Mach4 Galil Plugin Operations Guide

10/26/22

Contents

Figure Reference

1 Introduction

The purpose of this manual is to teach the basic operation and functionality of the Mach4 Galil Plugin software. This manual is based on the Galil Plugin product build 229. The appearance of the interface may vary depending on the build number.

1.1 Before You Begin

Any machine tool is potentially dangerous. Computer controlled machines are potentially more dangerous than manual ones because, for example, a computer is quite prepared to rotate an 8" unbalanced cast iron four-jaw chuck at 3000 rpm, to plunge a panel-fielding router cutter deep into a piece of oak, or to mill away the clamps holding your work to the table. Because we do not know the details of your machine or local conditions we can accept no responsibility for the performance of any machine or any damage or injury caused by its use. It is your responsibility to ensure that you understand the implications of what you design and build and to comply with any legislation and codes of practice applicable to your country or state. If you are in any doubt, be sure to seek guidance from a professionally qualified expert rather than risk injury to yourself or to others.

1.2 What is Galil for Mach4?

The Galil plugin is an interface between Mach4 and one or more Galil controllers. The plugin translates movement, trajectory profiles, and output signals from Mach4 (G code) into a format compatible with the Galil controllers and receives feedback from the Galil for the various Mach4 functions and input signals.

1.3 Supported Galil Controllers

Controller	Interface	Control Mode
DMC-40x0	Ethernet	Contour/Linear
DMC-41x3	Ethernet	Contour/Linear
DMC-18x6	PCI	Contour/Linear
DMC-20x0	Ethernet	Linear
DMC-21x0	Ethernet	Linear
DMC-21x2/3	Ethernet	Linear
DMC-18x0	PCI	Linear
DMC-18x2	PCI	Linear
RIO-471xx	Ethernet	I/O only
RIO-4720x	Ethernet	I/O only
RIO-47300	Ethernet	I/O only

Of the two control modes, Contour mode is the preferred method of control . Linear interpolation mode will work with the older controllers but it is highly recommended to use a newer controller to make use of the Contour mode.

How many Galil Axes are needed?

You will want one Galil axis for every motor that the system will have. Plus one extra to optionally control a spindle motor. For instance, if you have a 3 axis mill with 1 motor per axis and a variable spindle, then a 4 axis controller will be sufficient. However, if a 4th "A" axis is be needed, then a 5 axis Galil controller will be needed.

1.4 Preparing the Galil Controller

The Galil plugin does not provide an interface to fully configure the Galil controller. For instance, servo tuning must be configured on the Galil controller and "burned" into the NVRAM storage. The suggested method of getting a machine operational is to first get the machine moving using only the Galil. While Galil programs can co-exist with the Galil plugin for Mach4, they must not interfere with the movement of any Mach controlled axes. Therefore it is highly recommended to not have any Galil programs running on the controller during initial setup.

Initial tuning and testing should be done with the Galil provided software. The suggested software from Galil is called "Galil Suites". The free version is quite useful and has a terminal and signal viewer. For a modest price, the pro version includes a scope and tuning routines to aid in tuning servo systems .

Special Considerations

1) If the use of a probe for part location, inspection, or tool offsets, then the general purpose inputs that control the position latches should be left available for this purpose. See the Galil AL and RL commands for your Galil model for an explanation of these inputs.

Special Considerations (Cont)

2) The default Galil firmware requires that a latch input be configured for every Galil axis for which a position is to be captured. This means that a mill with X, Y , and Z axis control will need to wire three of the latch inputs to the probe in parallel to capture each axes positions simultaneously.

3) Most 2 axis tables home to the top right hand corner and define that position as machine 0. The polarity of the LSCOM does have an effect on which way an axis will home. Please refer to the Galil operators manual for more information on this.

4) It is suggested that the target machine be equipped with limit switches on both ends of each axis and with a separate home switch for each axis. The G code program is then loaded and executed by Mach4.

The standard Mach4 screen is designed to be easy to navigate for first time users. Figure 2-1 shows the layout of the main page. This is the screen that will be loaded when Mach4Gui.exe is launched and the Mach4Mill profile is selected. Each part will be discussed in more detail in separate sections.

2 Installing the Galil Plugin

Installing the Galil plugin can be accomplished in two ways: via a plugin package or manually placing the files in the appropriate directory.

2.1 Galil Plugin Package Installation

1. Start the Mach4GUI and copy the Mach4Mill profile by selecting the "Copy Profile" button.

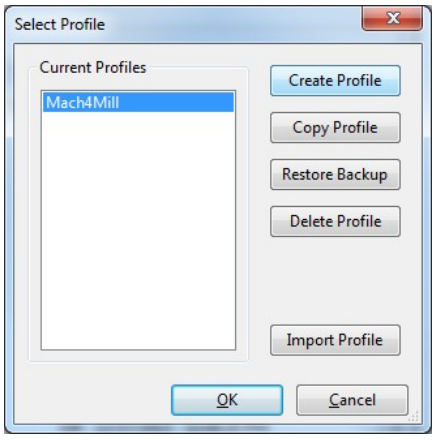


Figure 1 - Mach Profile Selection Dialog

2. Enter the name of the new profile. E.g. "Mach4Galil" and choose a screen set.

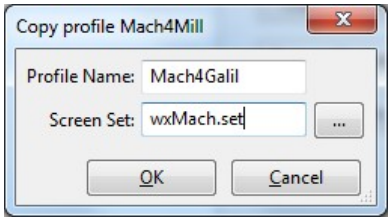


Figure 2 - Copy Profile Dialog

3. Select the new profile and click "OK".

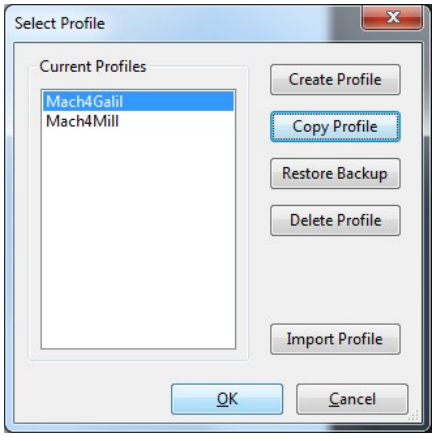


Figure 3 - Selection of the new profile

1. When Mach starts, navigate to and click the Configure->Mach... menu item. Then choose the Plugins Tab.

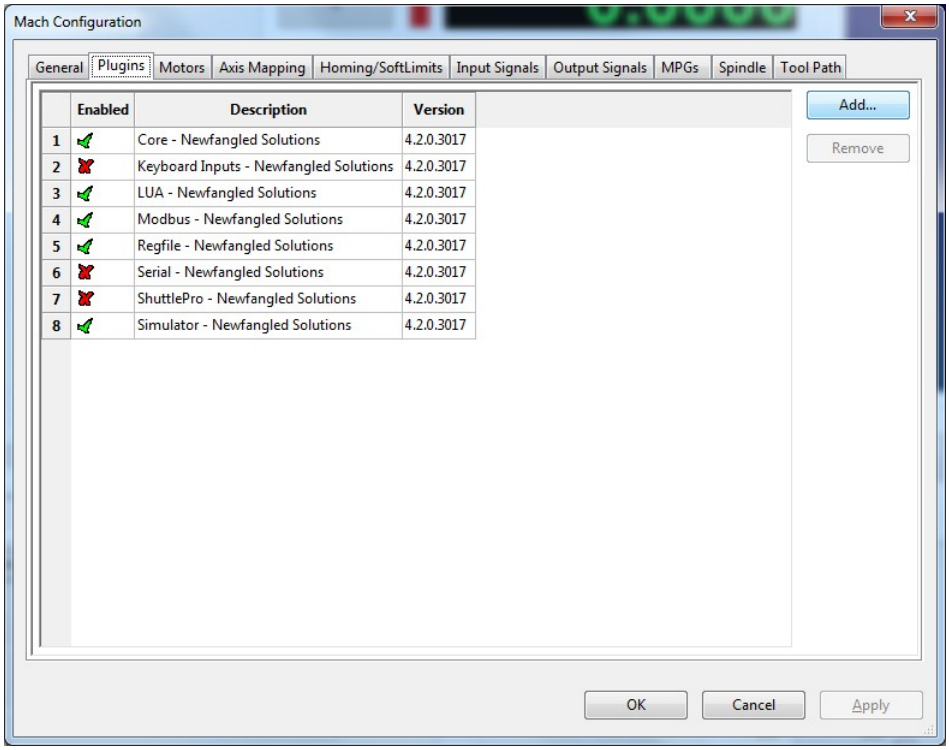


Figure 4 - Mach configuration Plugins Tab

2. Click on the "Add" button and navigate the file system to the Galil plugin package file. Then click "Open".

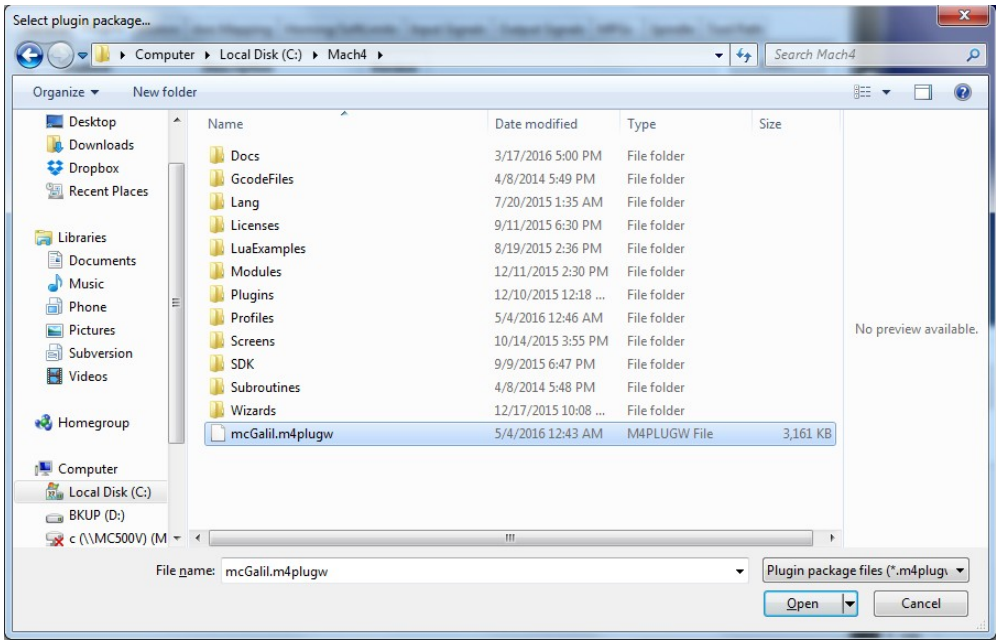


Figure 5 - Plugin package selection dialog

4. Disable the Sim plugin, enable the Galil plugin, and Click "OK". A message will notify you that a restart is necessary. When Mach is restarted, the Plugin will be fully installed. The next step is to configure the plugin.

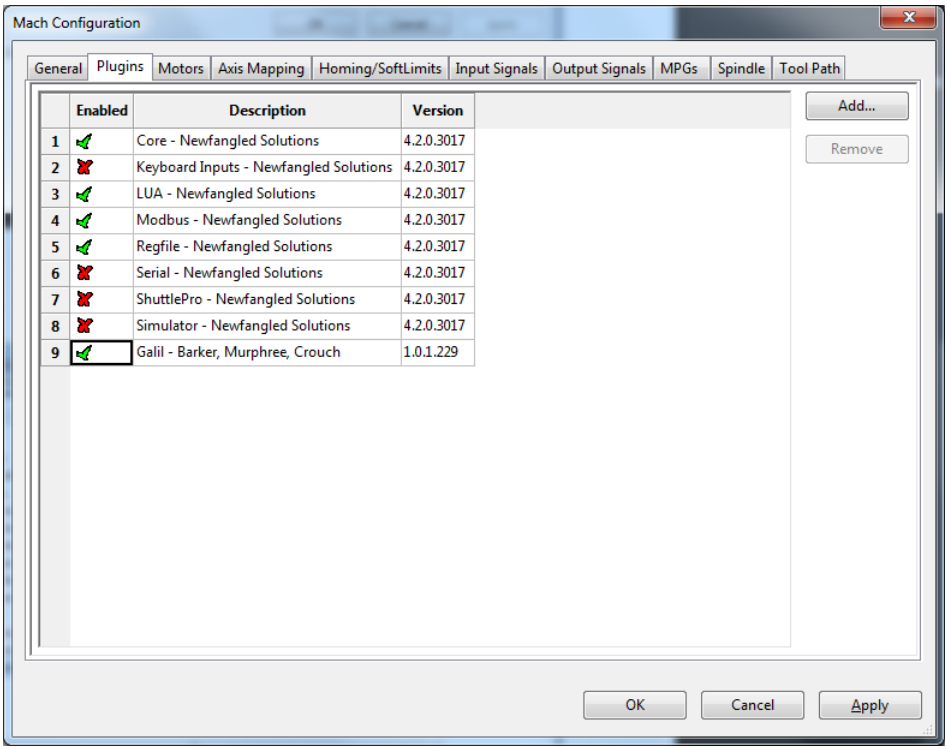
5. 

Figure 6 - Enabling the Galil plugin

2.2 Manual Galil Plugin Installation

The plugin packages are really just zip files with different extensions. *. m4plugw for Windows, *.m4plugl for Linux, and *.m4plugm for Macintosh. To access the zip file archive, all one has to do is change the extension to .zip and use any archive viewer/extractor.

Sometimes a developer may send you just a zip file if you are working with them to resolve a problem. In this case, there is no reason to change the extension.

1. Shutdown Mach4 if it is running.
2. Locate the Mach4 installation directory. There will be a Plugins directory under it.

3. Back up any existing Galil plugin files so that you can get back to that version if needed. The Two main files will be the Galil plugin itself (mcGalil.m4pw for Windows, mcGalil.m4pl for Linux, and mcGalil.m4pm for Macintosh) and mcGalil.sig.
4. Extract the contents of the developer provided zip file to the Plugins directory.
5. Restart Mach4.

3 Configuring the Galil Plugin

1. Once the plugin has been installed, exit Mach and restart. Navigate to the Config->Plugins menu path. Select the Galil plugin configuration menu item. The left list will show all discovered Galil controllers. If no controllers are in the list or the one you need is not there, then there is a communication problem that needs to be resolved on the Galil or the host PC.

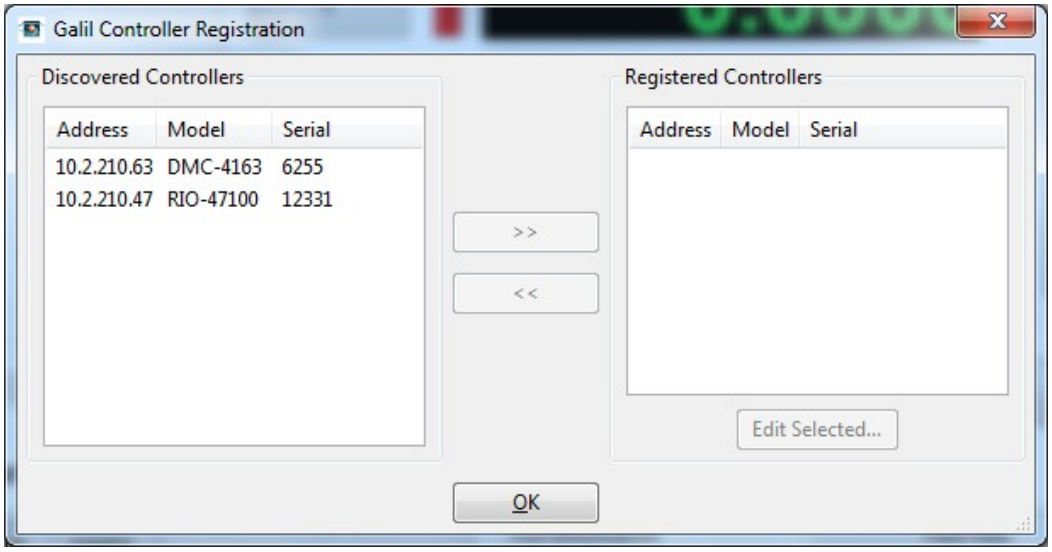


Figure 7 - Galil controller registration

2. Choose the Galil controller that is desired for motion control first and move it to the registered controllers column via the ">>" button. Then choose any other controllers and move them to the registered controllers as desired. NOTE: Only the first controller will control motion!

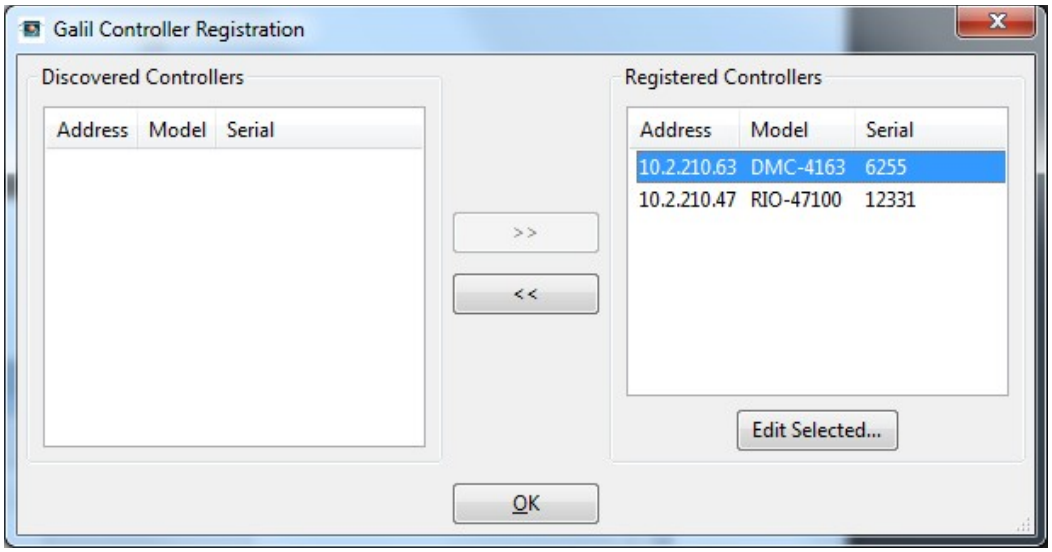


Figure 8 - Selecting the controller to edit

3.1 The General Tab

Select the desired controller to configure and click the "Edit Selected..." button. The following dialog will appear. Depending on the controller model, some options will not be enabled or not even shown. Below is a dialog that pertains to an Ethernet controller. PCI controllers will lack the Ethernet Parameters section.

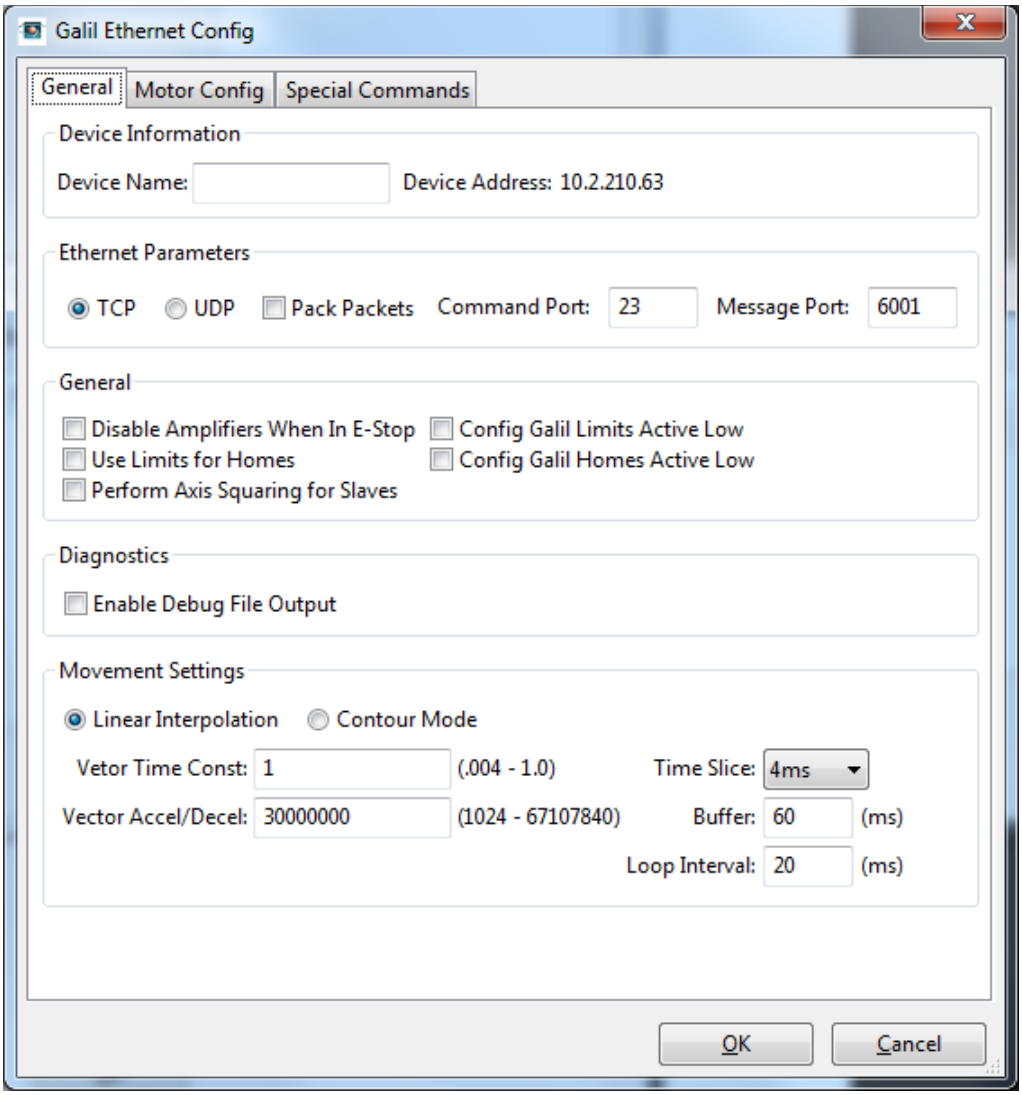


Figure 9 - Plugin configurations dialog General tab

Device Information:

Device information is common to all controller models with the exception of the Device Address (not editable) which will differ between Ethernet and PCI based controllers.

Device Name

Enter a Device Name for the Galil controller. The Device Name will be how Mach references this controller. Device names are typically all one word and end with a number. e.g. "Galil0".

Ethernet Parameters:

Ethernet Parameters control how the plugin talks to an Ethernet Controller. The available communication options are TCP and UDP. TCP is more reliable and should be used on a congested network. However, TCP has proven to be slower and may not provide the performance that is desired.

It is suggested to connect the Galil straight to the PC controller with a dedicated Ethernet card and use UDP.

Packed Packets

Packed Packets is an option that really should be used for UDP connections only. If provides the fastest means of communications to the host PC controller.

Command And Message Ports

The Command Port and Message Port are the TCP/UDP ports that the Galil will use for communication. They can be left at the default settings and should only be changed if there is a conflict.

General:

Disable Amplifiers in E-Stop

Disable Amplifiers in E-Stop can be selected if the Galil ENx signals should be used to shut down the servo drive amplifiers in an E-Stop condition. This is considered purely redundant because an E-Stop circuit should remove power to the amplifiers in any event.

Use Limits for Homes

Use Limits for Homes can be selected if the machine does not provide home switches for the axes. However, it is good practice to use home switches and it is highly recommended!

Perform Axis Squaring for Slaves

Performa Axis Squaring for Slaves can be selected if two or more Galil axes are used to move a gantry style machine axis. The homing routine will ensure that the gantry is not canted after a homing operation. It is not magic. Home switches on each gantry motor are required!

Config Galil Limits Active Low

Config Galil Limits Active Low is used to make the plugin agree with the wiring of LSCOM (See the Galil Operators Manual for more information. This is one of the most challenging aspects of setting up a Galil so a thorough understanding of how the Galil operates is essential.

Config Galil Homes Active Low

Config Galil Homes Active Low is similar to the above. Depending on if the limit switch is NO or NC, this setting may need to be changed. Due to the nature of the opto-isolated inputs on the Galil, a NC switch may appear in the data record as being held low.

Diagnostics:

Enable Debug File Output

Sometimes it may be useful to see what is actually being sent to the Galil controller. Selecting Enable Debug File Output will create a GalilDebug-<serial number>.txt file in the Plugins directory which may be viewed to determine an issue or sent to a developer for diagnostics. It is recommended to run with the option disabled for performance reasons.

Movement Settings:

Depending on the controller model, there may be two movement control options. Linear Interpolation or Contour Mode.

Linear Interpolation:

Linear Interpolation is the only option for older controllers like a DMC-12x3 or any Optima series controllers. There are two parameters that affect the Linear Interpolation mode. Vector Time Constant and Vector Accel/Decel The defaults will suffice for most installations.

Vector Time Constant

Vector Time Constant can be used to smooth the motion if necessary at the expense of increasing the amount of time the operation would normally take.

Vector Accel/Decel

Vector Accel/Decel can be lowered to allow the Galil trajectory planner to have more input on the movement, if needed.

Both of these settings are considered tune-able settings and the user should try various values in each to obtain desired movement smoothness. Again, the defaults work well for the majority of installations.

Contour Mode:

Contour Mode does not have any parameters.

Time Slice

Time slice determines the width of time that every movement segment takes. The smaller the time slice, the more granular the movement segments will be. The list is populated with the range of time slices that the chosen controller model is capable of. However, just because the controller may be capable of 1 millisecond time slices doesn't mean the system as a whole is capable of maintaining that amount of granularity. The PC and communications method may also pose issues with the lower time slices. That is why it is a settable parameter so the user can find what works on their machine.

Generally, a 4 millisecond time slice is sufficient for a metal cutting machine as segment faceting will not be apparent at metal cutting speeds. If cutting material at feed speeds approaching 300 inches per minute (7500 mm per minute), a 2 millisecond time slice may be warranted.

Buffer:

The Buffer is the amount of time in the motion buffer that should reside on the controller. The default of 60 milliseconds works fine for most machines. But this may need tuning depending on the host PC and communication conditions. This setting directly affects the time it takes feed hold to take effect. The larger the buffer, the longer it takes for feed hold to come to a stop. Generally, anything less than 100 milliseconds seems instantaneous to humans.

Loop Interval:

The loop interval specifies the interval in milliseconds that the plugin will try to fill the motion buffer up to the buffer setting above. The default value (20ms) is 3 times less than the buffer time. This gives the host PC three shots at maintaining the buffer without running the buffer dry. It is also the frequency at which Mach is provided feedback. Some host PC and Galil controller combinations can manage a 10 millisecond loop interval, but it is cutting it close. If you are getting movement stutters, increase the loop interval 5 milliseconds at a time while also increasing the buffer time by a multiple of 3 or more of the loop interval.

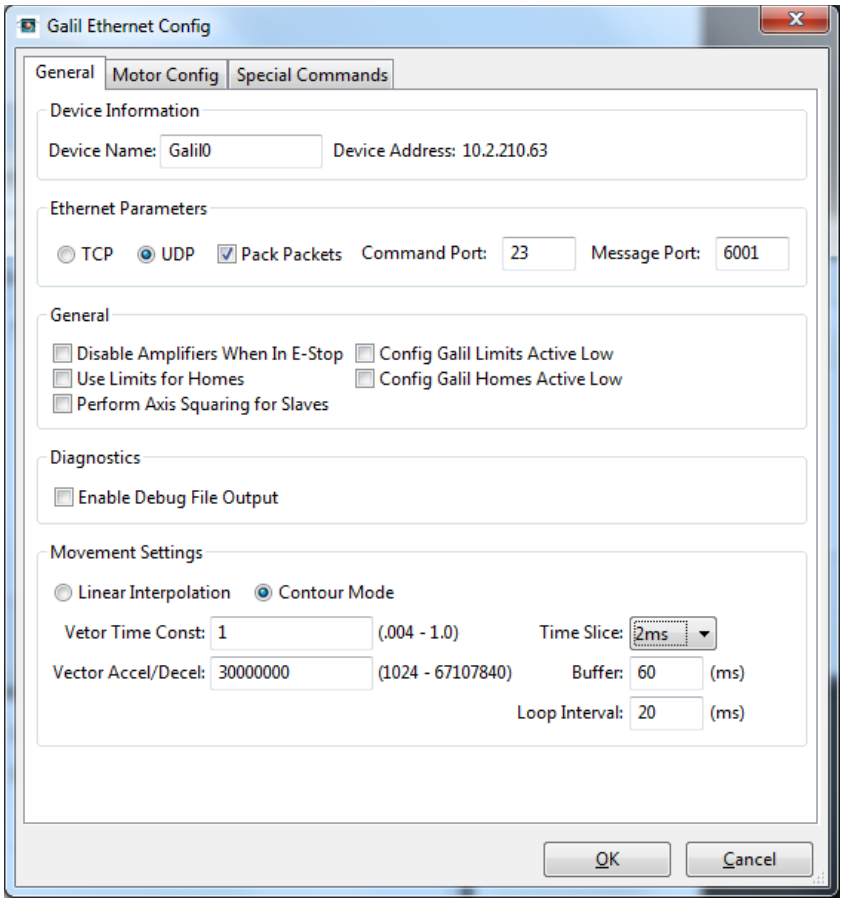


Figure 10 (Completed General Tab)

3.2 Motor Config Tab

The Motor Config Tab is where the Mach motor to Galil axis mapping is performed. There are also parameters that describe the motor types and feedback methods and feedback preference.

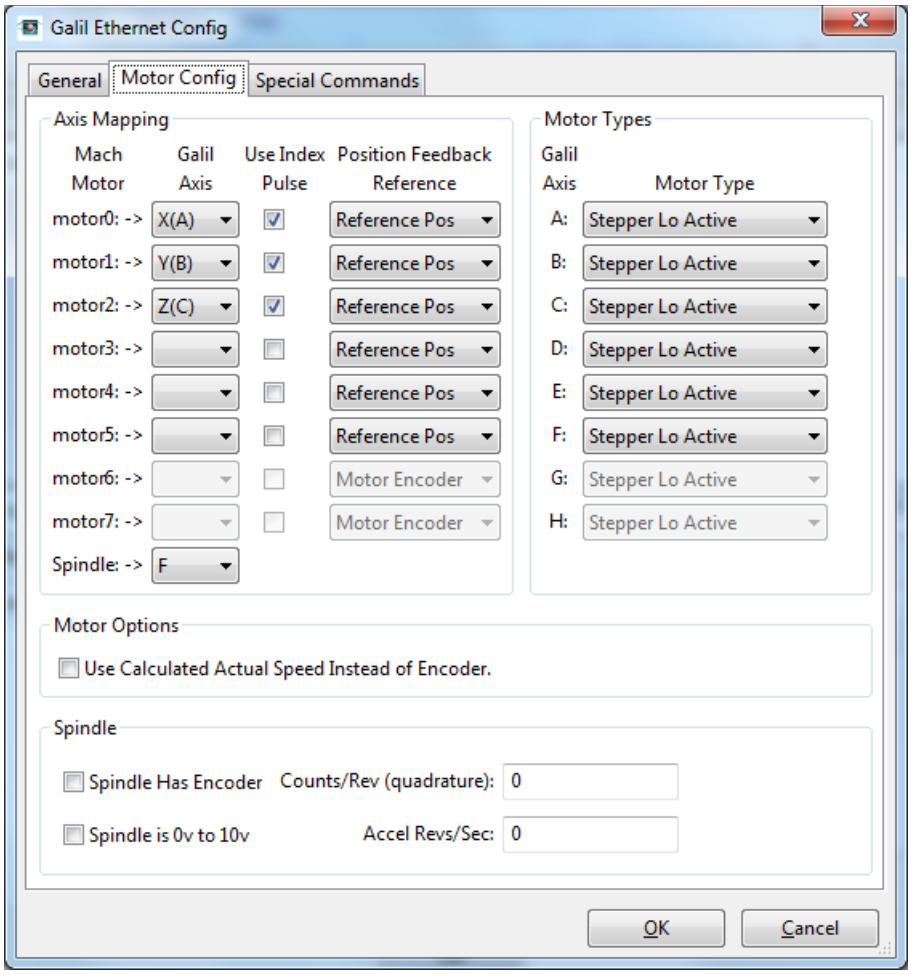


Figure 11 - Plugin configurations dialog Motor Config tab

Axis Mapping

Mach Motor

Mach Motor represents the motor device object in Mach4 that which a Galil axis can be mapped. The Galil Controller can have a maximum of 8 axes so only Motor0 to Motor7 are shown. A spindle is also defined. The method of spindle control will be discussed later.

Galil Axis

The Galil Axis field is a drop down list of available Galil axes. The list is capped by the available number of axes on the Galil controller that is being configured. As the axes are mapped to Mach motors, the are removed from the list of available Galil axes.

Use Index Pulse

The Use Index Pulse check box is related to homing. If your encoders have index pulses, then selecting this option will increase the repeatability of the homing operation.

Position Feedback Reference

Position Feedback can come from several Galil data record sources.

1. Motor Encoder: This is the main motor encoder. Most servo systems will use this.
2. Aux Encoder: The is the auxiliary encoder. Possibly used in a dual feedback backlash elimination system.
3. Reference Pos: This should be used if the Galil axis is controlling a stepper motor. Otherwise, the step and direction signals should be tied to the main motor encoder and that main motor encoder should also be configured with the CE command to accept the step and direction signals. See the Galil user manual for more information on this subject.
4. User Var: The newer Galil controllers provide a user variable per Galil axis that appears in the data record. Special position feedback can be accomplished with a Galil program that fills this user variable with any value. See the Galil user manual for more information on this subject.

Motor Types

Motor types define the motor that is attached to the Galil axis. These settings correspond to the Galil MT command.

1. Servo Motor (MT1)
2. Servo Reverse (MT-1)
3. Stepper Hi Active (MT-2)
4. Stepper Low Active (MT2)
5. Stepper Reversed Hi Active (MT-2.5)
6. Stepper Reversed Low Active (MT2.5)

Motor Options

Use Calculated Speed Instead of Encoder

Select this option if your coordinated axes system includes a stepper motor. Otherwise, you will need to wire the step and direction signals to the main motor encoder as described in Position Feedback Reference

Spindle

The default spindle control is via a Galil axis with the PID loop turned off. This essentially uses the Galil axis as an analogue output via the Galil OF command. The default is +-10v range with negative values for reverse revolutions and positive values for forward revolutions.

Spindle Has Encoder

Select this option if your spindle has an encoder directly mounted on the spindle itself or on the spindle motor if it is direct drive. To perform rigid tapping operations, the spindle **must** have an

encoder and this option **must** be selected.

With this option selected the Galil controls the spindle in the same manner as it would a servo drive. The spindle VFD must allow for a +-10v control input.

Counts/Rev (quadrature)

If the spindle has an encoder, the Galil plugin needs to know how many encoder counts per revolution. Input this value in this field.

Accel Rev/Sec

This controls the acceleration and deceleration of the spindle when the spindle uses an encoder.

Spindle is 0 to 10v

This option is used if the VFD controlling the spindle only accepts a 0 to 10v speed reference. Another means of controlling the direction with an output signal from Mach will also need to be provided. This options cannot be used in conjunction with a spindle with an encoder.

3.3 Extended I/O Tab

If the controller is capable of controlling extended I/O via either built in or daughter card, then the Extended I/O tab will be shown.

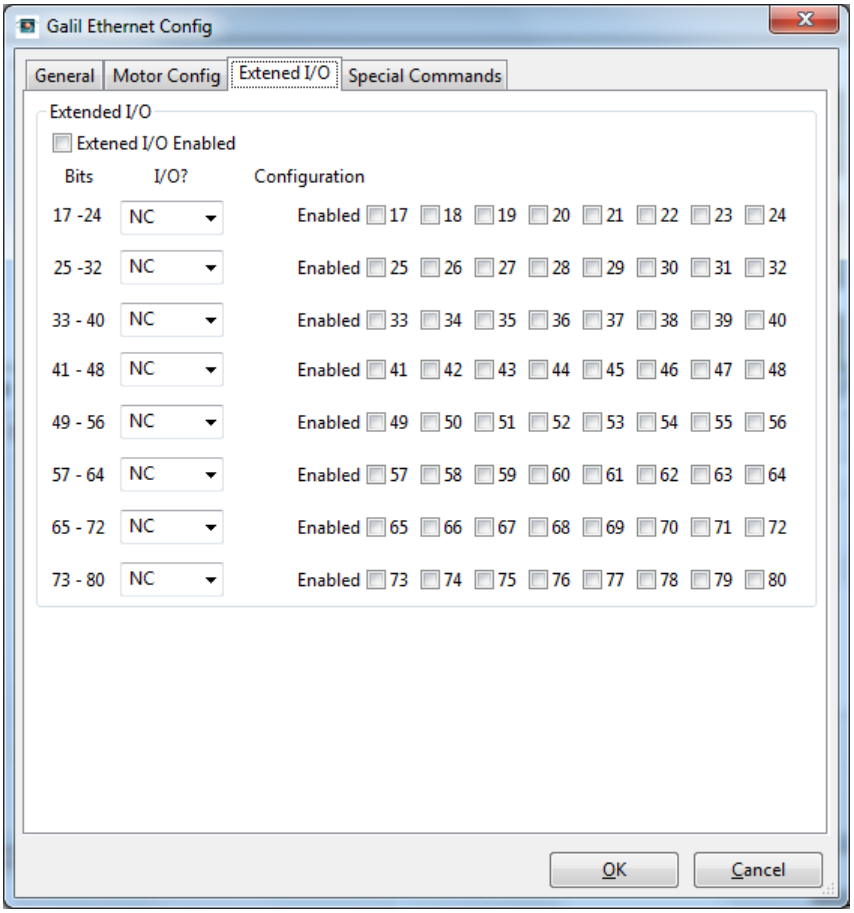


Figure 12 - Galil Plugin configurations dialog I/O tab

Extended I/O

Extended I/O Enabled

If extended I/O is not going to be used, the overhead required to process it can be diminished by not enabling the option.

Bits

These represent the bit range of the extended I/O block. The Galil uses the SB or CB commands to set or clear the bits controlling output blocks. If the block is an input block, the Galil command @IN[bit] can be used to read the state of the input..

I/O?

This setting determines if the block is configured as a block of inputs or a block of outputs or not configured at all.

Configuration

Each bit of a block and be enabled or disabled via the Enabled check box.

3.4 Special Commands

Special Commands are Galil commands that get sent to the controller are predefined times to enhance the power of the Mach/Galil combination.

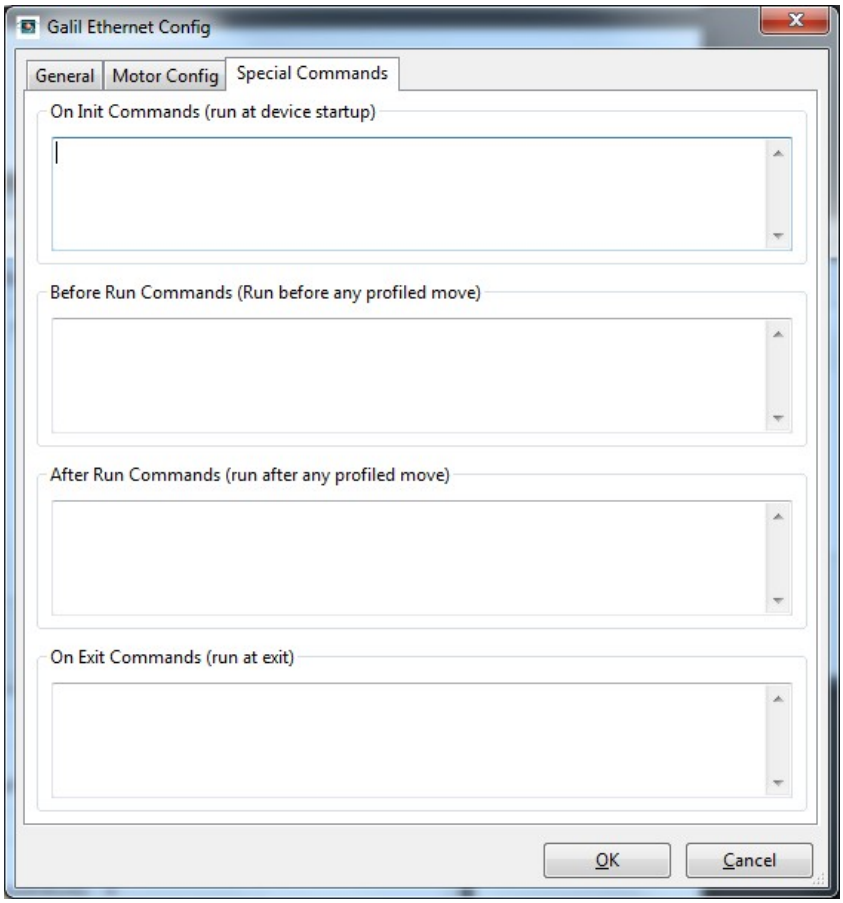


Figure 13- Plugin configurations dialog Special Commands tab

On Init Commands

These commands are run when the plugin is initialized. Possibilities include starting other Galil programs in other Galil threads, etc...

Before Run Commands

These commands are run before any trajectory profile is started. For example, they can be used to orient the spindle for a tangential knife.

After Run Commands

These commands run after the completion of a trajectory profile.

One Exit Commands

These commands will run at plugin shutdown so that any housekeeping can be done in a controlled manner.

3.5 A Galil RIO Configuration Example

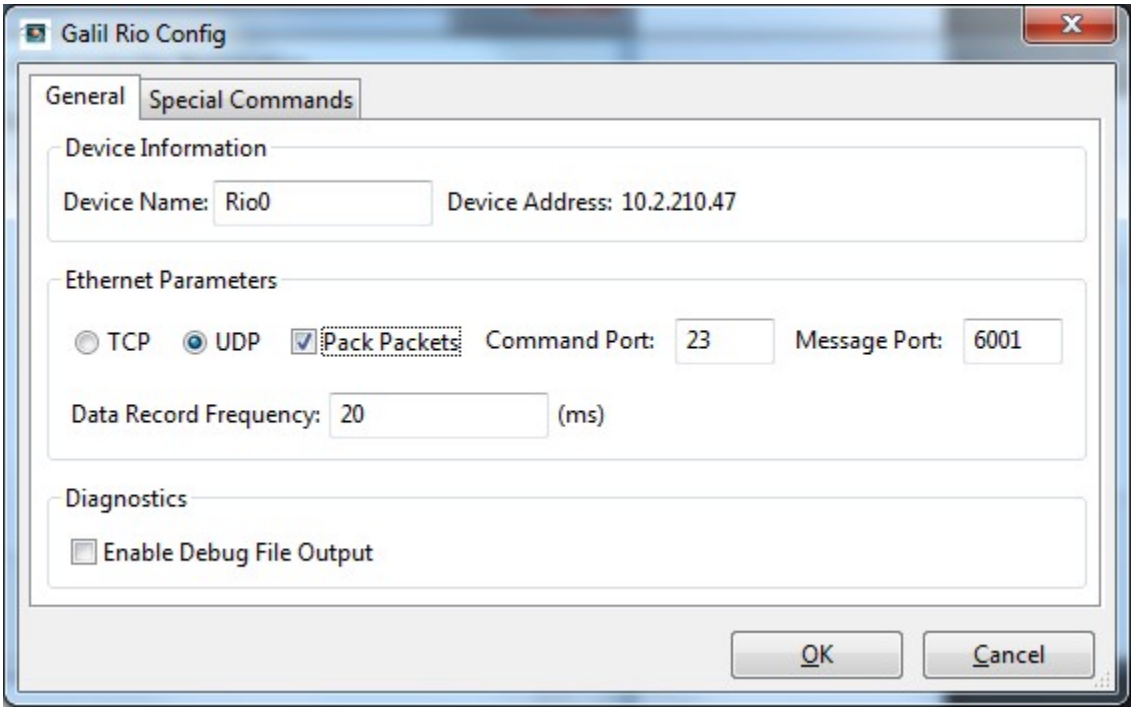


Figure 14 - Plugin configurations dialog RIO example

4 Finalizing the Mach/Galil Configuration

Close the Galil plugin configuration and navigate to the Configure->Select Motion Dev... menu item.

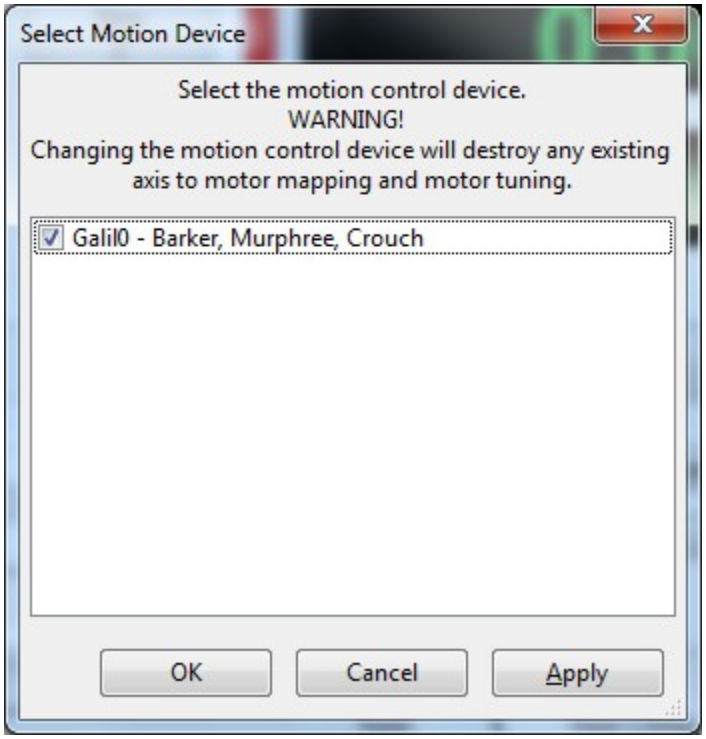


Figure 15 - Mach Motion Device selection dialog

Make sure the Galil device is selected as the motion device and click OK. At this point Mach should be configured according to the Mach Operators Manual. Setting motor parameter information, etc...

5 Registers and I/O objects exported by the Galil Plugin

The Galil plugin exports registers and I/O objects and makes them available for mapping operations in Mach.

5.1 Registers

Registers typically hold values that are not binary in nature such as numbers and strings. Registers can be accessed from Mach by a "device/register" path. For example, if our Galil device is named Galil0, then the register path to the main encoder would be "Galil0/MainEncA". The path for an analog input #1 would be "Galil0/AnalogInput1". All of the register that are exported to Mach are viewable with the RegFile Plugin's Diagnostics window.

Encoder Registers

The Galil Plugin will export a main and auxiliary encoder register for each axis available on the Galil controller. The register names will be formatted as follows:

- MainEnc**X** (Where **X** is in the range of A-H)
- AuxEnc**X** (Where **X** is in the range of A-H)

Analog Input Registers

The Galil Plugin will export one analog register per available axis on the Galil Controller. The Register names will be formatted as follows:

- AnalogInput**X** (Where **X** is in the range of 1-8)

The Command Register

The Galil plugin exports a special register called the Command register. It can be used to send Galil commands to the controller on an ad hock basis in macro or screen scripts with the mcRegSendCommand() API function call.

The register path for the command register will always be "DeviceName/Command". If our device is named Galil0, the "Galil0/Command" would be the path to the Command register.

5.2 I/O Objects

The Galil plugin will export a number of I/O objects depending on the model and configuration of the controller. I/O objects can be accessed by Mach references with a "device/object" path. These object populate the drop down lists in the Mach Input and

Limit Switches

The Galil plugin will export a forward (positive) and reverse (negative) limit switch per available axis on the controller. The limit switch names will be formatted as follows:

- LimitPos**X** (Where **X** is in a range of A-H)
- LimitNeg**X** (Where **X** is in a range of A-H)

Home Switches

The Galil plugin will export a home switch per available axis on the controller. The home switch names will be formatted as follows:

- Home**X** (Where **X** is in a range of A-H)

Inputs

The Galil plugin will export 8 general purpose inputs on a 1-4 axis Galil and 16 general purpose inputs on a 4-8 axis controller. If the controller has extended I/O, the number of inputs can be increased depending on the extended IO block configurations. The input names will be formatted as follows:

- Input**X** (Where **X** is in a range of 1-max inputs)

Enable Outputs

The Galil plugin will export an enable output per axis available on the Galil controller. The enable output names will be formatted as follows:

- Enable**X** (Where **X** is in a range of A-H)

Please refer to the Galil User's manual for information on how to wire and configure the enable (ENx) signals. A lot of the time, the ENx signals are Galil Interconnect Module (ICM) dependent.

Outputs

The Galil plugin will export 8 general purpose output on a 1-4 axis Galil and 16 general purpose output on a 4-8 axis controller. If the controller has extended I/O, the number of outputs can be increased depending on the extended IO block configurations. The output names will be formatted as follows:

- Output**X** (Where **X** is in a range of 1-max inputs)

6 Credits

The Galil Plugin has quite a storied existence. Without certain things (planetary and solar alignments) in the past happening at just the right moments in time, it may never have existed.

6.1 Programmers

1) Brian Barker
2) Steve Murphree (Documentation Author)
3) Ken Crouch
4) Jeff Birt

6.2 Testers

1) Ken Crouch (The master brave one)
2) T & S Machine
3) Mach Motion

6.3 Technical Consultants

Not only is the Galil a great motion controller, the Galil company is blessed with a great team from the top down. Here is a short list of Galil people who help us make this happen, but I'm sure there were many more behind the scenes.

1) Kaushal Shah (Galil)
2) Daniel Roberts (Galil)
3) Troy Miller (Galil)
4) Cory Bowdach (Galil)