PATHPILOT USER GUIDE FOR PLASMA

RELEASE NOTES, INTERFACE OVERVIEW, TOOLS AND FEATURES, PROGRAMMING

TORMACH®

TORMACH.COM

Copyright

Notice

Information is subject to change without notice by Tormach, Inc. For the most recent version of this document, see tormach.com/support.

You're welcome to make copies of this document for evaluating, learning about, and/or using the machine. You may not charge for any copies you make beyond the cost of printing.

Unless otherwise noted, companies, names, and various data used in examples are fictitious.

To the Reader

We're dedicated to continually improving our documentation and products, and welcome any clarifications, corrections, or suggestions.

Credits

Tormach®, 1300PL®, and PathPilot® are trademarks or registered trademarks of Tormach, Inc. Our milling machines and accessories are covered by one or more of the following U.S. Patents: 7,386,362; D606,568; D612,406; D621,859; and other patent(s) pending.

Other product or company names may be the trademarks of their respective owners.

PREFACE

ABOUT THIS DOCUMENT

PURPOSE AND SCOPE

This document is intended to provide sufficient information to allow you to use your PathPilot controller. It assumes that you have appropriate experience and/or access to training for any computer-aided design or manufacturing software for use with the machine.

GETTING HELP

We provide no-cost technical support through multiple channels. The quickest way to get the answers you need is normally in this order:

1. Read this document.
2. Read related documents at tormach.com/support.
3. If you still need answers, gather the following information so that we may help you as quickly as possible:
    - Your phone number, address, and company name (if applicable).
    - Machine model and serial number, which are located next to the Main Disconnect switch.
    - The version of PathPilot that you're running.
    - Any accessories that you have for your machine.
    - A clear and concise description of the issue.
    - Any supporting media and information that you can share with us. For example, you could:
        - Analyze what might have changed since the machine last worked correctly.
        - Record a short video.
        - Take a picture of a part.
        - For software, share log data .zip files, screen captures, or program files.
          For information, see "Share Log Data .zip Files" (below).
        - From the PathPilot interface, on the Status tab, record any available information.
        - Use a digital multimeter for voltage readings.
4. Once you've gathered the information in Step 3, contact us in the following ways:
    a. Create a support ticket: Go to tormach.com/how-to-submit-a-support-ticket
    b. Phone: (608) 849-8381 (Monday through Friday, 8 a.m. to 5 p.m. U.S. Central Standard Time)

SHARE LOG DATA .ZIP FILES

The controller keeps log data on how the machine has been working, which you can export as a .zip file. This information helps us troubleshoot software situations much faster.
To share log data .zip files:

1. Put a USB drive into the PathPilot controller.
2. From the PathPilot controller, on the **Status** tab, select **Log Data**.
   PathPilot creates a file called **logdata_[*TODAY'S-DATE*].zip**, and saves it on your USB drive.
3. Remove the USB drive from the controller. Create a support ticket with Tormach Technical Support at tormach.com/how-to-submit-a-support-ticket for guidance on how to proceed.

ADDITIONAL INFORMATION

For additional technical information and support videos, see tormach.com/support.

LIABILITY DISCLAIMER

We've made every effort to provide comprehensive and accurate information, but no warranty or fitness is claimed or implied. All information provided is on an as is basis. The authors, publisher, and Tormach, Inc. ("we", "us", and so on) shall not have any liability for, or responsibility to, any person or entity for any reason for any loss or damage arising from the information contained in this document.
This document provides guidance on safety precautions and techniques, but because the specifics of any one workshop or other local conditions can vary greatly, we accept no responsibility for machine performance or any damage or injury caused by its use. It's your responsibility to verify that you fully understand the implications of what you're doing and comply with any legislation and codes of practice applicable to your city, state, or nation.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

WHAT'S NEW?

IN THIS SECTION, YOU'LL LEARN:

- About the enhancements and fixed issues in the most recent version of PathPilot.

RELEASE NOTES FOR PATHPILOT V2.10.0

Fall 2023

Enhancements

All

We added:

- Support for a new on-screen keyboard. (PP-3497)
- Support for USB cameras.
  - "Dashcam" automatic 30 second video recording prior to hitting the emergency stop button. (PP-3525)
  - Manual video recording and image capture.
  - Video recording and image capture through the addition of three new M-codes.
  - A new M00/M01 workflow to simplify the creation of image files that can be shown during pause. (PP-3526)
- A new screen layout for portrait (vertical) orientation on 1920 × 1080 widescreen monitors. (PP-3522)
- Block delete functionality. You can now prefix any line of G-code with a slash (for example, `/G0 X0Y0Z0`). When you select the Block Delete button (toggle it on), PathPilot skips any line that begins with a slash (`/`). If Block Delete is toggled off, the lines execute as normal. (PP-3978)

Mills

- We split the **Coolant** button into a **Flood** and a **Mist** button on M, M+, and MX mills. For instructions on adding an additional mist outlet on your existing 1100/770 mill, refer to Tormach document SB10828. (PP-4139)

Fixed Issues

All

- We fixed an issue where saving files with a quotation mark (`"`) in the name prevented the file from loading. (PP-3701)

Mills

- We fixed an issue where incorrectly scaled feed rate values were shown in the DRO with G21 active during a pure A-axis motion. (PP-4115)

Lathes

- We fixed an issue where, in rare situations, the spindle on the 8L didn't stop when the enclosure door was opened. (PP-4091)

PATHPILOT INTERFACE OVERVIEW

IN THIS SECTION, YOU'LL LEARN:

- How PathPilot is organized, and where you can access each tool or feature.

CONTENTS

TORMACH®

ABOUT PATHPILOT

PathPilot is a combination hardware and software system that you use to control your machine. The controller hardware runs the PathPilot software.

The PathPilot interface is divided into sections: the Notebook section is in the top half of the screen, and the Persistent Controls section is in the bottom half.

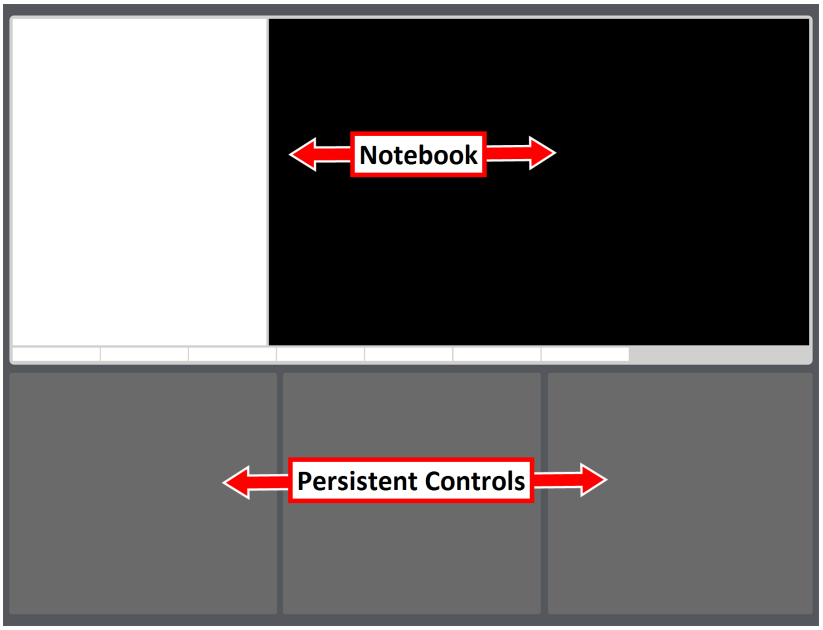


Figure 1: Sections in the PathPilot interface.

NOTEBOOK SECTION

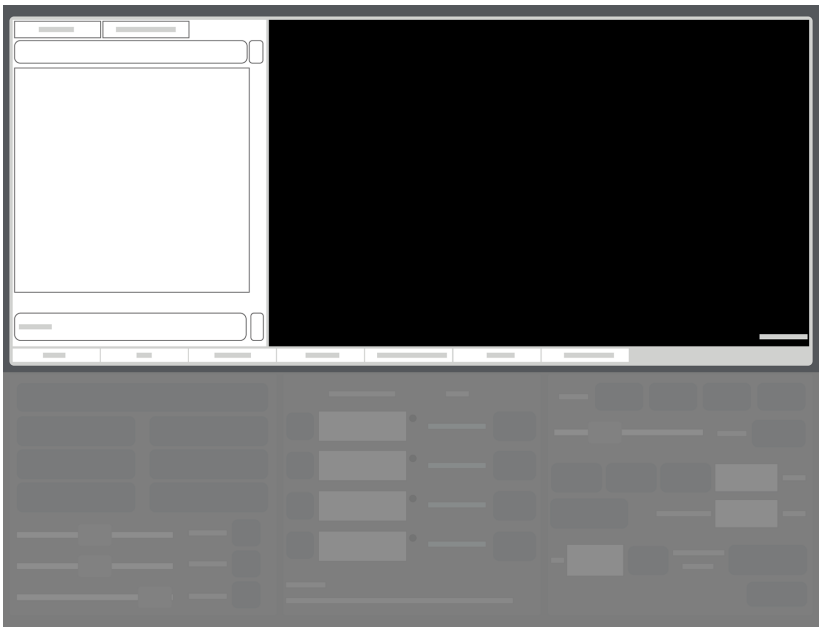


Figure 2: Notebook section.

The areas displayed in the Notebook section change depending on the activity that you're doing. Activities are grouped into the following tabs:

Main Tab

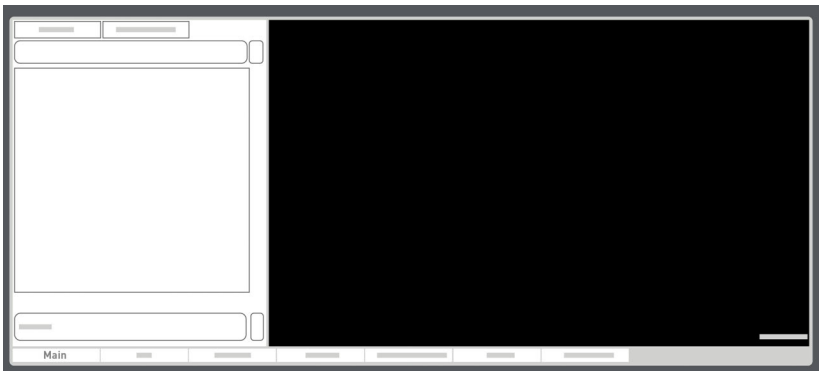


Figure 3: Main tab.

By default, the Main tab is active when you power on the PathPilot controller. From the Main tab, you can do the following activities:

- Access G-code files that are already loaded into PathPilot, and open or close them.
  For information, see "Access Recent G-Code Files" (page 37); "Close the Current Program" (page 37).
- Send G-code commands directly to the machine using the Manual Data Input (MDI) Line DRO field.
  For information, see "Manually Enter Commands" (page 97).
- In a G-code program, do tasks like finding specific terms in the code, reading the code, or viewing the generated tool path.
  For information, see "Search in the Code" (page 40); "Expand the G-Code Tab" (page 39); "Change the View of the Tool Path Display" (page 42).

File Tab

Figure 4: File tab.

From the File tab, you can do the following activities:

- Transfer G-code files into the PathPilot controller.
  For information, see "Transfer Files to and From the Controller" (page 35).
- Edit G-code files.
  For information, see "Edit G-Code with a Text Editor" (page 38).
- Load .nc files into PathPilot to run a program.
  For information, see "Load G-Code" (page 35).
- Move files within the system.
  For information, see "Preview G-Code Files" (page 36); "Manage System Files" (page 102).

Settings Tab

Figure 5: Settings tab.

From the Settings tab, you can do the following activities:

- Change the network name with which you're using PathPilot.
  For information, see "Change the Network Name" (page 53).
- Change the screen's layout orientation (landscape or portrait).
  For information, see "Change the Screen Orientation" (page 53).
- Configure PathPilot for the accessories you're using.
  For information, see "Enable the On-Screen Keyboard" (page 60); "Enable the USB M-Code I/O Interface Kit" (page 62); "Use a USB Camera" (page 62).
- Disable the limit switches for troubleshooting.
  For information, see "Disable Hard Stop Referencing" (page 59).
- Specify the way in which you want to use a G30 move.
  For information, see "Limit G30 Moves" (page 59).
- Identify the available G-code modes that you can use.
  For information, see "View Available G-Code Modes" (page 81).
- Adjust plasma settings.
  For information, see "Set Torch Touch Trigger Depth" (page 57); "Set Minimum Touchoff Spacing" (page 58); "Enable Ohmic Touchoff" (page 58); "Enable Torch Height Control" (page 58); "Enable Arc-Ok Checking" (page 58).

Offsets Tab

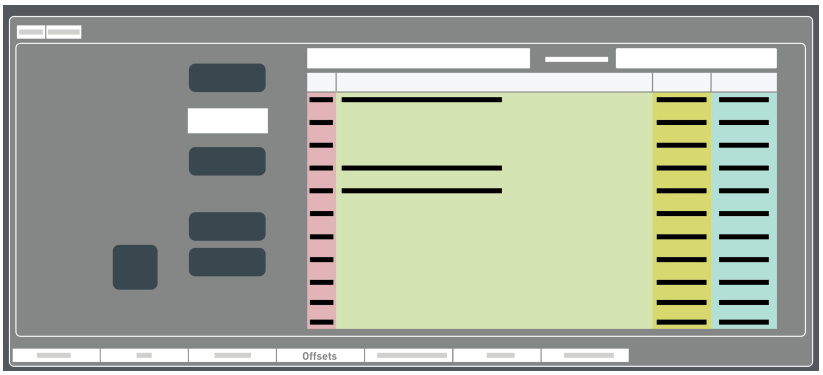


Figure 6: Offsets tab.

From the Offsets tab, you can do the following activities:

- Make and restore backup files of your settings.
  For information, see "Create Backup Files" (page 102); "Restore Backup Files" (page 105).
- Import and export .csv files of your tool table.
  For information, see "Import and Export the Tool Table" (page 106).
- Work with a table of tool descriptions and tool offsets.
  For information, see "Set Tool Length Offsets" (page 74).
- Use an Electronic Tool Setter (ETS) to measure tools.
  For information, see "Use an Electronic Tool Setter (ETS) to Measure Tools" (page 77).
- Preset a G30 position.
  For information, see "Use a G30 Position" (page 95).
- Read the currently programmed work offsets.
  For information, see "View Work Offsets" (page 81).

Conversational Tab

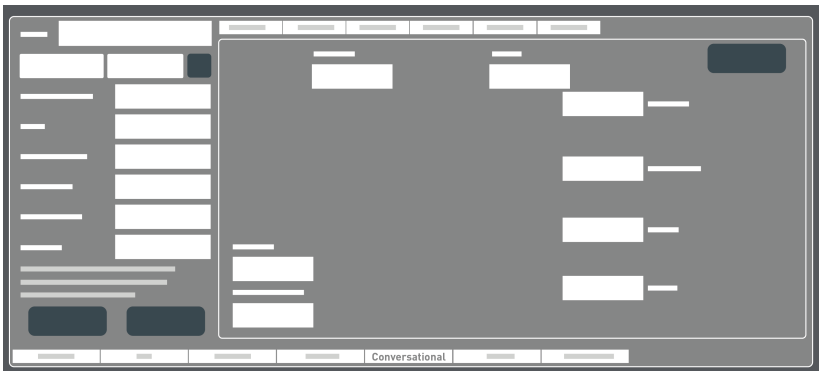


Figure 7: Conversational tab.

From the Conversational tab, you can do the following activities:

- Import a .dxf file.
  For information, see "Import a DXF File" (page 44).

Probe Tab

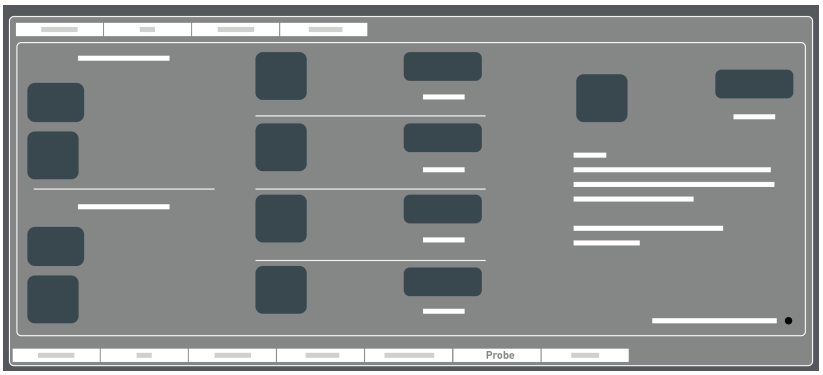


Figure 8: Probe tab.

From the Probe tab, you can do the following activities:

- Configure and control a probe to help perform certain functions. For information, see "Use a Probe with PathPilot" (page 68).

Status Tab

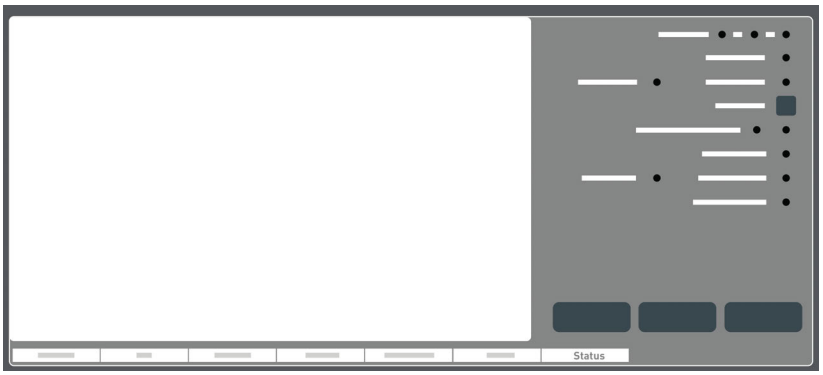


Figure 9: Status tab.

From the Status tab, you can do the following activities:

- View diagnostic machine information.
- Read error messages.
- Configure your internet connection.
  For information, see "Enable an Internet Connection" (page 49).
- Update or install a previous version of PathPilot.
  For information, see "Manage PathPilot Versions" (page 29).

PERSISTENT CONTROLS

Figure 10: Persistent Controls section.

The areas that display in the Persistent Controls section don't change (unlike the Notebook section). They display regardless of the activity you're doing. Activities are grouped into the following areas:

Program Control Area

Figure 11: Program Control area.

From the Program Control area, you can do the following activities either before starting or while running a G-code program:

- Reset the machine.
  For information, see "Bring the Machine Out of Reset" (page 82).
- Start, stop, or pause a G-code program.
  For information, see "Start a Program" (page 87); "Stop Machine Motion" (page 88); "Use the Feed Hold Function" (page 90).
- Use overrides to change the feed rate, voltage, and maximum velocity.
  For information, see "Use the Feed Rate Override Function" (page 91); "Use the Maxvel Override Function" (page 92); "Use the Voltage Override Function" (page 94).
- Manually control a G-code program.
  For information, see "Use M01 Break Mode" (page 92); "Use Single Block Mode" (page 93).

Position Status Area

Figure 12: Position Status area.

From the Position Status area, you can do the following activities either before starting or after running a G-code program:

- Reference the machine axes.
  For information, see "Reference the Machine" (page 86).
- Create work offsets.
  For information, see "Set Work Offsets" (page 79).
- Understand how you're jogging the machine.
  For information, see "View the Active Axis to Jog" (page 82); "View the Current Machine Position" (page 86); "View the Distance to Go" (page 89).
- Quickly determine which G-code modes are active.
  For information, see "View the Active G-Code Modes" (page 88).

Manual Control Area

Figure 13: Manual Control area.

From the Manual Control area, you can do the following activities either before starting or after running a G-code program:

- Move the machine axes.
  For information, see "Jog the Machine" (page 83).
- Use automatic look-up tables to find feed rate, amperage, pierce parameters, and torch height control voltage for the material you're cutting.
  For information, see "Use the AutoFS Material Picker" (page 99).

KEYBOARD SHORTCUTS

The following table lists the keyboard shortcuts in PathPilot.

Keyboard Shortcut	Use to...
Alt+E	Edit the currently loaded G-code program (from any tab in the PathPilot interface)
Alt+Enter	Use the Manual Data Input (MDI) Line DRO field
Alt+R	Start a program
Esc	Stop a program
Shift+Alt+E	From the Main tab, quickly edit a G-code program with conversational programming
Space Bar	Feed hold the machine

MANAGE PATHPILOT VERSIONS

You don't need to install updates sequentially. You can update from any previous version to the current version of PathPilot. Depending on what you want to do, refer to the following sections:

- "Download and Install an Update File from the Controller" (below)
- "Install an Update File from a USB Drive" (on the next page)
- "Install a Previous Version of an Update File" (page 31)

Download and Install an Update File from the Controller

1. Confirm that the PathPilot controller is powered on and out of **Reset** mode.
2. Downloading and installing an update file requires an Internet connection. From the **Status** tab, confirm that the **Internet** button LED light is on. Then, select **Update**.

Figure 14: Update button on the Status tab.

3. From the **Software Update** dialog box, select **Check Online**.

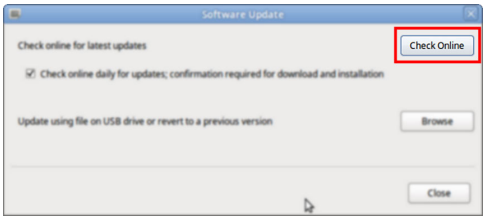


Figure 15: Software Update dialog box.

4. Select **Install**.

Figure 16: Install button on the Software Update dialog box.

The update file is downloaded, and a notification dialog box displays.

5. From the dialog box, select **OK**.
   The update file is installed on the PathPilot controller.
6. Follow the on-screen instructions to restart the PathPilot controller.

Install an Update File from a USB Drive

1. From the PathPilot support center, download the most recent PathPilot update file.
2. Transfer the PathPilot update file to a USB drive.
3. Put the USB drive into the PathPilot controller.
4. Confirm that the PathPilot controller is powered on and out of **Reset** mode.
5. From the **Status** tab, select **Update**.

Figure 17: Update button on the Status tab.

6. From the **Software Update** dialog box, select **Browse**.

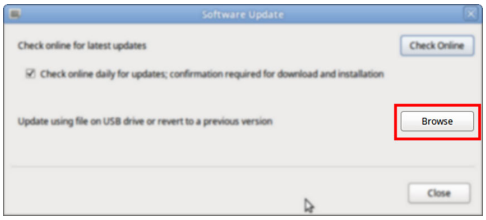


Figure 18: Software Update dialog box.

7. From the **Browse** dialog box, select **USB**.

Figure 19: Browse dialog box.

8. Select the desired update file, and then select **Update**.
   The update file is installed on the PathPilot controller.
9. Follow the on-screen instructions to restart the PathPilot controller.

Install a Previous Version of an Update File

1. Confirm that the PathPilot controller is powered on and out of **Reset** mode.
2. From the **Status** tab, select **Update**.

Figure 20: Update button on the Status tab.

3. From the **Software Update** dialog box, select **Browse**.

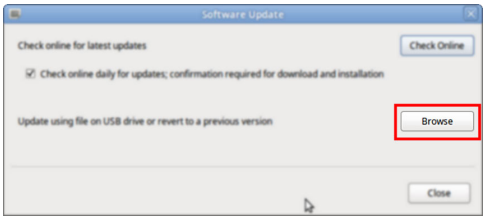


Figure 21: Software Update dialog box.

4. From the **Browse** dialog box, select **Previous Versions**.

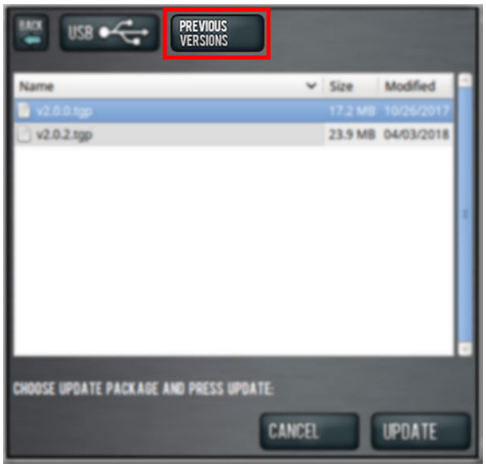


Figure 22: Browse dialog box.

5. Select the desired update file, and then select **Update**.
   The update file is installed on the PathPilot controller.
6. Follow the on-screen instructions to restart the PathPilot controller.


TITLE: externalThread-Mill
Aluminum
-any-
WORK OFFSET: G59
TOOL: 2
SPINDLE RPM: 555
FEEDRATE: 100.0
Z FEEDRATE: 100.0
Z CLEAR: 0.6000
(Flutes: N/A Chip load per tooth: N/A)
(SFM: 0 – MRR: N/A)
POST TO FILE
APPEND TO FILE
Face
Profile
Pocket
Drill / Tap
Thread Mill
Engrave
DXF (Mill)
EXTERNAL
INTERNAL
DEPTH
X (FROM DRILL TABLE)
Y (FROM DRILL TABLE)
NUMBER OF PASSES:
THREADS/IN:
3.0
Z START:
3.0000
Z CLEAR
PITCH (IN):
0.3333
2.0000
Z END:
LEFT
RIGHT
Main
File
Settings
Offsets
Conversational
Probe
Status
CYCLE START
SINGLE BLOCK
M01 BREAK
FEEDHOLD
STOP
COOLANT
RESET
100%
FEED 100%
100%
RPM 100%
100%
MAXVEL 100%
POS IN G55
DTG
ZERO X
5.1931
0.0000
REF X
ZERO Y
0.6512
0.0000
REF Y
ZERO Z
0.0000
0.0000
REF Z
ZERO A
0.0000
0.0000
REF A
STATUS:
G55 G90 G20 G80 G40 G94 G97 G99 G91.1
STEP:
REV
SPINDLE RANGE
T
0
05:06:02

PATHPILOT TOOLS AND FEATURES

IN THIS SECTION, YOU'LL LEARN:

- How to use PathPilot, depending on the activity that you want to do.

CONTENTS

CREATE AND LOAD G-CODE FILES

To get started with PathPilot, you must first load or create a G-code file.

Load G-Code

To run a G-code program on a PathPilot controller, you must first verify that the file is on the controller. For more information on transferring and moving files, see "Transfer Files to and From the Controller" (below).
To load G-code:

1. From the **File** tab, in the **Controller Files** window, select the desired .nc file.
2. Select **Load G-Code**.

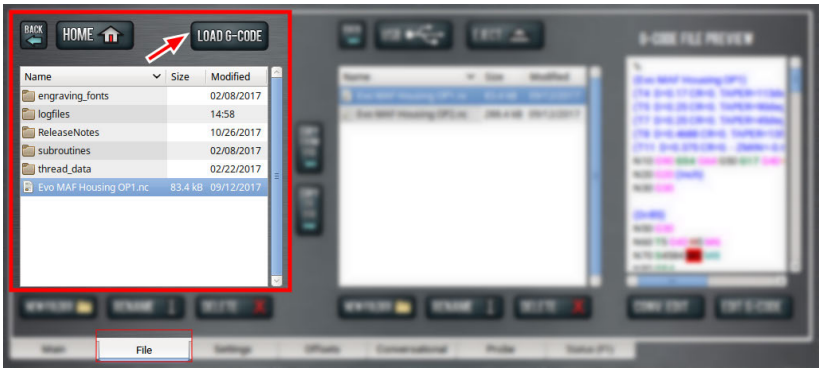


Figure 1: Controller Files window on the File tab.

> **Note:** This function is only available for files stored on the PathPilot controller.

PathPilot loads the G-code file and opens the **Main** tab.

Transfer Files to and From the Controller

To run a G-code program, you must transfer the files to the PathPilot controller. For information, see "About System Files" (page 102).
To transfer files to and from the controller:

1. Insert a USB drive into any open USB port.
2. From the **File** tab, select the file to transfer (either in the **USB Files** window or the **Controller Files** window).

> **Note:** Select **Back** to move backward and either **Home** or **USB** to move to the highest level.

3. Select the location to which you want to copy the transferred file.

4. Select either **Copy From USB** or **Copy to USB**.

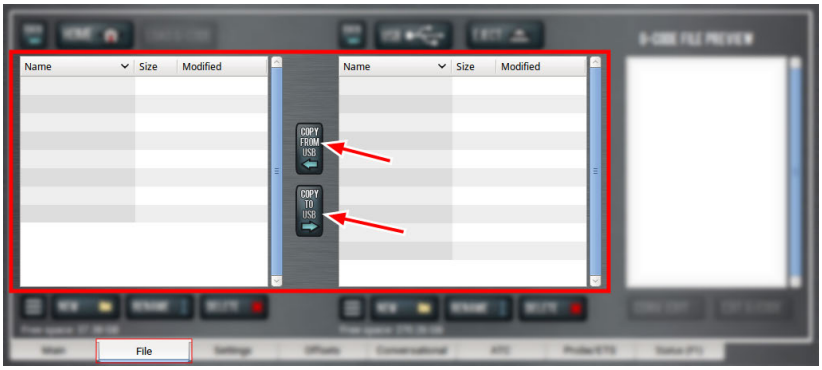


Figure 2: File tab.

> **Note:** The file must have a unique name. If it doesn't, you must either overwrite the file, rename the file, or cancel the file transfer.

5. Select **Eject**.
   It's safe to remove the USB drive from the controller.

Preview G-Code Files

You can preview an .nc file that's either on the PathPilot controller or on a USB drive.
To preview G-code files:

- From the **File** tab, in the **Controller Files** window or the **USB Files** window, select an .nc file.
  The text displays in the **Preview** window.

Figure 3: File tab.

Access Recent G-Code Files

You can load a recently loaded G-code file from the Main tab. For information, see "About the G-Code Tab" (page 39).
To access recent G-code files:

1. From the **Main** tab, in the **G-Code** tab, select the **Recent Files** menu.

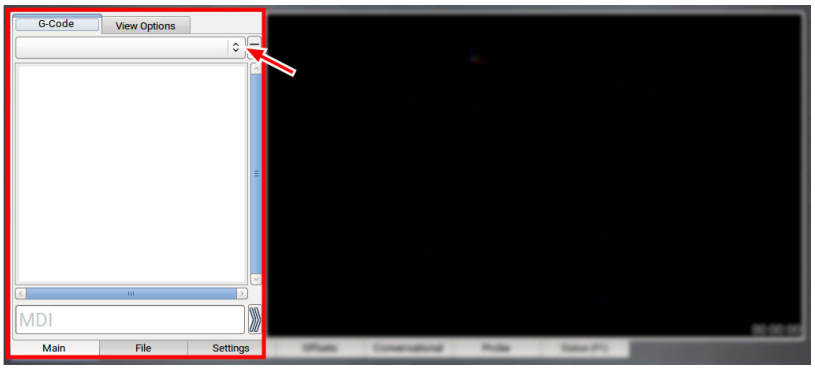


Figure 4: Recent Files menu on the Main tab.

   The last five program files loaded into PathPilot display.

2. Select the name of the desired G-code program.
   The G-code program loads.

Close the Current Program

1. From the **Main** tab, on the **G-Code** tab, select the **Recent Files** menu.
2. Select **Clear Current Program**.

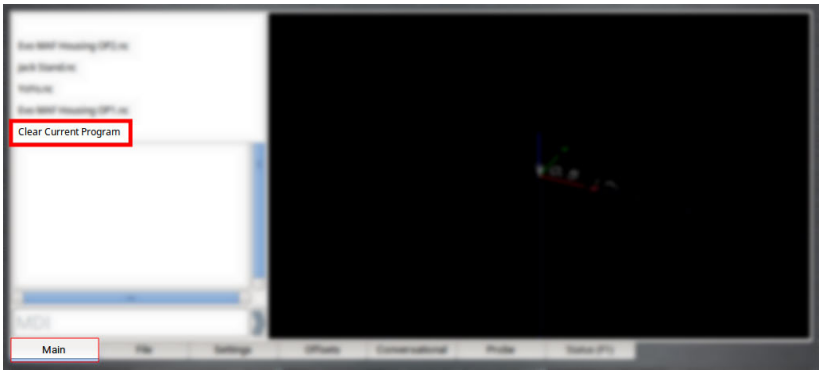


Figure 5: Recent Files menu on the Main tab.

   The currently loaded G-code program closes.

Edit G-Code with a Text Editor

You can edit .nc files that are on the PathPilot controller. If the .nc file is in the USB Files window, you must first transfer it to the controller; go to "Transfer Files to and From the Controller" (page 35).
To edit G-code with a text editor:

1. From the **Controller Files** window, highlight the .nc file and select **Edit G-code**.

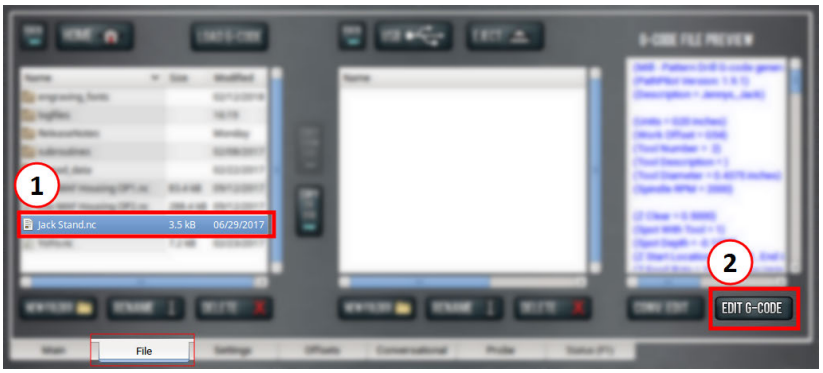


Figure 6: Edit G-code button on the File tab.

   The file opens in a text editor.
2. Make and save the appropriate changes to the file.
3. Close the text editor.

> **Tip!** To quickly edit an already loaded G-code program from the Main tab, you can use a keyboard shortcut: Shift+Alt+E.

Read G-Code

Once your G-code file is loaded into PathPilot, you can read it in the following ways:

Expand the G-Code Tab

You can change the size of the G-Code tab if you need more space to view the code. For more information on using the G-Code tab, see "About the G-Code Tab" (below).
To expand the G-Code tab:

- Select the **Window Expander**.

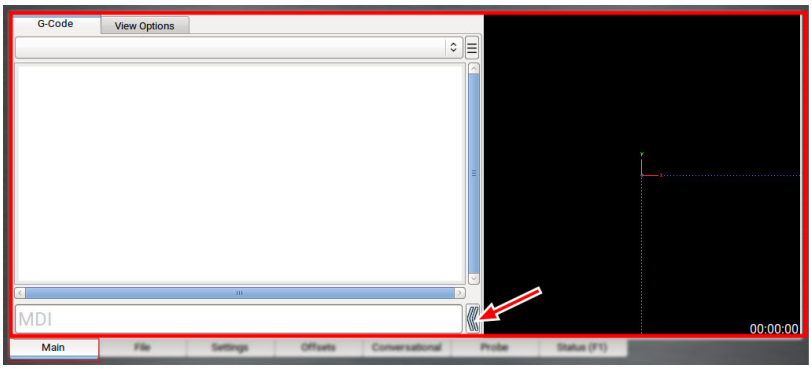


Figure 7: Window Expander on the Main tab.

The **Tool Path** display shrinks.

About the G-Code Tab

The G-Code tab displays the code of the currently loaded program file. Use the scroll bars to view the entire file. You can make the G-Code tab larger. For information, see "Expand the G-Code Tab" (above).
PathPilot highlights certain lines of code of interest. When running a G-code program in single block mode, there may be as many as two lines of G-code highlighted, both with a different color:

- **Green Line** Indicates the start line (the line from which PathPilot starts the program).
  To change the start line, go to "Set a New Start Line" (page 41).
- **Orange Line** Indicates the line of code that PathPilot is currently executing.

Search in the Code

You can use PathPilot to search the text of a G-code program file for specific numbers, codes, or other items of interest (like tools, feeds, and speeds).
To search in the code:

1. From **Main** tab, on the **G-Code** tab, select any line of code to use as a starting point.
2. In the **MDI Line** DRO field, type `FIND` followed by one of the following:
   - Any text. PathPilot searches for instances of the specific number or code.

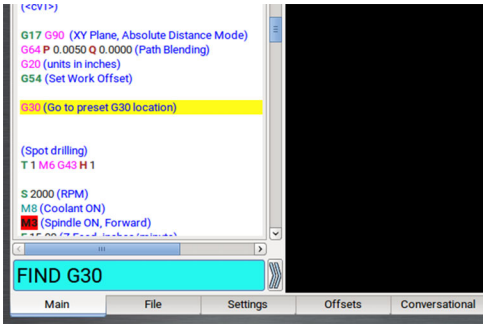


Figure 8: Search for a text command.

   - `FEED`. PathPilot searches for instances of the actual word `Feed` and any `F` G-code command.

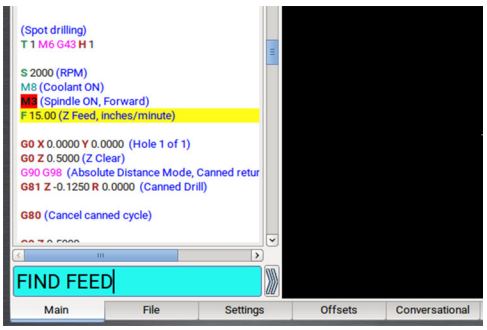


Figure 9: Search for a feed command.

   - `SPEED`. PathPilot searches for instances of the actual word `Speed` and any `S` G-code command.
   - `TOOL`. PathPilot searches for instances of the word `Tool` and any `T` G-code command.

> **Note:** The find command is not case-sensitive.

3. Select the **Enter** key.
   If PathPilot finds the information, the searched term is scrolled to and highlighted in the **G-Code** tab.
4. *(Optional)* Select **Enter**.
   PathPilot finds the next instance of the searched text.

5. *(Optional)* Select **Enter+Shift**.
   PathPilot finds the previous instance of the searched text.

> **Note:** When the search reaches the end of the G-code file, it starts again from the beginning.

Set a New Start Line

The start line (the line from which PathPilot starts the program) is, by default, the first line of code in the program. To set a new start line:

1. From the **Main** tab, on the **G-Code** tab, do one of the following:
   - Right-click any line in the program.

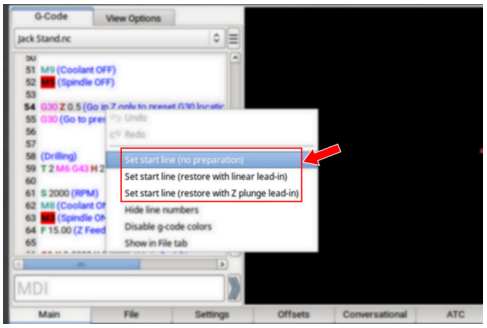


Figure 10: Accessing the Options menu by right-clicking.

   - Tap the line. Then, select the **Options** menu.
2. Select the desired lead-in move. For information, see "Lead-In Moves" (below).

Lead-In Moves

- **Set start line (no preparation)** Keep the current tool in the spindle, with the current tool length applied. The machine executes the start line from the current position.

> **Note:** We don't recommend this option for starting partway through a cut.

> **EXAMPLE**
>
> - Starting the program at a tool change.
> - Starting the program with a different tool in the spindle than the program calls for (like if your tool broke, which you've replaced, but you'd rather not edit the entire program or the tool table entry).

- **Set start line (restore with linear lead-in)** Perform a tool change (as required). The machine rapids in X and Y, then Z to the current position, then feeds in a straight linear line to the start line position.

  Note: This option assumes that the current position is the lead-in position.

  E X A M P L E
  Quickly resuming work after stopping the program to make an adjustment to the machine setup (like clearing chips, removing an object, or turning on the coolant pump). Because the machine's already set up, you can position the tool near the stopping point.

- **Set start line (restore with Z plunge lead-in)** Perform a tool change (as required). The machine rapids in Z to G30 clearance height, rapids in X and Y to the start line position, then feeds in Z to the start line position.

  E X A M P L E
  Running a sub-section of a large program when the correct tool isn't loaded (and positioning the tool tip near the starting point is difficult, like with a long tool or fly cutter loaded). This option doesn't require you to jog to the exact lead-in position.

Change the View of the Tool Path Display

1. From the **Main** tab, do one of the following:
   - Right-click the **Tool Path** display.

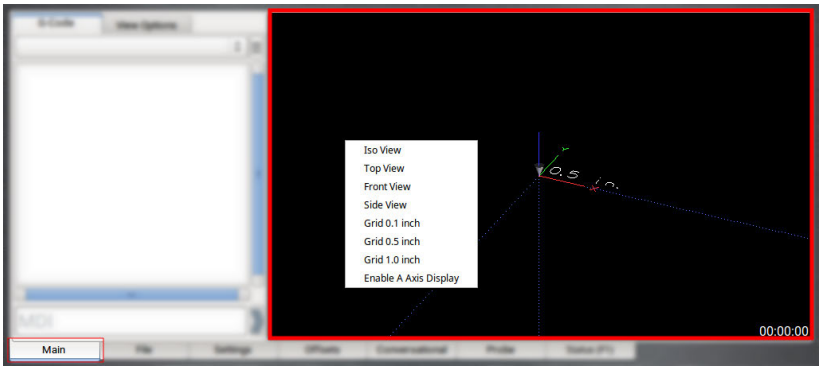


Figure 11: Tool Path display on the Main tab.

- Select the **View Options** tab.

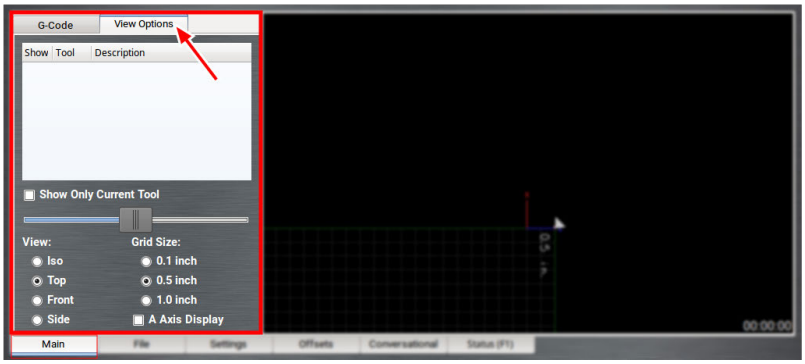


Figure 12: View Options tab on the Main tab.

2. Select a new view.
   For information, see "About the Tool Path Display" (below).

About the Tool Path Display

The Tool Path display is a graphical representation of the currently loaded G-code file's tool path.
Depending on which programming mode you're in (G20 or G21), PathPilot defaults to one of the following grid line spacings:

- **G20 Mode** 1/2 in. intervals
- **G21 Mode** 5 mm intervals

In the Tool Path display, there are four different line types:

- **Dotted Blue Lines** Indicate the boundary box (the ends of travel of the axes).
- **Red Lines** Indicate the tool path as it is cut.

> **Note:** The Tool Path display shows the program extents — the furthest points to which the tool will travel while running the program — of the currently loaded G-code file alongside the tool path lines.

- **White Lines** Indicate the preview lines.
- **Yellow Lines** Indicate the jogging moves.

To erase the jogging moves (yellow line) or the tool path (red lines), do one of the following:

- Double-click anywhere in the Tool Path display.
- Select Reset.

Import a DXF File

You can import a .dxf file (Drawing Exchange Format) into PathPilot to generate G-code, which can then cut the shape (or shapes) described in the .dxf file. For example, you could use this feature to engrave logos or artwork.

1. From the **Conversational** tab, select the **DXF** tab.

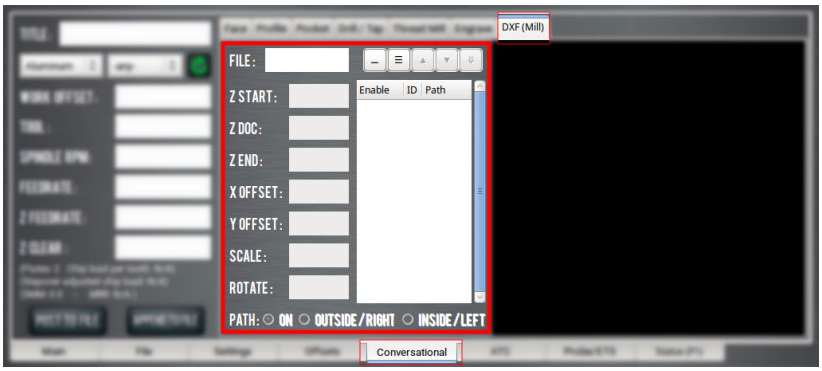


Figure 13: DXF tab on the Conversational tab.

2. Select the **File** DRO field.
   The **File Selector** dialog box opens.
3. Select the .dxf file, and then select **Open**.
4. The shapes from the selected file are loaded into the **Preview** window.

> **Note:** The .dxf file must already be transferred to the PathPilot controller. For information, see "Transfer Files to and From the Controller" (page 35).

5. In the **X Offset** DRO field and the **Y Offset** DRO field, type the offset value added in the **XY** direction from the bottom left corner of the .dxf drawing.
6. In the **Scale** DRO field, type the scale factor for the drawing. The value typed in the **Scale** DRO field is used as a multiplier for the .dxf dimensions, and is used for the entire drawing.

> **EXAMPLE**
> If you type `1.0` in the **Scale** DRO field, the .dxf is scaled at 100%.
> If you type `2.0` in the **Scale** DRO field, the .dxf is scaled at 200%.

7. In the **Rotate** DRO field, type the rotation angle in degrees.
   The rotation angle is applied around the Z-axis of the drawing's origin.
8. Select one of the following to set the cutter compensation to be applied to the tool path:
   - **On**: The tool moves along the path.
   - **Outside / Right**: Offsets the tool path right of the drawing path, seen from the direction where the tool enters the path.
   - **Inside / Left**: The opposite of **Outside / Right**.

Working with Layers and Shapes

The .dxf file contains shapes grouped into layers.
In the **Shape Selection** tree view window, you can enable or disable individual layers and complete layers. You can select shapes either from the tree view window or in the **Preview** window.

Change the Layer or Shape Cut Order

- Use the **Up Arrow** and **Down Arrow** buttons above the **Shape Selection** tree view window.
  Shapes or layers higher in the tree view window are cut earlier than those below it. The order in which the shapes are cut is the same as the order of the enabled shapes in the tree view window and the cyan path in the **Preview** window.

> **Note:** If a layer is selected, the whole layer is moved up or down. Shapes can't be moved between layers.

Adjust the Tree View Window

- Use the **Fold** and **Unfold** buttons to collapse and expand the layer and shape tree in the tree view window.

Working in the Preview Window

The **Preview** window uses the following colors:

- **Cyan** Selected paths
- **Gray** Disabled paths
- **White** Drawing path
- **Magenta** Cut path
- **Dark Cyan Stippled Line** Tool path between cuts
- The coordinates use the following colors:
  - **Red** X-axis
  - **Green** Y-axis
  - **Blue** Z-axis

Create and Add Shape Library Templates

Starting with PathPilot v2.9.0, you can create and add new templates to the Shape Library system.
Templates consist of two files:

- **.template**: A parametric G-code file that serves as the actual template.
- **.png**: A thumbnail image to identify it in the shape library.

Tip! We recommend that you create templates externally in your favorite G-code editor and then copy them to your PathPilot controller like you would any other file.

Create a Template

1. Create a new file named ***[shape_name]*.template**, where ***[shape_name]*** is your desired template name.

   E X A M P L E
   To create a template for a welding bracket, make the file name **welding_bracket.template**.

2. Inside the file that you created in Step 1, you can write regular G-code with the addition of special parametric variables that PathPilot uses to display adjustable parameters in the Shape Library.
   The following example is a template file that creates a line from one (X, Y) location to another:

```
line.templateText#<start_x> = 1.0 (PARAMFLOAT)
#<start_y> = 1.0 (PARAMFLOAT)
#<end_x> = 5.0 (PARAMFLOAT)
#<end_y> = 5.0 (PARAMFLOAT)
G20 (set units to inches)
G91.1 (Restore Incremental Arc Distance Mode)
G90 (Absolute Distance Mode)
G30
M200
G0 X[#<start_x>] Y[#<start_y>]
G15
G16
(Cut to end of line)
G1 X[#<end_x>] Y[#<end_y>]
M205
```

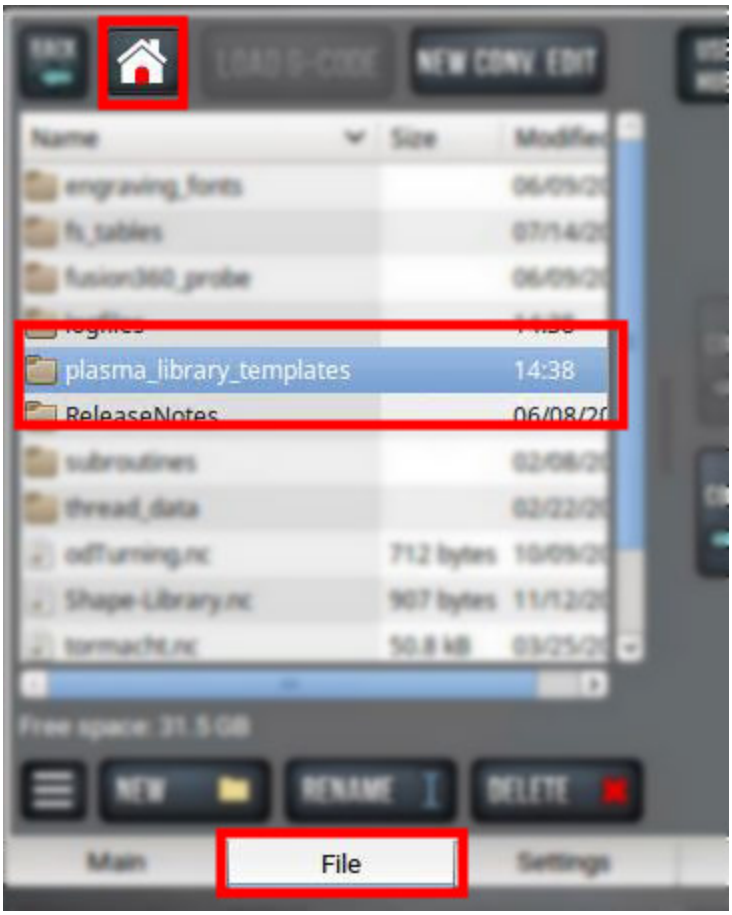
   Note: The special lines at the top each create a parameter that PathPilot recognizes and displays on screen. Each parameter can either have the type `PARAMFLOAT` or `PARAMINT` for floating point numbers or integers, respectively.

3. *(Optional)* Once you've written the G-code for your template, you can create a **.png** image to use as the thumbnail. The image must:
   - Be sized to 256 pixels × 256 pixels.
   - Use a file name that matches the template name.

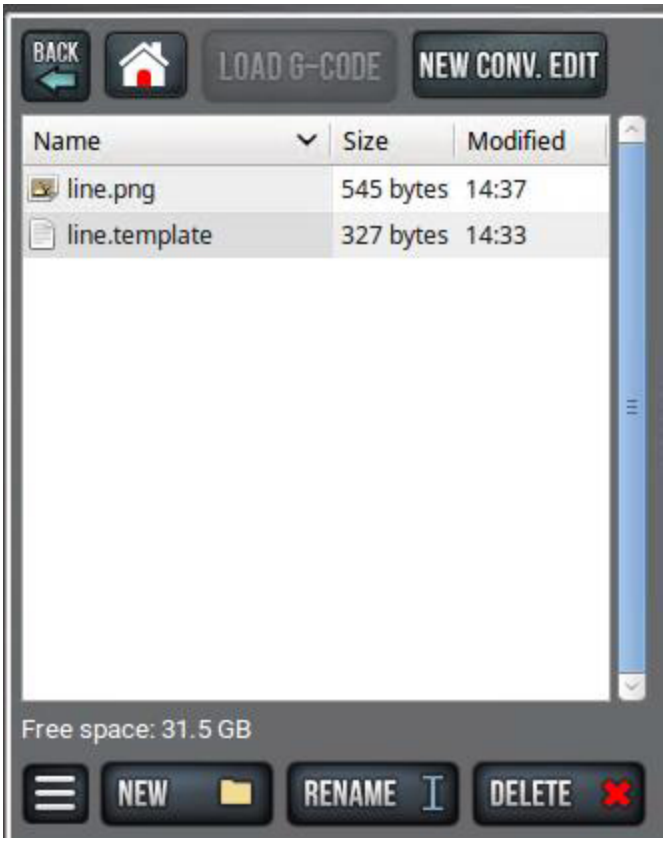
   > **EXAMPLE**
   > A thumbnail for the template **welding_bracket.template** requires an image with the file name **welding_bracket.png**.

Add a Template

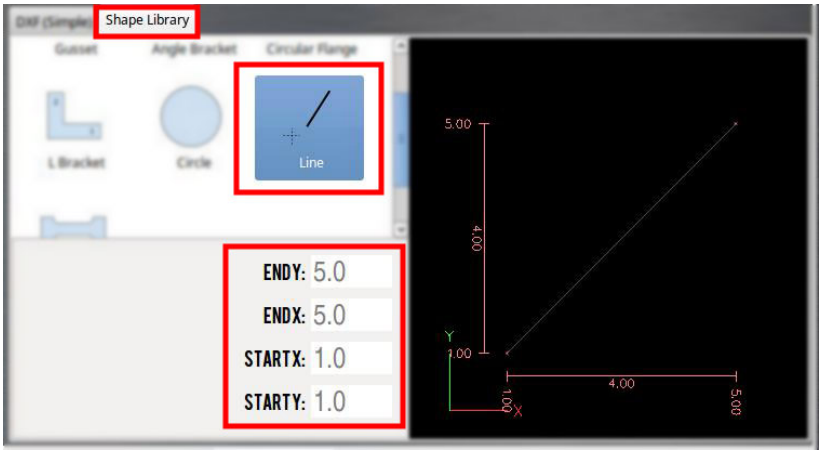
1. From the PathPilot controller, on the **File** tab, find the **plasma_library_templates** folder inside the **Home** directory. If you don't see this folder, make sure that you've updated to the latest version of PathPilot.



2. Copy the **.template** file and the **.png** file that you created into the **plasma_library_templates** folder.



3. Restart PathPilot.
   Your new template displays on the **Shape Library** tab, with any defined parameters from the **.template** file displayed as configurable options.



MACHINE SETTINGS AND ACCESSORIES

Before running a G-code program, you must first make sure that the machine settings are properly configured.

Enable an Internet Connection

If desired, you can enable an internet connection on your PathPilot controller. An internet connection allows you to receive automatic PathPilot updates and transfer files with Dropbox instead of a USB drive.
To enable an internet connection:

1. From the PathPilot interface, on the **Status** tab, select **Internet**.

Figure 14: Internet button on the Settings tab.

The **Network Configuration** dialog box displays.

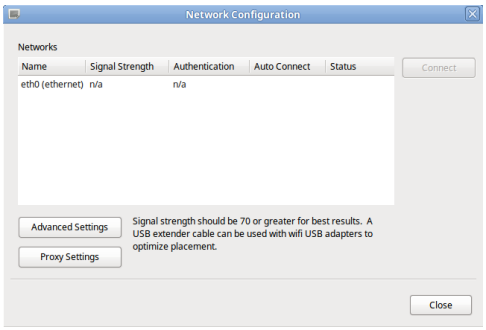


Figure 15: Network Configuration dialog box.

2. From the **Network Configuration** dialog box, in the **Networks** list, select the network you want to use. Then, select **Connect**.

   Note: Wi-Fi connection signal strengths are indicated on a scale of 0 to 100, with 100 being the strongest. PathPilot continually refreshes the signal levels to help you find the best placement for your Wi-Fi network adapter. Ethernet connections are indicated by a prefix in the following format: eth*[NUMBER]*. For example, **eth1**.

   The PathPilot operating system connects to the internet using the network you specified. It continues to detect and connect to the Wi-Fi network, even after power cycles.

3. Once connected, you can use the Dropbox and automatic updates features. Depending on what you want, see the following procedures:
   - "Connect to Dropbox" (below)
   - "Enable Automatic Updates" (page 52)

Connect to Dropbox

Note: Dropbox requires an internet connection. If you haven't yet enabled it, go to "Enable an Internet Connection" (on the previous page).

If desired, you can connect your PathPilot controller to a Dropbox account to easily synchronize your G-code files, which eliminates the need to transfer them with a USB drive.

Note: Dropbox stops synchronizing once the PathPilot controller's internal drive has less than 500 MB of free space. To avoid this, we recommend that you organize your Dropbox account on a separate computer before you connect to Dropbox with PathPilot. Only store files in the top-level that you want synchronized to your PathPilot controller.

To connect to Dropbox:

1. From the PathPilot interface, on the **Main** tab, in the **MDI Line** DRO field, type `ADMIN DROPBOX`.
2. Select the **Enter** key.
   The **Dropbox Configuration** application displays.

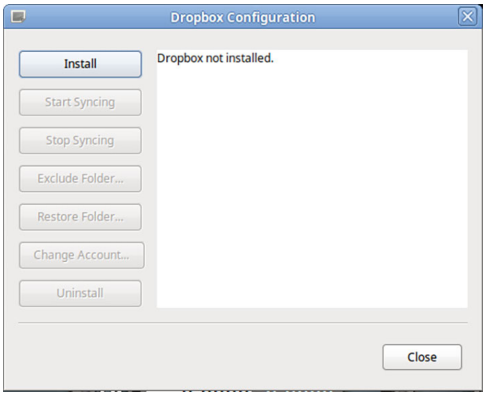


Figure 16: Dropbox Configuration application.

3. Select **Install**.
   The **Dropbox Installation** dialog box displays.
4. Select **OK**.

Figure 17: Dropbox Installation dialog box.

   The Dropbox installation starts and continues for about a minute. When done, a web browser displays.
5. From the web browser, sign in or create a Dropbox account.
   The PathPilot controller connects to the Dropbox account, creates a local **Dropbox** folder that is visible in **File** tab, and synchronizes the folder.

6. Because Dropbox stops synchronizing once the PathPilot controller's internal drive has less than 500 MB of free space, we recommend that you exclude large or unrelated folders from synchronization. Select **Exclude Folder...**.

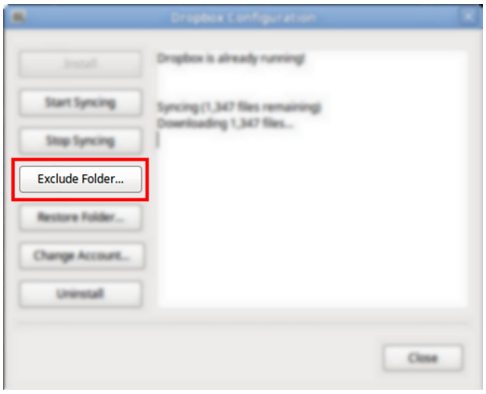


Figure 18: Exclude Folder button.

The **Add Folder to Excluded Set** dialog box displays.

7. In the **Add Folder to Excluded Set** dialog box, type the name of the folder to exclude.

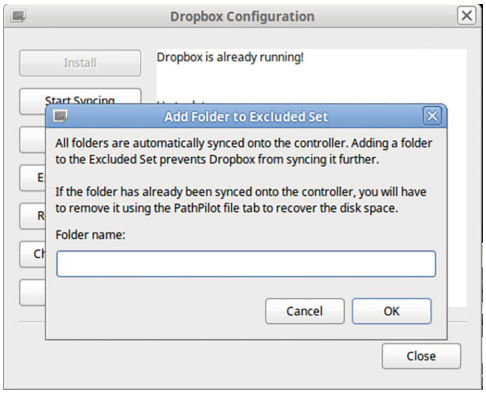


Figure 19: Add Folder to Excluded Set dialog box.

> **Note:** You can only exclude folders, not individual files.

8. Select **OK**.
9. Select **Close**.
   The PathPilot controller is now connected to Dropbox.

Enable Automatic Updates

> **Note:** Automatic updates require an internet connection. If you haven't yet enabled it, go to "Enable an Internet Connection" (page 49).

If desired, you can enable automatic updates for PathPilot.
To enable automatic updates:

1. From the PathPilot interface, on the **Status** tab, select **Update**.
   The **Software Update** dialog box displays.

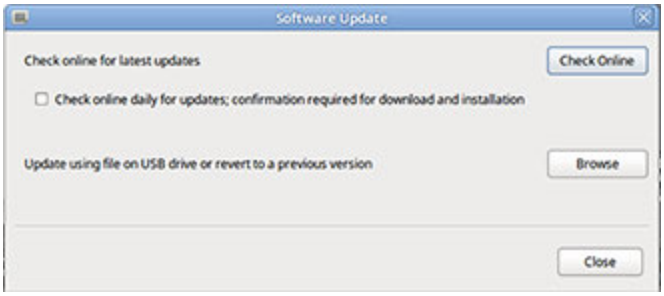


Figure 20: Software Update dialog box.

2. From the **Software Update** dialog box, select the **Check online daily for updates; confirmation required for download and installation** checkbox.
3. Select **Close**.
   When future updates are available, the **Status** tab displays a notification.

Change the Network Name

If you're connected to a network using either the Ethernet jack or the (optional) **Wireless Network Adapter (PN 38207)**, the PathPilot controller appears on your network as **network-attached storage**. The default network name of the controller is **TORMACHPCNC**.
To change the network name:

1. From the **Network Name** field, type a new network name.

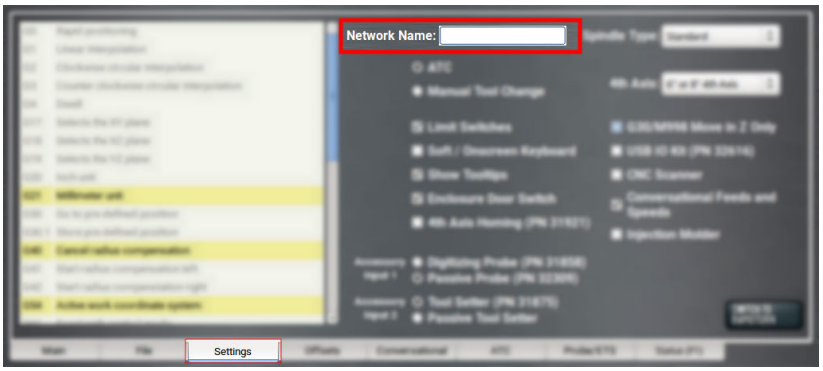


Figure 21: Network Name field on the Settings tab.

> **Note:** The network name must be unique within your network.

2. Select the **Enter** key.
3. For the change to take effect, you must restart the controller.

Change the Screen Orientation

A vertical orientation for 1920 × 1080 monitors is supported in PathPilot v2.10.0 and later. For more information on the portrait layout, go to "About Portrait Screen Layout" (on the next page).

To change the screen orientation:

1. From the PathPilot interface, on the **Settings** tab, select **Portrait** from the **Layout** drop-down menu. Restart the controller.

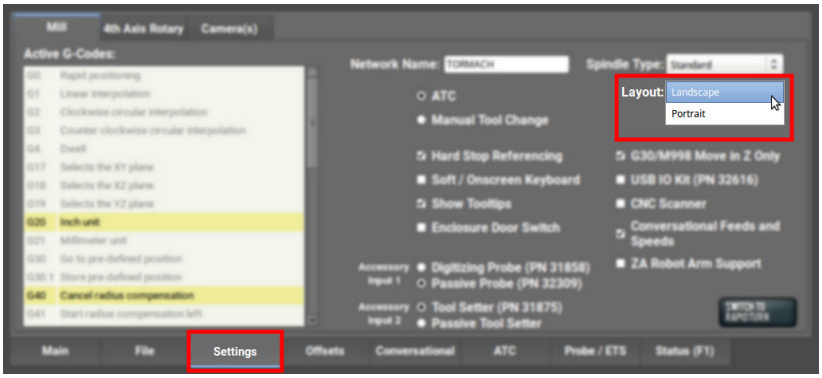


Figure 22: Layout drop-down menu on the Settings tab.

2. Rotate the monitor to the portrait orientation. You can rotate it either left or right, depending on what's easier for your setup.
3. While the controller is restarting, specify which direction you've rotated the monitor. Select **Apply**. If the result is unexpected, click **Restore Previous Configuration** on the confirmation dialog and choose a rotation direction again.

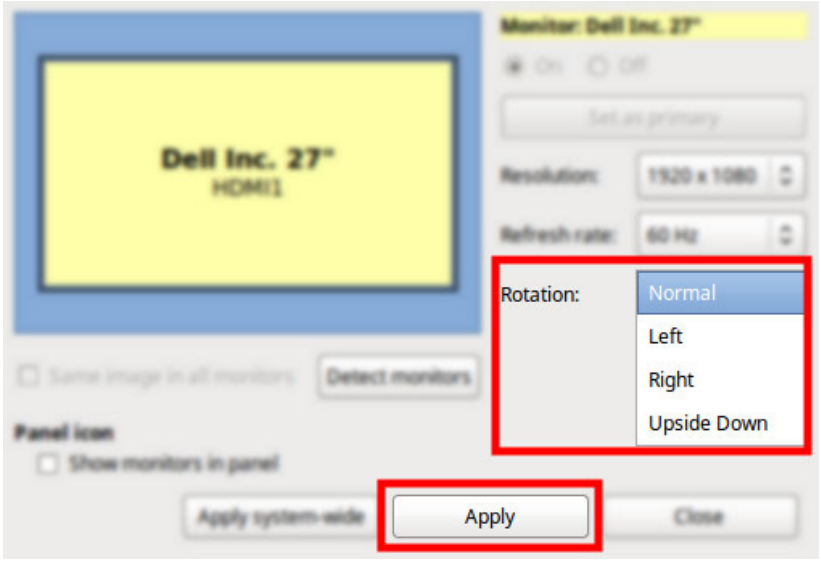


Figure 23: Monitor configuration dialog box.

The controller restarts in portrait layout.

About Portrait Screen Layout

Portrait layout provides some key advantages:

- A larger tool path window that's always visible at the top of the screen, regardless of which tab you have active.

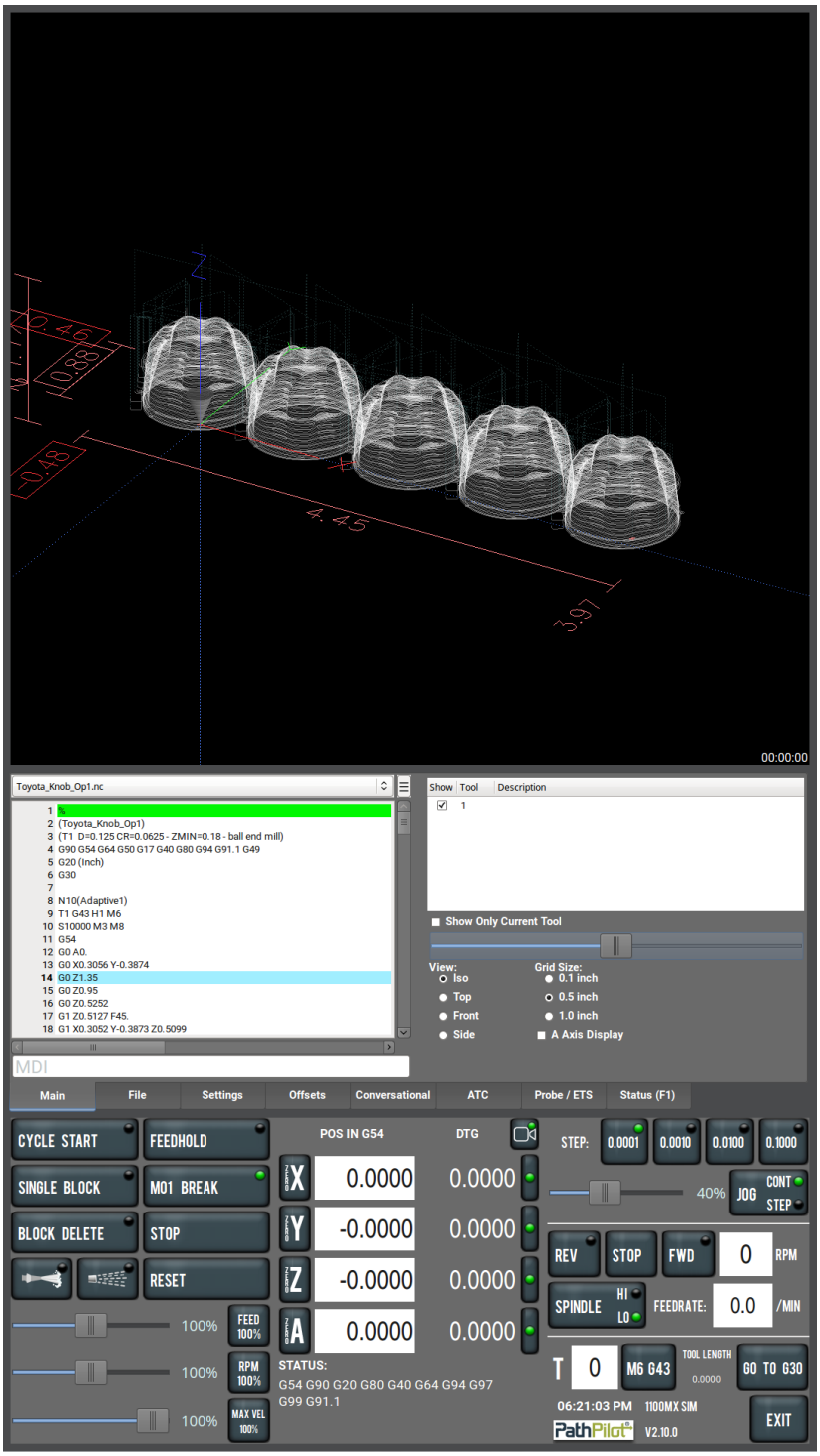


Figure 24: Tool Path window in portrait screen layout.

- A wider G-code window to more easily read the loaded G-code file and, if enabled, line numbers.
- The tool path window's view options are always visible for much easier access.

- When browsing G-code files using the File tab, file previews display on the top portion of the screen.

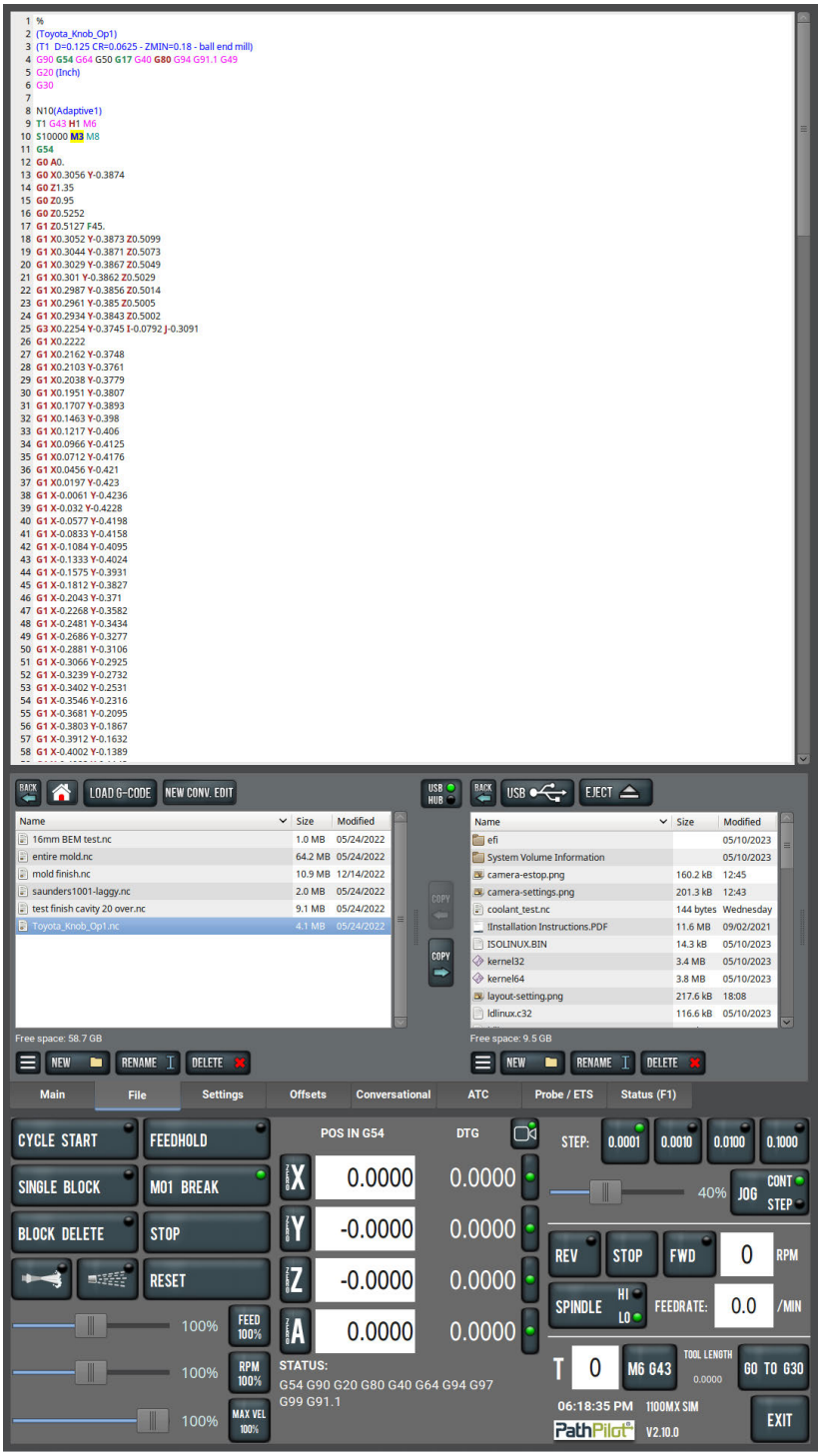
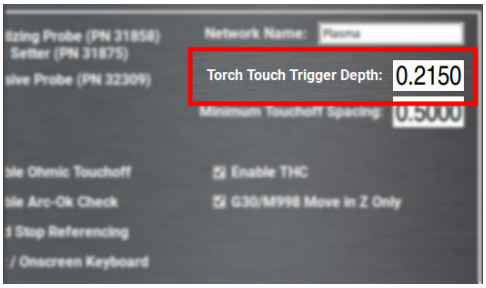


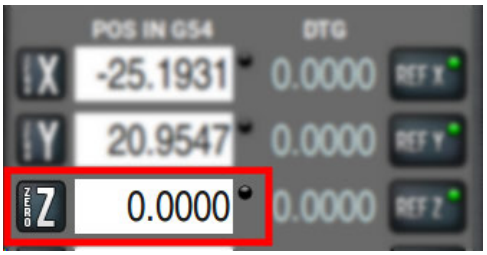
Figure 25: File tab G-code preview in portrait screen layout.

Set Torch Touch Trigger Depth



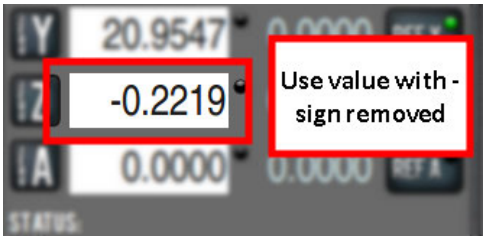
When using the physical touchoff switch for material height sensing, there's a certain amount of lost motion in the Z-axis between when the torch head touches the material and when the micro-switch in the torch lifter triggers. It's important to set an accurate value for the distance between torch touch and touch switch triggering so that the initial pierce height can be correctly computed.
To set torch touch trigger depth:

1. Power on the machine and reference all axes.
2. From the PathPilot interface, on the **Settings** tab, disable **Ohmic Touchoff**.
3. Put a piece of material on the machine table and position the torch above it.
4. Using a slow jog speed (~5 IPM), jog the torch down until it's just contacting the material.
5. Zero your Z-axis at this position.



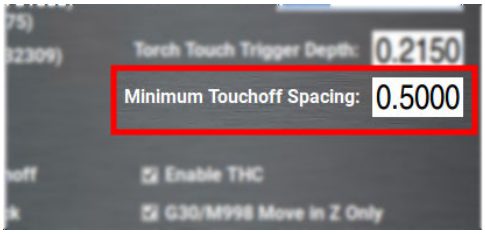
6. Switch to the **Status** tab and watch the **Touch Switch** LED. Very slowly, jog downwards until the LED comes on.

7. Note the value in the **Z** DRO field. This is your Torch Touch Trigger Depth. Convert this value to a positive number and enter it on the **Settings** tab.



8. Re-enable **Ohmic Touchoff** on the **Settings** page.

Set Minimum Touchoff Spacing



Minimum touchoff spacing is a time saving setting designed to avoid unnecessary probing during programs with a large number of individual cuts.
After the machine has probed for initial cut height, if the next cut begins within this distance it will be assumed that the material height is the same. This allows the machine to skip the probing routine and proceed straight to pierce height.
If your material is very flat and uniform, you can use a large distance for this setting (>2 in.). If your material is uneven or you would simply prefer to probe for height every time a cut begins, enter zero for this value.

Enable Ohmic Touchoff

This setting disables the ohmic sensing system entirely and relies solely on the physical touch switch for height sensing. This can be useful for cutting programs that result in water being repeatedly splashed into the torch nozzle, causing spurious probing trips.

Note: Do not disable ohmic touch-off when cutting very thin material. The weight of the torch will bend the material during probing and cause inaccurate pierce height.

Enable Torch Height Control

Disabling this setting stops all dynamic height adjustment during cutting. The torch will stay at the same height for the length of a cut and not adjust to follow the contours of the material.

Enable Arc-Ok Checking

Hypertherm plasma sources supply an arc-ok signal to PathPilot when the cutting arc is fully established. When this setting is enabled, the machine will pause during a `G16` pierce until is receives an arc-ok signal.
Disable this setting if your plasma source does not supply an arc-ok signal.

Disable Hard Stop Referencing

To provide a temporary workaround for a malfunctioning limit switch circuit, you can disable the limit switches.

Note: By default, the Hard Stop Referencing checkbox is selected.

To disable :

1. From the **Settings** tab, clear the **Hard Stop Referencing** checkbox.

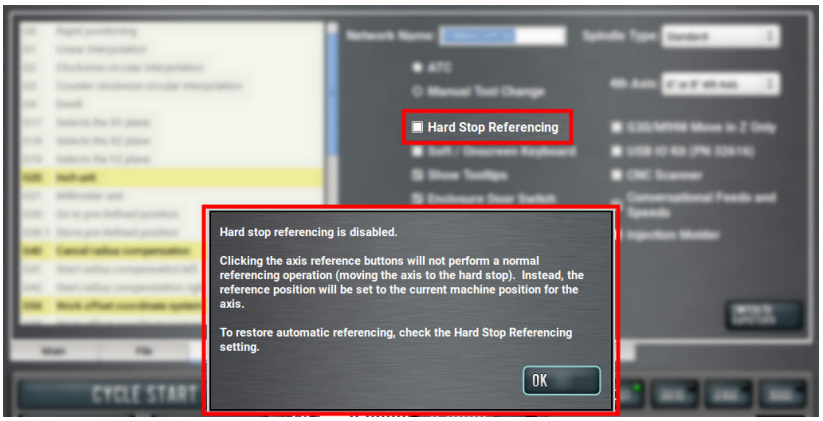


Figure 26: Hard Stop Referencing checkbox on the Status tab.

2. Select **OK**.
   The machine completes a unique referencing procedure after selecting the axis reference buttons: rather than moving each axis to the end of its travel, the reference position is set as the machine's current position.

   Tip! This is useful for troubleshooting, because you're now able to move the axis.

Limit G30 Moves

You can limit G30 moves so that only the Z-axis moves. For information, see "About G30" (page 96).
To limit G30 moves:

➢ From the **Settings** tab, select **G30/M998 Move in Z Only**.

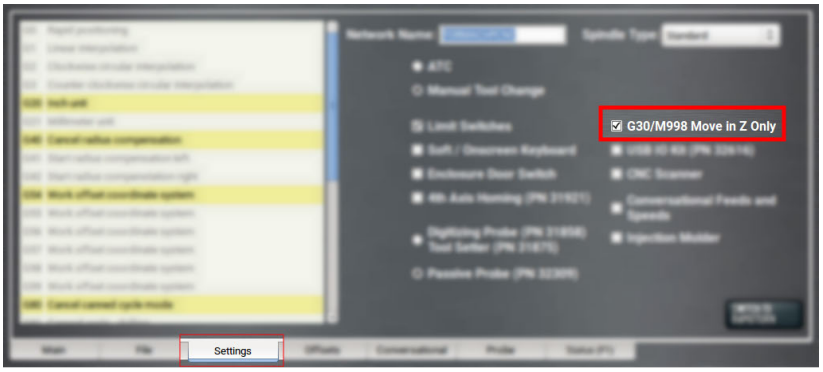


Figure 27: Settings tab.

About G30

A G30 command in a G-code program moves the machine to a preset position. For more information on setting a G30 position, see "Use a G30 Position" (page 95).
Use a G30 move to start a coordinated movement of the axes. You can limit the movement to only the Z-axis. For information, see "Limit G30 Moves" (on the previous page).

Tip! It's useful to program a G30 move right before a tool change so that the machine can jog to a safe tool change position.

Enable the On-Screen Keyboard

If you have an (optional) **Touch Screen Kit (PN 35575)**, you can use a soft keyboard to type information in the PathPilot interface. For information, see "About Soft Keyboards" (on the next page).
To enable and use the soft (on-screen) keyboard:

1. From the **Settings** tab, select **Soft / On-Screen Keyboard**.

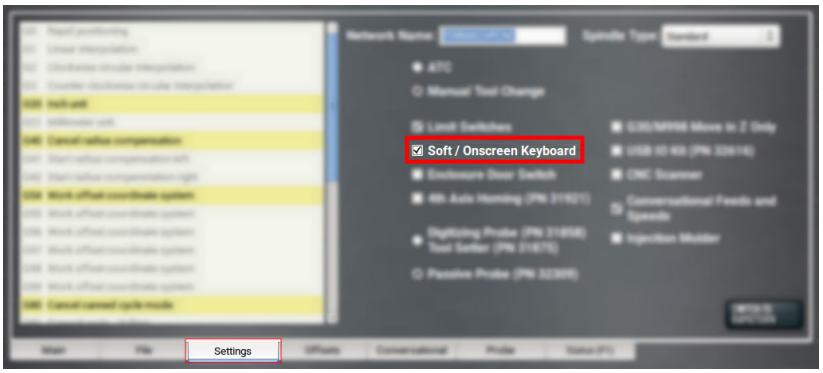


Figure 28: Settings tab.

2. To resize the keyboard, select a corner of the keyboard and drag.
3. To reposition the keyboard, select the **Anchor** key and drag the keyboard anywhere on the screen.

4. To close the keyboard, select the **X** key.

About Soft Keyboards

If you enabled a soft keyboard (on-screen keyboard) in the PathPilot interface to use with an optional touch screen or operator console, a keyboard opens when you select any field where keyboard input is required.
The keyboard displays a wide range of keys: both uppercase and lowercase, symbols, arrow keys, caps lock, backspace and delete, and more.

Figure 29: Soft (on-screen) keyboard.

Enable the USB M-Code I/O Interface Kit

If you have a **USB M-Code I/O Interface Kit (PN 32616)**, you must first enable it in the PathPilot interface.
To enable the USB M-Code I/O Interface Kit:

➢ From the **Settings** tab, select **USB IO Kit (PN 32616)**.

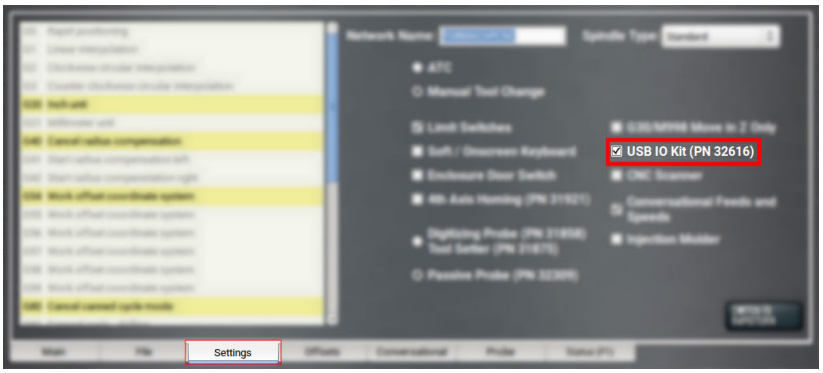


Figure 30: Settings tab.

Enable Tooltips

PathPilot displays expandable tooltips for many areas of the interface. Hovering over an item, like a DRO field or a button, displays helpful information about the item.
To enable or disable tooltips:

1. From the Settings tab, select or clear **Show Tooltips**.

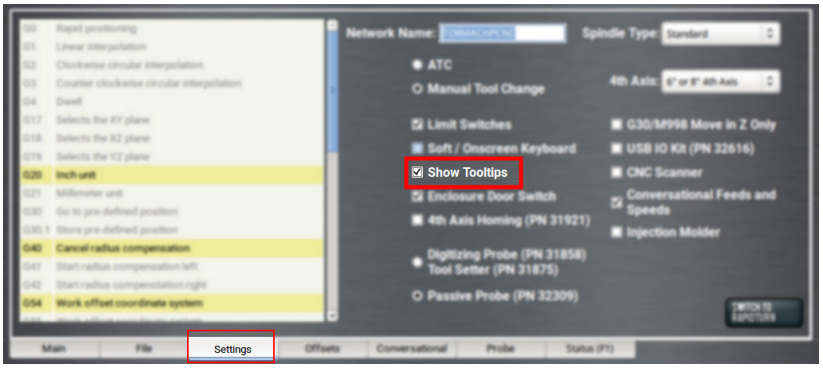


Figure 31: Show Tooltips checkbox.

> **Note:** If you disable the tooltips, you can still display them for specific items. Hover over an area of the interface, and select the Shift key on the keyboard.

Use a USB Camera

After plugging in the USB camera, navigate to the camera settings. From the PathPilot interface, in the **Settings** tab, open the **Camera(s)** tab. Identify the **Camera Status** read-only dialog box.

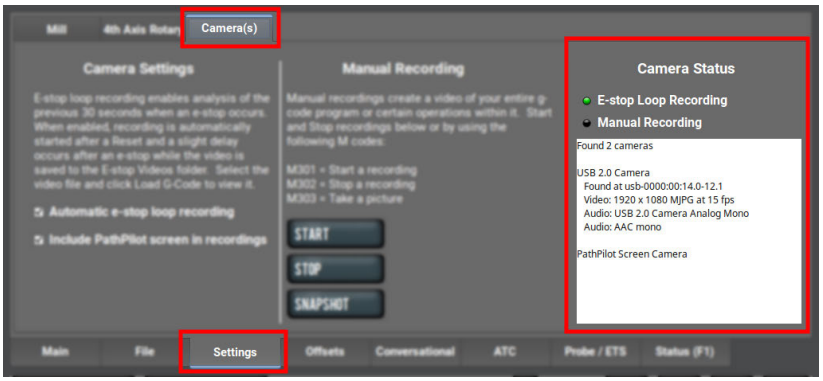


Figure 32: USB camera status.

As cameras are plugged in and unplugged, the **Camera Status** area is refreshed. To test compatibility of any USB camera, plug it in and watch the **Camera Status** area for the camera name and details.

Note: If a camera isn't shown after plugging it in or starting a video recording, it might require too much power from the USB ports on the controller. This is very likely when more than one camera is used. Try using a powered USB hub to add the camera(s).

When a USB camera is plugged in, it's analyzed for supported video and audio formats, frame sizes, and frame rates. If the camera supports it, PathPilot uses H.264 compression; otherwise, it uses Motion JPEG.
If the USB camera has a microphone, PathPilot records audio as well as video. The preferred format is compressed AAC, but uncompressed PCM is used as a fallback.

About USB Cameras

Recording video and audio from USB cameras is supported in PathPilot v2.10.0 and later. You can use up to four cameras simultaneously to record from different vantage points.

Note: All cameras are started and stopped at the same time — if you don't want a camera to record, you must unplug it.

USB cameras are compatible with all machine types, but older controllers may lack the processing power and memory needed for camera support. Controllers require 4GB of memory for camera functionality. Use the ADMIN MEMORY MDI command to verify the memory size of a controller.
You can purchase a Tormach USB Camera (PN 51240) with a metal case, mounting bracket, and 15-foot USB cable. Other USB cameras may work (see below), but do not include any technical support.

Manual Recording

To start or stop a manual recording, either:

- Use the controls in the **Manual Recording** area of the **Camera(s)** tab.
  When a manual recording is stopped, a file save-as dialog appears prompting you for the file name base to use.

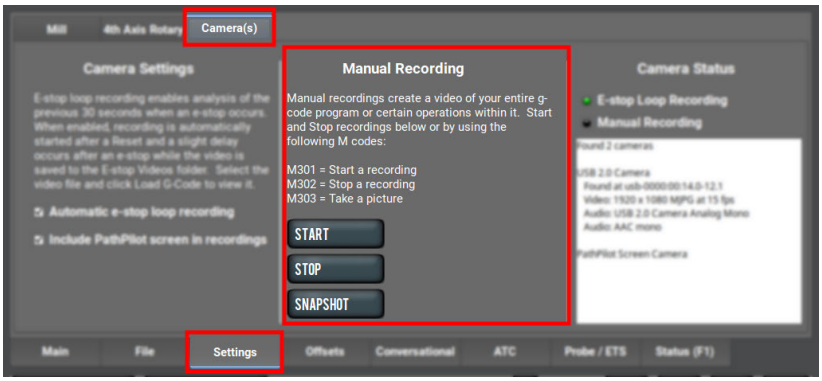


Figure 33: Manual recording controls.

- Select the **Video Camera Recording** button in the **Persistent Controls** section.

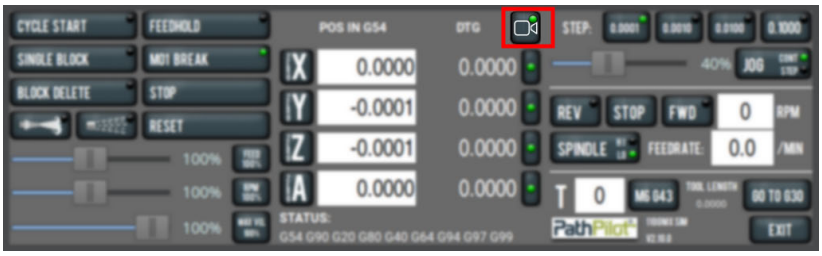


Figure 34: Video Camera Recording button.

Whenever PathPilot is recording from a USB camera and/or the virtual screen camera, the LED on this button is green. If PathPilot is recording and the button is pressed:

- If a program is running and not paused at an M00/M01, the recording is aborted.
- If a program is not running, but the machine is moving, the recording is aborted.
- Otherwise, if a manual recording is in progress, it is stopped and a file save as dialog will appear. If an automatic e-stop loop recording is in progress, it is aborted since no e-stop occurred.

To include a screen recording:

1. Toggle the **Include PathPilot screen in recordings** checkbox in the **Camera Settings** area of the **Camera(s)** tab to enable or disable screen recording.

Figure 35: Camera settings.

To take a picture (using all of the USB cameras at once):

1. Select **Snapshot** in the **Manual Recording** area of the **Camera(s)** tab.
   The **Main** tab displays.
2. Review the camera images, which display on top of the **Tool Path** area. The camera images refresh every 0.5 seconds.
3. Align the cameras or adjust lighting to your preference, and then select the **Shutter** button.

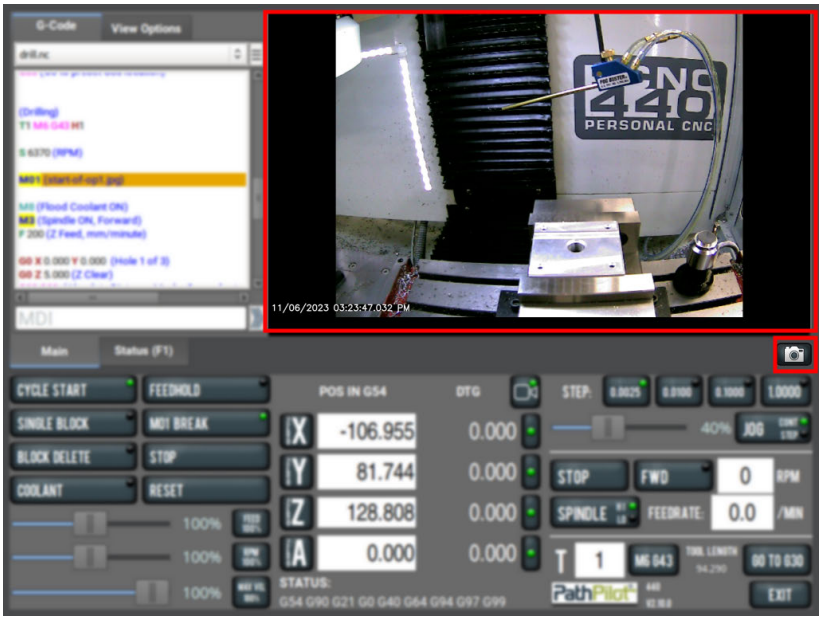


Figure 36: Example of taking a photo.

Automatic E-Stop Loop Recording ("Dashcam")

E-stop loop recording enables analysis of the previous 30 seconds after an E-stop. When enabled, recording is automatically started after reset.
To enable or disable the recording of emergency stops:

1. Toggle the **Automatic e-stop loop recording** checkbox in the **Camera Settings** area of the **Camera(s)** tab.

   Note: This feature is enabled by default.

   Automatic E-stop loop recording starts when the **Reset** button is selected. If you selected **Video Camera Recording** to abort a previous E-stop loop recording, select **Reset** to start it again.

To view E-stop videos:

1. A slight delay occurs after an E-stop while the video is saved to the **E-stop Videos** folder. Select the video file, and then select **Load G-Code** to view it.

Note: The E-Stop Videos folder is automatically monitored for internal drive space use. If the folder size grows beyond 5 GB, the oldest video files are automatically deleted until the folder size becomes less than 5 GB.

Review Video and Image Files

1. On the **File** tab, select the video or image file and select **Load G-Code**.
   A video player application starts or the image preview is displayed.
   Alternatively, you could transfer the video or image files to a Windows or macOS computer for review.

File Naming Convention

For manual and automatic E-stop recordings, the base file name for the recording has automatically chosen suffixes appended for each camera.
For example, if you stop a manual recording of two cameras, specify “Left Bracket Op1” as the name, and enabled screen recording, you'll see the following files:

File Name	Description of File
Left Bracket Op1_0.mp4	Camera 0 mp4 video file
Left Bracket Op1_0.log	Troubleshooting log for camera 0
Left Bracket Op1_1.mp4	Camera 1 mp4 video file
Left Bracket Op1_1.log	Troubleshooting log for camera 1
Left Bracket Op1_PP.mp4	PathPilot screen recording mp4 video file
Left Bracket Op1_PP.log	Troubleshooting log for screen recording

G-Code Commands

PathPilot supports three new M-codes to control cameras within G-code programs: M301, M302, and M303.
Example use cases:

- Record only across each M01 stop where the operator needs to flip a workpiece or change a tool.
- Create short videos that focus on unique aspects of the program to reduce later video editing.
- Record USB IO integration operations with robots or other devices (pneumatic vises, etc.).
- Monitor progress on a workpiece by including M303 throughout the program.

File Naming Conventions

Recordings or pictures created by M301/M302/M303 have automatically generated file names, with the base file name taken from the running G-code file. Video files are saved alongside the G-code file. The suffix for each file uses a time stamp format. This makes it easy to distinguish multiple runs of the same G-code program.

For example, if engrave.nc is running and uses M301 and M302 to create one recording on a machine with one camera, and screen recording is enabled, you'll see the following files:

File Name	Description of File
engrave_2023-02-21_16_58_33_0.mp4	Camera 0 mp4 video file
engrave_2023-02-21_16_58_33_0.log	Troubleshooting log for camera 0
engrave_2023-02-21_16_58_33_PP.mp4	PathPilot screen recording mp4 video file
engrave_2023-02-21_16_58_33_PP.log	Troubleshooting log for screen recording
engrave_2023-02-21_17_43_22.jpg	Picture taken by a single M303 later in the program

Use M01 to Take Pictures

In addition to displaying information like pictures or messages during an M01 break, you can also use a USB camera (if installed) to take a picture.
To use M01 to take pictures:

1. Add `M01 (op1_setup.jpg)` into your G-code program.
2. Run the G-code program.
3. When PathPilot executes the M01 it looks to see if the comment contains a file name.
   - If there isn't a file name: The comment is shown as instructional text across the tool path.
   - If there is a file name, but the file doesn’t exist yet and the extension is .jpg, .png, or .jpeg: The USB cameras are initialized and shown in the tool path display.
4. Select the **Shutter** button to take the picture and create the op1_setup.jpg file.
   In future runs of the G-code program, **op1_setup.jpg** will display to the operator for instructional purposes on the workpiece.

For more information, see "Display Information and Capture Images During an M00 or M01 Break" (page 143).

SET UP G-CODE PROGRAMS

Before running a G-code program, you must first make sure that the machine is properly set up for the specific G-code program.

Use a Probe with PathPilot

Use the Probe tab in the PathPilot interface to automate functions with a probe.

Set Up the Probe

Before using the functions on the Probe tab, you must first do the following:

1. Verify that tool number 99 (the probe tool) is in the spindle.
2. Disable the spindle to prevent any accidental spindle starts with the probe in the spindle.
3. Verify that the feed rate is appropriate for probing moves.

   Note: All probing moves occur at a feed rate specified by the DRO fields on the Probe Setup tab.

4. Press the probe tip and make sure that, from the PathPilot interface, on the Probe tab, the Accessory Input light comes on.
   This indicates that the probe polarity is correctly specified.
   If the Accessory Input light does not come on, you must change the probe polarity setting.

Use a Probe to Find a Feature's Location

To find the location of a workpiece or vise in the current work offset coordinates:

1. From the PathPilot interface, on the **Probe** tab, select the **X/Y/Z Probe** tab.
2. Position the probe near the workpiece or vise.
3. One at a time, select **Find X+**, **Find X-**, **Find Y+**, **Find Y-**, or **Find Z-**.

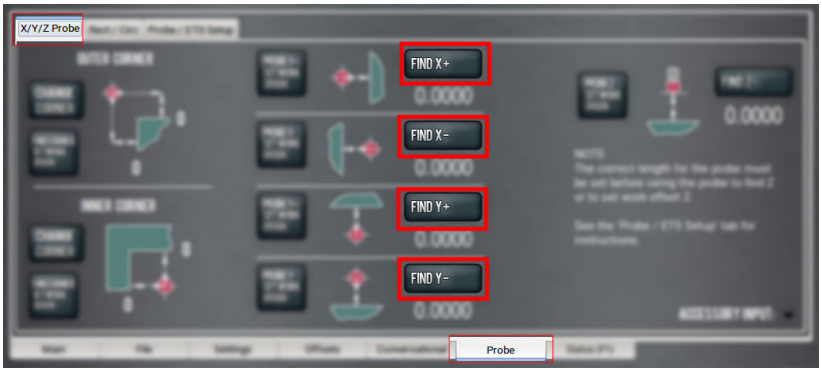


Figure 37: Probe tab.

The axis is probed, and the location of the probed surface is displayed.

Use a Probe to Set Work Offset Zeroes

You can set the work offsets of a workpiece or vise jaw using a probe.

Set the X and Y Work Offset Zero on the Corner of a Feature

1. From the PathPilot interface, on the **Probe** tab, select the **X/Y/Z Probe** tab.
2. Position the probe so that it is below the surface of the feature and 1 in. away from the vice jaw corner in the **X** and **Y** directions.
3. Select **Find Corner, Set Work Origin**.

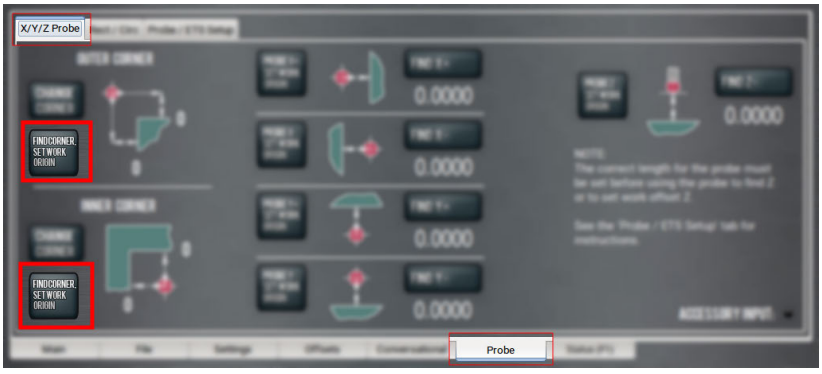


Figure 38: Probe tab.

The axes are probed, and the location of the probed surface is set as the current work offset's X/Y origin.

> **Note:** Select **Change Corner** to change the corner on which to probe.

Set the Work Offset Zeroes on a Feature

1. From the PathPilot interface, on the **Probe** tab, select the **X/Y/Z Probe** tab.
2. Position the probe near the workpiece or vise.
3. One at a time, select **Probe X+, Set Work Origin**, **Probe X-, Set Work Origin**, **Probe Y+, Set Work Origin**, **Probe Y-, Set Work Origin**, or **Probe Z-, Set Work Origin**.

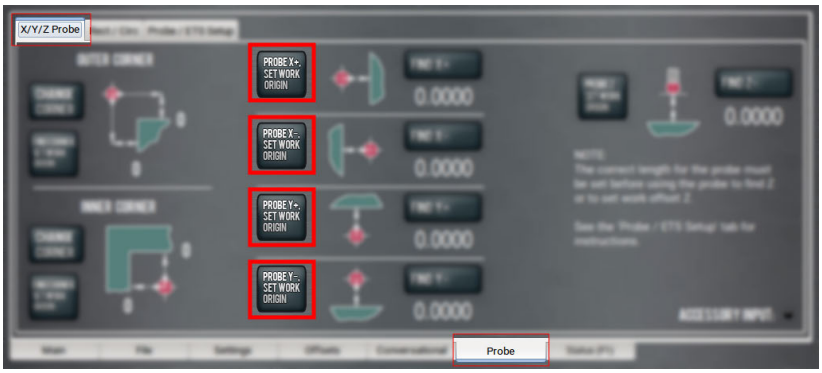


Figure 39: Probe tab.

The axis is probed, and the location of the probed surface is set as the current work offset's origin.

Use a Probe to Find the Center of a Feature

You can find the center of a pocket, slot, or boss on a part using a probe.

Find the Center of a Pocket

1. From the PathPilot interface, on the **Probe** tab, select the **Rect/Circ** tab.
2. Position the probe near the center of the pocket.
3. Select **Find Center, Set Work Origin** as shown in the following image.

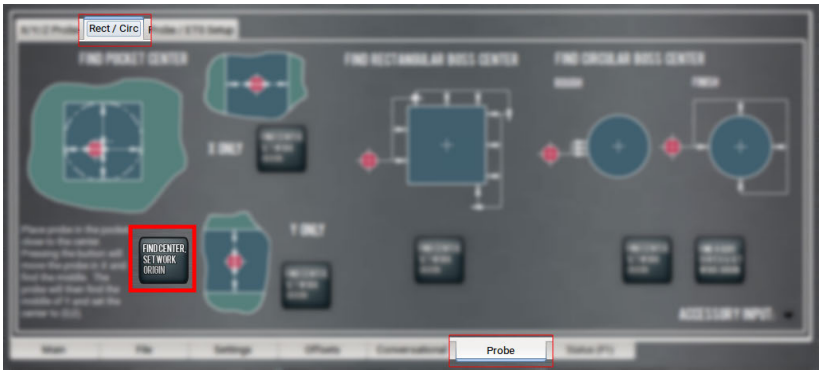


Figure 40: Probe tab.

Find the Center of a Slot

1. From the PathPilot interface, on the **Probe** tab, select the **Rect/Circ** tab.
2. Position the probe near the center of the slot.

3. Depending on the slot, do one of the following:
   - To probe the slot in the **X** direction only, select **Find Center, Set Work Origin** as shown in the following image.

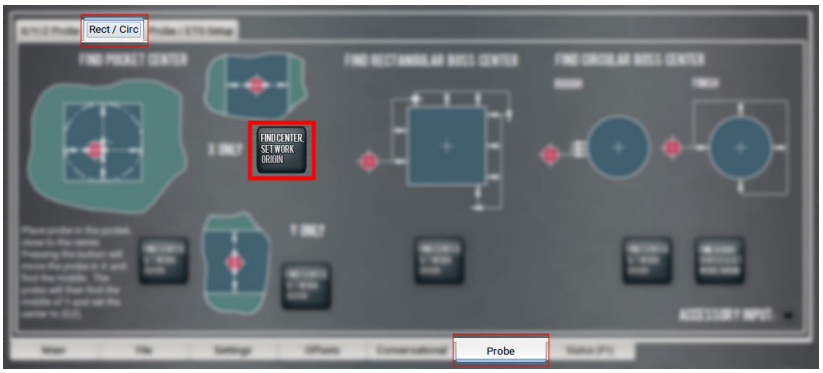


Figure 41: Probe tab.

   - To probe the slot in the **Y** direction only, select **Find Center, Set Work Origin** as shown in the following image.

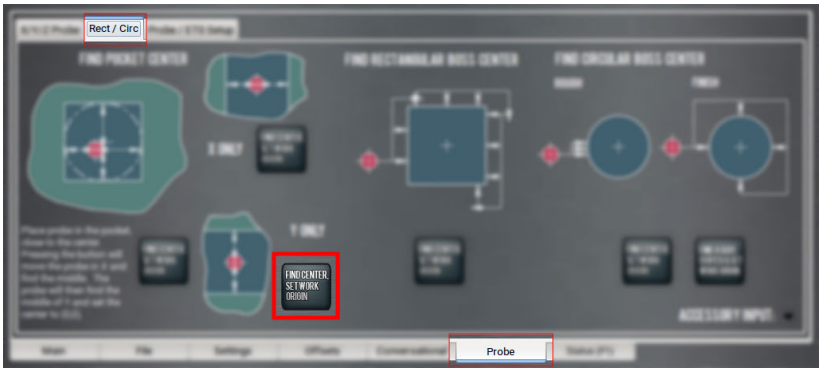


Figure 42: Probe tab.

Find the Center of a Rectangular Boss

1. From the PathPilot interface, on the **Probe** tab, select the **Rect/Circ** tab.
2. Position the probe below the top surface of the boss and on the left-hand side.

3. Select **Find Center, Set Work Origin** as shown in the following image.

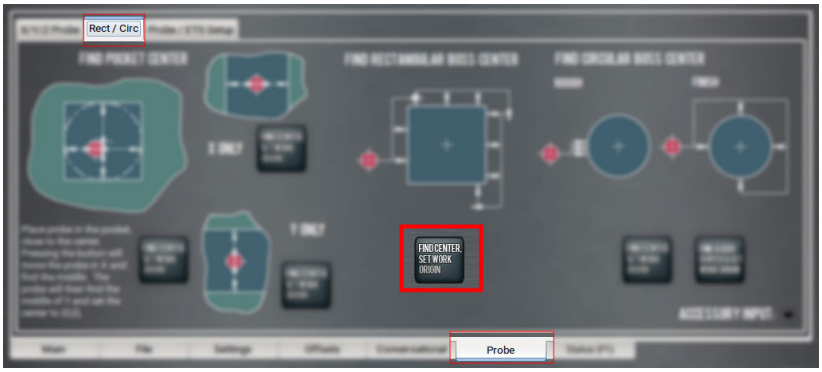


Figure 43: Probe tab.

The probe moves around the edge of the workpiece to find the center.

Find the Center of a Circular Boss

1. From the PathPilot interface, on the **Probe** tab, select the **Rect/Circ** tab.
2. Position the probe below the top surface of the boss and on the left-hand side.
3. Select **Find Center, Set Work Origin** as shown in the following image.

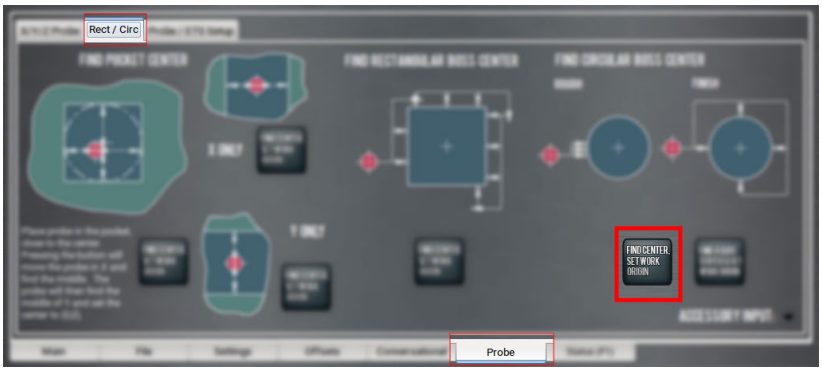


Figure 44: Probe tab.

The probe moves around the workpiece three times to determine the approximate center of the curve, and then makes four additional move to confirm the center of the circle.

Find the Center Rotation of an A-Axis

1. From the PathPilot interface, on the **Probe** tab, select the **Rect/Circ** tab.
2. Position the probe directly above the A-axis center of rotation.

3. Select **Find A Axis Center & Set Work Origin**.

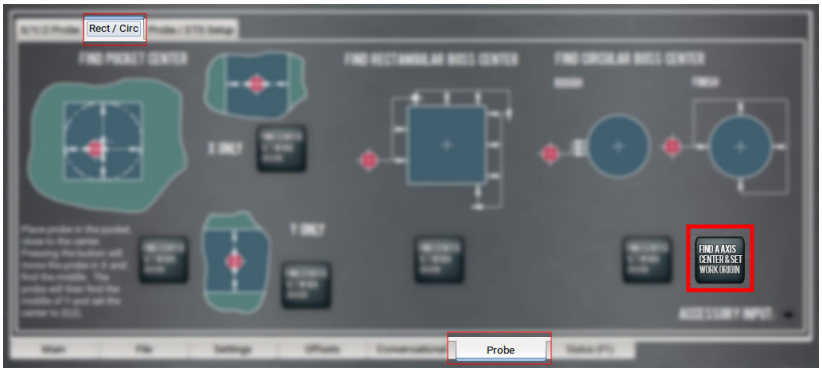


Figure 45: Probe tab.

The probe moves around the round workpiece mounted in the A-axis to find the center rotation of the A-axis.

Set Tool Length Offsets

Before running a G-code program, PathPilot must know the length of the tools that are required for the program. For more information on using tool length offsets, see "About Tool Offsets" (below).

Note: You can import a .csv file with tool offset data. For information, see "Import and Export the Tool Table" (page 106).

To set tool length offsets:

1. Verify that the machine is powered on and out of reset.
2. Put a tool into a tool holder, and set it aside to measure.
   For information, see Set Up Tooling.
3. From the PathPilot interface, on the **Offsets** tab, verify that the **Tool** tab is selected.
4. Find the **Tool Table** window.

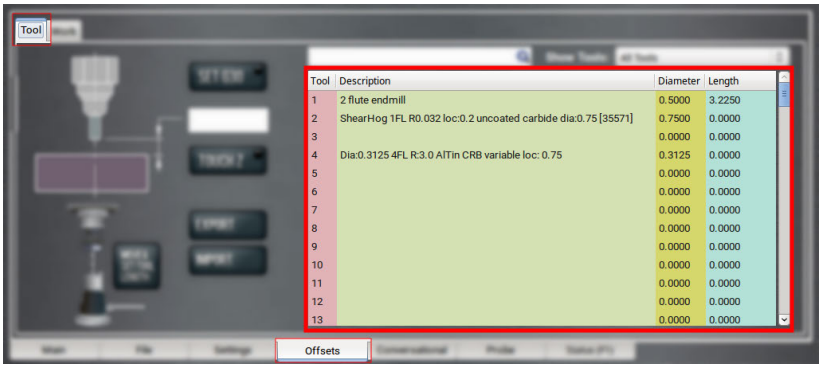


Figure 46: Tool Table window on the Offsets tab.

5. Depending on your workflow, you can measure your tools using any of the following methods:
   - **Use an Electronic Tool Setter** For information, see "Use an Electronic Tool Setter (ETS) to Measure Tools" (page 77).
   - **Touch Off of a Known Reference Height** For information, see "Touch Off the Tool Length Offsets" (on the next page).

About Tool Offsets

Tool offsets allow you to use various tools while still programming with respect to the workpiece. Tools can have different lengths (and, while using cutter radius compensation, different diameters).
The most common tool offset is the tool length offset: when you change tools, PathPilot must account for the difference in tool length. In CNC machines, the tool length offset is applied using a G43 command.
Before you begin a G-code program, you must verify the lengths of the tools in the program, and make sure that the lengths agree with the tool length offsets set in PathPilot:

- Each time you change tools, you must apply a new tool length offset in PathPilot.
- Each time you replace a tool, you must remeasure its length, and apply a new tool length offset in PathPilot.

> ***NOTICE!*** You must always verify that the physical length of a tool agrees with the tool length offset value set in PathPilot. If you don't, there's a risk that the tool length offset misrepresents the currently active tool in the spindle, which may result in a machine crash or damaged tooling, workpieces, or fixtures.

Touch Off the Tool Length Offsets

Touch off the tool length offsets by using a reference surface with a known height, which gives you a basis to measure any other tool lengths. Use any surface that is parallel (within 0.02 mm) to the machine table. For example:

- A **1-2-3 Block Set (PN 31950)**
- Box parallel

There are two steps to touch off the tool offsets. Complete the following steps in the order listed:

Set a Known Reference Height

This procedure sets a new Z zero position for the currently selected work offset.
To set a known reference height:

1. Identify a precision surface to use as a reference surface (like a 1-2-3 Block Set), and put it below the spindle on the machine table. Verify that there's a clear path from the spindle to the machine table.
2. Verify that the drive dogs won't contact the reference surface before the end face of the spindle.
3. Set a new, unused work offset (like G55). From the PathPilot interface, on the **Main** tab, in the **MDI Line** DRO field, type a work offset. Then select the **Enter** key.
   For information, see "Set Work Offsets" (page 79).
4. If there's already a tool in the spindle, remove it.
5. From the PathPilot interface, in the **Tool** DRO field, type `0`. Then select the **Enter** key.
6. Slowly jog the Z-axis down (-Z) until it's 0.04 in. (1 mm) from the reference surface.
7. Measure the thickness of a piece of paper, and put the paper on the reference surface. Note the thickness of the paper for later.

8. While moving the paper back-and-forth across the reference surface, slowly step the Z-axis down (-Z) until you feel a light pull on the piece of paper. This indicates that the paper is contacting the end face of the spindle.

   Note: It's easier to use step jogging for this task. For information on step jogging, see "About Step Jogging" (page 85).

9. From the PathPilot interface, in the **Z-axis** work offset DRO field, type the thickness of the piece of paper. Then select the **Enter** key.

Figure 47: Z-axis work offset DRO field.

   The reference surface is now set as the Z zero position in the current coordinate system.

10. To set the tool length offset, go to Measure Tools Using a Known Reference Height.

Measure Tools Using a Known Reference Height

This procedure sets the tool length offset using a known reference height. If you have not yet done so, you must first set the Z zero position; go to Set a Known Reference Height.
To measure tools using a known reference height:

1. Verify that the reference surface is still on the machine table with the piece of paper.
2. From the PathPilot interface, on the **Offsets** tab, find an unused tool number in the **Tool Table** window. Then, type a description for the tool you're measuring.
3. Put the tool holder into the spindle.
4. From the PathPilot interface, in the **Tool** DRO field, type the number of the tool. Then select the **Enter** key.

Figure 48: Tool DRO field.

5. Slowly jog the Z-axis down (-Z) until it is 0.04 in. (1 mm) from the reference surface.
6. Continue to slowly jog the Z-axis while slowly moving the piece of paper back-and-forth on the reference surface.

7. Stop jogging the Z-axis when you feel a light pull on the piece of paper, which indicates that it is in contact with the tool.
8. From the PathPilot interface, on the **Offsets** tab, in the **Tool Table**, select the tool for which you previously wrote a description.
9. In the **Touch Z** DRO field, type the thickness of the piece of paper. Then select the **Enter** key.

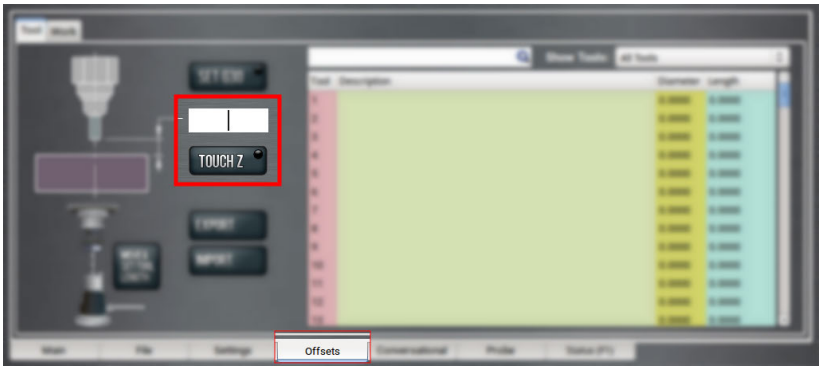


Figure 49: Touch Z DRO field and button.

10. Select **Touch Z**.
    The length of the tool is stored in the **Tool Table** window.
11. From the **Tool Table** window, in the **Length** column, verify that the length of the tool is correct.
12. In the **Diameter** column, type the diameter of the tool. Then select the **Enter** key.
13. Jog the Z-axis up (+Z).
    You've completed the procedure to measure a tool offset. Repeat this procedure for any remaining tooling you have. Once you're done adding tool length offsets, switch back to your work coordinate system.

Use an Electronic Tool Setter (ETS) to Measure Tools

An is a device used to measure the length of a cutting tool.
To use an ETS to measure tools:

1. Plug in the ETS to the **Accessory 2** port.
2. Put the ETS on the known reference surface below the spindle.
3. From the PathPilot interface, on the **Offsets** tab, in the **Tool Table** window, in the **Description** column, type a description for the tool.
4. In the **Diameter** column, type the diameter of the tool. Then select the **Enter** key.
5. Put a tool holder into the spindle.

6. From the PathPilot interface, type the tool number in the **Tool** DRO field. Then select the **Enter** key.

Figure 50: Tool DRO field.

7. Jog the Z-axis down (-Z) until it is above the ETS.
8. From the **Offsets** tab, on the **Tool** tab, select **Move and Set Tool Length**.

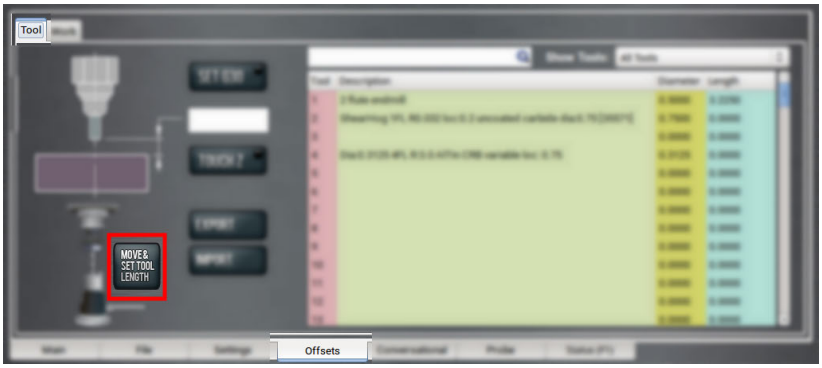


Figure 51: Tool tab on the Offsets tab.

> **Note:** Regardless of the initial feed rate, the final touch off feed rate while using an ETS is 2-1/2 in. per minute (IPM).

9. From the **Tool Table** window, in the **Length** column, verify that the length of the tool is correct.

Set Work Offsets

To set the current axis location to zero in the active work coordinate system:

- Select **Zero [Axis]**.

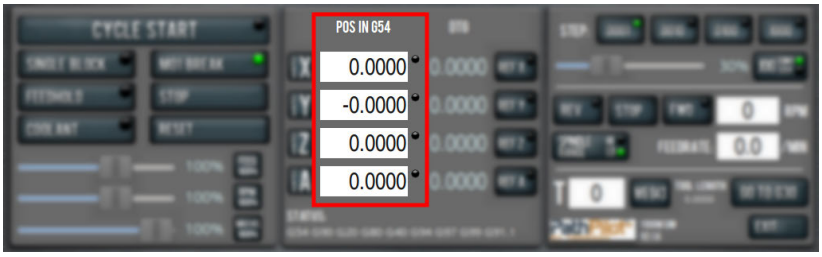


Figure 52: Work Offset DRO fields.

To change work offsets:

1. On the **Main** tab, in the **MDI Line** DRO field, type the new work offset to activate (for example, `G55`). Then select the **Enter** key.
2. The new work offset displays in the following locations in the PathPilot interface:
   - The **Status** read-only DRO field.
   - Above the **Work Offset** DRO fields.

Figure 53: Work offset indicated in the PathPilot interface.

> **Note:** The values in the **Work Offset** DRO fields update to indicate the new location of each axis in the new work offset.

For more information on using work offsets, see "About Work Offsets" (on the next page).

About Work Offsets

Work offsets allow you to think in terms of X, Y, and Z coordinates with respect to the part, rather than thinking of them with respect to the machine position. This means that you can jog the machine to an arbitrary location (like the end of a workpiece) and call that location zero.
You can save up to 500 work offsets in PathPilot. The naming structure varies based on the offset number, as detailed in the following table.

Work Offset Naming		
Offsets 1-9 (Use either name)		
Offset	**Extended Name**	**Name**
1	G54.1 P1	G54
2	G54.1 P2	G55
3	G54.1 P3	G56
4	G54.1 P4	G57
5	G54.1 P5	G58
6	G54.1 P6	G59
7	G54.1 P7	G59.1
8	G54.1 P8	G59.2
9	G54.1 P9	G59.3
Offsets 10-500 (Use extended name)		
Offset	**Extended Name**	**Name**
10	G54.1 P10	Not used
11	G54.1 P11	Not used
...		
499	G54.1 P499	Not used
500	G54.1 P500	Not used

View Work Offsets

To view the current work offset:

- From the **Offsets** tab, on the **Work** tab, identify the **Work Offsets Table** window.

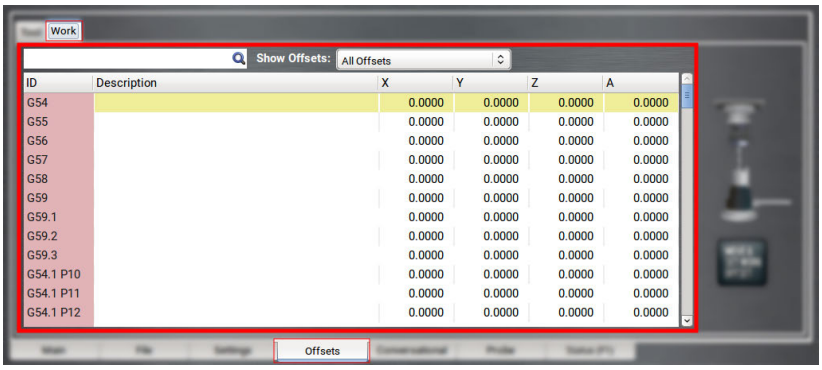


Figure 54: Work Offsets Table window.

The active work offset is highlighted.

To change the current work offset, go to "Set Work Offsets" (page 79).

View Available G-Code Modes

The G-Code Description window shows a list of all available G-code modes.
To view available G-code modes:

- From the **Settings** tab, find the **G-Code Description** window.

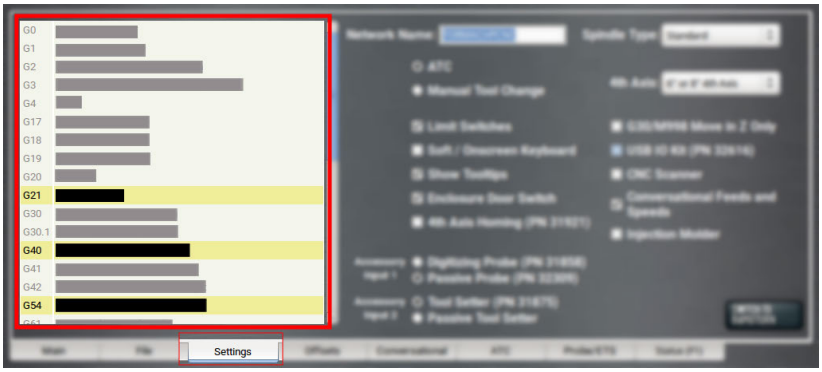


Figure 55: G-code Description window on the Settings tab.

RUN G-CODE PROGRAMS

While running a G-code program, use the following controls:

Bring the Machine Out of Reset

➢ Select **Reset**.

Figure 56: Reset button.

For more information on reset mode, see "About Reset Mode" (below).

About Reset Mode

When the machine is first powered on, or after an emergency stop, the Reset button flashes. When you select the flashing Reset button, PathPilot verifies communication to the machine and does the following activities:

- Brings the machine out of an emergency stop condition
- Clears alarms
- Clears the tool path backplot
- Resets all modal G-codes to their normal state
- Rewinds the currently loaded G-code program
- Stops machine motion, but is **not** a replacement for the Emergency Stop button

You can select the Reset button any time while the machine is on.

View the Active Axis to Jog

To find which axis is active while jogging your machine:

➢ Identify the light next to the **Work Offset** DRO fields.

Figure 57: Work Offset DRO fields.

For information, see "Jog the Machine" (below).

Jog the Machine

To switch between jogging modes:

➢ From the **Manual Control** area, in the **Jog** group, select **Jog**.
PathPilot toggles between continuous velocity mode and step mode.

Figure 58: Jog button.

When the **Cont** green light is on, continuous velocity mode is selected.
When the **Step** green light is on, step mode is selected.

To use continuous velocity mode:

➢ Set the velocity: drag the **Jog Speed** slider.

Figure 59: Jog Speed slider.

For more information on continuous velocity mode, see "About Continuous Velocity Jogging" (on the next page).
To use step mode, select the step size. Do one of the following, depending on your accessories:

- In the **Manual Control Area**, in the **Jog** group, select the step size.
The **Step** button's light comes on, indicating which step size is active.

Figure 60: Step buttons (in G20 mode).

- On the (optional) Jog Shuttle, press the Step button to toggle the currently selected step size. In the PathPilot interface, the **Step** button's light comes on, indicating which step size is active.

For more information on step mode, see "About Step Jogging" (on the next page).

Jog in Continuous Velocity Mode

In continuous mode, the machine jogs at a continuous velocity.
To select continuous velocity mode:

➢ In the **Manual Control** area, select **Jog**.

Figure 61: Continuous velocity jogging controls.

When the **Cont** green light is on, continuous velocity mode is selected.
When the **Step** green light is on, step mode is selected.

To set the velocity:

➢ Drag the **Jog Speed** slider.

Figure 62: Jog Speed slider.

About Continuous Velocity Jogging

While jogging in continuous velocity mode, the machine moves at a constant speed for as long as:

- A keyboard key is pressed
- The Jog Shuttle outer ring is twisted away from the neutral position

This is useful when you're doing things like:

- Roughly positioning the machine (for example, to move the spindle head away from the workpiece).
- Moving the machine a certain distance at a constant speed.

Jog in Step Mode

In step mode, the machine jogs in steps, which range based on the programming mode you're using:

- **Imperial (G20) Mode** 0.0001 in. to 0.1000 in.
- **Metric (G21) Mode** 0.01 mm to 10 mm

To select the step size:

➢ In the **Manual Control Area**, select the step size.
The **Step** button's light comes on, indicating which step size is active.

Figure 63: Step buttons (in G20 mode).

About Step Jogging

While jogging in step mode, the machine moves one step each time you either press a jog key on the keyboard or click the inner wheel of the Jog Shuttle. The jog step sizes range depending on the programming mode you are using:

- **Imperial (G20) Mode** 0.0001 in. to 0.1000 in.
- **Metric (G21) Mode** 0.01 mm to 10 mm

Step jogging mode is useful to finely move the machine, like when you're indicating a workpiece or manually setting tool lengths.
The jog keys on the keyboard only move the machine in steps when step mode is indicated in PathPilot. The inner wheel on the jog shuttle always moves the machine in steps, regardless of which mode is indicated in PathPilot.

View the Current Machine Position

➢ Identify the **Work Offset** DRO fields.

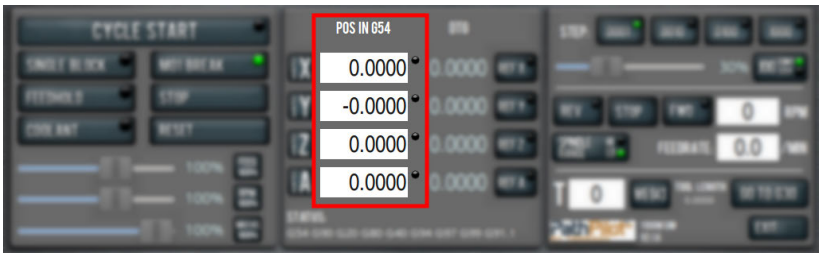


Figure 64: Work Offset DRO fields.

The position is expressed by the currently active work offset coordinate system (like G54 or G55).

When the machine isn't moving, you can edit the DRO fields. For more information on setting work offsets, go to "Set Work Offsets" (page 79).

Reference the Machine

1. Verify that the machine can freely move to its reference position (at the ends of travel).
2. To verify that the tooling is clear of any possible obstructions, reference the Z-axis before referencing the other axes: from the PathPilot interface, select **Ref Z**.

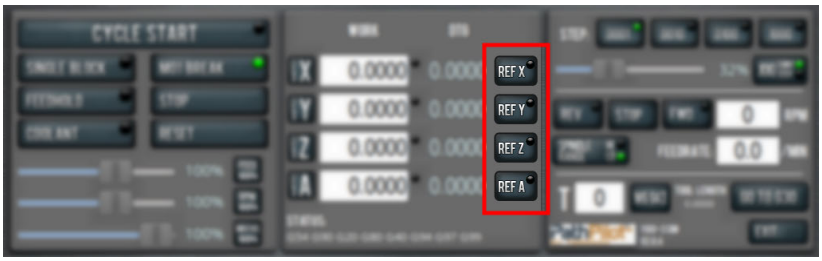


Figure 65: Reference buttons.

3. Once the spindle is clear of any possible obstructions, continue referencing all axes.

> **Note:** You can select the buttons one after another. Once the machine references one axis, it'll move on to the next.

After each axis is referenced, its button light comes on.

For more information on referencing the machine, see "About Referencing" (below).

About Referencing

You must reference the machine to establish a known position for PathPilot. The position that's set while referencing the machine is the origin of the machine coordinate system. Without referencing the machine, PathPilot won't know the current position of the machine axes.
You must reference the machine at the following times:

- After you power on the machine
- After you push in the Emergency Stop button
- Before running a G-code program
- Before using MDI commands
- Before setting work or tool offsets
- After a collision or an axis stall/fault

When referencing, the machine moves each axis to the end of its travel. The machine stops at the limit switch, which sets the axis' reference position.

Start a Program

➢ From the PathPilot interface, in the **Main** tab, select **Cycle Start**.

Figure 66: Cycle Start button.

For more information on starting a program, see "About Cycle Start" (below).
If you can't start a program, go to "Cycle Start Reference" (on the next page).

About Cycle Start

While a program is running, the Cycle Start button's light is on.
The Cycle Start button's light flashes if motion is paused during the program. The following modes may pause motion during a program:

- Single block
- Feed hold
- M01 break

If machine motion pauses a single block, feed hold, or M01 break, the Cycle Start button flashes until it's selected again.

Cycle Start Reference

The Cycle Start button doesn't operate if you select it:

- While you're not in the Main tab. For information, see "Main Tab" (page 17).
- Before you've loaded a G-code program. For information, see "Load G-Code" (page 35).
- Before referencing the machine. For information, see "Reference the Machine" (page 86).

Stop Machine Motion

➢ From the **Program Control** area, select **Stop**.

Figure 67: Stop button.

View the Active G-Code Modes

To find the currently active G-code modes and the currently active tool at a glance:

➢ Identify the **Status** read-only DRO field.

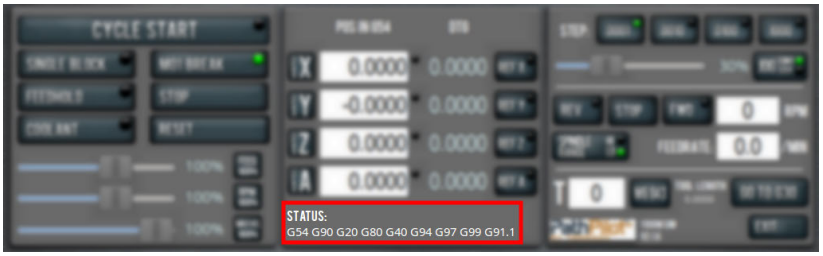


Figure 68: Status read-only DRO field.

For more information on G-code modes, go to "View Available G-Code Modes" (page 81).

View the Distance to Go

To view the distance to go:

- Identify the **DTG** read-only DRO fields.

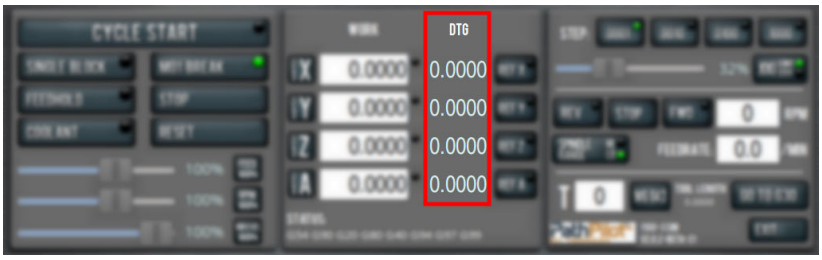


Figure 69: DTG read-only DRO fields.

The value is the remaining distance in any programmed move.

For more information on using the **DTG** read-only DRO fields, see "About Distance to Go" (below).

About Distance to Go

While a program is running, the DTG read-only DRO fields show the remaining distance in each move.
After using the feed hold function or the maxvel override function, look at the distance to go. This read-only DRO field is useful to prove out a part program.

CONTROL G-CODE PROGRAMS

If necessary, use the following controls to add to your G-code program:

Use the Feed Hold Function

➢ Select **Feed Hold**.

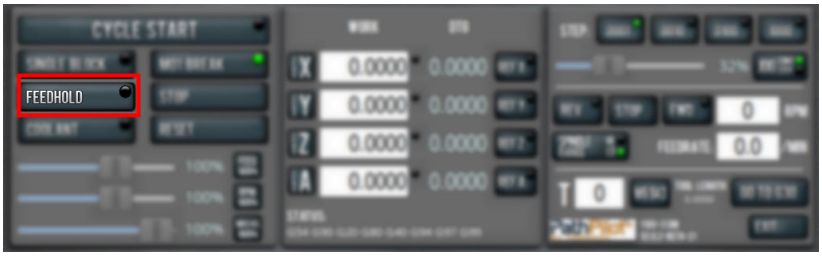


Figure 70: Feed Hold button.

> **Tip!** Use the **Spacebar** key to quickly activate the feed hold function.

For more information on using the feed hold function, see "About Feed Hold" (below).

About Feed Hold

When the feed hold function is active, the Feed Hold button's light is on.
The feed hold function pauses machine motion — aside from the spindle — and the Cycle Start button flashes. For information, see "About Cycle Start" (page 87).

> **Note:** If the machine isn't moving, the feed hold function doesn't have an effect.

You can use the feed hold function either while a program is running or while you are using manual data input (MDI) commands. If the program is running a spindle-synchronized move, the feed hold function is delayed until the move is complete.

Use the Feed Rate Override Function

To use the feed rate override function:

- Using the **Feed Rate Override** slider, change the programmed feed rate by a specific percentage.

Figure 71: Feed Rate Override slider.

> **Note:** Percentages range from 1-200%.

To remove the feed rate override function:

- Select **Feed 100%**.
  The feed rate returns to 100% of its programmed value (it's no longer overriden).

For more information on the feed rate override function, see "About Feed Rate Override" (below).

About Feed Rate Override

You can use the feed rate override function while you're doing any of the following activities:

- Using manual data input (MDI) commands
- Jogging
- Running a program with G01, G02, or G03 commands

The feed rate override function does not affect G00 (rapid) commands. It's ignored if:

- The program is running a spindle-synchronized move
- An M48 (disable feed and speed overrides) command is used

To indicate lack of motion or unusual levels, the slider turns yellow when it's either at 0% or above 100%.
The Feed Rate Override slider and Feed 100% button work similarly to the spindle override controls. They affect the commanded feed rate by a percentage from 1-200%. The feed rate override works for MDI, jogging, and G-code program G01/G02/G03 moves. The override has no effect on G00 (rapid) moves.

Use M01 Break Mode

➢ Select **M01 Break**.

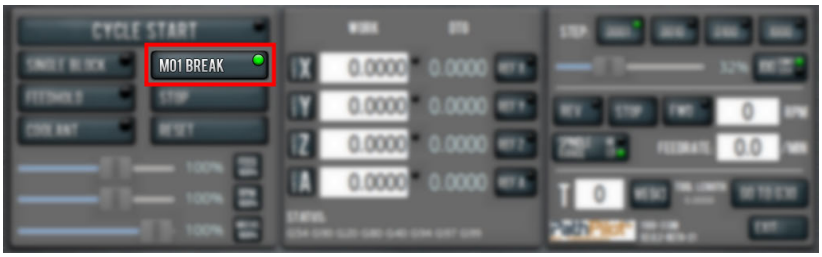


Figure 72: M01 Break button.

For more information on using M01 break mode, see "About M01 Break" (below).

About M01 Break

When the M01 break mode is active, the M01 Break button's light is on. When the M01 break mode is inactive, the M01 Break button's light is off.
M01 break mode enables any M01 (optional stop) commands that are programmed in the G-code file. You can turn M01 break mode on or off either before starting a program or while a program is running.

- **When M01 Break is Active** Machine motion stops after PathPilot reaches an M01 command, and the Cycle Start button flashes. For information, see "About Cycle Start" (page 87).
- **When M01 Break is Inactive** PathPilot ignores all programmed M01 commands.

Use the Maxvel Override Function

To use the maxvel override function:

➢ Using the **Maxvel Override** slider, change the maximum velocity by a specified percentage.

Figure 73: Maxvel Override slider.

To remove the maxvel override function:

➢ Select **Maxvel 100%**.

For more information on using the maxvel override function, see "About Maxvel Override" (on the next page).

About Maxvel Override

The maxvel override function affects G00 and G01 commands, and it's useful for:

- **Running a Program for the First Time** Drag the Maxvel Override slider to 0% to verify that all DRO fields look appropriate.
- **Safety** If you're running a spindle-synchronized move, a maxvel override isn't ignored.
  Verify that the maxvel override value allows the machine to use the programmed feed rate during spindle-synchronized moves. If it can't, the spindle-synchronized move won't produce the results you want.

To indicate lack of motion or unusual levels, the slider turns yellow when it's either at 0% or above 100%.

Use Single Block Mode

➢ Select **Single Block**.

Figure 74: Single Block button.

For more information on using single block mode, see "About Single Block" (below).

About Single Block

While single block mode is active, the Single Block button's light is on.
Single block mode runs one line of G-code at a time. After each line, motion is paused, and the Cycle Start button flashes. For information, see "About Cycle Start" (page 87).
You can turn single block mode on or off either before starting a program or while a program is running. For information, see "Use Single Block Mode" (above).

Note: Single block mode ignores non-motion lines, like comment lines or blank lines.

Use the Voltage Override Function

To use the voltage override function:

- Using the **Voltage Override** slider, change the programmed voltage by a specific percentage.

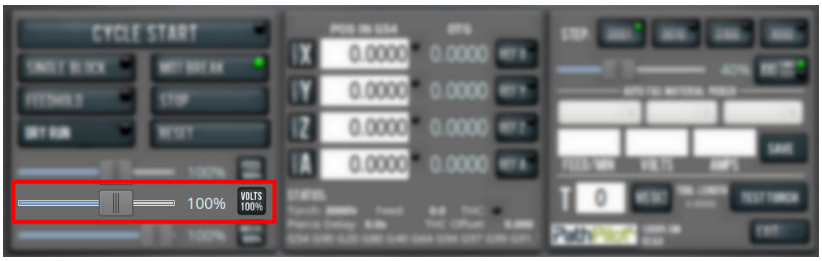


Figure 75: Voltage Override slider.

> **Note:** Percentages range from 1-200%.

To remove the voltage override function:

- Select **Volts 100%**.
  The voltage returns to 100% of its programmed value (it's no longer overriden).

Change the Tool Number

The Tool DRO field shows the tool currently in the spindle.

Figure 76: Tool DRO field.

To change the tool number (and apply its tool length offset):

1. In the **Tool** DRO field, type a number (the valid range is from 0-1000). Then select the **Enter** key.

> **Note:** You can also select **M6 G43**. For information, see "About M6 G43" (on the next page).

About M6 G43

The M6 G43 button is a shortcut used to do the following:

- Change the number of the currently-loaded tool in the spindle to the number typed in the Tool DRO field. This is the equivalent of an M06 command.
- Apply the tool length offset for that tool typed in the Tool DRO field. For more information on tool length offsets, see "Set Tool Length Offsets" (page 74). This is the equivalent of a G43 command.

Use a G30 Position

The Go to G30 button moves the machine to a predefined G30 position. For information, see "About G30" (on the next page).
To set a G30 position:

1. Jog the machine to the desired G30 position.
2. From the **Offsets** tab, select **Set G30**.

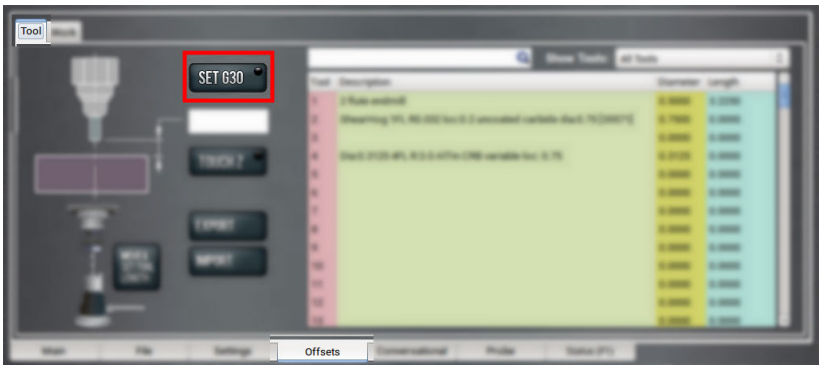


Figure 77: Set G30 button.

To go to a set G30 position:

- Do one of the following:
  - Use a `G30` command in a G-code program.
  - Select **Go To G30**.

Figure 78: Go to G30 button.

Note: The G30 position defaults to only moving the Z-axis.

About G30

A G30 command in a G-code program moves the machine to a preset position. For more information on setting a G30 position, see "Use a G30 Position" (on the previous page).
Use a G30 move to start a coordinated movement of the axes. You can limit the movement to only the Z-axis. For information, see "Limit G30 Moves" (page 59).

Tip! It's useful to program a G30 move right before a tool change so that the machine can jog to a safe tool change position.

View the Tool Length

- Identify the **Tool Length** read-only DRO field.

Figure 79: Tool Length DRO field.

If the tool offset matches the number of the tool in the **Tool** DRO field, the text is light blue on a gray background.
If the tool offset doesn't match the number of the tool in the **Tool** DRO field, the text is orange on a red background.

Manually Enter Commands

You can send G-code commands directly to the machine by using the MDI Line DRO field. For information, see "About the MDI Line DRO Field" (below).
To manually enter commands:

1. Select the **MDI Line** DRO field.

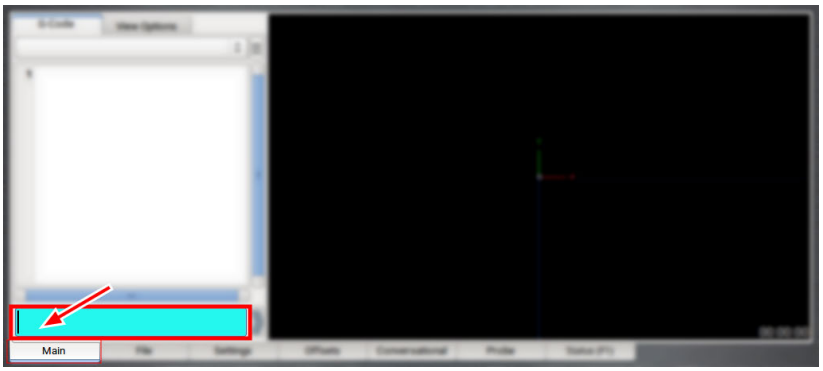


Figure 80: MDI Line DRO field.

The DRO field highlights.

2. Type the command.

> **Note:** You can use the **Backspace**, **Delete**, **Left Arrow**, and **Right Arrow** keys to correct typing errors.

3. You must press the **Enter** key to execute the command. To abandon the command, press **Esc**.

About the MDI Line DRO Field

The MDI Line DRO field allows you to send commands (or, manual data input) directly to PathPilot. For information, see "Manually Enter Commands" (above).
The MDI Line DRO field saves up to 100 of your most recent commands, which are saved after a power cycle.
When you select the MDI Line DRO field, all keystrokes are used within the field — so, you can't jog the machine.

Admin Commands Reference

Use the following commands in PathPilot:

Admin Command	Use to...
ADMIN CALC	Open the calculator.
ADMIN CONFIG	Change the configuration of the PathPilot interface.
ADMIN DATE	Customize the PathPilot controller's date and time.
ADMIN DISPLAY	Customize the PathPilot controller's screen display.
ADMIN DROPBOX	Connect your controller to a Dropbox account.
ADMIN KEYBOARD	Customize the PathPilot controller's keyboard layout.
ADMIN MEMORY	Determine how much total RAM is on your controller.
ADMIN MOUSE	Change the mouse preferences, like pointer speed and right- or left-hand button mapping
ADMIN NETWORK	Configure a Wi-Fi network.
ADMIN SETTINGS BACKUP	Create a backup of tool offset and fixture information to store externally.
ADMIN SETTINGS RESTORE	Restore tool offset and fixture information backup from an external location.
ADMIN TOOLTIP DELAYMS	Set the milliseconds prior to displaying the tooltip (and then again for the expanded tooltip). The default is 1200 milliseconds.
ADMIN TOOLTIP MAXDISPLAYSEC	Limit the amount of time the expanded tooltip displays. The default is 15 seconds.
ADMIN TOUCHSCREEN	Configure the optional Touch Screen Kit.

Copy Recently Entered Commands

1. From the **MDI Line** DRO field, press either the **Up Arrow** key or the **Down Arrow** key.
   The previously entered command displays.
2. You must press the **Enter** key to execute the command. To abandon the command, press **Esc**.

For information, see "Manually Enter Commands" (on the previous page).

Use the AutoFS Material Picker

PathPilot on the 1300PL is designed so that you don't need to manually program feeds and speeds in your G-code. Instead, feed rate, amperage, pierce parameters, and torch height control voltage are all set at program run-time using automatic look-up tables based on the material you're cutting.
AutoFS is designed so that you can export a program once from your post-processor and run it multiple times with different materials, all without re-posting. All material-specific machine parameters are set by the M200 AutoFS M-code (rather than directly programmed in the G-code).

Writing G-Code with AutoFS

A standard block of G-code before a cutting operation might look like this:

```
M210 P123 (Set THC voltage to 123v)
M211 P45 (Set plasma source to 45A)
M207 P0.15 (Set pierce height to 0.15")
M208 P0.08 (Set cut height to 0.08")
M209 P2 (Set a two second pierce delay)
F225 (Set cutting feed rate of 225 IPM)

G15 (Perform ohmic probe)
G16 (Pierce and start cutting)
```

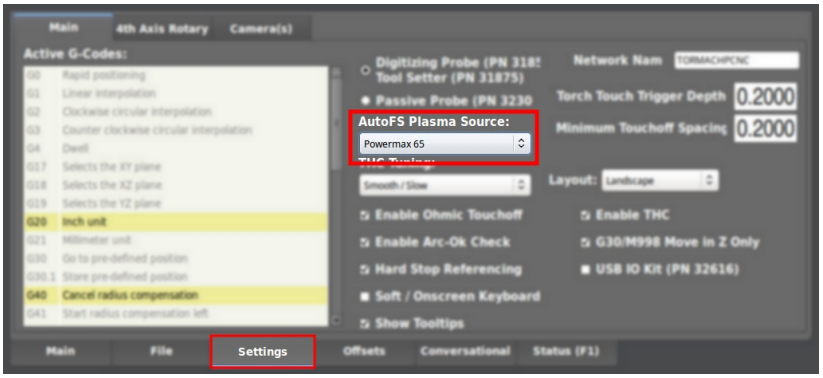
The block of code shown above is totally valid and can be used if you prefer to hard code your cutting parameters. Alternatively, the code shown above can be replaced with the following when AutoFS is being used:

```
M200 (Set cutting parameters from AutoFS tables)

G15 (Perform ohmic probe)
G16 (Pierce and start cutting)
```

Using AutoFS

1. On the **Settings** tab, select which plasma source you're using from the **Auto FS Plasma Source** drop-down menu.



2. Before running your program, select the material you're cutting from the AutoFS dropdowns on the **Main** tab of PathPilot.
   The DROs turn green, which indicates that they were automatically set.

With a material selected, the next time an `M200` code is encountered in a program, PathPilot will set all of the cutting parameters for that material.

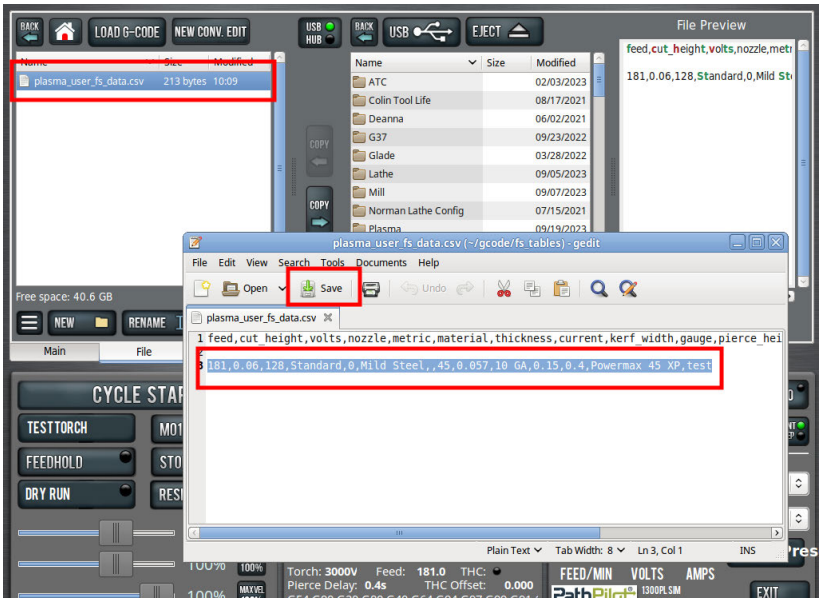
Creating Custom AutoFS Presets

Since your shop air supply and cutting conditions might differ from those used to create the AutoFS presets, you can save your own custom settings for a material at any time.
To create new AutoFS presets:

1. From the **Main** tab, select the material you're cutting from the AutoFS dropdowns, and then type your cut parameters into the DRO fields.
2. Select **Save**.
   The values are stored as a new preset for that material.

Deleting Custom AutoFS Presets

1. From the File tab, navigate to the fs_tables folder, and open the plasma_user_fs_data.csv file.
2. Highlight the line of the preset you want to delete and delete it. Then, save the file.



> **Note:** This file is read on boot only, so the preset remains visible until the controller is power cycled.

Use Cycle Counters (M30 and M99)

On the Main tab, the Tool Path display shows M30 and M99 cycle counters. They're useful to count parts completed during unattended operation. For each M-code, there's an A and B counter. This provides more flexibility, because you can reset them to 0 independently.
For example, you could use M30 A to count parts each shift, and M30 B to count parts each week. The cycle counters persist across the controller's power cycles.

Monitor Cycle Counters

- In the **MDI Line** DRO field, type `ADMIN CYCLECOUNTER` to show or hide the counters and to reset them to 0.

Change Cycle Counter Values

The cycle counters are implemented as read-only persistent G-code numbered parameters, as detailed in the following table. If needed, the cycle counter value can be read in G-code.

Cycle Counter	Parameter
M30 A	#5650
M30 B	#5651
M99 A	#5652
M99 B	#5653

To change a counter value explicitly, use a G10 command: `G10 L99 P~ Q~`

- P~ selects the cycle counter to change. Use any of the values detailed in the following table.

Cycle Counter	P~
M30 A	0
M30 B	1
M99 A	2
M99 B	3

- Q~ specifies the value to set the cycle counter. If Q~ is omitted, the cycle counter is incremented by 1. For example, if you program `G10 L99 P2`, the M99 A cycle counter increments by 1.

SYSTEM FILE MANAGEMENT

To keep the files on your system backed up and organized, use the following controls:

Manage System Files

Use the File tab to manage system files on the PathPilot controller. For information, see "About System Files" (below).
To manage system files:

- From the PathPilot interface, on the **File** tab, do any of the following from the **Controller Files** window:
  - Select a file, and then select **New Folder**, **Rename**, or **Delete**.
  - Select a file, and go to the **Options** menu. Then, select **Copy**, **Cut**, or **Paste**.

To navigate through the system files:

- Select **Back** or **Home**.

About System Files

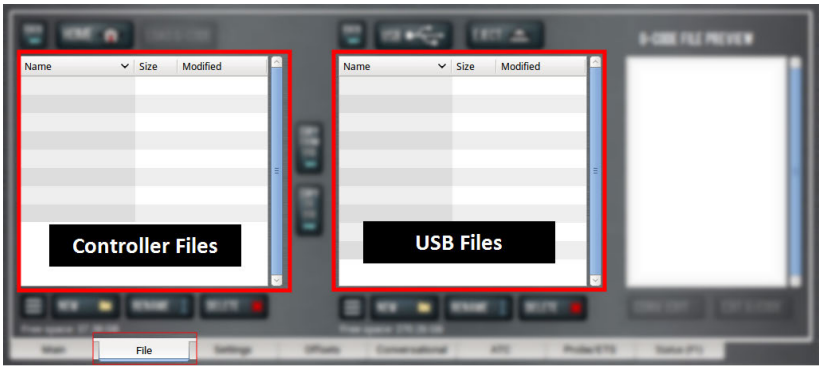


Figure 81: File tab.

PathPilot doesn't run G-code program files from a USB drive. You must first transfer files to the PathPilot controller. For information on transferring files, see "Transfer Files to and From the Controller" (page 35).

Create Backup Files

1. Insert a blank, formatted USB drive into the PathPilot controller.

> **Note:** To prevent errors when backing up and restoring files, only use a blank, formatted USB drive.

2. From the PathPilot interface, on the **Main** tab, in the **MDI Line** DRO field, type `ADMIN SETTINGS BACKUP`. Then select the **Enter** key.
   PathPilot generates a backup .zip file, and the **Admin Settings Backup** dialog box displays.

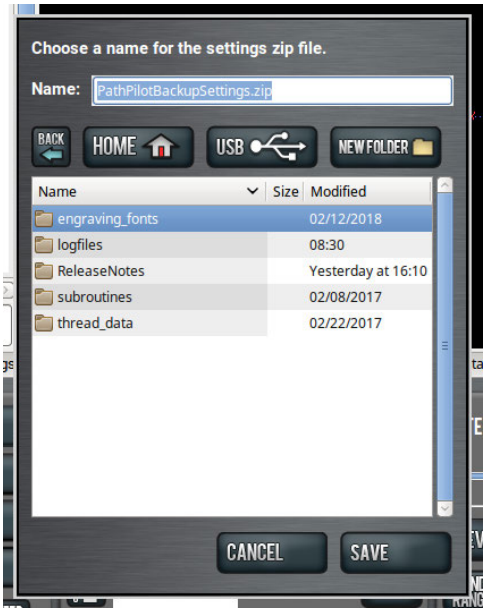


Figure 82: Admin Settings Backup dialog box.

3. From the **Admin Settings Backup** dialog box, specify where (on the PathPilot controller or on a USB drive) to save the backup .zip file.
4. Select **Save**.
   The backup .zip file is saved in the location you specified in Step 3.
5. If you saved the backup .zip file on the PathPilot controller, you must manually transfer it — along with other files you want to back up (like G-code programs) — to a USB drive. From the PathPilot interface, on the **File** tab, in the **Controller Files** window, select the backup .zip file and any other files you want to back up.

Figure 83: Controller Files window on the File tab.

> **Note:** Files must have unique names. If they don't, PathPilot prompts you to overwrite or rename files, or cancel the file transfer.

6. To prevent errors, make sure you don't include the following folders:
    - **logfiles**
    - **media**
    - **ReleaseNotes**
    - **subroutines**
    - **USB**
7. Select **Copy to USB**.
   The files are copied and display in the **USB Files** window.
8. Eject the USB drive from the PathPilot controller.
9. From the PathPilot interface, select **Exit**.
10. Verify that all files are properly saved: insert the USB drive on a device other than the PathPilot controller, and review the list of files on the USB drive.
11. *(Optional)* As an extra precaution, copy all the files onto the device.

About Backup Files

Make a regular backup of all tool offset and fixture information and machine settings stored on your PathPilot controller. Store the file externally to use if you replace your controller or restore it to factory settings.

Restore Backup Files

1. Insert the USB drive with your backup files into the PathPilot controller.
2. From the PathPilot interface, on the **Main** tab, in the **MDI Line** DRO field, type `ADMIN SETTINGS RESTORE`. Then select the **Enter** key.
   The **Admin Settings Restore** dialog box displays.

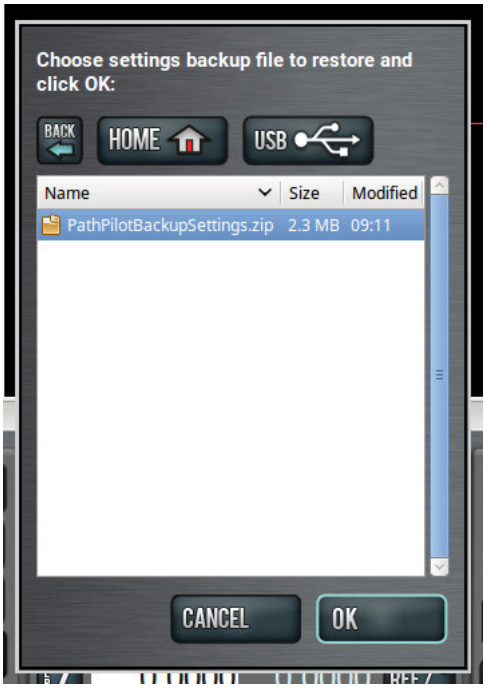


Figure 84: Admin Settings Restore dialog box.

3. From the **Admin Settings Restore** dialog box, navigate to the backup .zip file on the USB drive, and then select **OK**.
   The PathPilot operating system restores the backup, then restarts.

4. If you backed up any other files onto the USB drive, you must manually transfer the files to the PathPilot controller. From the PathPilot interface, on the **File** tab, in the **USB Files** window, select the files you want to transfer.

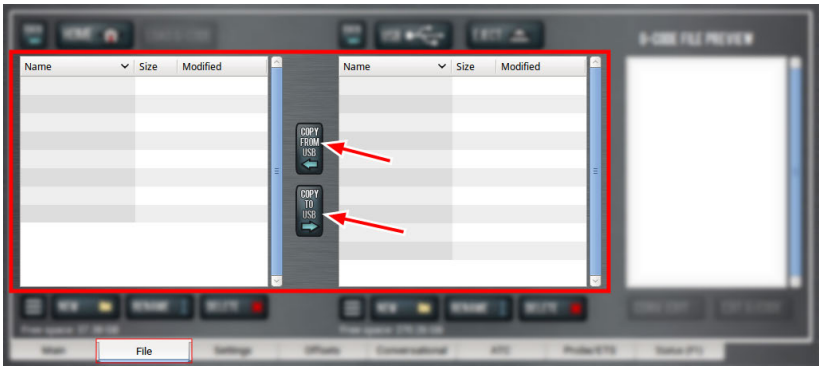


Figure 85: USB Files window on the File tab.

> **Note:** To navigate backward, select **Back**. To navigate to the top level, select **USB**.

5. From the **Controller Files** window, select the folder into which you want to copy the files.
6. Select **Copy From USB**.
   The files display in the **Controller Files** window.

> **Note:** Files must have unique names. If they don't, PathPilot prompts you to overwrite or rename files, or cancel the file transfer.

Import and Export the Tool Table

You can manage the tool table using an external .csv file.

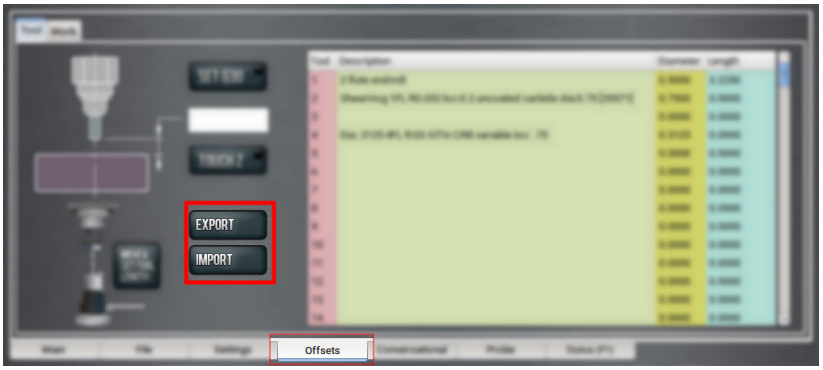


Figure 86: Export and Import buttons on the Offsets tab.

Import a .csv File

1. Transfer the .csv file to a USB drive.
2. Insert the USB drive into the PathPilot controller.
3. Confirm that the PathPilot controller is on.

4. From the **Offsets** tab, select **Import**.
   The **Import** dialog box displays.

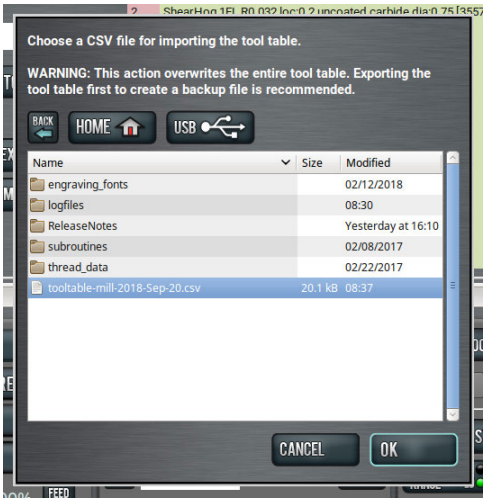


Figure 87: Import dialog box.

5. Navigate to the .csv file on the USB drive. Then, select **OK**.
   The .csv file updates the tool table.

Export the Tool Table as a .csv File

1. From the **Offsets** tab, select **Export**.
   PathPilot generates the .csv file, and the **Export** dialog box displays.

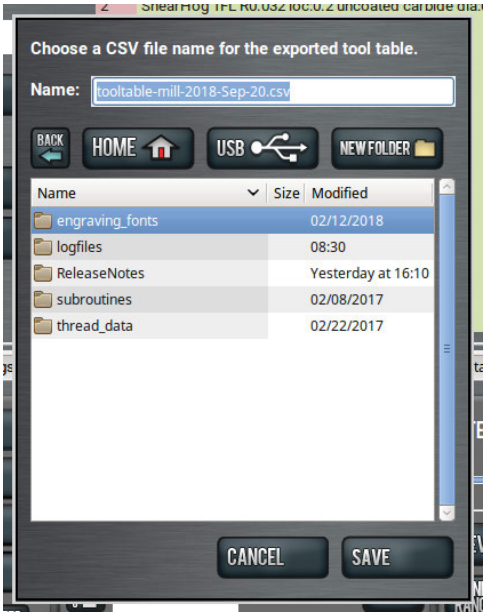


Figure 88: Export dialog box.

2. In the **Name** DRO field, type the name for the .csv file.
3. Select **Save**.
   The .csv file is saved in the **File** tab.

4. From the **File** tab, select the newly created .csv file, and then select **Copy to USB**.
5. Select **Eject**.
   It's safe to remove the USB drive from the controller.


TITLE: externalThread-Mill
Aluminum
-any-
WORK OFFSET: G59
TOOL: 2
SPINDLE RPM: 555
FEEDRATE: 100.0
Z FEEDRATE: 100.0
Z CLEAR: 0.6000
(Flutes: N/A Chip load per tooth: N/A)
(SFM: 0 – MRR: N/A)
POST TO FILE
APPEND TO FILE
Face
Profile
Pocket
Drill / Tap
Thread Mill
Engrave
DXF (Mill)
EXTERNAL
INTERNAL
DEPTH
X (FROM DRILL TABLE)
Y (FROM DRILL TABLE)
NUMBER OF PASSES:
THREADS/IN:
3.0
Z START:
3.0000
Z CLEAR
PITCH (IN):
0.3333
2.0000
Z END:
LEFT
RIGHT
Main
File
Settings
Offsets
Conversational
Probe
Status
CYCLE START
SINGLE BLOCK
M01 BREAK
FEEDHOLD
STOP
COOLANT
RESET
100%
FEED 100%
100%
RPM 100%
100%
MAXVEL 100%
POS IN G55
DTG
ZERO X
5.1931
0.0000
REF X
ZERO Y
0.6512
0.0000
REF Y
ZERO Z
0.0000
0.0000
REF Z
ZERO A
0.0000
0.0000
REF A
STATUS:
G55 G90 G20 G80 G40 G94 G97 G99 G91.1
STEP:
REV
SPINDLE RANGE
T
0
05:06:02

PROGRAMMING

IN THIS SECTION, YOU'LL LEARN:

- About the languages that are understood and interpreted by PathPilot.

CONTENTS

BEFORE YOU BEGIN

- **Referring to This Section** Use this section only for reference. To learn about the principles of the control language (so that you can write programs by hand from first principles, for example), we recommend that you consult an introductory textbook on G-code programming.
- **Creating and Editing G-Code Files** We recommend using a text editor like Gedit or Notepad++. Don't use a word processor to create or edit G-code files — it'll leave unseen codes that could cause problems or prevent a G-code file from working.

PROGRAMMING OVERVIEW

Read the following sections for a G-code overview:

About G-Code Programming Language

A G-code program is made up of one or more lines of code. Each line of code is called a block, and can include commands to the machine. Blocks are collected into a file, which makes a program.
A block is normally made up of an optional line number at the beginning, followed by one or more words, which groups the elements together into a single statement.
A word is a letter followed by a number (or, something that evaluates to a number). A word can either give a command or provide an argument to a command.
A program is one or more blocks, each separated by a line break. Blocks in a program are executed either:

- Sequentially (from the top of the program to the bottom)
- Until an end command (M02 or M30) is encountered

E X A M P L E :
`G01 X3` is a valid line of code with two words:

- `G01` is a command: the machine should move in a straight line at the programmed feed rate.
- `X3` provides an argument value: the value of X should be 3 at the end of the move.

Most commands start with either G (general) or M (miscellaneous) — G-codes and M-codes.
There are two commands (M02 and M30) that end a program. A program can end before the end of a file. If there are lines in a file after the end of a program, they're not meant to be executed in the normal flow (they're generally parts of subroutines).

G-Code Formatting Reference

A permissible block of input code is made up of the following programming elements, in order, with the restriction that there is a maximum of 256 characters allowed on a line:

1. *(Optional)* Block delete character (`/`)
2. *(Optional)* Line number
3. Any number of words, parameter settings, and comments
4. End of line marker (carriage return or line break)

Programs are limited to 999,999 lines of code.
Spaces and tabs are allowed anywhere on a line of code and do not change the meaning of the line, except inside comments. Blank lines are allowed in the input, but they're ignored. Input is not case sensitive (except in

comments), so any letter outside a comment may be in uppercase or lowercase without changing the meaning of a line.

EXAMPLE
`G00 x +0. 12 34y 7` is equal to `G00 x+0.1234 y7`

A line may have:

- Any number of G words, but two G words from the same modal group may not appear on the same line.
- Zero to four M words, but two M words from the same modal group may not appear on the same line.
- For all other legal letters, a line may have only one word beginning with that letter.

Any input not explicitly allowed is illegal, and causes the interpreter to either signal an error or ignore the line. PathPilot omits blocks of code that are prefixed with a block delete character (`/`).
PathPilot sometimes ignores things it doesn't understand. If a command doesn't work as expected, or does nothing, make sure that it's correctly typed. PathPilot doesn't check for excessively high machining feeds or speeds, and it doesn't detect situations where a legal command will do something unfortunate (like machining a fixture).

Line Numbers

A line number is indicated by the following, in the order listed:

1. The letter N
2. An integer (with no sign) between 0 and 99,999,999 (which must be written without commas)

Line numbers may be repeated, or used out of order, but that's rare in normal practice. A line number isn't required, and is often omitted.

Words

A word is indicated by the following, in the order listed:

1. A letter other than N or O
2. A real value

Letters

Words may begin with any of the following letters, except N or O:

Note: Several letters (I, J, K, L, P and R) may have different meanings in different contexts.

Letter	Description
A	A-axis
B	B-axis

Letter	Description
C	C-axis
D	Tool radius compensation number
F	Feed rate
G	General function
H	Tool length offset index
I	X-axis offset for arcs
J	Y-axis offset for arcs
K	Z-axis offset for arcs
L	Number of repetitions in canned cycles and subroutines, or key used with G10
M	Miscellaneous function
N	Line number
O	Subroutine label number
P	Dwell time in canned cycles, dwell time with G04, key used with G10, or tapping depth in M871 through M874
Q	Feed increment in a G83 canned cycle, or repetitions of subroutine call
R	Arc radius, or canned cycle retract level
S	Spindle speed
T	Tool selection
U	Synonymous with A
V	Synonymous with B
W	Synonymous with C
X	X-axis
Y	Y-axis
Z	Z-axis

Values

A real value is one of the following:

- An explicit number (like 341, or -0.8807)
- An expression (like [2+2.4])

- A parameter value (like #88)
- A unary operation value (like acos[0])

> **Note:** In the command examples that we use, the tilde symbol (~) stands for a real value. If `L~` is written in an example, the ~ is often referred to as the L number. Similarly the ~ in `H~` may be called the H number, and so on for any other letter.

A number is a subset of a real value. Processing a real value to come up with a number is called evaluating. An explicit number evaluates to itself.
Explicit numbers have the following rules (in this case, a digit is a single character, 0 through 9):

- A number must consist of the following, in the order listed:
  1. An optional plus or minus sign
  2. Zero to many digits
  3. *(Optional)* One decimal point
  4. Zero to many digits
- There must be at least one digit somewhere in the number.
- It must be either an integer (no decimal point) or a decimals (decimal point).
- It may have any number of digits (subject to line length limitations).

  > **Note:** PathPilot only keeps 17 significant figures, which is enough for all known applications.

- A non-zero number with no sign as the first character is assumed to be positive.

Initial zeros (a zero before the decimal point and the first non-zero digit) and trailing zeros (a zero after the decimal point and the last non-zero digit) are allowed, but not required. A number written with initial or trailing zeros has the same value when it is read as if the extra zeros were not there.
Numbers used for specific purposes by PathPilot are often restricted to some finite set of values, or to some range of values. In many uses, decimal numbers must be close enough to an integer to be accepted as a valid input. A decimal number which is supposed to be close to an integer is considered close enough if it is within 0.0001 of an integer.

Order of Execution

If a parameter setting of the same parameter is repeated on a line (like `#3=15 #3=6`), only the last setting takes effect. It's illogical, but not illegal, to set the same parameter twice on the same line.
The order of items on a line doesn't determine the order of execution on the commands.
Three types of items' order may vary on a line (as given earlier in this section):

- **Word** May be reordered in any way without changing the meaning of the line.
- **Parameter Setting** If it's reordered, there is no change in the meaning of the line unless the same parameter is set more than once. In this case, only the last setting of the parameter takes effect.

  EXAMPLE
  When the line `#3=15 #3=6` is interpreted, the value of parameter 3 is 6. If the order is reversed to `#3=6 #3=15` and the line is interpreted, the value of parameter 3 is 15.

- **Comment** If it contains more than one comment and is reordered, only the last comment is used. If each group is kept in order or reordered without changing the meaning of the line, then the three groups may be interleaved in any way without changing the meaning of the line.

  EXAMPLE
  `G40 G01 #3=15 (foo) #4=-7.0` has five items and means exactly the same thing in any of the 120 possible orders, like `#4=-7.0 G01 #3=15 G40 (foo)`, for the five items.

The order of execution of items on a line is critical to safe and effective machine operation. If items occur on the same line, they are executed in a particular order. To impose a different order (like to turn coolant off before the spindle is stopped), code the commands on separate blocks.
The order of execution is as follows:

1. Comment (including message)
2. Set feed rate mode (`G93`, `G94`, `G95`)
3. Set feed rate (`F`)
4. Set spindle speed (`S`)
5. Special I/O (`M62` to `M68`)

   Note: This is not supported.

6. Change tool (`T`)
7. Spindle on/off (`M03`, `M04`, `M05`)
8. Save State (`M70`, `M73`, restore state (`M72`), invalidate state (`M71`)
9. Coolant on/off (`M07`, `M08`, `M09`)
10. Enable/disable overrides (`M48`, `M49`, `M50`, `M51`, `M52`, `M53`)
11. Operator defined commands (`M101` to `M199`)
12. Dwell (`G04`)
13. Set active plane (`G17`, `G18`, `G19`)
14. Set length units (`G20`, `G21`)

15. Cutter radius compensation on/off (G40, G41, G42)
16. Tool table offset on/off (G43, G49)
17. Fixture table select (G54 through G58 and G59 P~)
18. Set path control mode (G61, G61.1, G64)
19. Set distance mode (G90, G91)
20. Set canned cycle return level mode (G98, G99)
21. Home, change coordinate system data (G10) or set offsets (G92, G94)
22. Perform motion (G00 to G03, G12, G13, G80 to G89 as modified by G53)
23. Stop (M00, M01, M02, M30, M60)

Modal Groups

G- and M-codes are, generally speaking, modal — they cause the machining system to change from one mode to another. The mode stays active until another command changes it implicitly or explicitly.

EXAMPLE
If coolant is turned on (M07 or M08), it stays on until it is explicitly turned off in the program (M09).

A few G-codes and M-codes are non-modal (like Dwell (G04)). These codes have effect only on the lines on which they occur.
Modal commands are arranged in sets, called modal groups. Only one member of a modal group may be in force at any given time. In general, a modal group contains commands for which it is logically impossible for two members to be in effect at the same time (like inch units (G20) vs. millimeter units (G21)).
A machining system may be in many modes at the same time, with one mode from each modal group being in effect.
For all G-code modal groups, when a machining system is ready to accept commands, one member of the modal group must be in effect. There are default settings for these modal groups. When the machining system is turned on or re-initialized, default values are automatically in effect.
Modal groups for G-codes are detailed in the following table.

Group	Commands	Group Description
Group 1	{G00, G01, G02, G03, G33, G38.x, G73, G76, G80, G81, G82, G84, G85, G86, G88, G89}	Motion (one always in effect)
Group 2	{G17, G18, G19, G17.1, G17.2, G17.3}	Plane selection
Group 3	{G90, G91}	Distance mode
Group 4	{G90.1, G91.1}	Arc distance mode

Group	Commands	Group Description
Group 5	{G93, G94}	Feed rate mode
Group 6	{G20, G21}	Length units
Group 7	{G40, G41, G42, G41.1, G42.1}	Cutter compensation
Group 8	{G43, G43.1, G49}	Tool length offset
Group 10	{G98, G99}	Return mode in canned cycles
Group 12	{G54, G55, G56, G57, G58, G59, G59.1, G59.2, G59.3}	Select work offset coordinate system
Group 13	{G61, G61.1, G64}	Path control mode
Group 14	{G96, G97}	Spindle control mode
Group 15	{G07, G08}	Lathe diameter mode

Modal groups for M-codes are detailed in the following table.

Group	Commands	Group Description
Group 4	{M00, M01, M02, M30, M60}	Program stop and program end
Group 7	{M03, M04, M05}	Spindle control
Group 8	{M07, M08, M09}	Coolant control (special case: `M07` and `M08` may be active at the same time)
Group 9	{M48, M49}	Override control

Non-modal G-codes are:

- **Group 0** {G04, G10, G28, G30, G53, G92, G92.1, G92.2, G92.3}

Comments

You can add comments to lines of G-code to help clarify the intention of the programmer. To embed a comment in a line, use parentheses. To add a comment to the end of a line, use a semicolon.

Note: The semicolon is not treated as the start of a comment when it's enclosed in parentheses.

Comments can appear between words, but they can't be between words and their corresponding parameter.

EXAMPLE:
`S100(set speed)F200(feed)` is okay, but `S(speed)100F(feed)` is not.

Supported G-Codes Reference

G-Code	Description
G00	Rapid linear motion
G01	Linear motion at feed rate
G02	Clockwise arc at feed rate
G03	Counterclockwise arc at feed rate
G04	Dwell
G07, G08	Diameter / radius mode **Note:** The 15L Slant-PRO lathe and the RapidTurn both use `G07` (X positions displayed in diameter values). `G08` is not used or supported in PathPilot.
G10 L1	Set tool table
G10 L2	Set coordinate system
G10 L10	Set tool table – calculated – workpiece
G10 L11	Set tool table – calculated – fixture
G10 L20	Set coordinate system
G15	Workpiece probe
G16	Pierce
G17, G18, G19	Plane selection
G20/G21	Length units
G28	Return to predefined position
G28.1	Return to predefined position
G30	Return to predefined position
G33	Spindle synchronized motion (like threading)
G33.1	Rigid tapping
G40	Cancel cutter compensation
G41/G42	Cutter compensation (left/right)
G41.1, G42.1	Dynamic cutter compensation

G-Code	Description
G43	Apply tool length offset
G49	Cancel tool length compensation
G53	Absolute coordinates
G54-G59.3	Select work offset coordinate system
G61/G61.1	Set exact path control mode
G64	Set blended path control mode
G73	High-speed peck drill
G76	Multi-pass threading cycle
G80	Cancel canned cycles
G81	Drilling cycle
G82	Simple drilling cycle
G83	Peck drilling cycle
G85	Boring cycle
G86	Boring cycle
G88	Boring cycle
G89	Boring cycle
G90, G90.1	Arc distance mode
G91, G91.1	Incremental distance mode
G92	Offset coordinates and set parameters
G92.x	Cancel G92, etc.
G93, G94, G95	Feed rate mode
G96, G97	Spindle control mode
G98	Initial level return / R-point level after canned cycles

PROGRAMMING G-CODE

Read the following sections as a G-code reference:

About the Examples Used

Many commands require axis words (`X~`, `Y~` ,`Z~`, or `A~`) as an argument. Unless explicitly stated otherwise, you can make the following assumptions:

- Axis words specify a destination point
- Axis words relate to the currently active coordinate system, unless explicitly described as being in the absolute coordinate system
- Where axis words are optional, any omitted axes retain their current value

Any items in the command examples not explicitly described as optional are required.

Rapid Linear Motion (G00)

For rapid linear motion, program: `G00 X~ Y~ Z~ A~`

- `X~` is the X-axis coordinate
- `Y~` is the Y-axis coordinate
- `Z~` is the Z-axis coordinate
- `A~` is the A-axis coordinate

This produces coordinated linear motion to the destination point at the current traverse rate (or slower, if the machine won't go that fast). It's expected that cutting won't take place when a `G00` command is executing. The `G00` is optional if the current motion mode is G00.
Depending on where the tool is located, follow these two basic rules:

1. If the Z value represents a cutting move in the positive direction (like out of a hole), the X-axis should be moved last.
2. If the Z value represents a move in the negative direction, the X-axis should be moved first.

Conditions

The motion differs if:

- Cutter radius compensation is active
- G53 is programmed on the same line

Troubleshooting

It's an error if:

- All axis words are omitted
  The axis words are optional, except that at least one must be used.
- `G10`, `G28`, `G30` or `G92` appear in the same block

Linear Motion at Feed Rate (G01)

For linear motion at feed rate (for cutting or not), program: `G01 X~ Y~ Z~ A~ F~`

- `X~` is the X-axis coordinate
- `Y~` is the Y-axis coordinate
- `Z~` is the Z-axis coordinate
- `A~` is the A-axis coordinate
- `F~` is the feed rate

This produces coordinated linear motion to the destination point at the current feed rate (or slower, if the machine won't go that fast). The `G01` is optional if the current motion mode is G01.

Conditions

The motion differs if:

- Cutter radius compensation is active
- G53 is programmed on the same line

Troubleshooting

It's an error if:

- All axis words are omitted
  The axis words are optional, except that at least one must be used.
- `G10`, `G28`, `G30`, or `G92` appear in the same block
- No F word is specified

Arc at Feed Rate (G02 and G03)

A circular or helical arc is specified using either `G02` (clockwise arc) or `G03` (counterclockwise arc). The axis of the circle or helix must be parallel to the X-, Y- or Z-axis of the machine coordinate system. The axis (or equivalently, the plane perpendicular to the axis) is selected with `G17` (Z-axis, XY-plane), `G18` (Y-axis, XZ-plane) or `G19` (X-axis, YZ-plane). If the arc is circular, it lies in a plane parallel to the selected plane.
If a line of code makes an arc and includes rotational axis motion, the rotational axes turn at a constant rate so that the rotational motion starts and finishes when the XYZ motion starts and finishes. This is rare.
The motion differs if cutter radius compensation is active.
Two formats are allowed for specifying an arc: the center format and the radius format. In both formats, the `G02` or `G03` is optional if it's the current motion mode.

Radius Format Arc

For a clockwise arc in radius format, program: `G02 X~ Y~ Z~ A~ R~`
For a counterclockwise arc in radius format, program: `G03 X~ Y~ Z~ A~ R~`

- `X~` is the X-axis coordinate
- `Y~` is the Y-axis coordinate
- `Z~` is the Z-axis coordinate
- `A~` is the A-axis coordinate
- `R~` is the radius of the arc

In radius format, the coordinates of the end point of the arc in the selected plane are specified along with the radius of the arc. A positive radius indicates that the arc turns through 180 degrees or less, while a negative radius indicates a turn of 180 degrees to 359.999 degrees.
If the arc is helical, the value of the end point of the arc on the coordinate axis parallel to the axis of the helix is also specified.
We don't recommend programming radius format arcs that are:

- Nearly full circles
- Semicircles
- Nearly semicircles

A small change in the location of the end point produces a much larger change in the location of the center of the circle (and the middle of the arc). The magnification effect is large enough that rounding error in a number can produce out-of-tolerance cuts.
You can program arcs that are:

- Up to 165 degrees
- Between 195 degrees to 345 degrees

EXAMPLE
`G17 G02 X 1.0 Y 1.5 R 2.0 Z 0.5` is a radius format command to mill an arc, which makes a clockwise (as viewed from the positive Z-axis) circular or helical arc whose axis is parallel to the Z-axis, ending where X = 1.0, Y = 1.5, and Z = 0.5, with a radius of 2.0. If the starting value of Z is 0.5, this is an arc of a circle parallel to the XY-plane; otherwise, it's a helical arc.

Troubleshooting

It's an error if:

- Both of the axis words for the axes of the selected plane are omitted
  The axis words are all optional except that at least one of the two words for the axes in the selected plane must be used.
- No R word is given
- The end point of the arc is the same as the current point
- `G10`, `G28`, `G30`, or `G92` appear in the same block

Center Format Arc

For a clockwise arc in center format, program: `G02 X~ Y~ Z~ I~ J~`
For a counterclockwise arc in center format, program: `G03 X~ Y~ Z~ I~ J~`

- `X~` is the X-axis coordinate
- `Y~` is the Y-axis coordinate
- `Z~` is the Z-axis coordinate
- `A~` is the A-axis coordinate
- `I~` is the center of arc (X coordinate)
- `J~` is the center of arc (Y coordinate)
- `K~` is the center of arc (Z coordinate)

In the center format, the coordinates of the end point of the arc in the selected plane are specified along with the offsets of the center of the arc from the current location. In this format, it's okay if the end point of the arc is the same as the current point.
The center is specified using the I, J, K words associated with the active plane. These specify the center relative to the current point at the start of the arc, defined in incremental coordinates from the start point.
It's an error if:

- When the arc is projected on the selected plane, the distance from the current point to the center differs from the distance from the end point to the center by more than 0.0002 inches (if you're programming in inches) or 0.002 millimeters (if you're programming in millimeters)
- `G10`, `G28`, `G30`, or `G92` appear in the same block

Arc in XY Plane

When the XY-plane is selected, program: `G02 X~ Y~ Z~ A~ I~ J~` (or, use `G03` instead of `G02`)
I and J are the offsets from the current location or coordinates – depending on arc distance mode (G90.1/G91.1) of the center of the circle (X and Y directions, respectively).
It's an error if:

- X and Y are both omitted
  The axis words are all optional except that at least one of X and Y must be used.
- I and J are both omitted
  I and J are optional except that at least one of the two must be used.

Arc in XZ Plane

When the XZ-plane is selected, program: `G02 X~ Y~ Z~ A~ I~ K~` (or, use `G03` instead of `G02`)
I and K are the offsets from the current location or coordinates – depending on arc distance mode (G90.1/G91.1) of the center of the circle (X and Z directions, respectively).
It's an error if:

- X and Z are both omitted
  The axis words are all optional except that at least one of X and Z must be used.
- I and K are both omitted
  I and K are optional except that at least one of the two must be used.

Arc in YZ Plane

When the YZ-plane is selected, program: `G02 X~ Y~ Z~ A~ J~ K~` (or, use `G03` instead of `G02`)
J and K are the offsets from the current location or coordinates – depending on depending on arc distance mode (G90.1/G91.1) of the center of the circle (Y and Z directions, respectively).
It's an error if:

- Y and Z are both omitted
  The axis words are all optional except that at least one of Y and Z must be used.
- J and K are both omitted
  J and K are optional except that at least one of the two must be used.

E X A M P L E
`G17 G02 X1.0 Y1.6 I0.3 J0.4 Z0.9` is a center format command to mill an arc in incremental arc distance mode (G91.1) that makes a clockwise (as viewed from the positive Z-axis), circular, or helical arc whose axis is parallel to the Z-axis, ending where X = 1.0, Y = 1.6, and Z = 0.9, with its center offset in the X direction by 0.3 units from the current X location and offset in the Y direction by 0.4 units from the current Y location. If the current location has X = 0.7, Y = 0.7 at the outset, the center is at X = 1.0, Y = 1.1. If the starting value of Z is 0.9, this is a circular arc; otherwise, it's a helical arc. The radius of this arc would be 0.5.

In the center format, the radius of the arc is not specified, but it may be found easily as the distance from the center of the circle to either the current point or the end point of the arc.

```
(Sample Program G02EX3:)
(Workpiece Size: X4, Y3, Z1)
(Tool: Tool #2, 1/4” Slot Drill)
(Tool Start Position: X0, Y0, Z1)
N2 G90 G80 G40 G54 G20 G17 G94 G64 (SAFETY BLOCK)
N5 G90 G20
N10 M06 T2 G43 H2
N15 M03 S1200
N20 G00 X1 Y1
N25 Z0.1
N30 G01 Z-0.1 F5
N35 G02 X2 Y2 I1 J0 F20 (ARC FEED CW, RADIUS I1,J0 AT 20 IPM)
N40 G01 X3.5
N45 G02 X3 Y0.5 R2 (ARC FEED CW, RADIUS 2)
```

```
N50 X1 Y1 R2 (ARC FEED CW, RADIUS 2)
N55 G00 Z0.1
N60 X2 Y1.5
N65 G01 Z-0.25
N70 G02 X2 Y1.5 I0.25 J-0.25 (FULL CIRCLE ARC FEED MOVE CW)
N75 G00 Z1
N80 X0 Y0
N85 M05
N90 M30
```

Dwell (G04)

For a dwell, program: `G04 P~`

- `P~` is the dwell time (measured in seconds)

Dwell keeps the axes unmoving for the period of time in seconds specified by the P number.

EXAMPLE
`G04 P4.2` (to wait 4.2 seconds)

Troubleshooting

It's an error if:

- The P number is negative

Set Offsets (G10)

Use the controls on the Offsets tab to set offsets. You can program offsets with the `G10` G-code command. Read the following sections for reference:

Set Tool Table (G10 L1)

To define an entry in the tool table, program: `G10 L1 P~ R~`

- `P~` is the tool number
- `R~` is the radius of tool

`G10 L1` sets the tool table for the P tool number to the values of the words. A valid `G10 L1` rewrites and reloads the tool table.

Troubleshooting

It's an error if:

- Cutter Compensation is on
- The P number is unspecified
- The P number is not a valid tool number from the tool table
- The P number is 0

Set Tool Table (G10 L10)

To change the tool table entry for tool P so that if the tool offset is reloaded with the machine in its current position and with the current G5x and G92 offsets active, program: `G10 L10 P~ R~`

- `P~` is the tool number
- `R~` is the radius of tool

The current coordinates for the given axes become the given values. The axes that are not specified in the `G10 L10` command are not changed. This could be useful with a probe move (`G38`).

Troubleshooting

It's an error if:

- Cutter Compensation is on
- The P number is unspecified
- The P number is not a valid tool number from the tool table
- The P number is 0

Set Tool Table (G10 L11)

`G10 L11` is just like `G10 L10`, except that instead of setting the entry according to the current offsets, it's set so that the current coordinates would become the given value if the new tool offset is reloaded and the machine is placed in the G59.3 coordinate system without any G92 offset active. This allows you to set the G59.3 coordinate system according to a fixed point on the machine, and then use that fixture to measure tools without regard to other currently active offsets.
Program: `G10 L11 P~ X~ Y~ Z~ R~`

- `P~` is the tool number
- `R~` is the radius of tool

Troubleshooting

It's an error if:

- Cutter Compensation is on
- The P number is unspecified
- The P number is not a valid tool number from the tool table
- The P number is 0

Set Coordinate System (G10 L20)

`G10 L20` is similar to `G10 L2`, except that instead of setting the offset/entry to the given value, it is set to a calculated value that makes the current coordinates become the given value.
Program: `G10 L20 P~ X~ Y~ Z~ A~`

- `P~` is the number of coordinate system to use (G54 = 1, G59.3 = 9)
- `X~` is the X-axis coordinate
- `Y~` is the Y-axis coordinate
- `Z~` is the Z-axis coordinate
- `A~` is the A-axis coordinate

Troubleshooting

It's an error if:

- The P number does not evaluate to an integer in the range 0 to 9
- An axis other than X, Y, Z, or A is programmed

Workpiece Probe (G15)

Probe vertically downwards towards the end of travel until the ohmic cap contacts the work piece.
Once the workpiece has been located, the Z coordinate of the current coordinate system is set to zero.
`G15` will trigger on either the ohmic cap making continuity with the workpiece or the physical touch switch in the torch lifter detecting upwards pressure on the torch from the material.

Pierce (G16)

Raise the torch to the pierce height set in the variable #<_PierceHeight>. The torch is turned on and the program pauses for the pierce delay value set by AutoFS or M209.
After the pierce delay, Torch Height Control (THC) is enabled and the program continues.

Plane Selection (G17, G18, G19)

To select the XY-plane as active, program: `G17`
To select the XZ-plane as active, program: `G18`
To select the YZ-plane as active, program: `G19`
The active plane determines how the tool path of an arc (`G02` or `G03`) or canned cycle (`G73`, `G81` through `G89`) is interpreted.

Length Units (G20 and G21)

To set length units to inches, program: `G20`
To set length units to millimeters, program: `G21`

Tip! Program either `G20` or `G21` near the beginning of a program, before any motion occurs. Avoid using either one anywhere else in the program. It's your responsibility to make sure that all numbers are appropriate for use with the current length units.

Return to Predefined Position (G28 and G28.1)

To make a rapid linear move from the current position to the absolute position of the values in parameters 5161-5166: `G28`

To make a rapid linear move to the G28.1 position by first going to the intermediate position specified by the X~, Y~, and Z~ words, program: `G28 X~ Y~ Z~`

Note: Any axis not specified won't move.

To store the current location of the tool in the G28.1 setting, program: `G28.1`
G28 uses the values stored in parameters 5161, 5162, and 5163 as the X, Y, and Z final points to move to. The parameter values are absolute machine coordinates in the native machine units of inches.
To store the current absolute position into parameters 5161-5163, program: `G28.1`

Troubleshooting

It's an error if:

- Cutter Compensation is turned on

Return to Predefined Position (G30 and G30.1)

G30 uses the values stored in parameters 5181 and 5183 as the X and Z final point to move to. The parameter values are absolute machine coordinates in the native machine units of inches.
To make a rapid traverse move from the current position to the absolute position of the values in parameters, program: `G30`
To make a rapid traverse move to the position specified by axes including any offsets, then make a rapid traverse move to the absolute position of the values in parameters 5181 and/or 5183, program: `G30 X~ Z~`

Note: Any axis not specified won't move.

To store the current absolute position into parameters 5181-5183, program: `G30.1`

Troubleshooting

It's an error if:

- Cutter Compensation is turned on

Automatically Measure Tool Lengths with an ETS (G37 and G37.1)

Use G37 and G37.1 with an Electronic Tool Setter (ETS) to enable automatic length measurement. For automated use, add a G37 command after an M6 tool change commands.
If you're using the ETS with a mill, the input port varies depending on your machine:

- **M Series Mills** Plug the ETS into the Accessory Input 2 port. You can still use the Accessory Input 1 port for other probes and accessories.
- **Older PCNC Mills** Plug the ETS into the (single) accessory input port.

Move to G37 Position Over ETS (G37.1)

To move to the G37 position (over the ETS), program: `G37.1`
To set the G37 position:

1. Jog the machine over the center of the ETS.
2. From the **Probe** tab, on the **ETS Setup** tab, select **Set G37 ETS Position**.
   The read-only DROs in the **ETS G37 Position Setup** group display the new position.

The G37 position is in G53 machine coordinate space. It defaults to (0, 0, 0), or the top left rear of machine travel (the same as the X-, Y-, and Z-axis reference position).
G37.1 supports X and Y tool offsets. If there are X or Y tool offsets present in the tool table (manually applied through a G10 L1 command), they offset the spindle position. This enables G37 for tools mounted in an auxiliary spindle installed on the spindle column.

Note: If G10 L1 is used to change the X or Y offset of the currently loaded tool, you must then apply the new offsets with a G43 command.

G37.1 performs as follows:

1. A rapid upward move to the Z clear position (which is always G53 Z = 0.0).
2. A rapid move in X and Y to the X and Y ETS coordinates.
3. A rapid downward move in Z to the Z ETS coordinate.

Note: The Z word saves time by rapidly moving closer to the ETS before the slower probing begins. You must use caution if you set this lower than G53 Z = 0.0. If you don't, there's a risk that a long tool could collide with the ETS and damage it and the tool.

Move and Measure Tool Length (G37)

To move and measure the tool length, program: `G37 H~ P~`

- `H~` saves the measured tool length to the H tool table entry instead of the current tool number's entry. You could use this to track tool wear between the two tool table entries, for example.

  Note: The newly measured tool length isn't applied, but it's stored in the tool table entry for tool number H.

- `P~` is positive or negative tolerance. It measures the tool length, but, instead of storing it in the tool table, compares it to the length in the tool table. If the difference exceeds the P tolerance, the G-code program stops.
  You could use this to detect broken or improperly inserted tools that are not fully seated in the spindle, for example.

G37 with no optional words moves to the G37 ETS position (through G37.1), probes the ETS, stores the new tool length in the tool table entry of the current tool, and applies the tool length offset.
G37 fails if the spindle nose hasn't been referenced to the ETS after a Z-axis reference. This sets a G53 coordinate at the ETS trigger point such that the measured tool length is the distance of the spindle nose to ETS reference. For more information, see the **ETS G37 Spindle Nose Reference** group on the **ETS Setup** tab.

So that tool length measurements have consistent results, G37 uses the fine probe feed rate of 2.5 in./min for the final ETS touch. G37 uses the rough probe feed rate for the first ETS touch.
G37 performs as follows:

1. Issues a G37.1 move to the ETS location.
2. A downward rough probe feed rate move until the tool triggers the ETS.
3. An upward retract move of 0.100 in. to back off the triggered ETS.
4. A downward slow ETS probe feed rate move until the tool triggers the ETS.
5. An upward retract move of 0.100 in. to back off the triggered ETS.
6. An upward rapid move to the G37 ETS Z position.

Straight Probe (G38.x)

`G38.2` probes toward the workpiece, stops on contact, and signals error if failure
`G38.3` probes toward the workpiece and stops on contact
`G38.4` probes away from the workpiece, stops on loss of contact, and signals error if failure
`G38.5` probes away from the workpiece and stops on loss of contact
`G38.6` moves away from the workpiece and ignores probe input
To perform a straight probe operation program: `G31 X~ Y~ Z~ A~`
Conventionally, the probe is tool #99. The rotational axis words are allowed, but it's better to omit them. If rotational axis words are used, the numbers must be the same as the current position numbers so that the rotational axes do not move. The tool in the spindle must be a probe.
In response to this command, the machine moves the controlled point (which should be at the end of the probe tip) in a straight line at the current feed rate toward the programmed point; if the probe trips, then the probe decelerates.
After successful probing, parameters 5061 to 5064 will be set to the coordinates of the location of the controlled point at the time the probe tripped (not where it stopped), or if it does not trip to the coordinates at the end of the move and a triplet giving X, Y, and Z at the trip is written to the triplet file.

Troubleshooting

It's an error if:

- The current point is less than 0.01 in. (0.254 mm) from the programmed point
- `G38` is used in inverse time feed rate mode
- Any rotational axis is commanded to move
- No X-, Y- or Z-axis word is used
  The linear axis words are optional, except that at least one of them must be used.
- Feed rate is zero
- The probe is already tripped

Use the Straight Probe Command

When you use the straight probe command, if the probe shank is kept nominally parallel to the Z-axis (i.e., any rotational axes are at zero) and the tool length offset for the probe is used, so that the controlled point is at the end of the tip of the probe, you may be able to find:

- Without additional knowledge about the probe, the parallelism of a face of a part to the XY-plane
- If the probe tip radius is known approximately, the parallelism of a face of a part to the YZ or XZ-plane
- If the shank of the probe is known to be well-aligned with the Z-axis and the probe tip radius is known approximately, the center of a circular hole

If the shank of the probe is known to be well-aligned with the Z-axis and the probe tip radius is known precisely, you can use the straight probe command for things like finding the diameter of a circular hole.

```
Example code:
o<probe_pocket> sub
(probe to find center of circular or rectangular pocket)
#<x_start> = #5420 (Current X Location)
#<y_start> = #5421 (Current Y Location)
#<x_max> = 1
#<x_min> = -1
#<y_max> = 1
#<y_min> = -1
#<feed_rate> = 30 (30 IPM)
F #<feed_rate>
G38.3 X #<x_max> (rough probe +X side of hole)
F [#<feed_rate>/30]
G38.5 X #<x_start> (finish probe)
#<x_plus>=#5061 (save results)
G00 X #<x_start> (return to start)
F #<feed_rate>
G38.3 X #<x_min> (probe -X side of hole)
F [#<feed_rate>/30]
G38.5 X #<x_start>
#<x_minus>=#5061 (save results)
G00 X #<x_start>
#<x_center> = [[#<x_plus>+#<x_minus>]/2]
G00 X #<x_center> (go to middle)
F #<feed_rate>
G38.3 Y #<y_max> (probe +Y side of hole)
F [#<feed_rate>/30]
G38.5 Y #<y_start>
#<y_plus>=#5062 (save results)
G00 Y #<y_start> (return to start)
```

```
F #<feed_rate>
G38.3 Y #<y_min> (probe -Y side of hole)
F [#<feed_rate>/30]
G38.5 Y #<y_start>
#<y_minus>=#5062 (save results)
G00 Y #<y_start>
#<y_center> = [[#<y_plus>+#<y_minus>]/2]
G00 Y #<y_center> (go to middle)
G10 L20 P1 X 0 Y 0 (set current location to zero)
F #<feed_rate> (restore original feed rate)
o<probe_pocket> endsub
M02
```

Cutter Compensation (G40, G41, G42)

To turn Cutter Compensation off, program: `G40`
It's okay to turn compensation off when it is already off.
It's an error if:

- A `G02`/`G03` arc move is programmed next after a `G40`
- The linear move after turning compensation off is less than twice the tool tip radius

To program Cutter Compensation to the left of the programmed tool path (as viewed looking down on the machine), program: `G41 D~`
To program Cutter Compensation to the right of the programmed tool path (as viewed looking down on the machine), program: `G42 D~`

- `D~` is the tool number associated with the diameter offset to be applied

The D word is optional — if there is no D word, the radius of the currently loaded tool is used. If no tool is loaded and no D word is given, a radius of 0 is used. If supplied, the D word is the tool number to use.
The lead in move must be at least as long as the tool radius. The lead in move can be a rapid move.
It's an error if:

- The D number is not a valid tool number, or it's 0
- Cutter Compensation is commanded to turn on when it is already on

Dynamic Cutter Compensation (G41.1 and G42.1)

To program dynamic Cutter Compensation to the left of the programmed tool path, program: `G41.1 D~`
To program dynamic Cutter Compensation to the right of the programmed tool path, program: `G42.1 D~`

- `D~` is the tip radius multiplied by two

`G41.1` and `G42.1` function the same as `G41` and `G42`, with the added scope of being able to ignore the tool table and to program the tool diameter.

Troubleshooting

It's an error if:

- Cutter Compensation is commanded to turn on when it is already on

Apply Tool Length Offset (G43)

To apply a tool length offset from a stored value in the tool table, program: `G43 H~`

- `H~` is the tool number associated with the length offset to be applied.

> **Note:** Generally speaking, the value of the H~ word should match the active tool number (T~ word).

It's okay to program using the same offset already in use, or to program without a tool length offset (if none is currently being used).

Troubleshooting

It's an error if:

- The H number is not an integer
- The H number is negative
- The H number is not a valid tool number

Engrave Sequential Serial Number (G47)

To engrave a serial number, either alone or added to the end of any text, program: `Z~ R~ X~ Y~ P~ Q~ D~`

- `Z~` is the depth of cut of the engraving
- `R~` is the retract height between character segments in the numbers
- `X~` is, if present, the starting X position, or the left side of the serial number
  If omitted, the current X position is assumed.
- `Y~` is, if present, the starting Y position, or the bottom side of the serial number
  If omitted, the current Y position is assumed.
- `P~` is, if present, the X extent (width) in current units (inches or millimeters) of the engraved number
- `Q~` is, if present, the Y extent (height) in current units (inches or millimeters) of the engraved number

- `D~` is, if present, the requested number of decimals of the engraved number
  If the requested D value exceeds the number of decimals in the serial number, the serial number will show leading zeros. If the requested D value is less than the number of decimals in the serial number, only the digits of the serial number will show.

  EXAMPLE
  A serial number of 10, where D = 4, engraves as 0010. A serial number of 9056, where D = 3, engraves as 9056.

Troubleshooting

It's an error if:

- Cutter Compensation is on
- The Z number is unspecified
- The R number is unspecified
- The Z number is greater than the R number
- The P number is too small (determined by the font used)
- The Q number is too small (determined by the font used)

Cancel Tool Length Compensation (G49)

To cancel tool length compensation, program: `G49`

Absolute Coordinates (G53)

For rapid linear motion to a point expressed in absolute coordinates, program: `G01 G53 X~ Y~ Z~` (or use with `G00` instead of `G01`)
All the axis words are optional, except that at least one must be used. The `G00` or `G01` is optional if it is in the current motion mode. `G53` isn't modal, and must be programmed on each line on which it is intended to be active. This produces coordinated linear motion to the programmed point. If `G01` is active, the speed of motion is the current feed rate (or slower if the machine won't go that fast). If `G00` is active, the speed of motion is the current traverse rate (or slower if the machine won't go that fast).

Troubleshooting

It's an error if:

- `G53` is used without `G00` or `G01` being active
- `G53` is used while cutter radius compensation is on

Select Work Offset Coordinate System (G54 to G54.1 P500)

You can save up to 500 work offsets in PathPilot. The naming structure varies based on the offset number.

- To select work offset 1, program: `G54` or `G54.1 P1`
- To select work offset 2, program: `G55` or `G54.1 P2`
- To select work offset 3, program: `G56` or `G54.1 P3`
- To select work offset 4, program: `G57` or `G54.1 P4`
- To select work offset 5, program: `G58` or `G54.1 P5`
- To select work offset 6, program: `G59` or `G54.1 P6`
- To select work offset 7, program: `G59.1` or `G54.1 P7`
- To select work offset 8, program: `G59.2` or `G54.1 P8`
- To select work offset 9, program: `G59.3` or `G54.1 P9`
- To select a work offset beyond the standard 9 (listed above), program: `G54.1 P###`, where `P###` is a parameter indicating the index of the work offset you want to use (work offset 10 through work offset 500).

  EXAMPLE
  To select the 124th work offset, program `G54.1 P124`.

For information, see "About Work Offsets" (page 80).

Troubleshooting

It's an error if:

- One of these G-codes is used while cutter radius compensation is on
- The X- and Z-axis work offset values are stored in parameters corresponding to the system in use (i.e., System 1 X=5221, Z=5223; System 2 X=5141, Z=5143; up to System 9 X= 5381, Z = 5383).

Set Exact Path Control Mode (G61)

To put the machining system into exact path mode, program: `G61`

Set Blended Path Control Mode (G64)

To attempt to maintain the defined feed velocity, program: `G64 P~ Q~`

- `P~` is, if present, the maximum acceptable tool path deviation to round corners to maintain speed. If P is omitted then the speed is maintained however far from the programmed path the tool cuts.
- `Q~` is, if present, the maximum deviation from collinearity that will collapse a series of linear G01 moves at the same feed rate into a single linear move.

It's okay to program for the mode that is already active.

Distance Mode (G90 and G91)

Interpretation of the operating system code can be in one of two distance modes: absolute or incremental.
To go into absolute distance mode, program: `G90`.

In absolute distance mode, axis numbers (X, Y, Z, A) usually represent positions in terms of the currently active coordinate system. Any exceptions to that rule are described explicitly in this section.
To go into incremental distance mode, program: `G91`.
In incremental distance mode, axis numbers (X, Y, Z, A) usually represent increments from the current values of the numbers. I and J numbers always represent increments, regardless of the distance mode setting. K numbers represent increments.

Arc Distance Mode (G90.1 and G91.1)

`G90.1` – Absolute distance mode for I and K offsets. When `G90.1` is in effect, I and K both must be specified with `G02`/`G03` for the XZ plane or it is an error.
`G91.1` – Incremental distance mode for I and K offsets. `G91.1` returns I and K to their default behavior.

Temporary Work Offsets (G92, G92.1, G92.2, and G92.3)

IMPORTANT! This is a legacy feature. Most modern programming methods don't use temporary work offsets.

To apply a temporary work offset, program: `G92 X~ Y~ Z~ A~`

- `X~` is the X-axis coordinate
- `Y~` is the Y-axis coordinate
- `Z~` is the Z-axis coordinate
- `A~` is the A-axis coordinate

`G92` reassigns the current controlled point to the coordinates specified by the axis words (X~, Y~, Z~, and/or A~). No motion takes place.
The axis words are optional, except that at least one must be used. If an axis word is not used for a given axis, the coordinate on that axis of the current point is not changed. Incremental distance mode (G91) has no effect on the action of `G92`.
When `G92` is executed, it is applied to the origins of all coordinate systems (`G54` through `G59.3`).

E X A M P L E
If the current controlled point is at X = 4, and there is currently no G92 offset active, and then `G92 X7` is programmed, this reassigns the current controlled point to X = 7 — effectively moving the origin of the active coordinate system -3 units in X. The origins of all inactive coordinate systems also move -3 units in X. This -3 is saved in parameter 5211.

`G92` offsets may be already be in effect when the `G92` is called. If this is the case, the offset is replaced with a new offset that makes the current point become the specified value.
It's an error if:

- All axis words are omitted

PathPilot stores the G92 offsets and reuses them on the next run of a program. To prevent this, you can program a G92.1 (to erase them), or program a G92.2 (to stop them being applied – they are still stored).
To reset axis offsets to zero and sets parameters 5211 - 5219 to zero, program: G92.1
To reset axis offsets to zero, program: G92.2
To set the axis offset to the values saved in parameters 5211 to 5219, program: G92.3

Feed Rate Mode (G93, G94, and G95)

To set the active feed rate mode to inverse time, program: G93
Inverse time is used to program simultaneous coordinated linear and coordinated rotary motion. In inverse time feed rate mode, an F word means the move should be completed in ***[1/F number]*** minutes.

> **EXAMPLE**
> If the F number is 2.0, the move should be completed in half a minute.

When the inverse time feed rate mode is active, an F word must appear on every line which has a G01, G02, or G03 motion, and an F word on a line that does not have G01, G02, or G03 is ignored. Being in inverse time feed rate mode does not affect G00 (rapid traverse) motions.
To set the active feed rate mode to units per minute mode, program: G94
In units per minute feed rate mode, an F word is interpreted to mean the controlled point should move at a certain number of inches per minute, or millimeters per minute, depending upon what length units are being used.
To set the active feed rate mode to units per revolution mode, program: G95
In units per revolution mode, an F word is interpreted to mean the controlled point should move a certain number of inches per revolution of the spindle, depending on what length units are being used. G95 is not suitable for threading, for threading use G33 or G76.

Troubleshooting

It's an error if:

- Inverse time feed rate mode is active and a line with G01, G02, or G03 (explicitly or implicitly) does not have an F word
- A new feed rate is not specified after switching to G94 or G95 canned cycle return level – G98 and G99

Spindle Control Mode (G96 and G97)

To set constant surface speed mode, program: G96 D~ S~

- D~ is the maximum spindle RPM.
  This word is optional.

- `S~` is the surface speed.

> **Note:** If `G20` is the active mode, the value is interpreted as feet per minute. If `G21` is the active mode, the value is interpreted as meters per minute

> **EXAMPLE**
> `G96 D2500 S250` (set constant surface speed with a maximum RPM of 2500, and a surface speed of 250).

When using `G96` (the most common mode of machine operation), X0 in the current coordinate system (including offsets and tool lengths) must be the spindle axis.
To set RPM mode, program: `G97`

Troubleshooting

It's an error if:

- S is not specified with `G96`
- A feed move is specified in `G96` mode while the spindle is not turning

PROGRAMMING M-CODE

Read the following sections for reference:

Supported M-Codes Reference

M-Code	Description
M00	Program stop
M01	Optional program stop
M02	Program end
M03, M04	Rotate spindle clockwise/counterclockwise
M05	Stop spindle rotation
M07, M08	Coolant on
M09	All coolant off
M30	Program end and rewind
M48	Enable speed and feed override
M49	Disable speed and feed override
M64	Activate output relays
M65	Deactivate output relays
M66	Wait on an input **Note:** `M64` through `M66` is only useful with a **USB M-Code I/O Interface Kit (PN 32616)**.

M-Code	Description
M98	Call subroutine
M99	Return from subroutine/repeat
M200, M203, M205, M207-M13	Plasma specific M-codes
M301, M302, M303	USB camera control

Program Stop and Program End (M00, M01, M02, and M30)

To stop a running program temporarily, regardless of the optional stop switch setting, program: `M00`
To stop a running program temporarily, but only if the optional stop switch is on, program: `M01`
It's okay to program `M00` and `M01` in MDI mode, but the effect probably won't be noticeable because normal behavior in MDI mode is to stop after each line of input.
If a program is stopped by an `M00`, `M01`, selecting Cycle Start restarts the program at the following line of the G-code program.
To end a program, program: `M02` or `M30`.
`M02` leaves the next line to be executed as the `M02` line. `M30` rewinds the G-code file. These commands can have the following effects:

- Axis offsets are set to zero (like `G92.2`) and origin offsets are set to the default (like `G54`)
- Selected plane is set to XY (like `G17`)
- Distance mode is set to absolute (like `G90`)
- Feed rate mode is set to units per minute mode (like `G94`)
- Feed and speed overrides are set to on (like `M48`)
- Cutter Compensation is turned off (like `G40`)
- The spindle is stopped (like `M05`)
- The current motion mode is set to G01 (like `G01`)
- Coolant is turned off (like `M09`)

No more lines of code in the file are executed after the `M02` or `M30` command is executed. Selecting Cycle Start starts the program back at the beginning of the file.

Display Information and Capture Images During an M00 or M01 Break

Display Information with Images

If the comment occurs on a line with `M00` or `M01`, and contains a file name with a .jpg or .png extension, PathPilot displays the image in the Tool Path display when the program reaches the M00 or M01 break.
To display an image during an M00 or M01 break:

1. Move an image file with a .jpg or .png extension to the PathPilot controller in one of the following locations:
   - In the same folder as the G-code program
   - In an images folder within the G-code program's folder
   - In an images folder within the home directory
2. Program an M00 or M01 break, and, using parentheses, type the full file name of the image (including its extension).

   EXAMPLE
   `M01 (Op1_Setup.jpg)` displays the image file on the Tool Path display.
3. The image file displays on the Tool Path display.

Display Information with Text

To display a message on the Tool Path display:

1. Program an M00 or M01 break, and, using parentheses, type a message that you'd like to display on the screen.

   EXAMPLE
   `M01 (Check coolant nozzles are pointed correctly)` displays *Check coolant nozzles are pointed correctly* across the bottom of the Tool Path display.
2. The message displays on the Tool Path display.

Capture Images with a USB Camera

In addition to displaying information like pictures or messages during an M01 break, you can also use a USB camera (if installed) to take a picture.
To use M01 to take pictures:

1. Add `M01 (op1_setup.jpg)` into your G-code program.
2. Run the G-code program.
3. When PathPilot executes the M01 it looks to see if the comment contains a file name.
   - If there isn't a file name: The comment is shown as instructional text across the tool path.
   - If there is a file name, but the file doesn't exist yet and the extension is .jpg, .png, or .jpeg: The USB cameras are initialized and shown in the tool path display.
4. Select the **Shutter** button to take the picture and create the op1_setup.jpg file.
   In future runs of the G-code program, **op1_setup.jpg** will display to the operator for instructional purposes on the workpiece.

For more information, see "Use a USB Camera" (page 62).

Spindle Control (M03, M04, and M05)

To start the spindle turning clockwise at the currently programmed speed, program: `M03`
To start the spindle turning counterclockwise at the currently programmed speed, program: `M04`
The speed is programmed by the S word.
To stop the spindle from turning, program: `M05`
It's okay to use `M03` or `M04` if the spindle speed is set to 0; if this is done, the spindle won't start turning. If later the spindle speed is set above 0, the spindle starts turning. It is permitted to use `M03` or `M04` when the spindle is already turning, or to use `M05` when the spindle is already stopped.

Tool Change (M06)

To execute a tool change sequence, program: `M06`
`M06` behaves differently depending on whether or not the machine is equipped with an Automatic Tool Changer (ATC):

- If you have an ATC:
  - If the requested tool (T number) is assigned to the carousel, `M06` initiates an automatic tool change.
  - If the tool is not assigned to the carousel, you're prompted to manually change the tool and select Cycle Start to confirm the tool change. This resumes the program.
- If you don't have an ATC:
  - `M06` commands the machine, stops the spindle, pauses program execution, and prompts operator to change tools by flashing Tool Change LED.
  - The program resumes after you select Cycle Start to confirm that the tool has been changed.

We recommend putting the `T~`, the `M06`, and the `G43 H~` on one line (block) of code.

EXAMPLE

```
N191 M06 T3 G43 H3
```

Coolant Control (M07, M08, and M09)

To turn coolant on, program: `M07`
To turn flood coolant on, program: `M08`
To turn all coolant off, program: `M09`
It's always okay to use any of these commands, regardless of what coolant is on or off.

Override Control (M48 and M49)

To enable the speed and feed override, program: `M48`
To disable both overrides, program: `M49`
It's okay to enable or disable the switches when they are already enabled or disabled.

Feed Override Control (M50)

To enable the feed rate override control, program: `M50 P1`

The P1 is optional.
To disable the feed rate control, program: `M50 P0`
When feed rate override control is disabled, the feed rate override slider has no influence, and all motion is executed at programmed feed rate (unless there is an adaptive feed rate override active).

Spindle Speed Override Control (M51)

To enable the spindle speed override control, program: `M51 P1`
The P1 is optional.
To disable the spindle speed override control, program: `M51 P0`
When spindle speed override control is disabled, the spindle speed override slider has no influence, and the spindle speed is equal to the value of the S word.

Set Current Tool Number (M61)

To change the current tool number while in MDI or manual mode, program: `M61 Q~`

- `Q~` is the tool number

Troubleshooting

It's an error if:

- `Q~` is not 0 or greater

Set Output State (M64 and M65)

Note: These commands are only useful when the machine is equipped with the USB M-Code I/O Interface Kit.

There are four output relays available on the USB I/O module.
To activate output relays (contact close), program: `M64`
To deactivate output relays (contact open), program: `M65`
There are four contacts, numbered from 0 to 3. The contact is specified by the P word.

EXAMPLE

- Activating the first relay: `M64 P0`
- Activating the second relay: `M64 P1`

The outputs are deactivated using M65 with the P word specifying the relay.

EXAMPLE

- Deactivating the second relay: `M65 P1`
- Deactivating the fourth relay: `M65 P3`

There is only one P word and one relay per line. Each relay command must be done on an individual line.
The following is legal:

```
M64 P0
M64 P2
M64 P3
```

The following is not legal:

```
M64 P023
M64 P0 P2 P3
```

Wait on Input (M66)

Note: This command is only useful when the machine is equipped with the USB M-Code I/O Interface Kit.

There are four digital inputs available on the USB I/O module.
`M66 P- | E- <L->`

- `P-` is the digital input number from 0 to 3.
- `L-` is the wait mode type:
  - Mode 0: IMMEDIATE – no waiting, returns immediately. The value of the input at that time is stored in parameter #5399.
  - Mode 1: RISE – waits for the selected input to perform a rise event.
  - Mode 2: FALL – waits for the selected input to perform a fall event.
  - Mode 3: HIGH – waits for the selected input to go to the HIGH state.
  - Mode 4: LOW – waits for the selected input to go to the LOW state.
- `Q-` is the timeout in seconds for waiting

The Q value is ignored if the L word is zero (IMMEDIATE). A Q value of zero is an error if the L word is non-zero.

Plasma Specific M-Codes

M-Code	Description	Set by M200	Parameters
M200	AutoFS - Apply the automatically generated feed, speed and cutting values for the material selected in the main PathPilot UI. AutoFS will set feedrate, THC voltage, and pierce delay.		
M203	Start Torch		
M205	Stop Torch and return to G30 Z Height		
M207	Set Pierce Height used when executing a G16 pierce. Set automatically when using M200 AutoFS.	Yes	P (Required): Pierce height in the current unit system (in. or mm.)
M208	Set Cut Height used when executing a G16 pierce. Set automatically when using M200 AutoFS.	Yes	P (Required): Cut height in the current unit system (in. or mm.)
M209	Set Pierce Delay used when executing a G16 pierce. Set automatically when using M200 AutoFS.	Yes	P (Required): Pierce delay in seconds
M210	Set THC voltage. Does not enable or disable THC.	Yes	P (Required): Desired THC Voltage
M211	Set plasma source cut current. If a Hypertherm adapter is not installed on the system, program run will be paused with an M01 break to allow the user to set the specified cut current.	Yes	P (Required): Desired Cut Current in Amps
M212	Set plasma source cut mode. If a Hypertherm adapter is not installed on the system, program run will be paused with an M01 break to allow the user to set the specified cut mode.	No	P (Required): Cut mode: Normal = 1 CPA = 2 Gouge = 3 Note: For the Hypertherm Powermax 45 Normal Mode and CPA Mode (Continuous Pilot Arc) are the same.
M213	Set plasma source air pressure. If a Hypertherm adapter is not installed on the system, program run will be paused with an M01 break to allow the user to set the specified air pressure.	No	P: Desired Cut Pressure in PSI Note: When the pressure is omitted or set to 0, the pressure control is set to Automatic.

PROGRAMMING INPUT CODES

Read the following sections for reference:

Feed Rate (F)

To set the feed rate, program: `F~`
Depending on the setting of the feed mode toggle, the rate may be in units-per-minute or units-per-rev of the spindle. The units are those defined by the G20/G21 mode. The feed rate may sometimes be overridden.

Spindle Speed (S)

To set the speed in revolutions per minute (rpm) of the spindle, program: `S~`
The spindle turns at the commanded speed when it has been programmed to start turning. It's okay to program an S word whether the spindle is turning or not. If the speed override switch is enabled and not set at 100 percent, the speed is different from what is programmed. It's okay to program S0, but the spindle does turn if that is done.

Troubleshooting

It's an error if:

- The S number is negative

Change Tool Number (T)

It's your responsibility to make sure that the machine is in a safe place for changing tools (for example, by using `G30`). This allows optimization of motion which can save time. You can provide a pause for manual intervention with `M00` or `M01` before the tool change.

Troubleshooting

It's an error if:

- A negative T number is used
- A T number larger than 1000 is used

ADVANCED PROGRAMMING

Parameter and expression programming language features are not used in common G-code application (hand coding), G-code created by PathPilot conversational programming, or the majority of third-party CAM-programming systems.
There are significant differences between controls in the way parameters work. Do not assume that code from another control works in the same way with the operating system. We don't recommend writing parametric G-code — i'ts difficult to debug, and difficult for another operator to understand. Modern CAM virtually eliminates the need for it.
Read the following sections for reference:

Parameters

Read the following sections for reference:

Parameters Reference

The RS274/NGC language supports parameters. Parameters are analogous to variables in other programming languages. PathPilot maintains an array of 10,320 numerical parameters. Many of them have specific uses. The parameters that are associated with fixtures are persistent over time. Other parameters are undefined when the operating system is loaded. The parameters are preserved when the interpreter is reset. Parameters 1 to 1000 can be used by the code of part-programs.
There are several types of parameters of different purpose and appearance. The only value type supported by parameters is floating-point; there are no string, Boolean or integer types in G-code like in other programming languages. However, logic expressions can be formulated with Boolean operators (AND, OR, XOR, and the comparison operators EQ, NE, GT, GE ,LT, LE), and the MOD, ROUND, FUP and FIX operators support integer arithmetic.

Parameter Syntax

There are three types of parameters, numbered, named local, and named global. The type of the parameter is defined by its syntax:

- Numbered - `#4711`
- Named local - `#<localvalue>`
- Named global - `#<_globalvalue>`

Parameter Scope

The scope of a parameter is either global or local within a subroutine. The scope of each parameter is inferred from its syntax. Subroutine parameters and named local parameters have local scope. Named global parameters and all numbered parameters starting from #31 are global in scope. RS274/NGC uses lexical scoping. In a subroutine, only the local parameters defined therein and any global parameters are visible. The local parameters of a calling procedure are not visible in a called procedure.

Behavior of Uninitialized Parameters

Uninitialized global parameters and unused subroutine parameters return the value zero when used in an expression. Uninitialized named parameters signal an error when used in an expression.

Parameter Mode

The mode of a parameter can either be read/write or read-only. Read/write parameters may be assigned values within an assignment statement. Read-only parameters cannot be assigned values. They may appear in expressions, but not on the left-hand side of an assignment statement.

Persistence and Volatility

Parameters can either be persistent or volatile. When the operating system is powered off, volatile parameters lose their values and are reset to zero. The values of persistent parameters are saved in a disc file and restored to their previous values when the operating system is powered on again. All parameters except numbered parameters in the current persistent range (5163 to 5390) are volatile.

Intended Use

Numbered parameters in the range #31-#5000, named global, and local parameters are available for general-purpose storage of floating-point values, like intermediate results, flags, etc., throughout program execution. They are read/write (can be assigned a value). Subroutine parameters, numbered parameters #1-#30, and system parameters are read-only and not available for general use. Subroutine parameters are used to hold the actual parameters passed to a subroutine. Numbered parameters in the range of #1-#30 are used to access offsets of coordinate systems. System parameters are used to determine the current running version and are read-only.

Numbered Parameters Reference

A numbered parameter is recognized by the pound symbol (`#`) followed by an integer between 1 and 5399. The parameter is referred to by this integer, and its value is whatever number is stored in the parameter. A value is stored in a parameter with the (`=`) operator.
Example: `#3 = 15` (set parameter 3 to 15)
A parameter setting does not take effect until after all parameter values on the same line have been found. For example, if parameter 3 has been previously set to 15 and the line: `#3=6 G01 X#3` is interpreted, a straight move to a point where X = 15 occurs before the value of parameter 3 is set to 6.
The # symbol takes precedence over other operations. For example, `#1+2` means the number found by adding 2 to the value of parameter 1, not the value found in parameter 3. Of course, `#[1+2]` does mean the value found in parameter 3.

The # character may be repeated; for example `##2` means the value of parameter whose index is the (integer) value of parameter 2. PathPilot maintains a number of read-only parameters. Only parameters for the relevant axes are maintained: (X Y Z A) for mill and (X Z) for lathe. The remaining parameters for unused axes are undefined.

Read-Only Parameters

- 1-30: Subroutine local parameters of call arguments. These parameters are local to the subroutine. For further information, see Programming with Subroutines later in this chapter
- 31-5000: G-code operator parameters. These parameters are global in G-code file
- 5061-5070: Result of G38.2 probe (X Y Z A B C U V W)
- 5161-5169: G28 home for (X Y Z A B C U V W)
- 5181-5189: G30 home for (X Y Z A B C U V W)
- 5210: 1 if G92 offsets are active, 0 if not
- 5211-5219: G92 offset (X Y Z A B C U V W)
- 5220: Current coordinate system number 1-9 for G54 - G59.3
- 5221-5230: Coordinate System 1, G54 (X Y Z A B C U V W R) – R denotes XY rotation angle around Z-axis
- 5241-5250: Coordinate System 2, G55 (X Y Z A B C U V W R)
- 5261-5270: Coordinate System 3, G56 (X Y Z A B C U V W R)
- 5281-5290: Coordinate System 4, G57 (X Y Z A B C U V W R)
- 5301-5310: Coordinate System 5, G58 (X Y Z A B C U V W R)
- 5321-5330: Coordinate System 6, G59 (X Y Z A B C U V W R)
- 5341-5350: Coordinate System 7, G59.1 (X Y Z A B C U V W R)
- 5361-5370: Coordinate System 8, G59.2 (X Y Z A B C U V W R)
- 5381-5390: Coordinate System 9, G59.3 (X Y Z A B C U V W R)
- 5399: Result of M66 – check or wait for input
- 5400: Current tool number
- 5401-5409: Tool offset (X Y Z A B C U V W)
- 5410: Current tool diameter
- 5411: Current tool front angle
- 5412: Current tool back angle
- 5420-5428: Current position including offsets in current program units (X Y Z A B C U V W)

Subroutine Parameters Reference

Subroutine parameters are specifically reserved for call arguments. By definition, these are parameters #1-#30 and are local to the subroutine.

Named Parameters Reference

Named parameters work like numbered parameters, but are easier to read and remember. All parameter names are converted to lowercase and have spaces and tabs removed. Named parameters must be enclosed with `< >` marks.

`#<named parameter here>` is a local named parameter. By default, a named parameter is local to the scope in which it is assigned.

You can't access a local parameter outside of its subroutine. This is so two subroutines can use the same parameter names without fear of one subroutine overwriting the values in another.

`#<_global named parameter here>` (i.e., name starting with an underscore) is a global named parameter. They are accessible from within called subroutines and may set values within subroutines that are accessible to the caller. As far as scope is concerned, they act just like regular numeric parameters. They are not made persistent by storage in a file.

The global parameters `_a`, `_b`, `_c`, . . . `_z` are reserved for special use. Do not use these parameters.

EXAMPLES

- `#<_endmill_dia> = 0.049` is a declaration of named global variable.
- `#<_endmill_rad> = [#<_endmill_dia>/2.0]` is a reference to previously declared global variable.
- `o100 call [0.0] [0.0] [#<_inside_cutout>-#<_endmill_dia>] [#<_Zcut>] [#<_feedrate>]` is mixed literal and named parameters.

Expressions

An expression is a set of characters starting with a left bracket (`[`) and ending with a right bracket (`]`). Located between the brackets are numbers, parameter values, binary operators, functions, and other expressions. An expression is evaluated to produce a number. An example of an expression is:

`[1 + acos[0] - [#3 ** [4.0/2]]]`

All expressions on a line are evaluated when the line is read and before anything on the line is executed.

Read the following sections for reference:

Binary Operators Reference

Binary operators only appear inside expressions. There are three types of binary operators: mathematical, logical, and relational.

There are four basic mathematical operations: addition (+), subtraction (-), multiplication (*), and division (/). In addition, the modulus operation (MOD) finds the remainder after division of one number by another number. The power operation (**) of raising the number on the left of the operation to the power on the right. There are three logical operations: non-exclusive or (OR), exclusive or (XOR), and logical and (AND).

The relational operators are equality (EQ), inequality (NE), strictly greater than (GT), greater than or equal to (GE), strictly less than (LT), and less than or equal to (LE).
Binary operators are divided into several groups according to their precedence as follows, from highest to lowest:

1. **
2. * / MOD
3. + -
4. EQ NE GT GE LT LE
5. AND OR XOR

If operations in different precedence groups are strung together, operations with a higher precedence are performed before operations with a lower precedence. If an expression contains more than one operation with the same precedence, the operation on the left is performed first.

EXAMPLE
`[2.0 / 3 * 1.5 - 5.5 / 11.0]` is equivalent to `[[[2.0 / 3] * 1.5] - [5.5 / 11.0]]`
which is equivalent to `[1.0 - 0.5]`
which is
`0.5`

The logical operations and modulus are to be performed on any real numbers, not just on integers. The number zero is equivalent to logical false, and any non-zero number is equivalent to logical true.

Functions Reference

The available functions are:

- `ATAN[Y]/[X]`: Four quadrant inverse tangent
- `ABS[arg]`: Absolute value
- `ACOS[arg]`: Inverse cosine
- `ASIN[arg]`: Inverse sine
- `COS[arg]`: Cosine
- `EXP[arg]`: e raised to the given power (ex)
- `FIX[arg]`: Round down to integer
- `FUP[arg]`: Round up to integer
- `ROUND[arg]`: Round to nearest integer
- `LN[arg]`: Base-e logarithm
- `SIN[arg]`: Sine
- `SQRT[arg]`: Square root

- `TAN[arg]`: Tangent
- `EXISTS[arg]`: Check named parameter

Subroutines

Subroutines are subprograms that are called from inside another program.
Read the following sections for reference:

Subroutines Reference

Subroutines are identified in a program by a unique subroutine label. The subroutine label is the letter `o` followed by an integer (with no sign) between 0 and 99999 written with no more than five digits (000009 is not permitted, for example) or a string of characters surrounded by `<>` symbols.
Examples of valid subroutine labels:

- `o123`
- `o99999`
- `o<my test code>`

Subroutine labels may be used in any order, but they must be unique in a program. Each subroutine label must be followed by a subroutine keyword. The subroutine keyword defines the action associated with the subroutine label. Valid subroutine keywords and their meanings are:

- `Sub`: Begin subroutine definition
- `Endsub:` End of subroutine definition
- `Call`: Call the subroutine
- `Do/while/endwhile`: Execute the subroutine while a condition is true
- `Repeat/endrepeat`: Execute the subroutine while a condition is true
- `If/elseif/else/endif`: Conditionally execute the subroutine
- `Break`: Break out of a while or if/elseif statement
- `Continue`: Skip remaining code and restart at top of while or repeat loop
- `Return`: Return a value

The `sub` and `endsub` keywords are used to define the beginning and end a subroutine. All lines of code between the `sub` and `endsub` keywords are considered to be part of the subroutine.

```
Example of sub, endsub, call:
o100 sub
G53 G00 X0 Y0 Z0 (rapid move to machine home)
```

```
o100 endsub
...
o100 call (call the subroutine here)
M02
```

Subroutines can either be defined in the program file or in a separate file. If the subroutine is defined in the same file as the main program that calls the subroutine, it must be defined before the call statement.
For example, this is valid:

```
o100 sub
G53 G00 X0 Y0 Z0 (rapid move to machine home)
o100 endsub
...
o100 call (call the subroutine here)
M02
```

But this is not:

```
o100 call (call the subroutine here)
M02
o100 sub
G53 G00 X0 Y0 Z0 (rapid move to machine home)
o100 endsub
...
```

A subroutine can be a separate file as long as:

- The file is named the same as your call.
- The file includes a `sub` and `endsub` in the file.
- The file is in the directory **/subroutines**.
- The file name only includes lowercase letters, numbers, dashes, and underscores.
- The file only contains a single subroutine definition.
- The file ends with the extension **.nc**.

Note: File names are lowercase letters only. `o<MyFile>` is converted to `o<myfile>` by the interpreter.

To execute a subroutine in a program, it must be called. To call a subroutine, program `o~ call` where ~ is the subroutine name. The subroutine name may be either a named file, a numbered file, or an expression that evaluates to a valid subroutine label.

- Expression example: `o[#101+2] call`
- Named file example: `o<myfile> call`
- Numbered file example: `o123 call`

`o~ call` takes up to 30 optional arguments, which are passed to the subroutine as #1, #2 , . . . , #N. Unused parameters from #N+1 to #30 have the same value as in the calling context.
Parameters #1-#30 are local to the subroutine. On return from the subroutine, the values of parameters #1 through #30 (regardless of the number of arguments) are restored to the values they had before the call.
The following calls a subroutine with three arguments: `o200 call [1] [2] [3]`
Because 1 2 3 is parsed as the number 123, the parameters must be enclosed in square brackets.
Subroutine bodies may be nested.

- Nested subroutines may only be called after they are defined.
- They may be called from other functions, and may call themselves recursively if it makes sense to do so.
- The maximum subroutine nesting level is 10.

Subroutines do not have return values, but they may change the value of parameters above #30 and those changes are visible to the calling G-code. Subroutines may also change the value of global named parameters.

Conditional Subroutines Reference

Subroutines can be conditionally executed using the if/endif or the if/else/elseif/endif keyword constructs.

if/endif

The if/endif conditional will execute a block of code following theif keyword only when the if argument evaluates to true.

```
If/endif example:
o100 sub
(notice that the if-endif block uses a different number)
o110 if [#2 GT 5]
(some code here)
o110 endif
(some more code here)
o100 endsub
```

if/elseif/else/endif

The if/elseif/else/endif conditional will execute the block of code following the if keyword when its argument evaluates to true. If the argument evaluates to false, then the code following each elseif is executed as long as the associated elseif argument evaluates to true. If no elseif keywords are present, or if all elseif arguments evaluate to false, than the code following the else keyword is executed.

```
If/elseif/endif example:
o102 if [#2 GT 5] (if parameter #2 is greater than 5 set F100)
F100
o102 elseif [#2 LT 2] (else if parameter #2 is less than 2 set F200)
F200
o102 else (else if parameter #2 is 2 through 5 set F150)
F150
```

```
o102 endif
```

Repeating Subroutines Reference

Subroutines can be repeated a finite number of times using the repeat/endrepeat keyword.

```
Repeat example:
(Mill 5 diagonal shapes)
G91 (Incremental mode)
o103 repeat [5]
... (insert milling code here)
G00 X1 Y1 (diagonal move to next position)
o103 endrepeat
G90 (Absolute mode)
```

Looping Subroutines Reference

Subroutines can be looped using the do/while or while/endwhile keyword constructs.

do/while

The do/while loop executes a block of code once and continues to execute the code block until the while argument evaluates to true.

```
Do/while loop example:
#1 = 0 (assign parameter #1 the value of 0)
o100 do
(debug, parameter 1 = #1)
o110 if [#1 EQ 2]
#1 = 3 (assign the value of 3 to parameter #1)
(msg, #1 has been assigned the value of 3)
o100 continue (skip to start of loop)
o110 endif
(some code here)
#1 = [#1 + 1] (increment the test counter)
o100 while [#1 LT 3]
M02
```

while/endwhile

The while/endwhile repeats a set of statements an indefinite number of times, as long as the while argument evaluates to true.

```
While/endwhile example:
(draw a sawtooth shape)
G00 X1 Y0 (move to start position)
#1 = 1 (assign parameter #1 the value of 0)
F25 (set a feed rate)
o101 while [#1 LT 10]
```

```
G01 X0
G01 Y[#1/10] X1
#1 = [#1+1] (increment the test counter)
o101 endwhile
M02 (end program)
```

The following statements cause an error message and abort the interpreter:

- A return or endsub not within a sub definition
- A label on repeat which is defined elsewhere
- A label on while which is defined elsewhere and not referring to a do
- A label on if defined elsewhere
- A undefined label on else or elseif
- A label on else, elseif or endif not pointing to a matching if
- A label on break or continue which does not point to a matching while or do
- A label on endrepeat or endwhile no referring to a corresponding while or repeat

PREVIOUS UPDATES

IN THIS SECTION, YOU'LL LEARN:

- About the enhancements and fixed issues in previous versions of PathPilot.

CONTENTS

RELEASE NOTES FOR PATHPILOT V2.9.6

May 2023

Enhancements

All

- We added support for Python-based diagnostic tests (.tsd files) to make some tech support steps quicker and easier. (PP-3698)

Mills

- We fixed an issue on MX mills where the Encoder self-test would, in rare cases, report false-positive failures during spindle acceleration. (PP-3917)

Plasma

- We added cut charts for Powermax 65, 85, and 105 SYNC plasma sources to the Auto FS material picker. (PP-3932)

Fixed Issues

Plasma

- We fixed an issue with set-start-line behavior on 1300PL plasmas, where arc voltage and feed settings were sometimes not updated at the start line. (PP-3476)

xsTECH

- We fixed an issue where PathPilot was prevented from functioning with some xsTECH routers. (PP-3938)

RELEASE NOTES FOR PATHPILOT V2.9.5

May 2023

Fixed Issues

- We fixed an issue introduced in PathPilot v2.9.4 where PathPilot was prevented from functioning with xsTECH routers. (PP-3914)
- We fixed an issue where some xsTECH routers E-stopped due to joint following errors on PathPilot 2.9.x. (PP-3667)

RELEASE NOTES FOR PATHPILOT V2.9.4

April 2023

Enhancements

- We added support for 7i92T machine interface boards, which are used with new PCNC 440 mills and 15L Slant-PRO lathes. (PP-3885)

RELEASE NOTES FOR PATHPILOT V2.9.3

February 2023

Enhancements

Lathes

- We restored the original default rapid speed on the 15L Slant-PRO lathe to 60 ipm because of infrequent, but ongoing, issues for some customers. (PP-3760)

Fixed Issues

All

- We fixed an issue where X/Y DROs were always in inches in rotated coordinate system. (PP-3774)
- Dropbox support now requires updating the Firefox browser on your controller. To easily update the browser, you can now use a script from the **Examples** folder in the **File** tab. (PP-3796)
  To learn how to use the technical support script, click here to read our how-to article.

RELEASE NOTES FOR PATHPILOT V2.9.2

October 2022

Enhancements

All

- Previously, loading a file larger than 2 MB disabled syntax highlighting and required you to manually enable it again. Now, syntax highlighting is restored to its previous setting when a file smaller than 2 MB is loaded. (PP-3675)

Fixed Issues

All

We fixed issues where:

- The previous tool live plot didn't clear when set start line was used with **Show Only Current Tool**. (PP-3595)
- The **Show Only Current Tool** setting sometimes reset after editing and reloading a G-code file. (PP-3596)
- In rare situations, a lot of continuous user interaction (like loading programs or conversational code/edits) resulted in low system resources and poor performance. (PP-3630)

Lathes

We fixed issues where:

- Conversational edits sometimes asked you to save changes again even though they had already been saved. (PP-3417)
- The minimum "X" extent of the tool path displayed incorrectly in the preview. (PP-3658)

Router

- We fixed an issue where, during occasional automated tool changes, a “bad number format” error appeared on the **Status** tab. (PP-3685)

RELEASE NOTES FOR PATHPILOT V2.9.1

August 2022

Fixed Issues

Mills

- We fixed an issue in PathPilot v2.9.0 where the **VFD Running** LED on the **Status** tab didn't illuminate on PCNC 440, PCNC 770, and PCNC 1100 mills. (PP-3632)

Lathes

- We fixed an issue in PathPilot v2.9.0 where, in G21 mode, some position and offset DROs displayed in the wrong units. (PP-3631)

RELEASE NOTES FOR PATHPILOT V2.9.0

July 2022

Enhancements

We added:

- An example spindle warmup program in the **Examples** folder in the **File** tab. Whenever the machine has been idle overnight or longer, running the spindle warmup program provides proper lubrication to the spindle bearings. For the 24R specifically, there's also a new spindle break-in program. (PP-3376)
- Driver support for new versions of USB WiFi adapters. (PP-3377)
- Detection of some error conditions in mills and lathes. A running program or MDI command now stops with an error if:
  - A motion requiring specific spindle rpm doesn't see the spindle reach the target speed within 20 seconds.
  - A motion requiring spindle synchronization doesn't see an index pulse within 10 seconds. (PP-3386)
- Support for the automatic tool changer (ATC) for 24R routers. (PP-3394)
- Support for the ZA6 robot. (PP-3395)
- Dropbox support now requires updating the Chrome browser on your controller. To easily update the browser, you can now use a script from the **Examples** folder in the **File** tab. (PP-3593)
  To learn how to use the technical support script, click here to read our how-to article.

Fixed Issues

All

We fixed issues where:

- In very rare situations, PathPilot unexpectedly shutdown while editing a conversational G-code program, though it could occur in other situations. (PP-3318)
- It wasn't possible to download files from PathPilot HUB if they were contained in a folder that didn't already exist on the local controller. (PP-3333)
- In some cases, the G71 path preview was shown incorrectly after a conversational edit session (program execution was not affected). (PP-3341)
- Introduced in PathPilot v2.8.2, a full-turn helical path could, in some cases, have too small of a pitch (due to an extra full turn). (PP-3354)
- An error could occur when downloading files from PathPilot HUB that are inside of folders. (PP-3390)
- Tool descriptions with colons were truncated during programs where manual tool changes were required. (PP-3431)

Mills

We fixed issues where:

- The maximum Z feed rate for conversational tapping was too small in G21, preventing you from saving an otherwise valid tapping program. (PP-3301)
- In some cases, G30 in conversational code failed with a limit error when the G30.1 Z position was at maximum Z height. (PP-3313)
- If G33.1 was called via MDI as the very first command after selecting **Ref** or **Reset**, it failed with an error, and caused the next MDI command to pause indefinitely before execution (which required you to select **Stop**/**Reset**). (PP-3328)
- A G-code error after a tool change in a running program could leave the previous tool's offset active. (PP-3357)
- On MX mills, calling a tool change reset the G28 position. (PP-3371)

Lathes

We fixed issues where:

- G71 conversational profile gouged if the profile crossed X0 with cutter compensation active. (PP-3277)
- On 8L lathes, warnings about incorrect tool offset X values for front/rear tool posts were reversed. (PP-3312)
- Ring-jogging in Y would eventually cause a soft emergency stop due to a joint following error on the non-existent Y axis. (PP-3334)
- Introduced in PathPilot v2.8.3, the G30 X position was incorrectly scaled. (PP-3344)
- Some rear tool post only tools were visible in the tool selection screen as possible tool types for the 8L lathe, which only uses front tool post tools. (PP-3369)
- In some cases, invalid tool numbers greater than 9999 were accepted in G-code, leading to spurious python exceptions in log files during conversational operations. (PP-3370)
- An explicit tool change to T0 didn't completely clear the tool offset if a wear offset was previously active (subsequent tool changes/offsets were unaffected). (PP-3378)
- In very rare situations, attempting to reference the X- and Z-axis at the same time on the 15L Slant-PRO lathe could result in one of the axes moving in the opposite direction it normally travels to reference. (PP-3383)
- In conversational profiling, front tool post tools weren't correctly configured and displayed. In some cases, the default front/back angles were incorrect, requiring an override to get proper profiling behavior. Also, we fixed the display of front tool post tool shapes in conversational profiling so that they match the lathe's orientation. (PP-3400)
- Recommended feed/speed/thread passes weren't populated for some materials on the 8L lathe. (PP-3427)

Plasma

We fixed issues where:

- In some cases, using set start line failed with a soft limit error if the start line was after a G15. (PP-3458)
- Unexpected AXIS_ABORT error messages sometimes appeared on the **Status** tab after a change in G20/G21 units during MDI actions or while running programs. (PP-3435)

RELEASE NOTES FOR PATHPILOT V2.8.3

September 2021

Enhancements

Mills and Routers

- For some setups (depending on the machine configuration and the model of the tool setter), we reduced the Z traverse feed rate in the G37 / G37.1 tool touchoff sequence. This reduces the risk of damage due to an incorrectly set G37 Z position. (PP-3291)

Mills

- We improved collision avoidance when using long tools in Automatic Tool Changers (ATCs). Previously, after a tool change, the final Z position was set to whichever was higher between G30 Z and tool change Z. Now, the Z position is saved at the beginning of a tool change, and later restored after the tool change is complete (if it's above the tool change height position). (PP-3286)

  Note: To toggle this behavior on or off, use the following commands in the MDI Line DRO field:
  `ADMIN ATC M6_EPILOG_RETRACT ON`
  `ADMIN ATC M6_EPILOG_RETRACT OFF`

xsTECH

- The noise made by the axis motors on xsTECH routers is now quieter. (PP-3256)

Fixed Issues

All

- We added updated graphics drivers for improved OpenGL performance on newer PathPilot controllers (Rev C). (PP-3279)
- We fixed an issue with very small G02 and G03 arcs and cutter compensation where, in some situations, the compensated arc used the wrong number of full turns. (PP-3278)
- The jog mode button's LED now reflects the current keyboard jogging mode. Previously, using the jog shuttle or console pendant could have side effects such that selecting the jog mode button appeared to have no effect in toggling the LED. (PP-3300)

Mills

- We fixed an issue with M98 subroutines/looping where the loop count was intermittently wrong after a program abort. (PP-1969)

- G30 now accepts an intermediate axis position in incremental (G91) coordinates. This makes it easier to retract individual axes. Previously, intermediate Z moves had to be to an absolute position. (PP-3273)

  EXAMPLE

  ```
  G91 G30 Z0
  G90
  ```

  Code breakdown: Move incrementally zero units in Z (no motion), then rapid in Z to the position specified in parameter 5183 (no other axes move).

Lathes

- We fixed an issue where, in some situations while using a console, cycling through jog axes from X to Z on the pendant reported a false error message that the maximum jog speed for the (non-existent) Y-axis was missing. (PP-3288)

  Note: This issue didn't affect keyboard jogging on the console.

RELEASE NOTES FOR PATHPILOT V2.8.0

June 2021

Enhancements

All

- We improved the behavior of the set start line function in two ways (PP-3090):
  1. If the correct tool is already in the spindle, skip the retract to tool change height.
  2. Retract to the `G30` position rather than `G53 Z0`. This saves time and allows for better control of retract position, like on lathes with a turret.
- We added a new, read-only parameter `# <_tool_offset_number>` to get the currently applied tool offset number at the interpreter read-ahead location for use in G-code programs. (PP-3249)

Mills

- We improved collision avoidance when using long tools in Automatic Tool Changers (ATCs). Now, after a tool change involving the ATC, the final Z position is set to whichever is higher between G30 Z and tool change Z. (PP-3254)

Plasma

- We added mild steel cut charts for the Powermax 85 plasma source. (PP-2941)
- We added a parametric conversational shape library. It includes things like common shapes, flanges, and brackets — along with conversational parameters — for quick and easy cutting. (PP-3193)

Fixed Issues

All

- We fixed issues where:
  - While using the pendant on the PathPilot operator console, the jog speed wasn't limited before the axes were referenced. (PP-3037)
  - The View Options tab (in the G-Code window) wouldn't refresh when a file modification was detected. (PP-3102)
  - Some G-code program errors would appear to leave the Tool Path display blank. (PP-3114)
  - When exiting a `G37` move, the machine moved to the initial Z position before moving to the Z clear position. (PP-3132)
  - While using the pendant on the PathPilot operator console, the axis motion stuttered when jogging at high speeds. (PP-3188)
  - After unplugging and re-connecting the touch screen monitor's USB cable, the calibration settings were removed. (PP-3245)

- Attempting to assign a value to a read-only parameter correctly raised an error, but failed to clear that error. This caused any future MDI or program runs to fail, and required a restart of PathPilot to resume normal operation. (PP-3252)

Mills

- We added rapid jogging support for rotary axes. (PP-2901)
- We fixed an issue where, if an ATC had low air pressure, selecting ATC FWD or ATC REV caused the PathPilot interface to freeze. This required a restart of PathPilot to resume normal operation. (PP-3253)

Lathes

- We fixed an issue where, after a spindle stop was commanded, a user command to unclamp the collet was allowed before the spindle had fully stopped. (PP-3219)

Plasma

- We fixed an issue where some AutoFS presets for mild steel used the incorrect amperage. (PP-3111)

RELEASE NOTES FOR PATHPILOT V2.7.4

May 2021

Enhancements

Mills

- On PCNC 440 Automatic Tool Changers (ATCs) that use the older, off-board stepper driver, we reduced the acceleration and deceleration parameters by 33% to dampen jerk during backlash compensation moves. (PP-3226)

Fixed Issues

All

- We fixed an issue where, while a program was running, you couldn't select the **Expand View** button in the **Notebook** section of the **Main** tab. This issue was introduced in PathPilot v2.7.3. (PP-3227)
- Previously, you could change the set-start-line options while a program was running, but the changes didn't take effect because they're cleared after program stop or completion. This has been resolved and now changes to set-start-line options aren't allowed while a program is running.

Mills

- We fixed issues where:
  - In some situations, using non-Tormach USB devices or hubs prevented the ATC firmware from updating. (PP-3221)
  - On PCNC and M series mills, some normal usages of the R / P word in G74 were incorrectly reported as an error. This issue was introduced in PathPilot v 2.5. (PP-3235)

RELEASE NOTES FOR PATHPILOT V2.7.3

April 2021

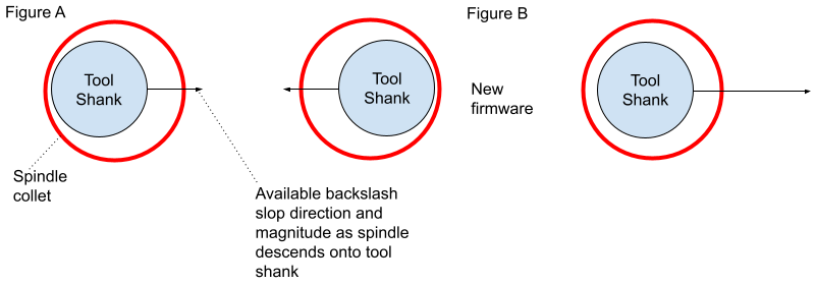
Enhancements

All

- We improved the diagnostic information that displays when the controller fails to power on because of a depleted CMOS battery. (PP-3173)
- While referencing an axis, you can now use the **Space Bar** key on the keyboard to use the feed hold function and stop axis motion. (PP-3178)
- We added driver support for newer Realtek 8152/8153 Ethernet chips, which provides wider support for USB to Ethernet adapters. (PP-3220)

Mills

- We updated the ATC firmware to make it easier to align the tool shank or BT30 pull stud with the spindle collet. Previously, adjusting the tray position required incremental rotations in both directions (see **Figure A**). Now, to compensate for any backlash in the tray position, the ATC always approaches the final target position from the same direction (see **Figure B**). For example, during forward moves, the ATC slightly overshoots the target position, and then reverses back into position.



> **IMPORTANT!** After you update the ATC firmware, we recommend that you check the adjustment of the tray position for a tool change. Before making any adjustments, keep the spindle collet above the tool shank or BT30 pull stud — when you're adjusting the target position using the **--** and **++** buttons on the **ATC** tab, the backlash compensation move still occurs.

If needed, the MDI command `ADMIN ATC BACKLASH OFF` disables the backlash compensation.

Fixed Issues

All

- We fixed issues where:
  - In conversational serial number engraving operations, G47 was placed after M30 when coolant was disabled. (PP-3156)
  - In conversational engraving operations, 'GG5x' was added if the text field was empty and the serial number box was checked. (PP-3162)
  - The preview in the **Tool Path** display wasn't updated if a program loaded with an error, which could be misleading. (PP-3163)
  - In simulation/PathPilot HUB configurations, limit switches were prevented from being disabled. This sometimes led to an unrecoverable state, like if the machine was jogged into a limit switch before all axes were referenced. (PP-3166)
  - A previously selected start line would sometimes be reset after running a long MDI command. (PP-3167)
  - During MDI command execution where keyboard jog commands would cause status messages and, in some cases, abort the MDI command in progress. (PP-3174)
  - Setting the G30 position to an axis G53 zero position sometimes caused G30 commands to generate limit errors that showed very small distance violations. (PP-3203)
  - The G30 position for the A-axis in G21 (metric) mode wouldn't position the axis correctly. (PP-3204)
  - Two display issues occurred with A-axis values in G21 mode:
    - A-axis work offsets were incorrectly scaled up by 25.4 in the **Work Offset** table (but correctly applied during program execution).
    - **Feed Rate** DRO field showed an incorrect value when only the A-axis was moving. (PP-3199)
- We improved the usability of the **Feed Hold** button (and its shortcut on the keyboard, the **Space Bar** key) in several conditions:
  - The **Cycle Start** button's LED now flashes when the machine is in feed hold during an MDI command
  - Feed hold causes a controlled stop of axis motion during jogging and referencing
  - We also fixed an issue where feed hold could be enabled during jogging, which prevented programs from running until explicitly cleared by a stop or selecting the **Reset** button. (PP-3178)

Mills

- We fixed issues where:
  - PCNC 440 configurations didn't support more than one USB I/O module. (PP-3157)
  - In certain situations, the conversational pocket operation could produce problematic tool paths. (PP-3172)

- When referencing PCNC 440 mills, the **Stop** and **Reset** buttons stopped the currently referencing axis, but didn't stop any other queued reference commands. (PP-3179)
- Set start line could, in some cases, report a false error when used with a program containing G37 with a non-zero P word. (PP-3195)
- Conversational Edit could silently fail to save a file if one or more operations were posted with Tool 0. (PP-3196)

Lathes

- We fixed a long-standing issue where, if a set start line was chosen after a G96 command, the maximum spindle speed (D word) would sometimes be ignored until the next explicit G96 line.
  We also improved lathe surface feed and RPM DRO fields so that they show the correct RPM-limited values during manual operations and jogging. Previously, the value in the RPM DRO field showed a value larger than the actual spindle speed for small X diameters. (PP-3165)
- We fixed an issue where the **Collet Clamped** button wasn't useable during M0/M01 stops. (PP-3217)

RELEASE NOTES FOR PATHPILOT V2.7.2

January 2021

Enhancements

All

- We improved the usability of toolpath extents display: the extents no longer include G28/G30 motions, or lead-in moves for set-start-line. (PP-3105)

Mills

- We optimized/reduced time spent waiting for the spindle to be at speed at the start of a G84 non-rigid tapped hole. (PP-3150)

Fixed Issues

All

- We fixed issues where:
  - The A-axis display selection wasn't enabled on start up, requiring you to explicitly disable and re-enable it. (PP-3119)
  - In some rare situations, or when setting up A-axis probing for the first time, the stock/axis center settings didn't match the setting shown in the interface. Also, A-axis probing now starts from the current A-axis position, rather than always rotating to A0, allowing probing of stock center at arbitrary A-axis angles. (PP-3128)
  - In PathPilot HUB, some older PCNC 1100 mill configurations were unable to switch to RapidTurn mode because of an INI file error. (PP-3133)
  - The alarm level of G-code errors detected at load was too low, making these errors less visible to users. This issue was introduced in v2.7.0/v2.7.1. (PP-3153)
- We slightly reduced the acceleration for microARC, and we reduced the G0 rapid speed from 20 rpm to 18 rpm. (PP-3144)

Mills

- We fixed issues where:
  - On Series 3 and older 1100 mills, program load/redraw reported a cryptic error message (program execution was unaffected). (PP-3120)
  - Mill conversational DXF failed to draw a preview for a positive Z End position. (PP-3125)

Lathes

- We fixed issues where:
  - For some metric thread pitches in conversational lathe tapping, the exported value didn't exactly match the desired value (for example, 1.4925 mm instead of 1.5 mm). (PP-3147)
  - The high belt spindle speed on 8L lathes was off by factor of 2. (PP-3151)

RELEASE NOTES FOR PATHPILOT V2.7.1

November 2020

Fixed Issues

Mills

- We resolved an issue, introduced in v2.7.0, which had the potential for certain error conditions using abort syntax to be ignored. Specifically, probing and some ATC errors were affected. (PP-3115)

Lathes

- We resolved an issue, introduced in v2.7.0, which had the potential for certain error conditions using abort syntax to be ignored. Specifically, automatic collet closer 'not clamped' detection before spindle start may not have functioned correctly. (PP-3115)

Routers

- We resolved an issue, introduced in v2.7.0, which had the potential for certain error conditions using abort syntax to be ignored. Specifically, the spindle chiller alarm wouldn't be reacted to properly. (PP-3115)

RELEASE NOTES FOR PATHPILOT V2.7.0

November 2020

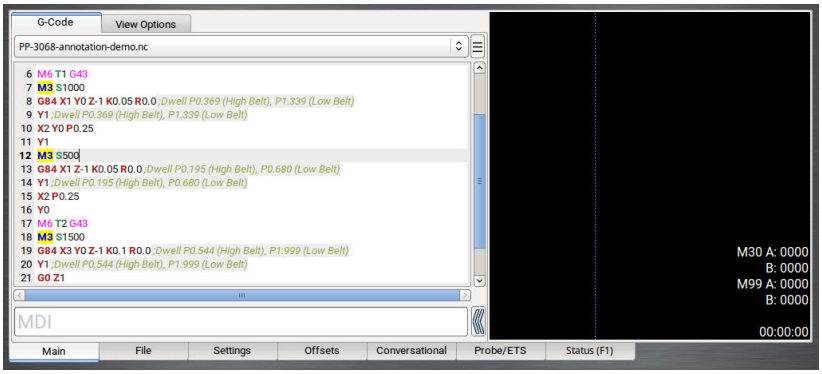
Enhancements

All

- For more consistent behavior with other computers, you can now abort a button click after pressing the mouse by moving off of the button before releasing the mouse. (PP-1852)
- G30 motions in a part program are now shown in the toolpath preview. (PP-2845)
- We added support for date and time stamping in G-code-based log outputs with a new #<_epochtime> parameter (which converts the current UTC epoch time in seconds). (PP-3006)
- We improved error detection in `G84` / `G33.1` rigid tapping that were previously reported as warnings. (PP-3029)
- We improved error checking in rigid tapping to allow shallower tap cycles. (PP-3043)
- G-code errors from program load now highlight the line causing the error in the G-code source view, and scroll to the error location. (PP-3045)
- We updated the caution message displayed on the **ETS Setup** tab to provide more detail on how to set the G37 position. (PP-3053)

Mills

- For G84 non-rigid tapping, an annotation is now shown in the source view with the calculated dwell time for the given spindle speed/belt position. This makes fine-tuning the dwell time easier. (PP-3049)



- We improved G-code colors in the Source View and editor, including previously missing G-codes, some comment syntax, M3/M30 ambiguity, and others. (PP-3013)
- The ATC set tool change position now logs old and new positions. (PP-3055)
- When setting the tool change position, the ATC now displays the acceptable range in metric if G21 is enabled. (PP-3056)
- We improved the usability of A-axis center probing with an option to find stock center without rotating the A-axis, and a visually distinct probe model in the live plot. (PP-3060)
- We added 0.5 seconds dwell between ATC drawbar activation and spindle raise. (PP-3093)

Fixed Issues

All

- We fixed issues where:
  - PathPilot opened a file with no extension in the editor rather than loading it as a program. (PP-3023)
  - Using the feature to find the A-axis center and set the origin (on the **Probing Rect / Circ** tab) didn't store the correct offset values. (PP-3059)
  - **Find A-axis center** (on the **Probing Rect / Circ** tab) sometimes returned incorrect values when used in G21 mode.
  - In some cases, the tool path live plot wasn't cleared if the tool view was filtered. (PP-3063)
  - If the controller was rebooted without coming out of reset, the current tool in spindle would not be restored the next time the **Reset** button was selected. (PP-3097)
  - Several buttons in the mill ATC and lathe tool touchoff tabs didn't respond after pressing the **Enter** key, needing an explicit mouse click instead. (PP-3098)

Mills

- We added support for A-axis positioning moves (part of traverse move to initial position) with G84 (PP-3014), and A-axis support for G30. (PP-3061)
- We fixed issues where:
  - G84 soft tapping cycles had an extra delay at the bottom of the hole, which could compress the tension-compression head on retract. (PP-3018)
  - Axis couldn't be referenced without the enclosure door closed if door switch/lock was installed. This behavior should have been limited to only CE configurations. (PP-3019)
  - In rare cases, G84 soft tapping generated a cryptic error message in long-running programs. (PP-3028)
  - A status message that displayed when the tool change position was out of range didn't accurately display the range. (PP-3057)
  - The A-axis display checkbox in the **View Options** menu could be unchecked while a program was running. The display would be unchanged, but the checkbox no longer matched the display state. (PP-3065)
  - Conversational circular pockets very close to the tool diameter would do a straight plunge instead of a helical ramp. (PP-3050)
  - Conversational rectangular pockets could be cut oversize in X if the tool radius was very close to the corner radius. (PP-3052)
  - Conversational rectangular/circular pockets could cause a plunge entry to extend past the -X edge of the pocket. (PP-3067)
- We updated the Fusion 360™ post-processor (provided by David Loomes), which has the following changes:
  - Added support for partial circular bore and partial circular boss probe operations.
  - Implemented size and position tolerance checking for the stock (set the **wrong size** or **Out of position** check boxes on the **Actions** tab for each probe operation in Fusion).
  - Option to print probe results to the status screen (set the **Print Results** check box on the **Actions** tab).
  - Implemented the **Measure tool** and **Tool break control** options in **Manual NC** CAM operations.
  - Implemented **Tool break control** when selected on the **Post Processor** tab in the Fusion 360 tool library.
  - Provided multiple options to control retraction at various stages during program execution. This replaces the old **Use G30** and **Use G28** options to give finer control. Default settings give the same results as Use G30 = Yes and Use G28 = No.

Tip! For more information on the changes made to the Fusion post-processor, we recommend watching David Loomes' YouTube video: https://youtu.be/qrJZmE5qgzI

Lathe

- Previously, conversational programs posted `G30 Z#5422`. This meant that G30 didn't move in X, even if the **G30 Move in Z Only** checkbox was cleared (on the **Settings** tab). Now, conversational programs post only `G30`. (PP-2930)
- We fixed an issue where the **Spindle RPM** slider in CSS mode wasn't uniform, reaching maximum spindle speed before the 100% position. (PP-3016)
- We added error detection to prevent G33.1 rigid tapping in G96 mode and protect against tap breakage. (PP-3042)
- We fixed an issue where set start line on the lathe didn't correctly track tool offsets. (PP-3087)

Plasma

- PathPilot now immediately recognizes a failed plasma arc and pauses machine motion until the torch can be relit. (PP-2999).

RELEASE NOTES FOR PATHPILOT V2.6.0

August 2020

Enhancements

All

- We added support for the 1300PL machine. (PP-1377)
- We added PathPilot HUB file transfer capabilities to the **File** tab. (PP-2613)

Mills

- We extended and improved the A-axis display in the **Tool Path** display, which makes it easier to understand the effective tool path relative to the workpiece. (PP-2862)
- On the **Settings** tab, there's now a **4th Axis Rotary** sub-tab. We consolidated the 4th axis rotary settings and configuration on this new tab. (PP-2894)
- G64 Naive CAM Detection now supports rotary axes, which smooths out 4th axis Fusion 360 programs. (PP-2907)

Fixed Issues

All

- We fixed an issue where there was a missing unit conversion in incremental polar coordinates. Previously, incremental polar angles were interpreted as radians instead of degrees. (PP-2905)
- We fixed a rare case with G95, where the first feed move immediately following a G33 move would use the wrong feed rate. This didn't occur if G33 was followed by other motions like G0, tool change, etc. (PP-2923)
- When you typed G or A in the **MDI Line** DRO field (for valid G-code commands), the auto-complete dropdown displayed. Now, you must type two characters for the auto-complete dropdown to display. (PP-2928)
- We fixed an issue with conversational edit where an edited coolant setting would be ignored if an operation was edited again before saving. (PP-2965)
- We fixed an issue with non-Tormach (unsupported) touch screens, where a calibration error occurred when the controller powered on. (PP-2980)
- We fixed an issue where the background grid for the front/side G-code view in the **Tool Path** display was incorrectly displaced when large tool offsets were active. (PP-2986)
- The time zone for Virtual PathPilot controllers on HUB is now set according to the user's HUB account preferences. (PP-3004)
- When you add a .txt file to PathPilot, you can now either load the file as G-code or view and edit it with the text editor. Previously, .txt files **only** opened in the text editor. (PP-3005)

Mills

- We fixed an issue, introduced in PathPilot v2.3.6, that prevented 4th axis homing from working. (PP-2955)
- We fixed an error in a Fusion 360 probing routine (f360_probing-y-channel.ngc) that incorrectly used x instead of y. (PP-2956)
- We fixed an error where, when you tried to set a reference height with a tool other than Tool 0 on the **Probe Setup** tab, the green LED came on after selecting **Set Reference Height** (which falsely indicated that the position was set). (PP-2991)
- We fixed an issue where, in the tool table, either missing tools (between tool 1 and tool 256) or duplicate tools (between tool 257 and tool 1000) weren't correctly displayed. (PP-2997)

Lathe

- We fixed an issue where, after re-editing an existing file that used conversational threading, the operation didn't update to reflect the number in the **Passes** DRO field. (PP-2925)
- We fixed an issue that prevented conversational lathe threading from using the maximum allowed number of threading passes (98). (PP-2964)


TITLE: externalThread-Mill
Aluminum
-any-
WORK OFFSET: G59
TOOL: 2
SPINDLE RPM: 555
FEEDRATE: 100.0
Z FEEDRATE: 100.0
Z CLEAR: 0.6000
(Flutes: N/A Chip load per tooth: N/A)
(SFM: 0 – MRR: N/A)
POST TO FILE
APPEND TO FILE
Face
Profile
Pocket
Drill / Tap
Thread Mill
Engrave
DXF (Mill)
EXTERNAL
INTERNAL
DEPTH
X (FROM DRILL TABLE)
Y (FROM DRILL TABLE)
NUMBER OF PASSES:
THREADS/IN:
3.0
Z START:
3.0000
Z CLEAR
PITCH (IN):
0.3333
2.0000
Z END:
LEFT
RIGHT
Main
File
Settings
Offsets
Conversational
Probe
Status
CYCLE START
SINGLE BLOCK
M01 BREAK
FEEDHOLD
STOP
COOLANT
RESET
100%
FEED 100%
100%
RPM 100%
100%
MAXVEL 100%
POS IN G55
DTG
ZERO X
5.1931
0.0000
REF X
ZERO Y
0.6512
0.0000
REF Y
ZERO Z
0.0000
0.0000
REF Z
ZERO A
0.0000
0.0000
REF A
STATUS:
G55 G90 G20 G80 G40 G94 G97 G99 G91.1
STEP:
REV
SPINDLE RANGE
T
0
05:06:02