scrapped Parts	148,55,63
	MDC Finished Cycles	70,40,209
	MDC Unfinished Cycles	238,250,5
	MDC Machine Hours	255,99,71
	MDC Scheduled Hours	160,32,240
	MDC Production Hours	0,255,0
	MDC Utilization	0,192,0
	MDC Quality	128,0,128
	MDC Scheduled Production	0,128,128
	MDC Performance Efficiency	255,128,64
	MDC OEE	0,0,255

Working with Gradient Colors in Touch HMI v12

Since we have added the new Gradient Color function for Buttons with Predator Touch HMI v12, we need to talk a bit about how to create good looking color gradients on Buttons.

Your first step is to choose a good color for your Button to start with. The new Predator Color Picker dialog can be a great help in selecting a color, as it displays all of the standard colors for the Predator products and the default colors for the MDC statuses. Once you have a color selected, you can copy the R,B,G values to the Button in the Properties Editor or simple select the color in the Color Picker and click on the **[OK]** Button to see what it looks like on your Button.

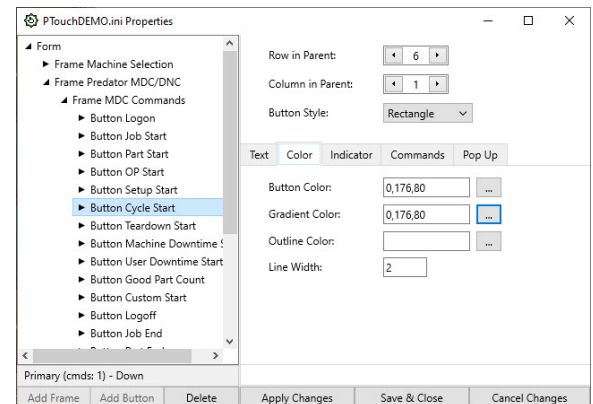
Now we need to address the 2nd step which is to pick a good complementing color for the Gradient Color field. When it comes to gradient colors you can go lighter towards the bottom of the button, or you can go darker. An easy trick to picking a complementing color is to use the alternate color format of H,S,L, or Hue, Saturation, and Luminance.



Let's look at the colors for the Buttons in the PTouchDEMO example INI file. If we open the configuration for the Cycle Start Button, we see that the Button Color is set to an R,B,G of **0,176,80** which is a bright Green color.



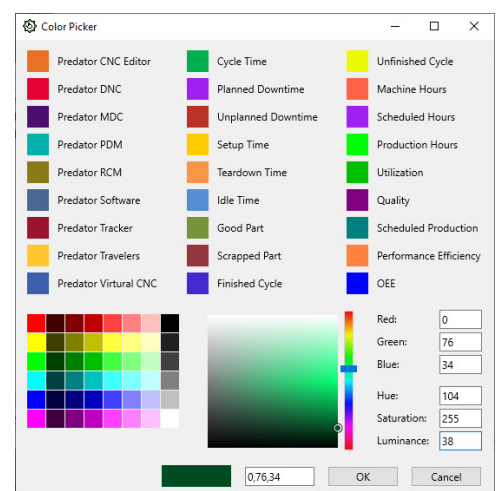
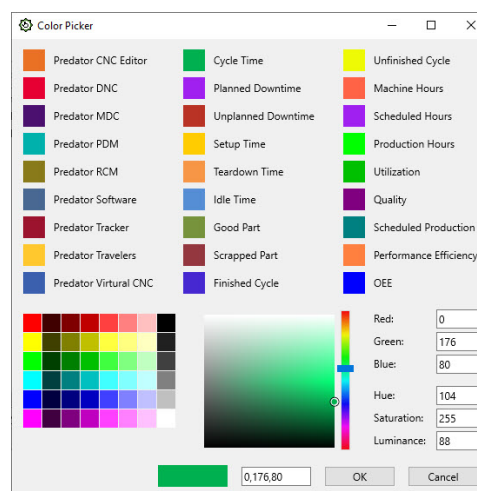
If we copy that color from the Button Color to the Gradient Color field and then open the Color Picker Tool, we can see that the same color has a HSL value of **104,255,88**.



If we simply lower the value for Luminance by 30 to 50 points, we will get a slightly darker shade of the same color. This makes a very nice-looking Gradient color on a Button.



Reversing the colors for the Button and Gradient fields gives the alternate gradient effect.



You will also notice that by making these changes the R,G,B values are nothing alike between the two colors, so we need to the HSL values to find a complementing color for the Gradient value.

This method of editing a Button color and drawing down the Luminance value for the Gradient side can lead to some very good looking buttons for your Touch HMI Panels.

Touch HMI v10 - v11 Interface Notes

The Predator Touch HMI Interface has evolved over the last few major revisions. This section describes the differences between the new Touch HMI v12 Interface and the older v10 and v11 versions.

Upgrading Touch HMI INI File from Version 10 to Version 11

In general version 10 INI files are upgradable to version 11 with minimal work. All Ethernet or Program buttons should work automatically. The only buttons that won't work are any close or exit buttons. With version 10 each button was assigned a specific type via the TYPE property. In version 11 this was eliminated because every button can run multiple function types. If a BUTTONS or TYPE property is found in the INI File, it will automatically be ignored by Version 11 of the Predator Touch HMI.

Eliminating the BUTTONS property was done to simplify HMI definition. Eliminating the TYPE property was done to support multiple functions per button. This change prevents existing exit buttons from working. In v10 an exit button could be defined with:

Type = Exit (v10)

The direct replacement to close the HMI is:

Exit = Silent (v11)

[NOTE: It is best practice to remove BUTTONS= and TYPE= lines when upgrading from Touch HMI v10 to v11.]

The v10 - v11 Application.ini File

Predator Touch HMI v10 and v11 Application.ini files do not support all of the functions described in the Touch HMI v12 Section. The v10 and v11 Application.ini files only had two main sections:

[Form]

[Button#]

Use of Comments

Within the .INI file it is often desired to comment out a line when testing a new .INI file configuration. Typing a semi colon character at the beginning of a line will comment out that line. In other words that line will be ignored by Predator Touch HMI.

For example:

;Color = 255,255,255

Color = 0,0,0

In the above example the first Color line will be ignored, and the second Color line will be honored.

The [Form] Section

The Touch HMI v10 and v11 **[Form]** section of the TouchHMI.ini file included the following property variables:

Left
Top
Width
Height
State
IP Address
Port
Rows
Columns
Timeout
Color
Stay on Top
Minimize
Maximize
Exit
Title
Serial Port
Baud Rate
Parity
Data Bits
Stop Bits
RFID Protocol
RFID IP Address
RFID Port

The **Left**, **Top**, **Width**, **Height**, and **State** variables are automatically updated whenever the Predator Touch HMI window is resized, maximized, or moved on the Windows Desktop. Although they can be changed manually within the .INI file it is not recommended, as these values will be overwritten when the Panel is moved on the Desktop.

***[NOTE:** If the “application.ini” file is set to read only then Predator Touch HMI will not be able to update the Left, Top, Width, Height, and State variables automatically. Often this is desired to force the application to always start in a known location, size, and state on the Windows Desktop.]*

***[NOTE:** If an “application.ini” file is copied from one PC to another with differing screen size or resolution, it is possible that the Touch HMI Panel may not be visible when run on the second PC. This due to one or more of the Left, Top, Width, and Height variables being set to a value that falls outside of the screen resolution of the new PC. This is one of the times were editing these values manually in the INI file is warranted.]*

The [Button#] Section

The Button Section of the Touch HMI v10 and v11 .INI file was similar to what we have today in Touch HMI v12 with a few exceptions:

- The **[Button#]** heading for each Button had a number associated with it in the following format **[Button1]** to indicate its order in a Touch HMI panel. If you edited a Touch HMI INI file to add or remove Buttons, you had to manually edit all of the button numbers to be in the order you wanted them to appear in the Panel. This functionality has been removed from Touch HMI v12 and any Button numbers found in an INI file with Touch HMI v12 will be ignored.
- Buttons in Touch HMI v10 and v11 did not support the new Pop-Up Menu option for Buttons in Touch HMI v12.

The following is an example of the Button Configuration of a Touch HMI v11 INI file:

[Button1]

Row = 1
Column = 1
Text = Line Down
Color = 185,51,37
Ethernet = MSLD

[Button2]

Row = 1
Column = 2
Text = Restarted Line
Color = 166,166,166
Ethernet = ME

[Button3]

Row = 2
Column = 1
Text = Send Queued
Color = 231, 0, 51
Serial = LOAD

[Button4]

Row = 2
Column = 2
Text = Send Tool Offsets\13\10to CNC
Color = 231, 0, 51
RFID=1
Ethernet = LOAD \RFID1.offset

Touch HMI v12 Interface Notes

The Predator Touch HMI v12 Interface has continued to evolve since the first versions were released. The latest version (v12.0.25.28) has a number of changes to the INI file that have taken place over the last few releases. This section describes the differences between the latest Touch HMI v12 INI Interface and earlier versions.

Converting v12 Touch HMI INI Files to Version 12.0.25.x

The latest version of the Touch HMI INI file has some changes that have been made to better support a wider range of functions. These include:

- Changes to a number of Variables including WindowsUser, ComputerName, TouchHMITitle, MachinePort, and IPv4 to name a few.
- The set of \A Replacement Variables were also changed from simply \Ax to \Argumentx. This happened due to some user data having the “\A” string in their arguments that could cause issues with the functions of a Touch HMI Command.
- Hex Variables were also changed from simply \xFF to \HexFF to make sure that the user understands that these are Hex values and not ASCII.

To accommodate these changes, a new Command Line Argument was created (-convert) to auto-update an existing Touch HMI v12 INI File when run with the -convert option. This option can be used within a Windows Shortcut to run the conversion when the file is opened. The syntax needed to use this option is:

```
"C:\Predator Software\Touch HMI 12.0\[TouchHMI INI filename].exe" -convert
```

When executed, a new “Version” option will be added to the [Form] Section of the INI File to indicate which version of the Touch HMI EXE file did the conversion (i.e. Version=12.0.25.28).

Next the conversion will update all of the new Argument Names in the entire INI file with the latest versions, so:

- A variable of \A1=Haas 101 would be changed to \Argument1=Haas 101.
- A variable of Port=11000 would be changed to MachinePort=11000.

... And so on.

Starting with the release of v12.0.25.25, the standard PTouchHMI.exe file has been updated to add the “-convert” command line argument so you can create a Desktop Shortcut with this command line option and use it to automatically update all of your TouchHMI.ini files with the latest variable names.

The Touch HMI Properties Editor Conventions

As of version v12.0.25.28 of the Touch HMI Properties Editor, we have made some additional changes to the auto-correct features of this tool. Here are the new features and format changes to the Touch HMI .INI file structure:

As of Touch HMI v12.0.25.28, the Touch HMI Properties Editor will support the following:

1. The **TouchHMI.ini** file can have any text or comments in the file as long as they appear either above the **[Form]** heading in the file, or below the **[EndForm]** heading in the file. No special formatting is required in these two areas of the .INI file for comments.
2. If you want to add comments within a **[Form]**, **[Frame]**, or **[Button]** section of a TouchHMI.ini file, they must begin with a **semicolon character (;)** to indicate that the line is a comment, and not part of the INI file to be processed. When a TouchHMI file is edited with this latest version of the Touch HMI Properties Editor, any comments found in the body of an INI file will be moved to the top of the section where they were found.
3. When a TouchHMI file is edited with this latest version of the Touch HMI Properties Editor, any lines in an INI file that do not match the current syntax for that version of TouchHMI v12 will be commented out using a **colon character (:)** and moved to the bottom of the section where the text was found. These lines will not be processed when the Panel runs and can be deleted by the user if they want to clean up the .INI file. This allows a user to update their old TouchHMI.ini files to the latest version and syntax without the need to go through each TouchHMI .ini file, line by line. As described above, lines beginning with a “;” will be ignored and left as comments in the file.

***[NOTE:** The actions of the Touch HMI Properties Editor listed above will only take place if a TouchHMI Panel is opened, edited, and then saved with the Properties Editor. If a file is opened and then closed with no changes made, then none of these changes to the .INI file will be processed.]*

Touch HMI v12 Examples

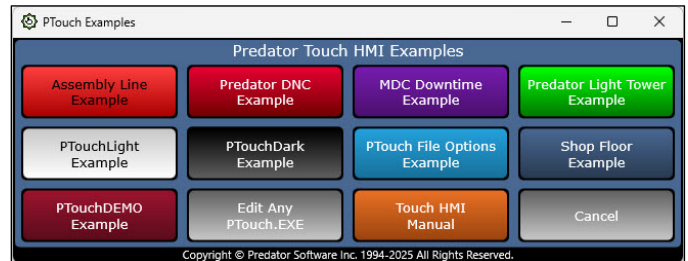
This section outlines some of the Touch HMI Examples (*.ini files) that are included as part of the deliverables with Touch HMI Version 12.

PTouchHMI.exe

The main Predator Demo Touch HMI EXE is called “**PTouchHMI.exe**”. It is delivered with a default INI file that will display the following Touch HMI Panel.

This Touch HMI panel has 12 Buttons with other examples listed and uses the new Touch HMI v12 Format. All of these examples are delivered with Touch HMI v12 as INI files ONLY.

The 12 Buttons are:



- **Assembly Line Example** - This Button copies the PTouchHMI.exe to **Assembly Line.exe** and runs the example Touch HMI Panel with the **-properties** argument and allows you to view and edit it in the new Touch HMI v12 Properties Editor.
- **Predator DNC Example** - This Button copies the PTouchHMI.exe to **DNC.exe** and runs the example Touch HMI Panel with the **-properties** argument and opens it in the new Touch HMI v12 Properties Editor.
- **MDC Downtime Example** - This Button copies the PTouchHMI.exe to **Downtime.exe** and runs the example Touch HMI Panel with the **-properties** argument and opens it in the new Touch HMI v12 Properties Editor.
- **Predator Light Tower Example** - This Button copies the PTouchHMI.exe to **Predator Light Tower.exe** and runs the example Touch HMI Panel with the **-properties** argument and opens it in the new Touch HMI v12 Properties Editor.
- **PTouchLight Example** - This Button copies the PTouchHMI.exe to **PTouchLight.exe** and runs the example Touch HMI Panel with **-properties** argument and opens it in the new Touch HMI v12 Properties Editor.
- **PTouchDark Example** - This Button copies the PTouchHMI.exe to **PTouchDark.exe** and runs the example Touch HMI Panel with **-properties** argument and opens it in the new Touch HMI v12 Properties Editor.
- **PTouch File Options Example** - This Button copies the PTouchHMI.exe to **PTouchFileOptions.exe** and runs the example Touch HMI Panel with **-properties** argument and opens it in the new Touch HMI v12 Properties Editor.
- **Shop Floor Example** - This Button copies the PTouchHMI.exe to **Shop Floor.exe** and runs the example Touch HMI Panel with **-properties** argument and opens it in the new Touch HMI v12 Properties Editor.
- **PTouchDEMO Example** - This Button copies the PTouchHMI.exe to **PTouchDEMO.exe** and runs the example Touch HMI Panel with **-properties** argument and opens it in the new Touch HMI v12 Properties Editor.
- **Edit Any PTouch EXE** - This Button will launch a Windows “FileOpen” Dialog Box prompting you to select any existing v12 Touch HMI “*.exe” file so that it can be launched with the new **-properties** argument and allow you to edit it in the new Touch HMI v12 Properties Editor. This Button can be very useful if you have a number of older v10 or v11 Touch HMI Panels and you want to update them with the new Touch HMI v12 format and features.

[NOTE: To use the new v12 Configuration Utility with an older v10 or v11 Touch HMI INI file, you will first have to replace the *.exe file with a copy of the latest Touch HMI v12 exe and then rename it to match the name of the older v10 or v11 INI File. If you don't do this first, this Option on this panel will simply open the older v10 or v11 Touch HMI Panel without the Configuration Utility as it did not exist in the v10 or v11 Touch.exe files.]

- **Touch HMI Manual** - This Button will launch this Touch HMI Manual in PDF format for your review.
- **Cancel** - This button will cancel this TouchHMI.exe panel and close the window.

Here is what the **TouchHMI.ini** file looks like for this Touch HMI Panel:

Copyright © 1994 - 2025 Predator Software Inc. All rights reserved.

```
[Form]
;This is the Main Touch HMI DEMO Panel delivered with Predator Touch HMI v12.
;It can be used to launch all of the other DEMO TouchHMI INI Files.
Version=12.0.25.28
Top=40.5
Left=35.42
Width=34.95
Height=25
Title=PTouch Examples
Color=0,0,0
Rows=1
Columns=1
Auto Center=True
Stay On Top=False
Minimize=True
Maximize=True
Exit=True
IP Address=127.0.0.1
Port=1024
Serial Port=COM1
Baud Rate=9600
Parity=EVEN
Data Bits=7
Stop Bits=2
Monitor=1
Padding=1

[Button Defaults]
Text Position=Center
Font Color=Auto
Indicator On Color=14,255,144
Indicator Off Color=14,255,144

[Frame]
Parent Row=1
Parent Column=1
Text=Predator Touch HMI Examples
Color=73,104,146
Outline=0,0,0
Line Width=1
Rows=3
Columns=4
Width=100%
Height=100%
Padding=5
```

```
[Button]
Row=1
Column=1
Style=Rectangle
Text=Assembly Line\13\10Example
Color=255,64,64
Gradient=170,0,0
Outline=0,0,0
Line Width=2
Copy=PTouchHMI.exe,Assembly Line.exe
Program=Assembly Line.exe
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=1
Column=2
Style=Rectangle
Text=Predator DNC\13\10Example
Color=231,0,51
Gradient=113,0,1
Outline=0,0,0
Line Width=2
Copy=PTouchHMI.exe,DNC.exe
Program=DNC.exe
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=1
Column=3
Style=Rectangle
Text=MDC Downtime\13\10Example
Color=119,28,175
Gradient=75,16,111
Outline=0,0,0
Line Width=2
Copy=PTouchHMI.exe,Downtime.exe
Program=Downtime.exe
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=1
Column=4
Style=Rectangle
Text=Predator Light Tower\13\10Example
Color=0,255,0
Gradient=0,120,0
Outline=0,0,0
Line Width=2
Copy=PTouchHMI.exe,Predator Light Tower.exe
```

```
Program=Predator Light Tower.exe
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=2
Column=1
Style=Rectangle
Text=PTouchLight\13\10Example
Color=200,200,200
Gradient=255,255,255
Outline=0,0,0
Line Width=2
Copy=PTouchHMI.exe,PTouchLight.exe
Program=PTouchLight.exe
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=2
Column=2
Style=Rectangle
Text=PTouchDark\13\10Example
Color=0,0,0
Gradient=96,96,96
Outline=0,0,0
Line Width=2
Copy=PTouchHMI.exe,PTouchDark.exe
Program=PTouchDark.exe
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=2
Column=3
Style=Rectangle
Text=PTouch File Options\13\10Example
Color=36,161,221
Gradient=23,112,153
Outline=0,0,0
Line Width=2
Copy=PTouchHMI.exe,PTouchFileOptions.exe
Program=PTouchFileOptions.exe
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=2
Column=4
Style=Rectangle
Text=Shop Floor\13\10Example
Color=73,104,146
Gradient=41,58,81
```

```
Outline=0,0,0
Line Width=2
Copy=PTouchHMI.exe,Shop Floor.exe
Program=Shop Floor.exe
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=3
Column=1
Style=Rectangle
Text=PTouchDEMO\13\10Example
Color=156,19,46
Gradient=91,12,28
Outline=0,0,0
Line Width=2
Copy=PTouchHMI.exe,PTouchDEMO.exe
Program=PTouchDEMO.exe
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=3
Column=2
Style=Rectangle
Text=Edit Any\13\10PTouch.EXE
Color=96,96,96
Gradient=200,200,200
Outline=0,0,0
Line Width=2
Program=\\OSFileOpen("Select *.EXE to Open","C:\\Predator Software\\Touch HMI 12.0\\")
Arguments=-properties
Pop-Up Button Text=False
```

```
[Button]
Row=3
Column=3
Style=Rectangle
Text=Touch HMI\13\10Manual
Color=234,113,37
Gradient=155,68,15
Outline=0,0,0
Line Width=2
Program="C:\\Predator Software\\Touch HMI 12.0\\Predator Touch HMI v12.0 Reference Manual.pdf"
Pop-Up Button Text=False
```

```
[Button]
Row=3
Column=4
Style=Rectangle
Text=Cancel
Color=96,96,96
Gradient=200,200,200
```

Predator Touch HMI



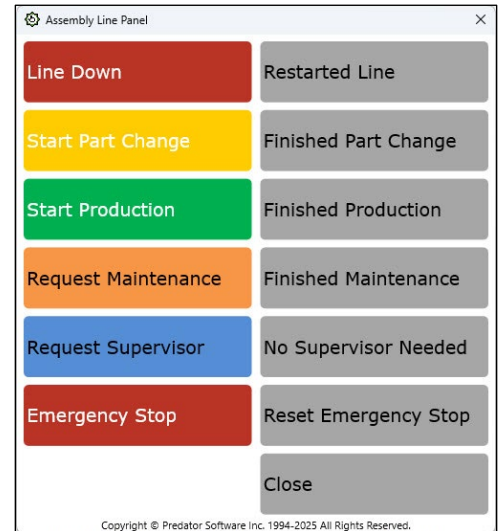
```
Outline=0,0,0  
Line Width=2  
Exit=Silent  
Pop-Up Button Text=False
```

```
[EndFrame]  
[EndForm]
```

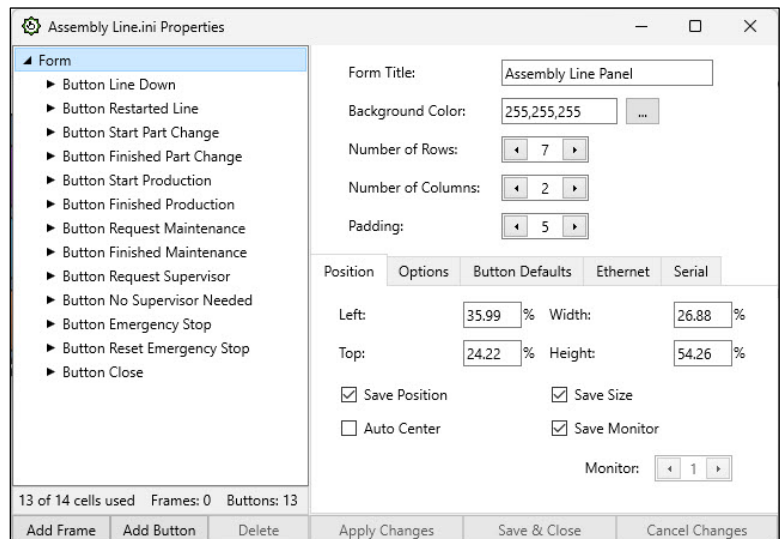
Assembly Line Example

The **Assembly Line** Demo Touch HMI Panel is an older Touch HMI v11 example that shows a simple Panel to enable the user to enter Assembly Line Requests from the Shop Floor. This INI file is included in the installation of Touch HMI v12 to help a new user visualize the many options available to them.

This Touch HMI Panel has no Frames and a total of 13 Buttons in the configuration.



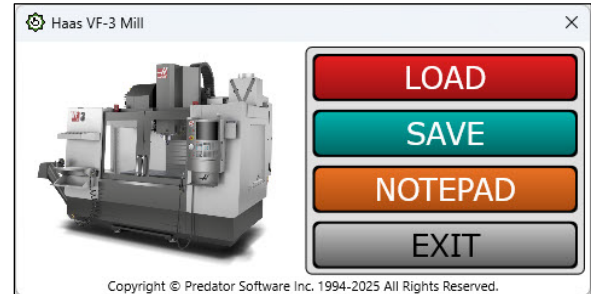
The Configuration Panel for this Touch HMI v12 example in the Touch HMI v12 Properties Editor (shown on the right) gives a much more understandable layout of the INI configuration format and allows the user to make changes to the panel without knowing all the exact syntax for the INI file required to make those changes.



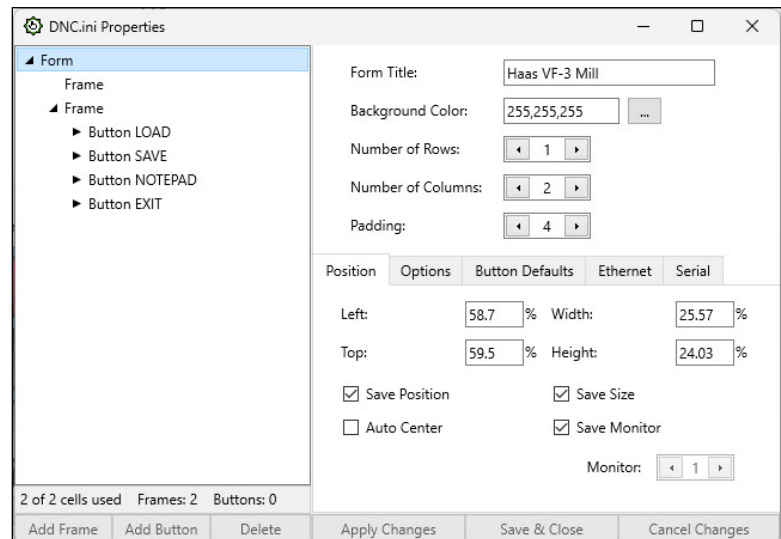
Predator DNC Example

The **Predator DNC** Demo Touch HMI Panel is a new Touch HMI v12 example that demonstrates the use of Frames and images, some of the new options that are available for the configuration in Touch HMI v12. This shows a simple Panel to enable the user to launch DNC Commands from the Shop Floor. This INI file is included in the installation of Touch HMI v12 to help a new user visualize the many options available to them.

This Touch HMI Panel shows the use of Frames with a total of 2 Frames in the configuration, one for an Image of the CNC machine and the 2nd for a set of 4 Buttons for DNC Commands. Two Buttons are for selecting LOAD and SAVE options and two more for launching the Windows Notepad to view NC and text files and a CANCEL Button to close the panel.



The Configuration Panel for this Touch HMI v12 example in the Touch HMI v12 Properties Editor (shown on the right) gives a much more understandable layout of the INI configuration format and allows the user to make changes to the panel without knowing all the exact syntax for the INI file required to make those changes.



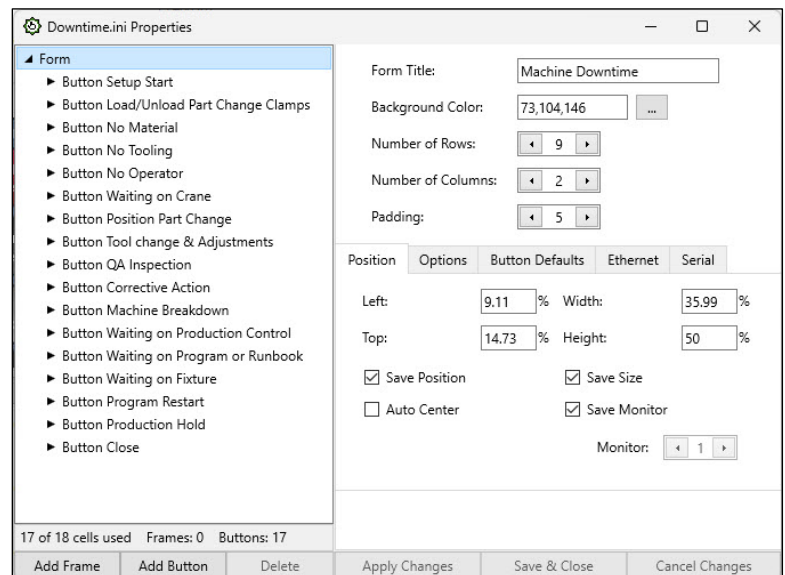
MDC Downtime Example

The **MDC Downtime** Demo Touch HMI Panel is another older Touch HMI v11 example that shows a simple Panel to enable the user to enter MDC Downtime Reasons from the Shop Floor. This INI file is included in the installation of Touch HMI v12 to help a new user visualize the many options available to them.

This Touch HMI Panel has no Frames and a total of 17 Buttons in the configuration.



The Configuration Panel for this Touch HMI v12 example in the Touch HMI v12 Properties Editor (shown on the right) gives a much more understandable layout of the INI configuration format and allows the user to make changes to the panel without knowing all the exact syntax for the INI file required to make those changes.



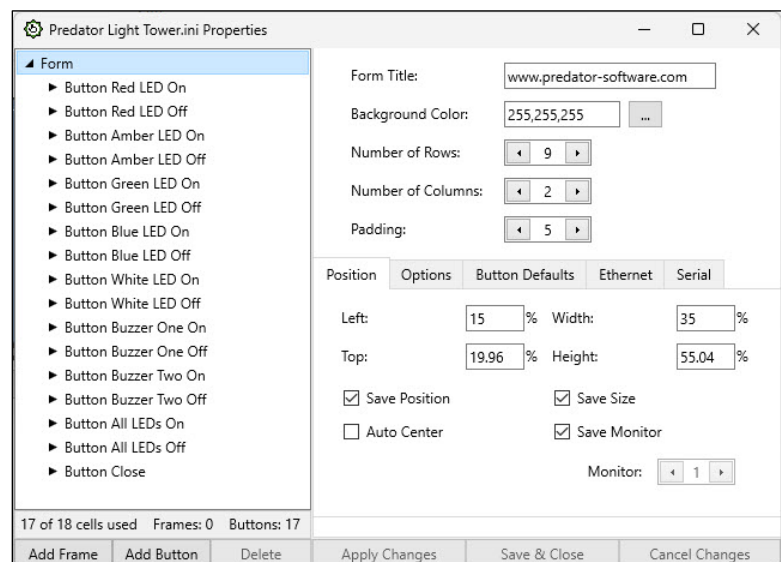
Predator Light Tower Example

The **Predator Light Tower** Demo Touch HMI Panel is another older Touch HMI v11 example that shows a simple Panel to enable the user to control the lights and buzzers on a Light Tower. This INI file is included in the installation of Touch HMI v12 to help a new user visualize the many options available to them.

This Touch HMI Panel has no Frames and a total of 17 Buttons in the configuration.



The Configuration Panel for this Touch HMI v12 example in the Touch HMI v12 Properties Editor (shown on the right) gives a much more understandable layout of the INI configuration format and allows the user to make changes to the panel without knowing all the exact syntax for the INI file required to make those changes.

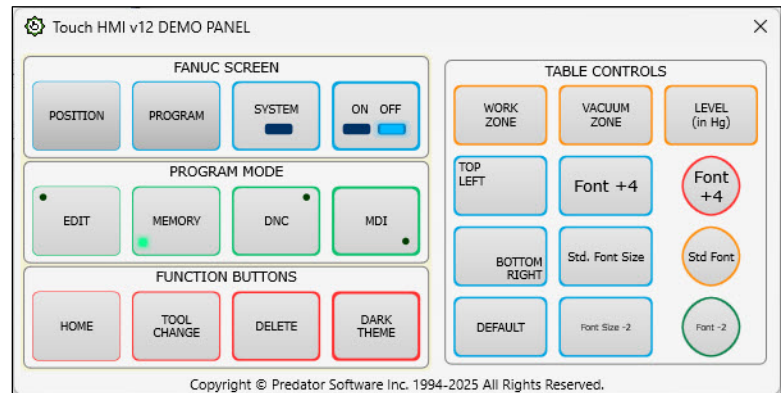


PTouchLight Example

The **PTouchLight** Demo Touch HMI Panel is a new Touch HMI v12 example that demonstrates the use of Frames and the many new options that are available for the configuration of Buttons in Touch HMI v12. This INI file is included in the installation of Touch HMI v12 to help a new user visualize the many options available to them.

This Touch HMI Panel shows the use of Frames with a total of 6 nested Frames in the configuration. It also has over 20 different Buttons in this example showing the many different configurations such as:

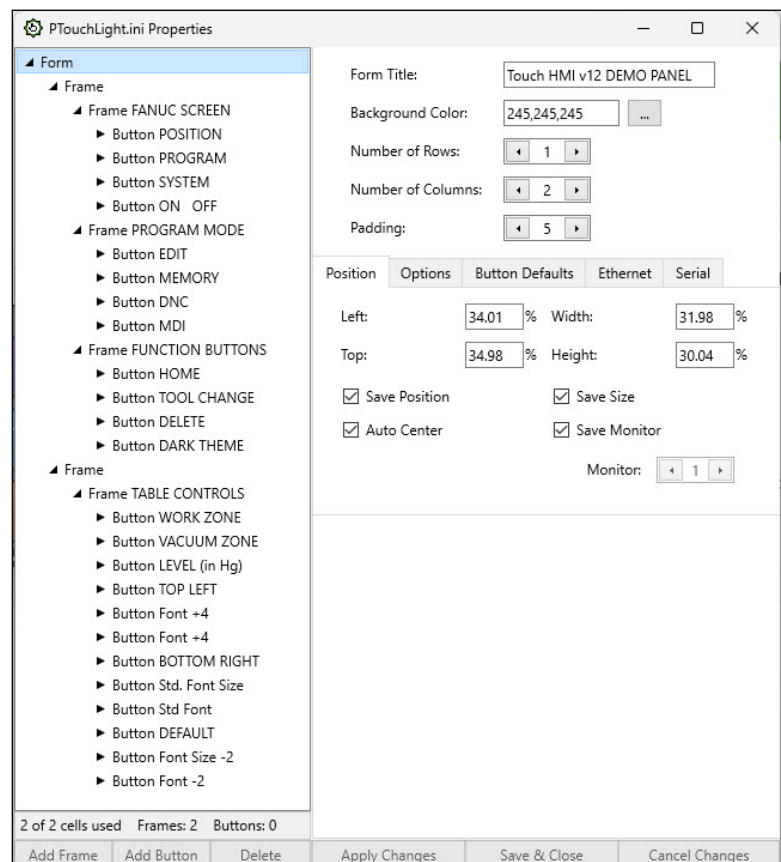
- Button Shape
- Button Color
- Button Gradient Color
- Button Indicators
- Text Size
- Text Location



The Configuration Panel for this Touch HMI v12 example in the Touch HMI v12 Properties Editor (shown on the right) gives a much more understandable layout of the INI configuration format and allows the user to make changes to the panel without knowing all the exact syntax for the INI file required to make those changes.

The **PTouchLight** Demo Touch HMI Panel is a Touch HMI v12 example that is the same as the **PTouchDark** Demo only with a light color background for the Frames and Buttons to highlight how a simple change like the background color can totally change the “look and feel” of a Touch HMI Panel.

The PTouchLight and PTouchDark examples also use the **ReloadINI** Command in one of their Buttons in the FUNCTION BUTTONS Frame to allow you to switch back and forth between the two INIs by clicking on the “THEME” Buttons.

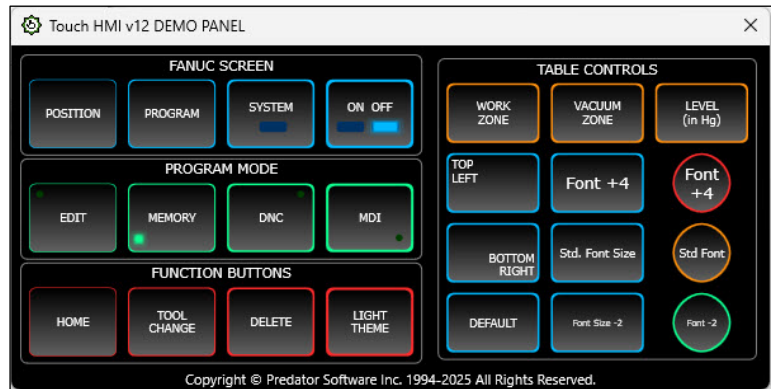


PTouchDark.exe

The **PTouchDark** Demo Touch HMI Panel is a new Touch HMI v12 example that demonstrates the use of Frames and the many new options that are available for the configuration of Buttons in Touch HMI v12. This EXE and INI combination are included in the installation of Touch HMI v12 to help a new user visualize the many options available to them.

This Touch HMI Panel shows the use of Frames with a total of 6 nested Frames in the configuration. It also has over 20 different Buttons in this example showing the many different configurations such as:

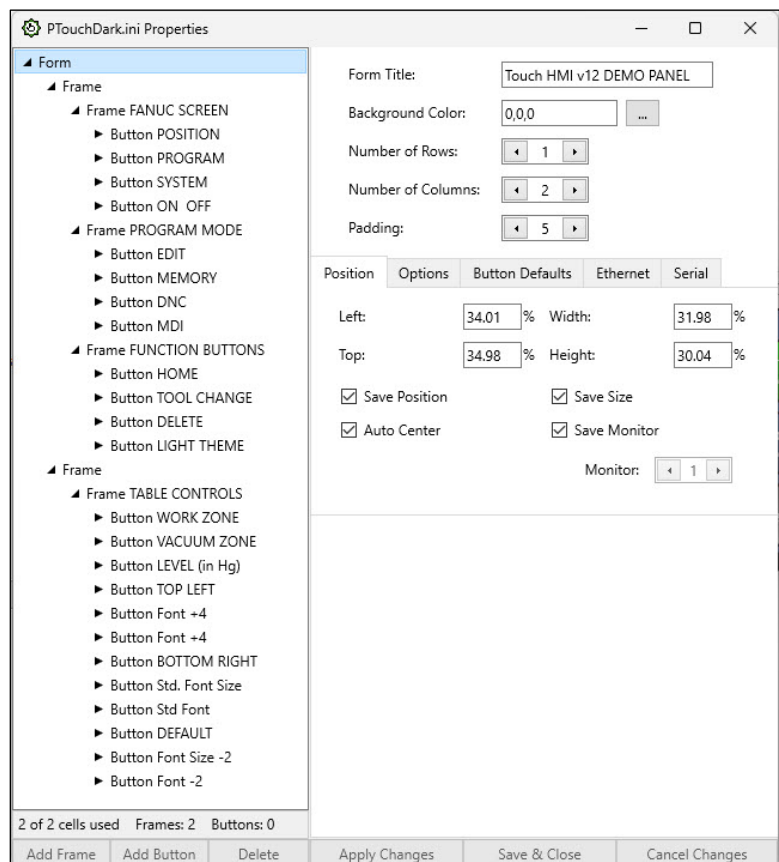
- Button Shape
- Button Color
- Button Gradient Color
- Button Indicators
- Text Size
- Text Location



The Configuration Panel for this Touch HMI v12 example in the Touch HMI v12 Properties Editor (shown on the right) gives a much more understandable layout of the INI configuration format and allows the user to make changes to the panel without knowing all the exact syntax for the INI file required to make those changes.

The **PTouchDark** Demo Touch HMI Panel is a Touch HMI v12 example that is the same as the **PTouchLight** Demo only with a dark color background for the Frames and Buttons to highlight how a simple change like the background color can totally change the “look and feel” of a Touch HMI Panel.

The PTouchLight and PTouchDark examples also use the **ReloadINI** Command in one of their Buttons in the FUNCTION BUTTONS Frame to allow you to switch back and forth between the two INIs by clicking on the “THEME” Buttons.

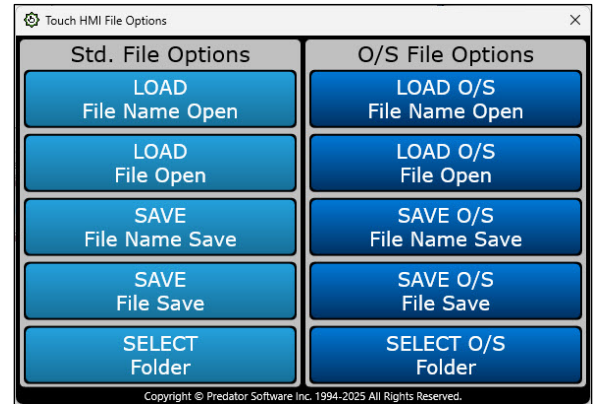


PTouch File Options Example

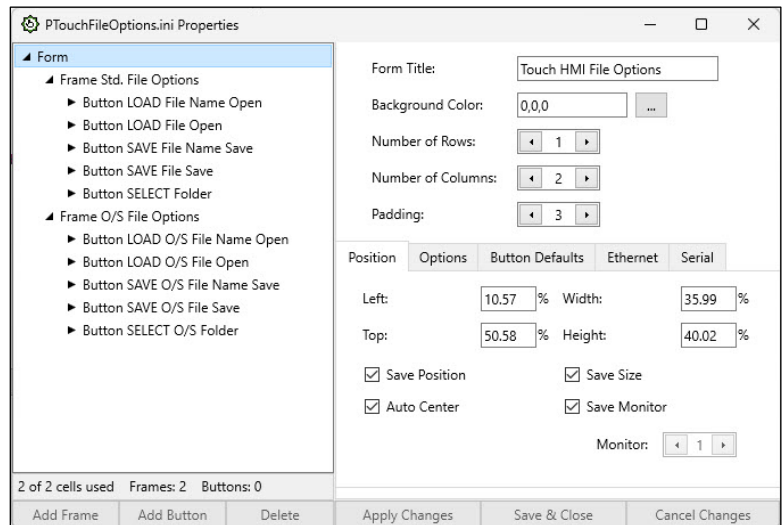
The **PTouch File Options** Demo Touch HMI Panel is a Touch HMI v12 example that shows all of the File Options and Prompts that can be used in Predator Touch HMI. This INI file is included in the installation of Touch HMI v12 to help a new user visualize the many options available to them.

This Touch HMI Panel has 2 Frames and a total of 10 Buttons in the configuration. The Buttons in the Left-side Frame are all using the standard Predator File command examples where the user is limited to where they can browse for accessing files.

The Buttons in the Right-side Frame are all using the Windows O/S File command examples where the user can access any file location that his current Windows User Rights allow. This example was designed to show the differences between these two sets of file access prompts in Predator Touch HMI v12.



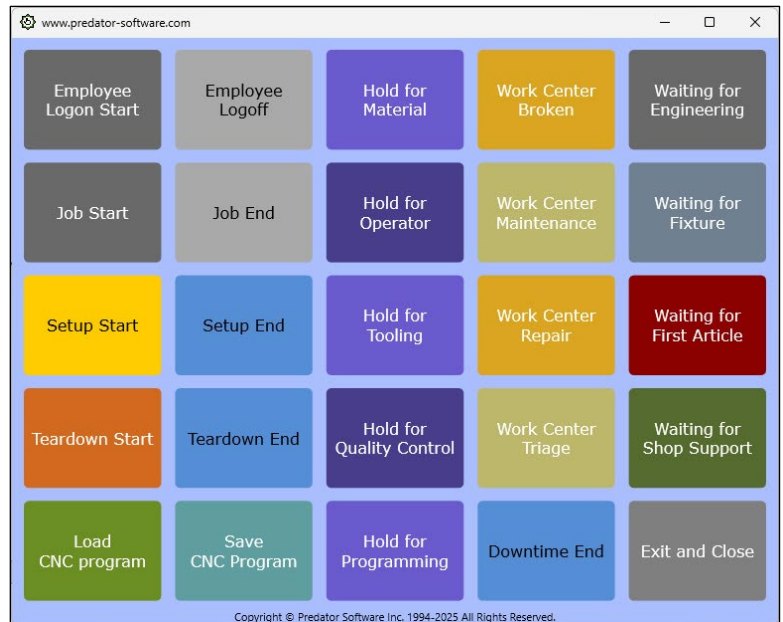
The Configuration Panel for this Touch HMI v12 example in the Touch HMI v12 Properties Editor (shown on the right) gives a much more understandable layout of the INI configuration format and allows the user to make changes to the panel without knowing all the exact syntax for the INI file required to make those changes.



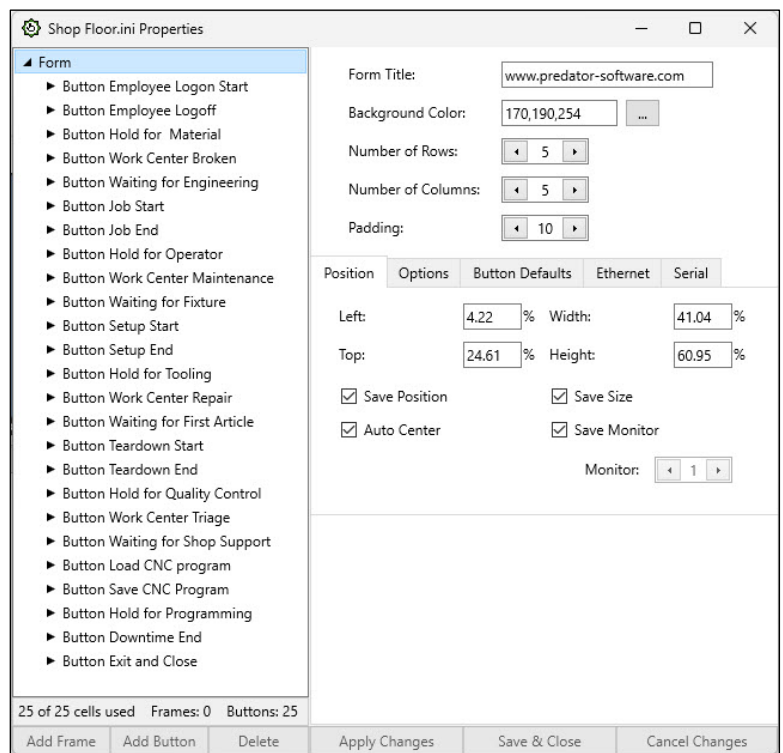
Shop Floor Example

The **Shop Floor** Demo Touch HMI Panel is another older Touch HMI v11 example that shows a simple Panel to enable the user to enter Shop Floor level status messages for Predator MDC. This INI file is included in the installation of Touch HMI v12 to help a new user visualize the many options available to them.

This Touch HMI Panel has no Frames and a total of 25 Buttons in the configuration.



The Configuration Panel for this Touch HMI v12 example in the Touch HMI v12 Properties Editor (shown on the right) gives a much more understandable layout of the INI configuration format and allows the user to make changes to the panel without knowing all the exact syntax for the INI file required to make those changes.



Predator Touch HMI



PTouchDEMO Example

The **PTouchDEMO** Touch HMI Panel is a new Touch HMI v12 example that demonstrates many of the new features in Touch HMI v12. This example covers the use of Frames, Drop-down Option Lists, and many more new features that are available for the configuration of Frames and Buttons in Touch HMI v12. This EXE and INI combination are included in the installation of Touch HMI v12 to help a new user visualize the many options available to them.

This Touch HMI Panel highlights the use of Frames, with a total of 4 nested Frames in this one Panel. It also has over 28 different Buttons available and 3 sets of Option Buttons.

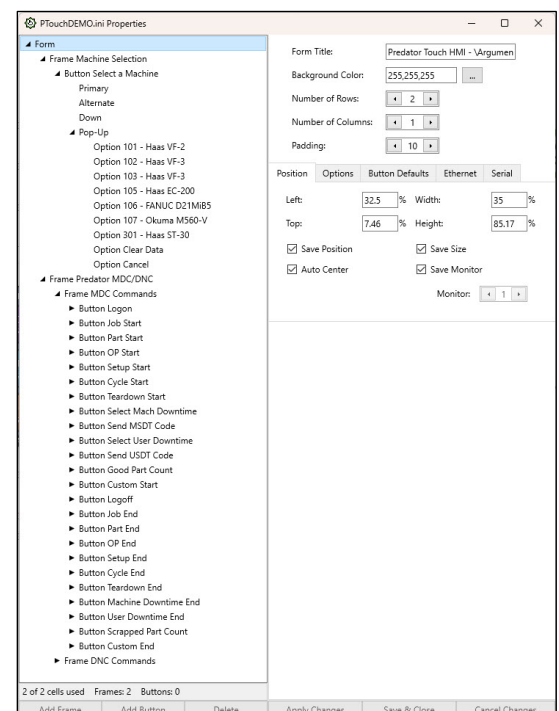
The first of these Frames enables a user to select a machine before executing any of the other commands. Once a machine is selected, all of the variables for that machine are set and processed via the buttons in the Frames below for MDC and DNC commands.

Then there is a second Frame below the first that has two additional nested Frames inside of it, one for sending MDC Commands for the selected machine and one for executing DNC Commands for that machine. The **“Select Mach Downtime”** and **“Select User Downtime”** Buttons also have Drop-down Lists where the user can select from a list of Downtime codes to send to Predator MDC.

With this Demo Panel we can provide the user with all the features they need to manage a set of CNC machines with only one TouchHMI.exe file and one TouchHMI.ini file. This would have taken an entire series of TouchHMI panels and configurations with multiple .EXE and .INI files with TouchHMI v10 or v11.

The Configuration Panel for this Touch HMI v12 example in the Touch HMI v12 Properties Editor (shown on the right) gives a much more understandable layout of the complex INI configuration format for this panel. It allows the user to make changes to the panel without knowing all the exact syntax for the INI file required to make those changes.

A detailed description of this Demo Panel is covered in the Tutorial on Creating a Drop-down Options List earlier in this manual.



Touch HMI Version History

Predator Software maintains a complete list of all the Released Versions for each product on our website. These lists contain each version that has been released, along with a list of the new features and any bug fixes that the version addressed.

Touch HMI v12.0.2x.x

The following is the link to the current list of releases for Predator Touch HMI v12:

https://www.predator-software.com/predator_touch_hmi_software_v12_history.htm

Touch HMI v11.0.22.7 (Oct. 12, 2022)

- **This was the final Touch HMI v11 release**
- Improved \Folder function to automatically convert mapped drives to UNC paths
- Improved \FileOpen function to automatically convert mapped drives to UNC paths
- Improved \FileSave function to automatically convert mapped drives to UNC paths

Touch HMI v10.0.5.27 (Nov. 23, 2015)

- **This was the final Touch HMI v10 release**
- Improved default install folder