



**bome software**

*everything sounds.*

# Bome's Midi Translator

## USER MANUAL

# 1 Welcome

Thank you for choosing Bome's Midi Translator!

## **This document contains:**

- [Midi Translator Quick Start](#)
- [Midi Translator Manual Documentation](#)

## **Announcement List:**

- As a legal owner of this software you may get updates of this Software. [Sign up to the announcement list](#) to get update notifications. The volume on this list is very low (one email per new version of Midi Translator).

## **User Support:**

- [Discussion Forums](#): get professional help for your questions
- [Contact Bome Software](#): user support for registered users

We hope you enjoy our software,

The Bome Software Team

**Note:** These help documents were created for the Windows version of Midi Translator. The concepts are the same, but the screenshots look a little different. Sorry for the inconvenience.

# 2 Quickstart

## 2.1 Before You Begin

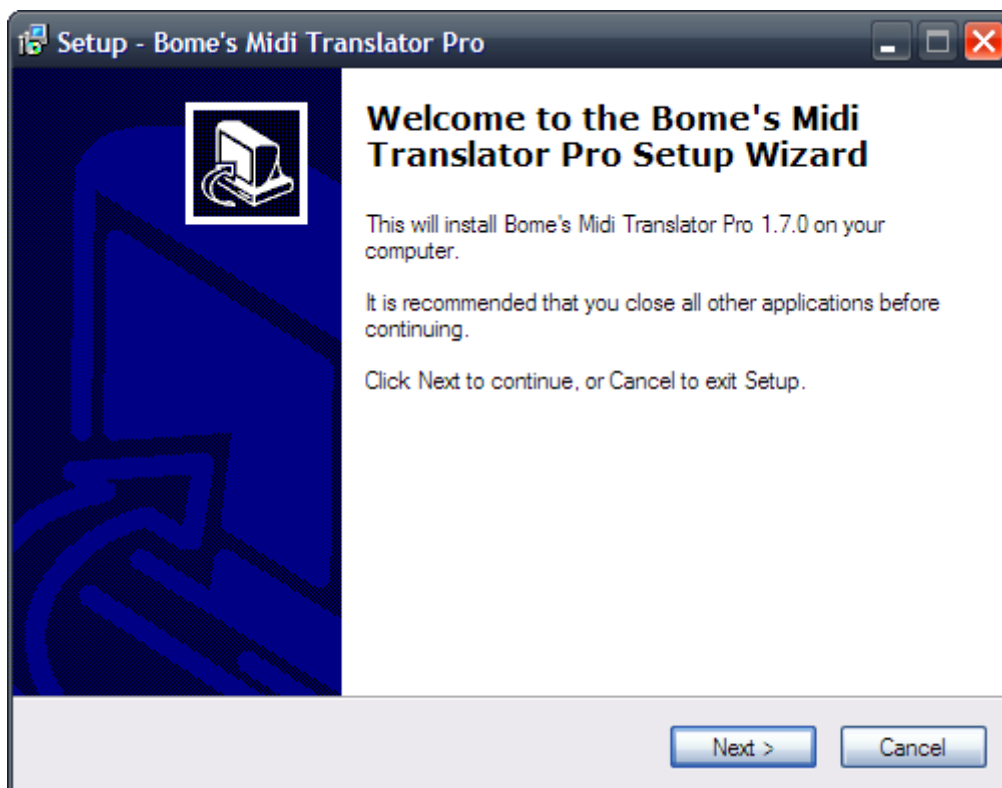
Install Bome's Midi Translator with the options that you require. If you wish to transmit MIDI messages to another application, make sure to install Bome's virtual MIDI port drivers during installation. On Mac computers the virtual port is installed by default.

## 2.2 Installation

### 2.2.1 General Installation

#### > INITIAL INSTALL SCREEN

After double-clicking the downloaded executable file, the installation wizard will begin. As stated, it is recommended that you close any other applications before beginning the installation.



***initial install screen***

## > LICENSE AGREEMENT

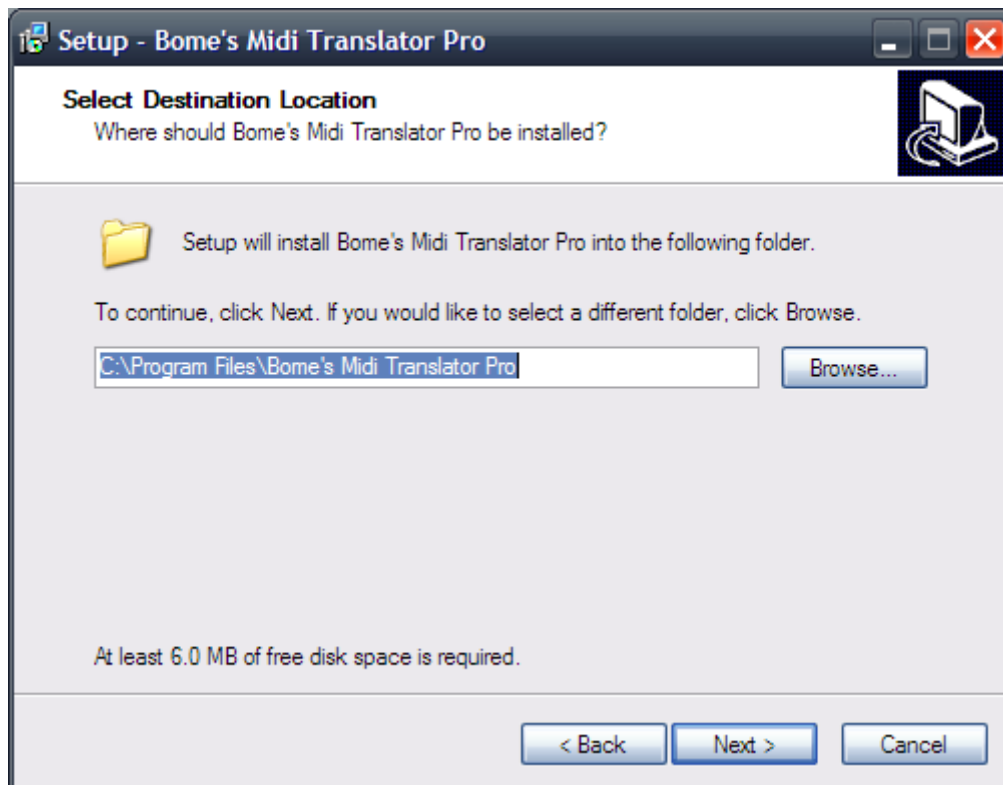
Please read the license agreement thoroughly. After reading, please select "I accept the agreement", and click 'Next' to acknowledge the license agreement and continue with installation.



***license agreement***

## > CHOOSE INSTALL DIRECTORY

Choose a directory to install Bome's MIDI Translator to. If you would like to install Bome's MIDI Translator to another directory or disk drive, click Browse and select the desired installation directory prior to clicking Next. If you have no preference, please click Next and continue with the installation process.

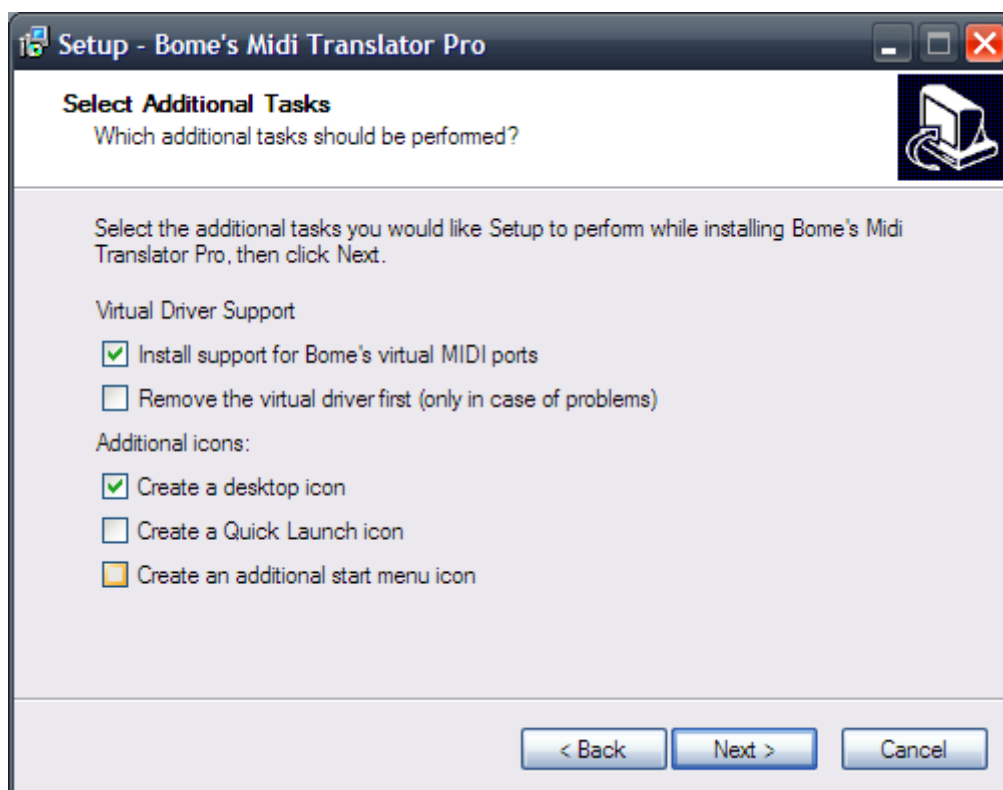


***choose install directory***

## > ADDITIONAL INSTALLATION TASKS

Choose whether or not you'd like to allow the installation to create a desktop icon and start menu icons so that you may easily start the program after it's installed. Also, choose whether or not to install the virtual MIDI port drivers. If you choose not to install them, please skip ahead to the Release Notes section. Otherwise, continue on to Virtual Port Installation.

**Note:** Bome's MIDI Translator Pro CAN operate without Bome's virtual MIDI ports, however program functionality will be limited. Virtual MIDI port devices allow Bome's MIDI Translator to communicate with software hosts that don't have direct MIDI port connections.

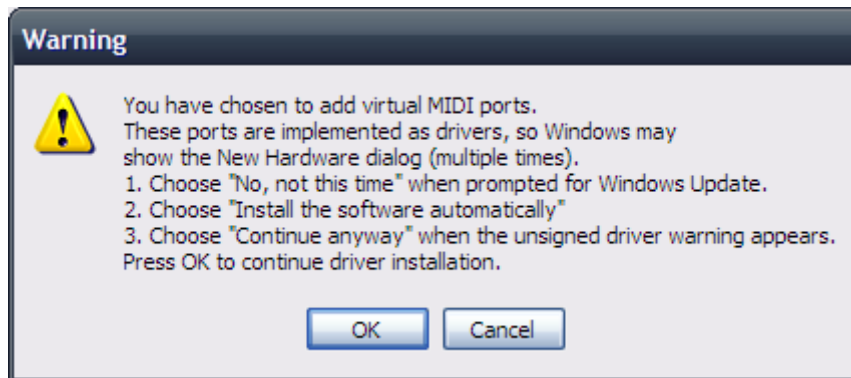


**menu options**

## 2.2.2 Virtual Port Installation

### > INSTALL WARNING

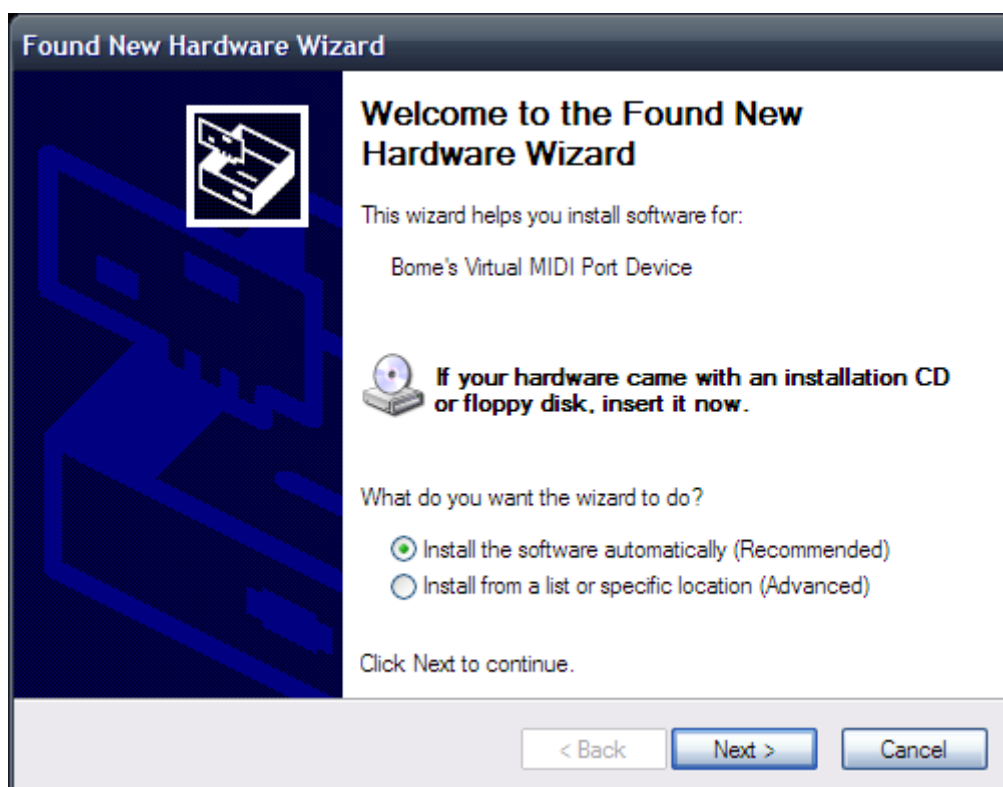
If you have chosen to install support for Bome's virtual MIDI ports, you will be presented with a dialog box detailing their installation process. Read the instructions carefully and then click OK.



### *virtual port install warning*

### > VIRTUAL PORT INSTALL

Bome's virtual MIDI ports are installed much like a real hardware device. Click "Install the software automatically" and then click Next to continue.



### *virtual port hardware install*

## > LOGO TESTING

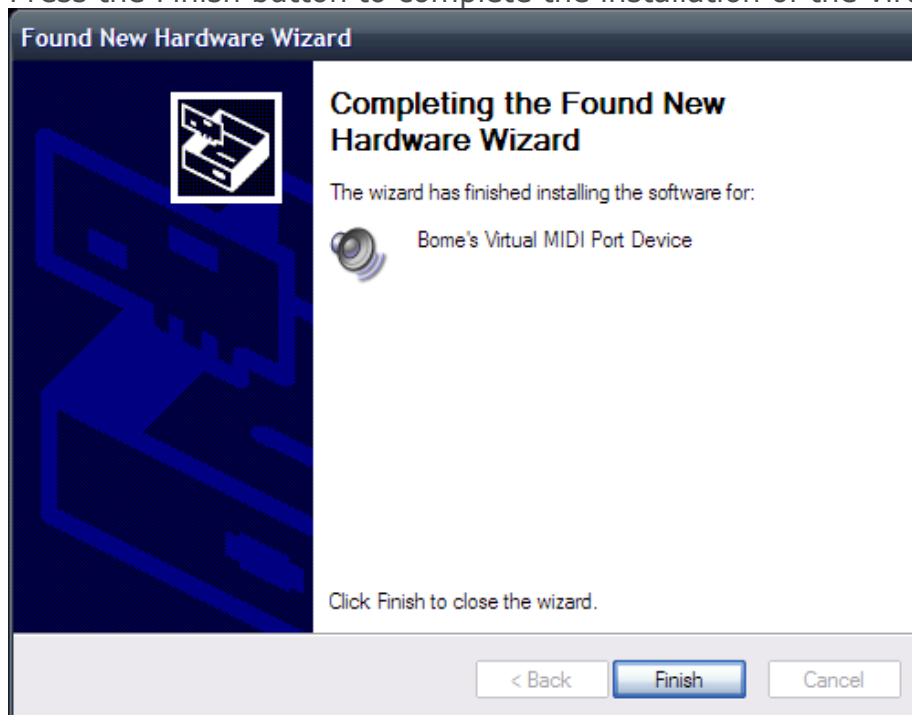
Be sure to click Continue Anyway when you are prompted about Windows Logo Testing.



*logo testing warning*

## > VIRTUAL PORT INSTALLATION COMPLETE

Press the Finish button to complete the installation of the virtual MIDI port.

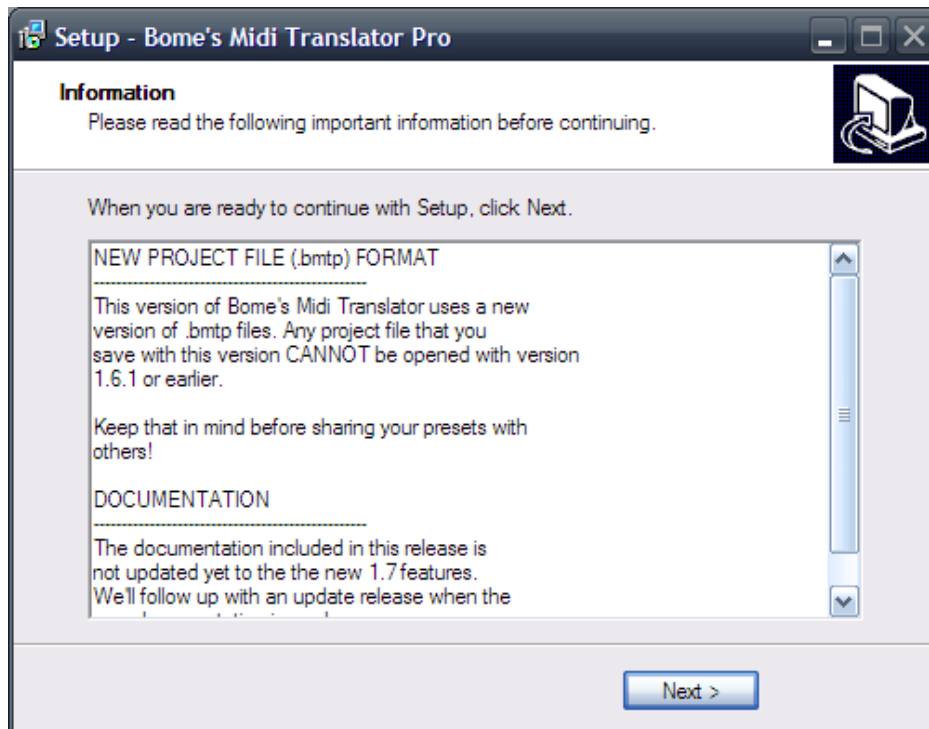


*virtual port installation complete*



## > RELEASE NOTES

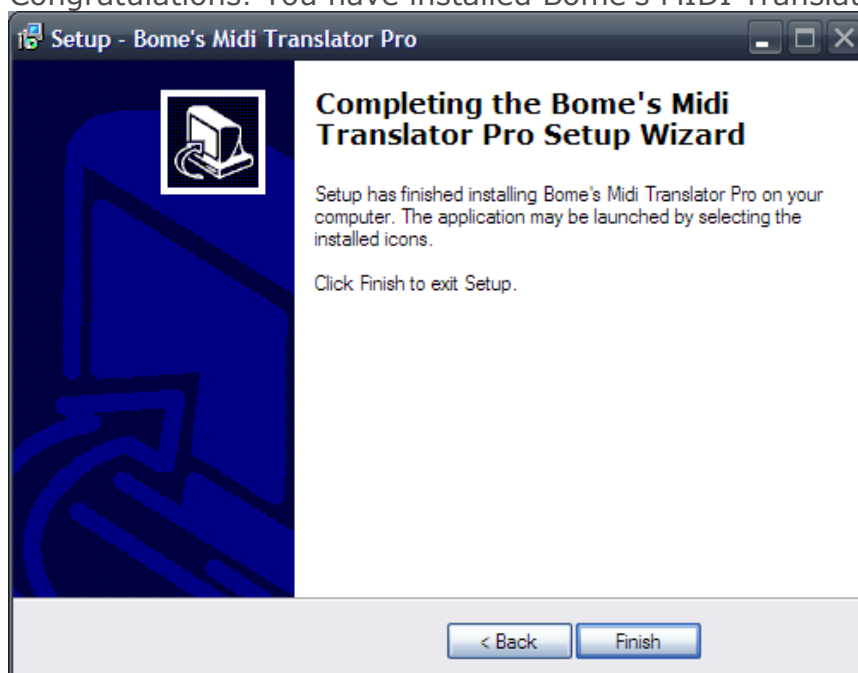
Please read the software release notes and then press Next.



### *release notes*

## > COMPLETION

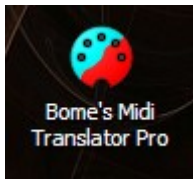
Congratulations! You have installed Bome's MIDI Translator successfully.



### *installation completion*

## > STARTING MIDI TRANSLATOR

Start Bome's MIDI Translator by double-clicking the desktop icon.

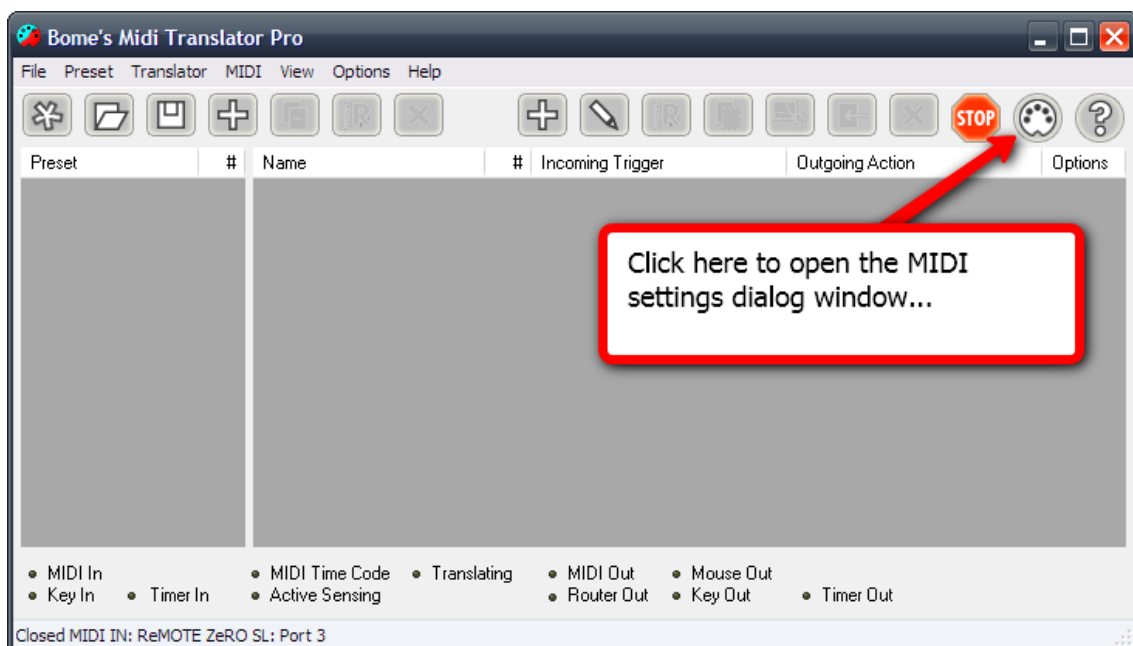


***starting midi translator***

## 2.3 MIDI Setup

### > MIDI SETTINGS

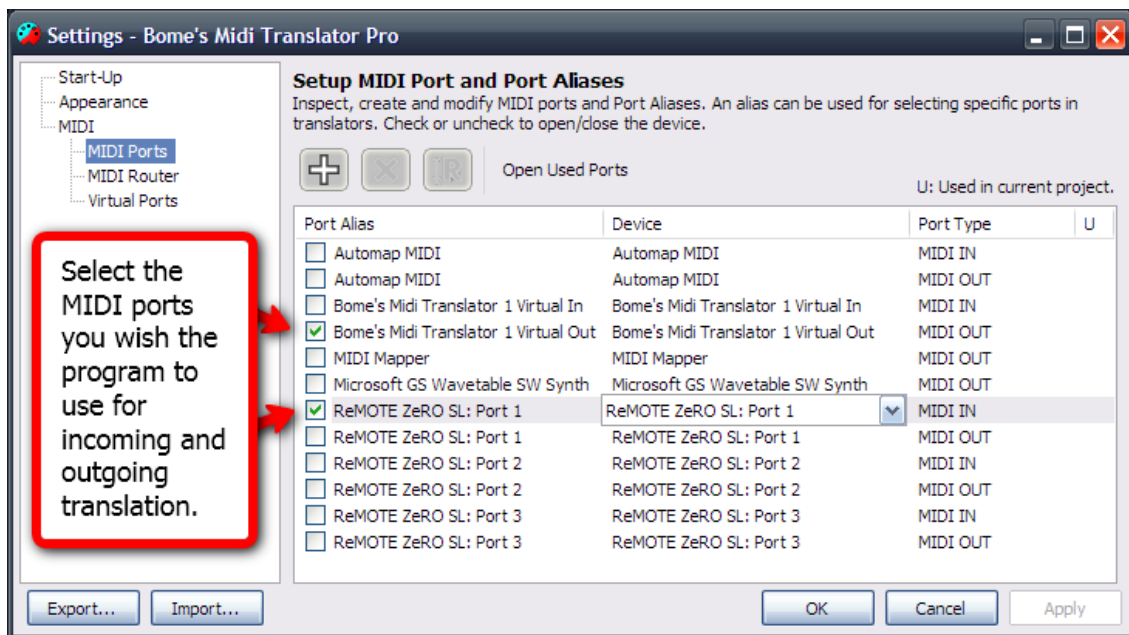
The first step in setting up Bome's MIDI Translator to work with your MIDI device is to define it in the program MIDI settings. To access the MIDI settings simply press the settings icon in the toolbar or select Options and then Settings in the menu.



***midi settings***

## > DEFINE MIDI PORTS

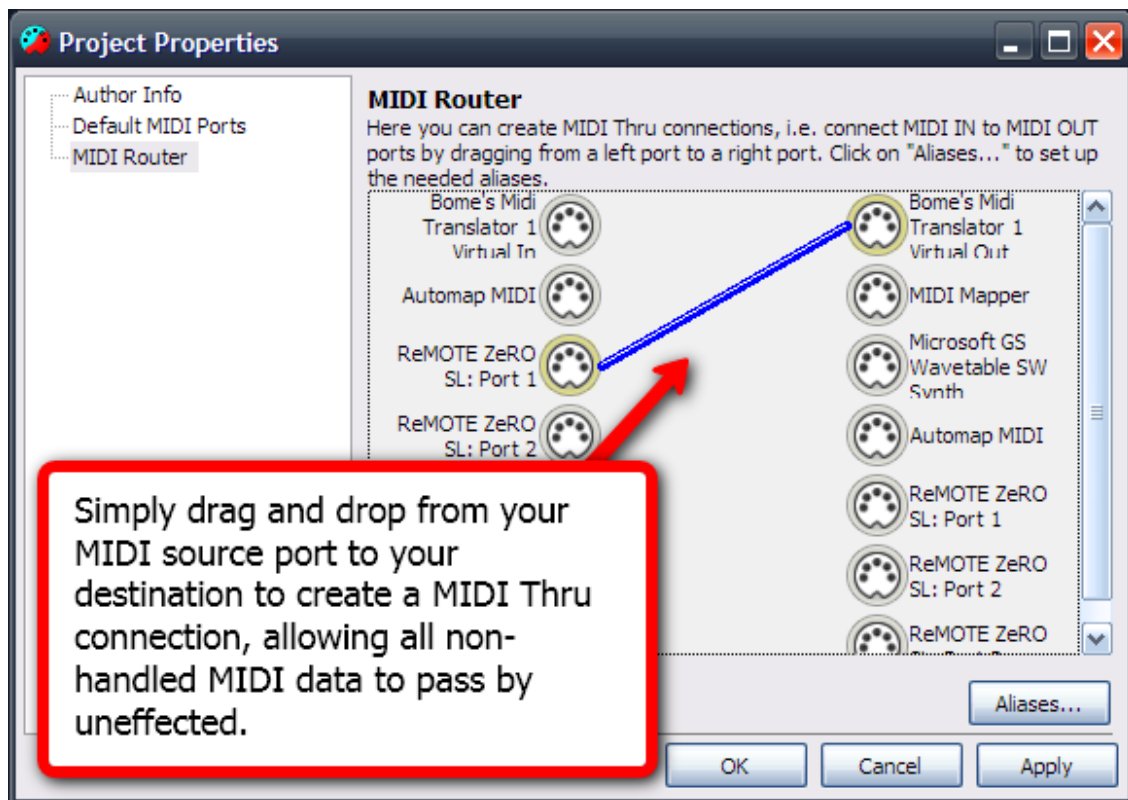
Next, specify the MIDI input and output ports you will be using. The MIDI OUT port will be the port translated MIDI messages are sent to. This is where your virtual MIDI port will be used if you are setting up a translator for a software sequencer or other audio application that interfaces with MIDI. Select your virtual MIDI port as the output device and then specify it as the MIDI input port in your 3rd party application in order to have Bomes Midi Translator control it. The MIDI IN port will be used as the source of MIDI data, typically connecting with an external USB controller or a MIDI interface on a sound card. Select the appropriate MIDI input source and then move on to MIDI thru routing...



***define midi ports***

## > MIDI THRU

Access the MIDI router by either clicking File/Project Properties/MIDI Router in the menu or by clicking MIDI Router in the Settings dialog and then clicking Open Project Properties. Click and drag a MIDI IN connection on the left side of the screen to the desired MIDI OUT connection to enable a MIDI thru connection on the ports. Any data NOT capture by Bome's MIDI Translator will be output to the designated output port.

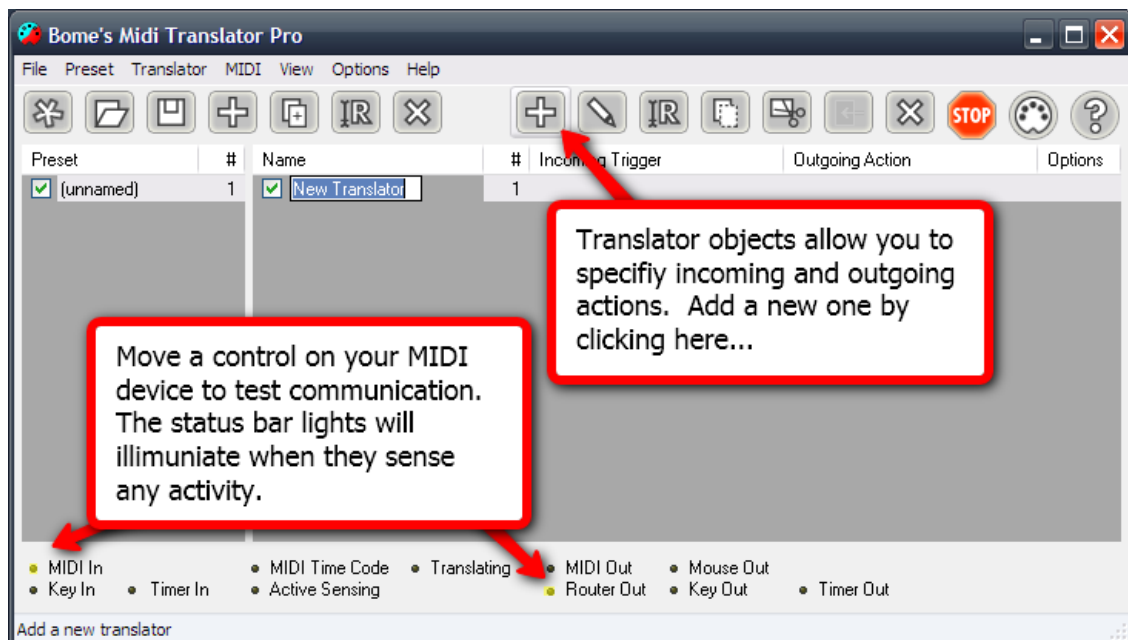


*define midi thru connections*

## > ADD TRANSLATOR

You now should have the MIDI interface settings properly configured. Test that they're working correctly by moving a controller on your MIDI device and checking to see if the corresponding light illuminates on the status bar.

Now you may begin adding translators. Click the Add Translator button on the toolbar to add a new blank translator. Name your translator something and then press the Enter key. You can now begin working with the translator. Double click the translator object to enter the Translator Edit screen.

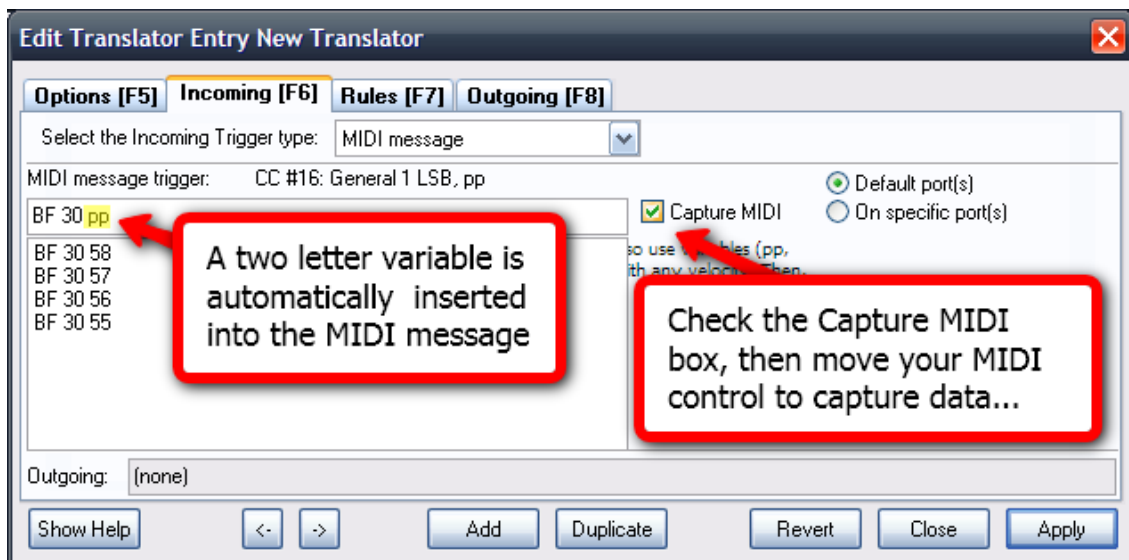


### ***adding translators***

## 2.4 Defining Translators

### > INCOMING ACTIONS

The Edit Translator screen is where you specify the incoming and outgoing actions that the translator operates with, as well as the rules and processes that take place between those actions. To start, we'll need to specify an incoming action to work with. Be sure the Incoming Trigger type is set to MIDI message, and click the Capture MIDI checkbox. Assuming your MIDI settings are correct you should see a list of MIDI messages scroll by the screen as you move a controller or press a key or button on your MIDI device. You may notice that a two letter variable has been inserted into your MIDI message. This variable is added if the detected MIDI message have altering velocity messages, as is the case with most continuous controllers such as sliders, potentiometers, keyboard keys, etc. This two letter variable can be used later in the rules section. Uncheck the Capture MIDI checkbox once you have detected MIDI data from your selected MIDI controller.

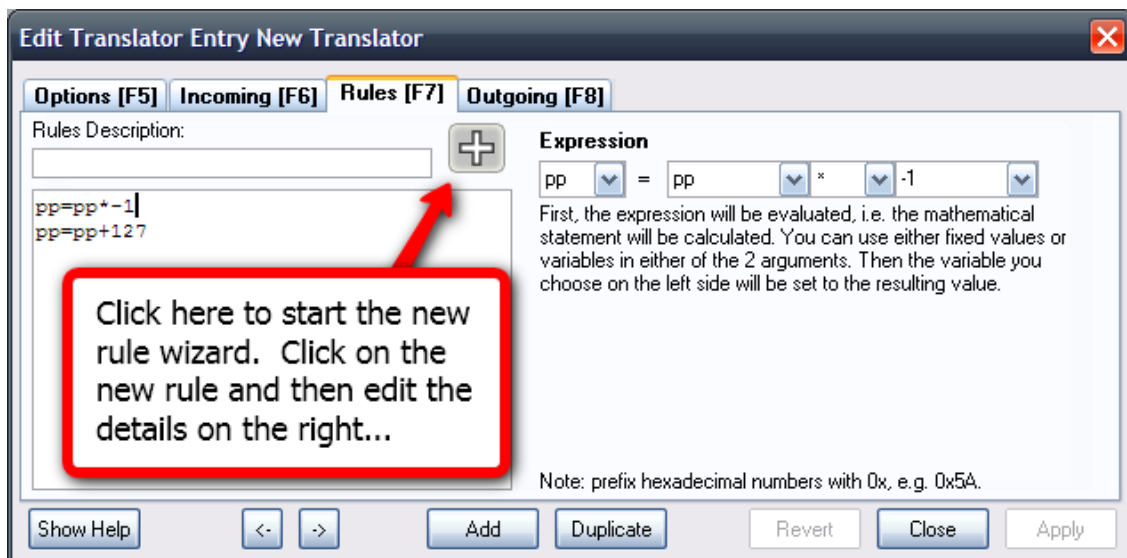


#### *translator incoming midi capture*

**Note:** Incoming MIDI messages are received and transmitted in their native hexadecimal values. Variables can be assigned to any part of an incoming MIDI message trigger string.

## > TRANSLATOR RULES

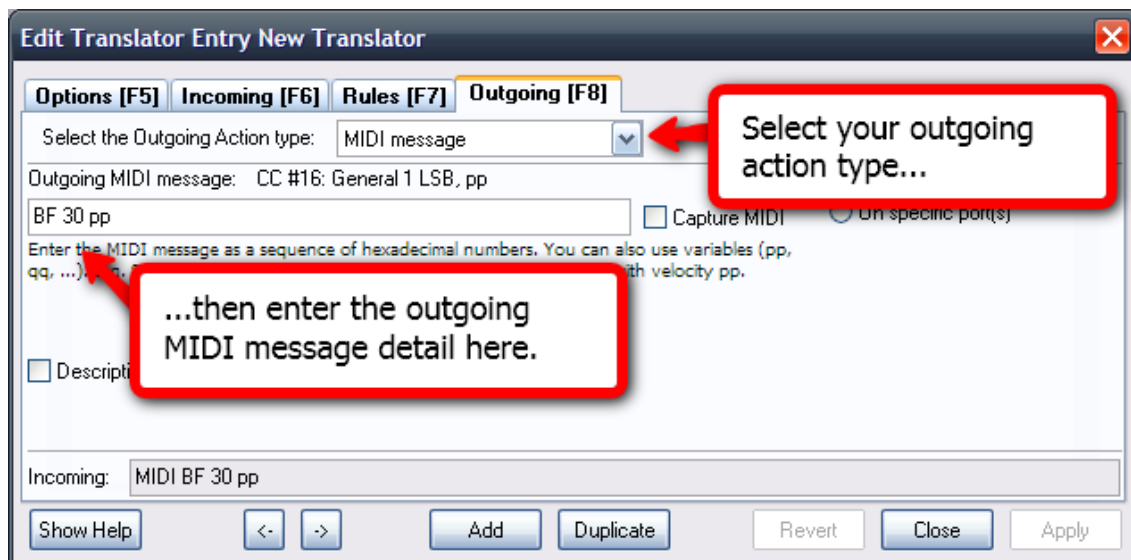
Next click on the Rules tab to view the rules entries for the translator. In this screen you can specify rules that effect the values local to the translator, or values stored in global variables. Click on the Add Rule button to select the type of rule that you wish to enter. After adding the basic rule, edit the rule parameters on the right side of the Rules dialog screen. Rules can also be entered directly into the text field of the Rules dialog. In the example, two rules have been entered that will reverse the controller value of a standard MIDI signal. The first rule takes the variable value of the incoming MIDI signal (for example, the turning of a MIDI knob going from 0 to 127) and multiplies it by -1. The following rule takes the negative value of the incoming signal from the prior rule and adds 127 to it, effectively reversing the signal. Experiment with adding your own rules and editing the rules parameters. Refer to the Translators section of the main manual to find out more about Rules.



### *translator rules entry*

## > OUTGOING ACTIONS

Now you need to specify what you want your outgoing action to be for the translator. Select an action type from the dropdown box and then enter the action details in the detail area below. Outgoing action types are varied and depend on the application you are working with and what you are trying to accomplish. Keep in mind that you can use both local and global variables in your translator entries. In the example we are duplicating the incoming MIDI signal's header information, so the outgoing action will have identical MIDI channel and controller information.



### *translator outgoing actions*



# 3 Midi Setup Guide

## > VIRTUAL MIDI PORTS

Beginning with Bome's MIDI Translator v1.7 and above, a virtual MIDI port driver is included as an optional install with the application.

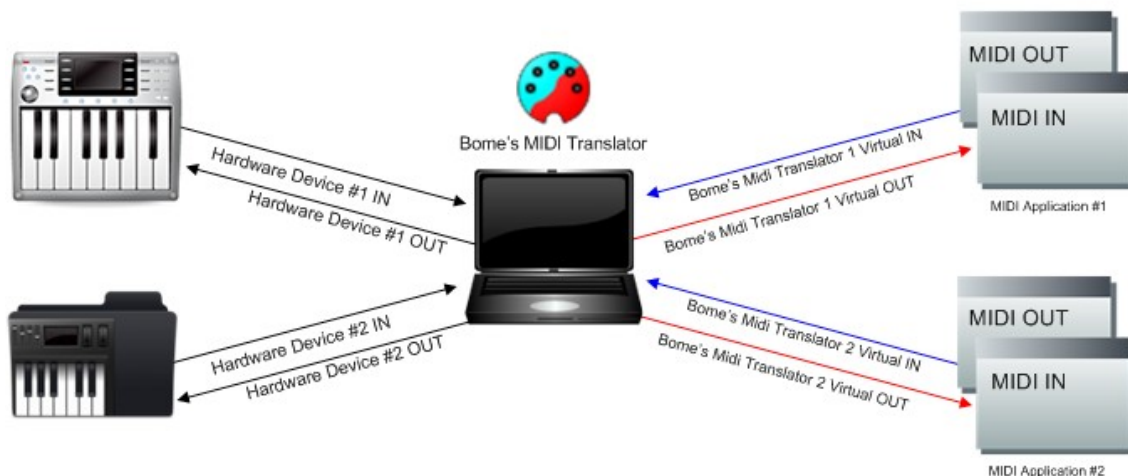
Virtual MIDI ports are available for use as unidirectional MIDI ports, requiring the MIDI Translator application on one end, and any another MIDI device destination on the other. Unlike "loopback" virtual MIDI ports which function as both IN and OUT ports simultaneously, Bome's virtual MIDI ports only pass through the Windows API once rather than twice to route MIDI data. This added efficiency makes Bome's MIDI Translator faster and more reliable, with the end result of reduced latency and jitter when operating with MIDI data. Bome's MIDI Translator also benefits from a high-speed MIDI data processing engine at it's core, delivering MIDI and translator action data at near realtime speeds.

As a result of the direct connection Bome's virtual MIDI ports have into the MIDI Translator's API, one end of the virtual MIDI port **MUST** be connected to the MIDI Translator application via translators or the MT Router. Direct use of the virtual MIDI ports between two applications is impossible.

Bome's MIDI Translator can communication directly with any MIDI device or application, allowing it to serve as a powerful hub for MIDI information. A common use for MIDI Translator's virtual ports would be to synchronize the MIDI clocks of two applications. Using MIDI Translator's virtual ports, this is a simple task of linking each application to a Bome's virtual MIDI ports, then connecting them in the MIDI Router.

### Bome's MIDI Translator – Virtual MIDI Ports

Virtual MIDI Ports can be used to allow MIDI communications between multiple external devices and multiple 3<sup>rd</sup> party applications. Linking one application's IN port to another application's OUT port can be configured easily in the MIDI Router section of the program settings. Bome's MT Virtual MIDI Ports are designed to be fast, stable and highly configurable.



**virtual port flow diagram**

## > DEVICES AND ALIASES

Devices and aliases represent the different MIDI sources and destinations available for MIDI Translator to send and receive MIDI data. Devices are automatically assigned an identically named alias, and are listed in the MIDI Ports configuration page in the program settings.

Aliases function as dynamic links to devices, allowing a MIDI Translator project to be shared amongst users with different MIDI hardware and software. When a Translator entry is created, default input and output MIDI ports are assigned to it based on settings in the Project, Preset and Translator default ports configuration pages. When a project file is opened by a user that has different hardware and software MIDI ports, the program will ask the user to reassign the used ports in the project to different MIDI devices that are available on the computer.

Standalone aliases can also be created, allowing you to create named ports in your projects to better organize MIDI communications. For example, it can be beneficial to create named ports in your project such as "KEYBOARD IN" and "PROJECT OUT" to make your projects more human readable.

Default MIDI Ports and/or Aliases can be assigned to different elements of your Bome's project, allowing flexibility in routing MIDI data to different devices. Default ports can be assigned at the overall Project level, which will dictate where newly created MIDI translators will receive and transmit MIDI data. After the Project level, default ports/aliases can be assigned at the Preset level, allowing you to override the Project default ports and have entire Presets dedicated to managing the MIDI data coming from or going to a particular device or devices. Lastly, individual Presets can have default port assignments that override both the Preset and Project default ports.

**Settings - Bome's Midi Translator Pro**

**Setup MIDI Port and Port Aliases**  
Inspect, create and modify MIDI ports and aliases. Check or uncheck to enable or disable specific ports in translators. Check or uncheck to enable or disable specific ports in the current project.

Use Aliases to create human-readable names for your MIDI sources and targets

Port Alias	Device	Port Type	U
<input checked="" type="checkbox"/> Project - Ableton OUT	Bome's Midi Translator 2 Virtual Out	MIDI OUT	U
<input checked="" type="checkbox"/> Project - Remote SL IN	ReMOTE ZeRO SL: Port 1	MIDI IN	U
<input checked="" type="checkbox"/> Project - Traktor IN	Bome's Midi Translator 2 Virtual In	MIDI IN	U
<input checked="" type="checkbox"/> Project - Traktor OUT	Bome's Midi Translator 1 Virtual Out	MIDI OUT	U
<input type="checkbox"/> Audio 8 DJ In	Audio 8 DJ In	MIDI IN	
<input type="checkbox"/> Audio 8 DJ Out	Audio 8 DJ Out	MIDI OUT	
<input type="checkbox"/> Automap MIDI	Automap MIDI	MIDI IN	
<input type="checkbox"/> Automap MIDI	Automap MIDI	MIDI OUT	

\* Devices represent actual hardware and software MIDI ports.

\* Aliases are pointers to these devices that can be reconfigured on the fly.

\* Default aliases can be specified via project settings, preset or individual translator.

Aliases are automatically created for any detected MIDI devices. If a device is changed or removed from your configuration, you can reassign any Translators using it to another device easily by pointing the Port Alias to a new device in the MIDI Ports configuration settings screen

OK Cancel Apply

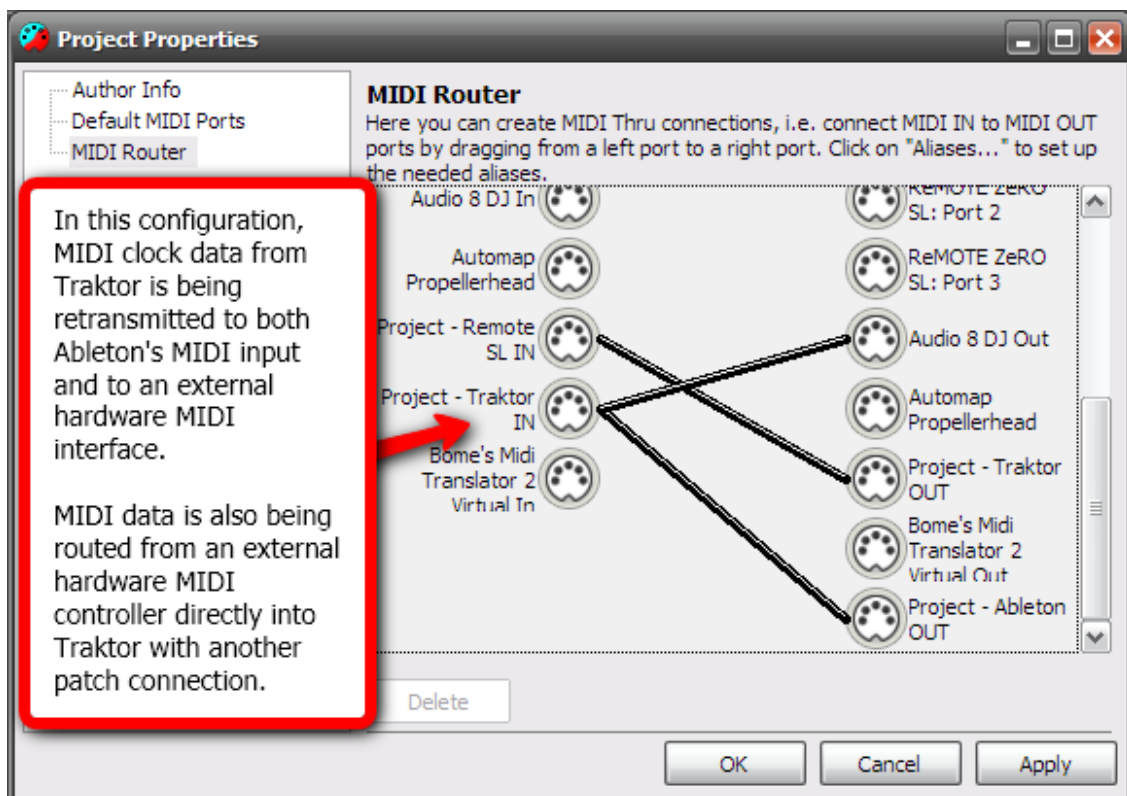
### **devices and aliases**

## > MIDI ROUTER

By default, MIDI Translator does NOT route ANY MIDI data. For MIDI data to be processed, either a Translator must be created for it, or a MIDI Router connection must be made.

The MIDI Router is a patch panel type setup screen that allows for "patch cords" to be created between available MIDI devices and aliases. Multiple connections can be made from a single source MIDI IN device allowing MIDI data to be replicated and send to multiple devices concurrently.

A MIDI Router patch connection in Bome's MIDI Translator can effectively be thought of as a MIDI Thru connection. Any MIDI data that is received by a source device in a MIDI Router patch connection is retransmitted to all connected destination devices.



***midi router setup***

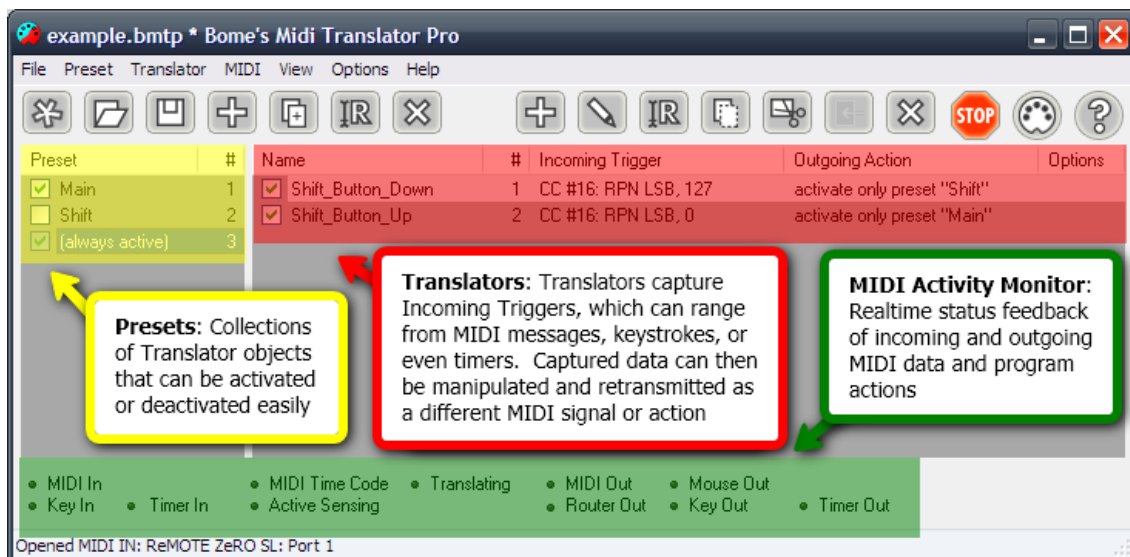
# 4 Program Interface

## > TRANSLATOR INTERFACE

From the main window of Bome's MIDI Translator, a user can manage projects, presets and translators from start to finish. The main interface is subdivided into four main sections; the Menu System, the Toolbar, the Translator Interface, and the Activity Monitor.

The translator interface defines which translator rules are available and activated, as well as allowing you to switch between available presets. Available presets are listed on the left pane window, along with the '[always active]' preset if defined. The right side of the Translator Interface contains the various rules that are defined in the template, along with a checkbox for activating/deactivating the translators and a brief rundown of the incoming and outgoing actions for each.















Each item in the translator interface has an appropriate context menu that is easily accessed by right-clicking the object. Also, global-level actions for either presets or translators can be accessed by right-clicking the background of the appropriate pane you wish to work with.



### *interface elements*

## > TOOLBAR

The Toolbar enables easy access to the most commonly used menu items in Bomes MIDI Translator. There is an equivalent menu item for each of these icons, which can be accessed via the menu system or via keyboard shortcut.

Icon	Name	Description
	New Project	Start a new empty project
	Open Project	Open a Project file from disk
	Save Project	Save the current Project to disk
	New Preset	Create a new Translator Preset
	Duplicate Preset	Create a new Preset with the same Translators of the current Preset
	Rename Preset	Rename selected Translator Preset
	Delete Preset	Delete selected Translator Preset
	New Translator	Add a new Translator
	Edit Translator	Edit the selected Translator
	Rename Translator	Rename the selected Translator
	Copy Translator	Copy the selected Translator(s) to the clipboard
	Cut Translator	Cut the selected Translator(s) to the clipboard
	Paste Translator	Paste Translator(s) from the clipboard
	Delete Translator	Delete selected Translator(s)
	Stop	Reset the MIDI out device (Panic)
	Settings	Show the Settings dialog for program configuration
	Help	Show Help Topics

## > ACTIVITY MONITOR

This is the event activity area of the screen, where you can quickly see what internal and external signals Bomes MIDI Translator is processing. The virtual LEDs will flash when the noted action is being performed, or the noted signal is being received or transmitted.

MIDI In      MIDI Time Code      Translating      MIDI Out      Mouse Out  
 Key In      Timer In      Active Sensing      Router Out      Key Out      Timer Out

### ***activity monitor***

# 5 Presets

## > PRESETS OVERVIEW

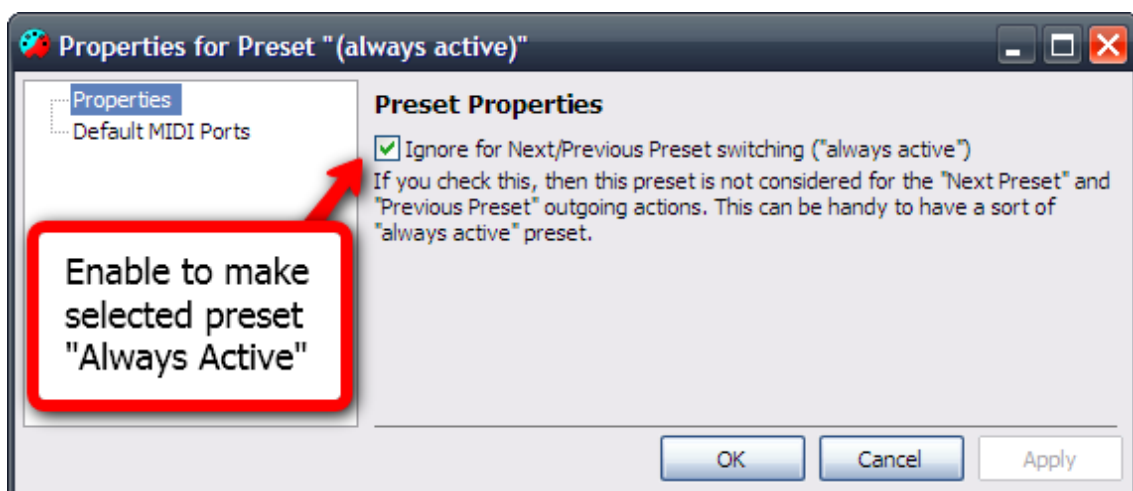
Bome's MIDI Translator includes the capability to encapsulate Translator entries into 'Presets' which can be managed at a more detailed level than a normal 'Global' setup where every translator is active all the time. A preset's active or inactive state can be determined by looking at the checkbox next to its name. If the checkbox is checked, the preset is active and its translators are being processed. If the preset's checkbox is unchecked, it is deactivated and no processing occurs.

Presets can be selected by clicking the preset's name, and they can be activated and deactivated via the mouse by clicking the checkbox. Presets can also be managed via the context menu accessed by right-clicking either the preset itself (copy, rename, delete, etc) or by right-clicking the Preset area's background.

Presets activation and deactivation can also be manipulated via Outgoing Actions of normal Translators.

## > ALWAYS ACTIVE PRESETS

Bome's MIDI Translator includes the capability to make a preset Always Active, rendering it exempt from the rules of Preset Change Outgoing Actions from translators. To make a preset Always Active, simply right-click it and select Properties, then from the Preset Properties Menu, enable the checkbox for Ignore for Next/Previous Preset Switching. Any presets with this property enabled will be processed by the program no matter what.



***always active presets***



## > CHANGING PRESETS

Presets can be activated and deactivated by clicking the checkbox next to the preset name. This is the most direct way of working with presets. However, mouse and GUI access is at many times at a premium, so Bome's MIDI Translator features the capability to switch and work with Presets via the 'Preset Change' outgoing Translator action.

Presets can be activated via Preset Change action by one of three primary methods;

- **Activate Previous/Next Preset**This outgoing action will cycle through the available presets in order, activating only one at a time. Presets must be arranged in the order of which the user desires to cycle them on and off. This is normally the preferred way of cycling through presets.
- **Activate/Deactivate By Name**This action will activate/deactivate a preset that is selected by name from a drop-down box. This is useful for most simple preset setups involving few presets. Features that involve key commands can be enabled/disabled on the fly so as not to interfere with regular keyboard operation when not needed.
- **Activate/Deactivate By Number**Presets can be enabled or disabled by number, which can be specified by a unique local or global variable (see the rules section for more information). This outgoing preset change action is useful for more complex MIDI Translator scripts that have many presets and change them on-the-fly depending on other Translator settings and variable states.

Presets also have the checkbox option available to 'Deactivate all other presets (except the always active)'. This can be enabled in any Preset Change outgoing action to automatically disable all other presets (except, of course, the 'always active' preset).

The screenshot shows the Bome's Midi Translator Pro interface. The main window displays a table of presets:

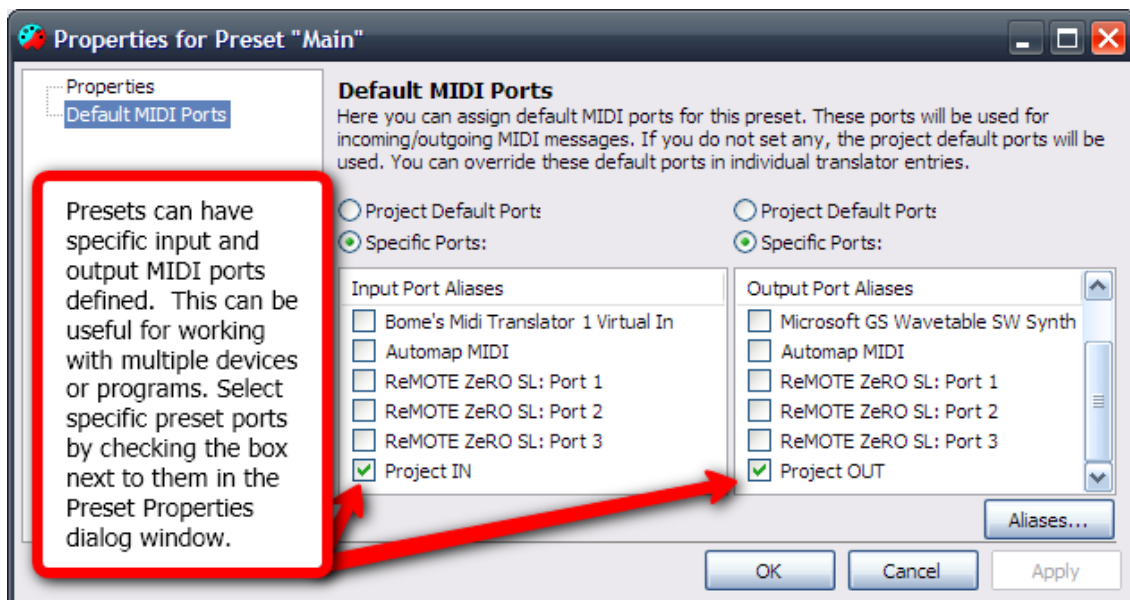
Preset	#	Name	#	Incoming Trigger	Outgoing Action	Options
<input checked="" type="checkbox"/> Main	1	<input checked="" type="checkbox"/> Shift_Button_Down	1	CC #16: RPN LSB, 127	activate only preset "Shift"	
<input type="checkbox"/> Shift	2	<input checked="" type="checkbox"/> Shift_Button_Up	2	CC #16: RPN LSB, 0	activate only preset "Main"	
<input checked="" type="checkbox"/> [always active]	3					

The 'Edit Translator Entry Shift\_Button\_Down' dialog box is open, showing the 'Outgoing [F8]' tab. The 'Select the Outgoing Action type:' dropdown is set to 'Preset Change'. The 'Deactivate all other presets (except the always active)' checkbox is checked. A red box highlights this checkbox, and a red arrow points to it from a text box that says 'A preset may be activated or deactivated as an outgoing action'.

### *changing presets via translator*

## > DEFAULT MIDI PORTS

Individual presets can also have Default Ports defined. Default Input MIDI ports are useful if you have multiple MIDI IN devices you wish to manage independantly from one another, or you have MIDI hardware on the same MIDI channel that you wish to separate. Default Output ports are useful if you are working with multiple software programs or outboard devices, and you wish to devide and manage translator data amongst them. Defining Default Ports for a preset is simple. First, right click on the preset you wish to change and select Properties to enter the Preset Properties screen again. Select the Default MIDI Ports page, and enable the checkboxes next to the MIDI IN and MIDI OUT ports you wish to use for the preset. Also note that you can select "Project Default Ports" to use the default project ports as defined in the Settings screen.



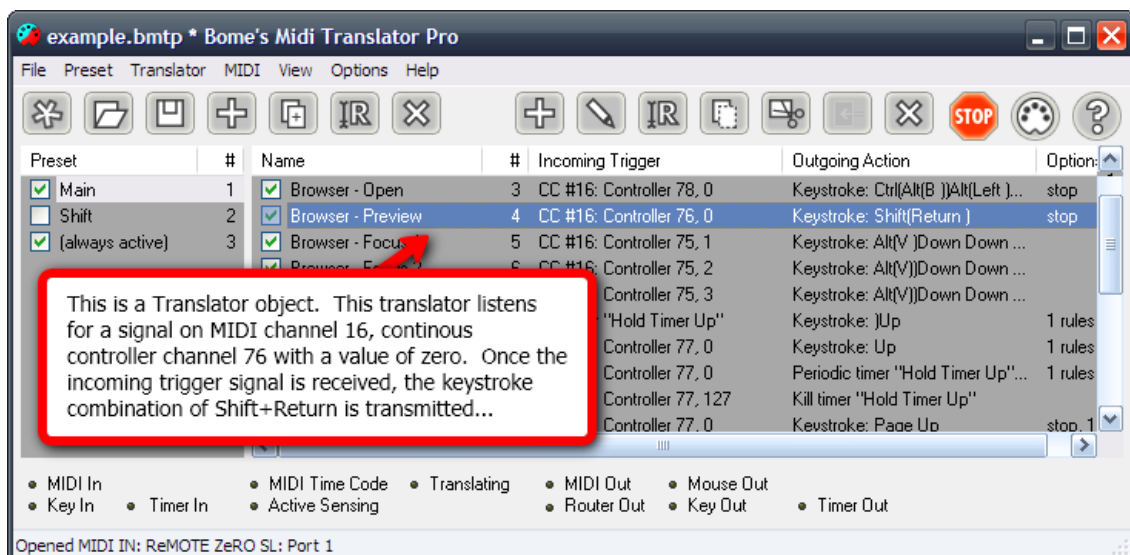
***default project MIDI ports***



# 6 Translators

## > TRANSLATORS OVERVIEW

Translators comprise the core functionality of Bomes MIDI Translator. In simple terms, translators listen for an 'incoming action', optionally do some processing on the incoming action (see the Rules section of the manual), and then optionally output an 'outgoing action.' Translators are limited to a single incoming action and outgoing action, but you can easily create multiple translators with identical incoming or outgoing actions depending on your needs. Translators can, however, transmit and receive on multiple MIDI ports, making interfacing with different devices easier. Also, the routing flexibility of outgoing actions and rules allow for a lot to be accomplished with a single translator.



### **typical midi translator setup**

## > TRANSLATOR OPTIONS

There are three main settings in the 'Options' screen for individual translators.

- **Name**

This is the simple descriptive property of the translator. It does not have any value other than for reference in presets and activation rules. The Translator name is not a unique value, therefore multiple translators can have the same name. It is recommended that the

Translator name be something simple that will make identifying multiple translators in large templates more easy.

- **Active**

This value determines whether the translator is actively being processed (listening for defined incoming action) or whether it is disabled (and therefore ignoring incoming actions). This parameter can also be changed from the main MIDI Translator interface.

- **Stop processing with this translator**

If this is enabled, successful completion of this translator's Outgoing Action will cause the rest of the translators in the current preset to be ignored. This is useful for multiple-part presets that have different processes depending on different defined actions.

A simple and useful use case for *Stop Processing* is to ignore the *Note Off* MIDI Message from your MIDI keyboard. For example, if you want to trigger a translator only when pressing the key, but not when releasing it, then create two translators:

1. In the first translator, specify the *Note On* MIDI message with 0 velocity (which is often used for *Note Off*), e.g. 90 45 00 with *Stop Processing* checked and **no outgoing action**. This translator will effectively "swallow" any Note Off (=key release) messages.
2. The second translator has as incoming MIDI message e.g. 90 45 qq. The outgoing action is the action that you want to do when *pressing* the key. Here, it does not matter if *Stop Processing* is checked or not.

With this trick, you get a translator (the 2nd one) which only reacts on key down.

Note: you can achieve the same with the Rules.



**translator options screen**

## 6.1 Incoming Triggers

### > INCOMING TRIGGERS

Incoming Trigger actions are events that Bomes MIDI Translator can detect and act on. Bomes MIDI Translator can recognize five different types of incoming actions.

- **MIDI Message**

MIDI Messages are defined as hexadecimal values in Bomes MIDI Translator. The hex values are broken up into three parts, the first being the associated MIDI channel, the second being the controller number, and the third being the controller CC value. The most simple way of defining a new MIDI incoming action is to use the 'capture midi' feature (covered in the Quick Start guide) to capture the incoming MIDI information while you are pressing a keyboard key, turning a knob or otherwise. If you need to assign multiple values to a single translator, variables can be used.

Translators can be assigned to listen on a specific port if so required. Select the MIDI ports you wish for the translator to listen on by clicking on the 'On Specific Port(s)' radio button and putting a checkmark next to each port you would like your translator to listen on.

Translators also have two additional options regarding MIDI messages as Incoming Actions:

- **Swallow MIDI Message / Do Not Route** - This option indicated that you wish NOT to retransmit the incoming MIDI message on to the MIDI Thru destinations specified in the Project Routing connection screen.
- **Description of MIDI Message** - This option allows you to write a short description of the Incoming Action for the translator that will show up in the main program interface. This can make complex translator setups much easier to work with and navigate.



### ***translator incoming – MIDI action***

Local variables are automatically defined in incoming MIDI message actions if the 'capture midi' checkbox is enabled and multiple values of any one control are detected. Variables are covered more completely in the Rules section. You can use local variables (defined as double-letter pairs such as 'oo' and 'xx' in the rules section to modify the incoming values to fit your needs. You can also use global variables (defined as variables beginning with the letters 'g' or 'h') to pass variables back and forth between translators.

Bome's MIDI Translator's online help system (click Show Help in the bottom left corner of the Translator window) includes a useful HEX<->DECIMAL translator that will assist in making proper MIDI assignments.

The following are some sample incoming MIDI string examples, along with description. For further instructions on Using Variables go to Using Rules and Variables.

<b>ACTION STRING</b>	<b>TYPE</b>	<b>CHANNEL</b>	<b>CONTROLLER</b>	<b>VALUE</b>
9F 6F oo	<b>Note On</b>	16	111	ANY
BA ww 7F	<b>Controller</b>	11	ANY	127
B4 xx pp	<b>Controller</b>	5	ANY	ANY
C4 nn	<b>Program Change</b>	5	ANY	NONE

#### o **Keystroke**

Keystroke-based incoming translator actions are defined by single keys and can be separated either by complete keypress cycles (down and up) or by individual 'Key Down' or 'Key Up' events. Having separate key 'Up' and 'Down' events lets the user define separate translators that occur when a key is pressed down (for example, to

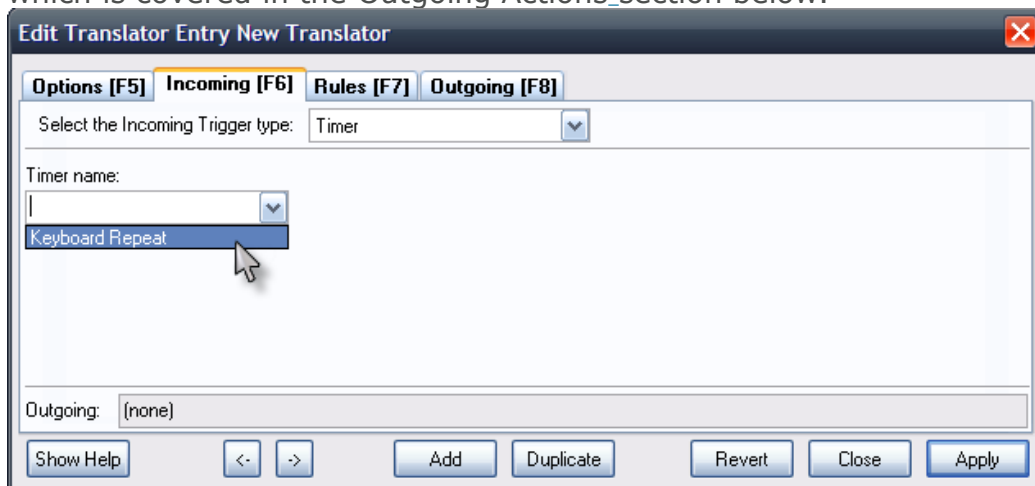
enable an audio effect), and when the key is released (disable previously activated effect).



***translator incoming – keystroke action***

### Timer

Incoming Timer actions are events that will trigger multiple times automatically depending on the Timer settings. Incoming Timer actions must be tied to an already-existing Outgoing Timer Action in order to work properly. Defining Incoming Timer actions is very simple - the only option is the selection of which Timer to use. Most of the Timer options are defined in the Outgoing Timer Action screen, which is covered in the Outgoing Actions section below.



***translator incoming – timer action***

### Preset Change

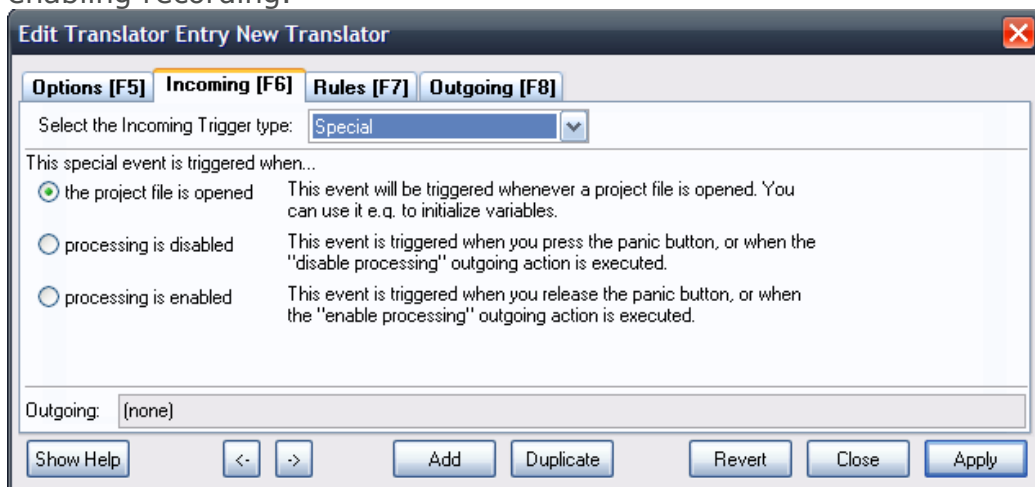
Preset change incoming actions are actions that are activated when the preset is changed through some means. Preset Change Actions are useful for "one-off" type of actions that only occur once at the very beginning of the preset change. These actions are often 'reset' actions that could either redefine global variables or reset controllers

to default values.

### ***translator incoming – preset change action***

- **Special**

Special Actions are actions that occur in response to program events. 'Startup' actions are best defined with Special actions such as when the project file is loaded. Certain actions that need to take place first before any other actions, and only once are best defined by using Special actions, such as Full-Screening your application and perhaps enabling recording.



### ***translator incoming – special action***

## 6.2 Outgoing Actions

### > OUTGOING ACTIONS

Bomes MIDI Translator can natively output seven types of outgoing actions, as well as function with translators that are composed solely of rules with no defined outgoing action.

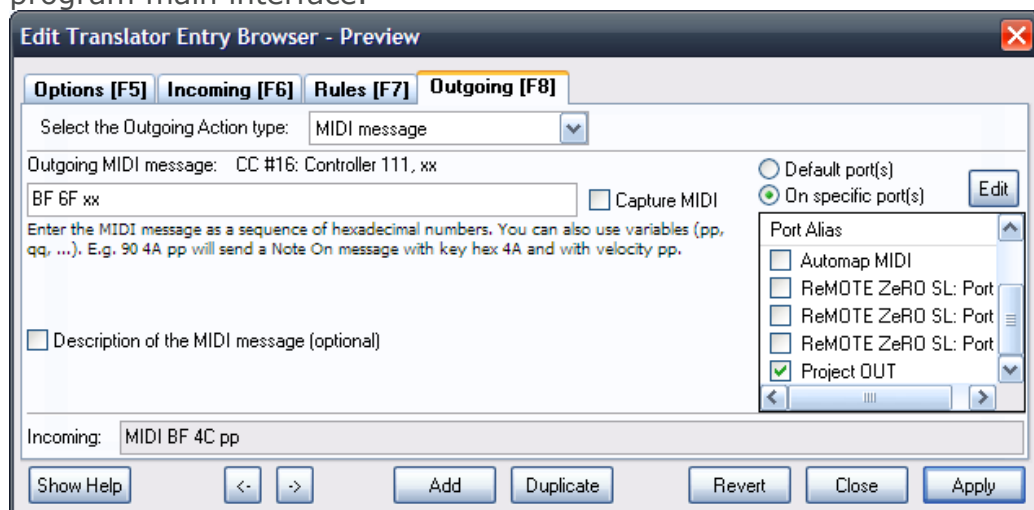
The Outgoing Actions supported are: **MIDI Message, Keystroke Emulation, Mouse, Timer, Preset Change, Special, WM\_Messages** and **None**.

#### o **MIDI Message**

One of the most powerful features of Bomes MIDI Translator Pro is the capability to translate MIDI messages and apply rules to them. In the screenshot below, the output value of the translator's outgoing MIDI message includes an 'xx' variable statement, meaning that any number of rules could have been used to assign value to this variable depending on many factors.

Outgoing MIDI actions can be transmitted on the Project Default MIDI ports, or on specific MIDI ports unique to the individual translator object. To select specific MIDI ports for transmitting outgoing MIDI data, simply select 'On Specific Port(s)' from the radio button selector on the right and put a check next to the box for each MIDI OUT port you wish to use.

Also, an optional Description can be entered for each Outgoing Action that will give a plain text description that can be viewed from the program main interface.



***translator outgoing – MIDI***

### • **Keystroke Emulation**

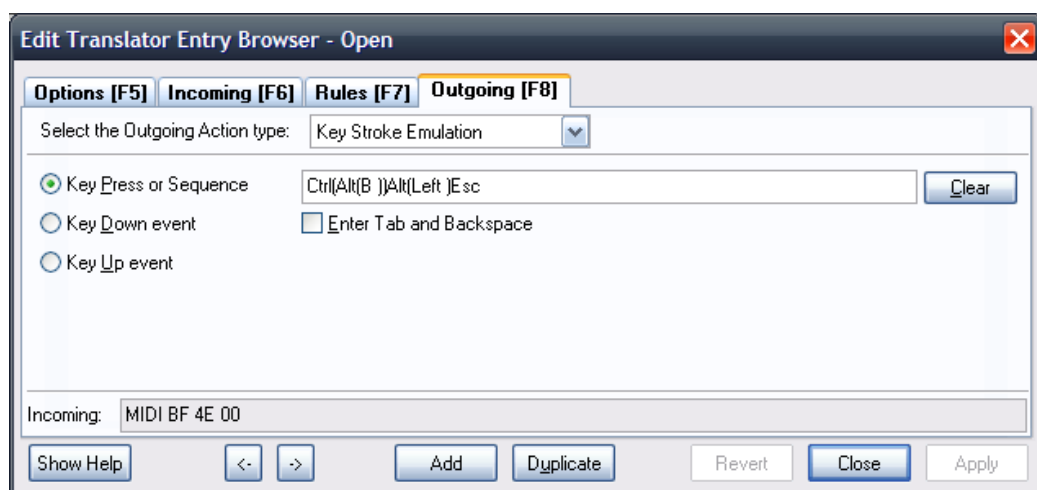
Outgoing keystroke emulation actions are the cornerstone of Bomes MIDI Translator. Incoming actions can be translated to any combination of keystrokes, for use in complex macro routines to control 3rd party program functions. The best place to start with programming keyboard combination macros is to consult your software's instruction manual and determine what keyboard shortcuts are available to you.

Keyboard Emulation outgoing actions can either be whole key press/key combination events, or individual Key Up / Key Down events. For outgoing keyboard commands that require the use of either the 'Tab' or 'Backspace' keys, a checkbox is provided that overrides the default behavior of those keys and allows them to be entered into the sequence box.

If you want to execute a key event, and have the key repeated as long as you press a key on your MIDI Keyboard, you will need two translators:

1. The first translator's incoming MIDI message is e.g. 90 45 00 (i.e. it reacts on the Note Off event) with *Stop Processing* activated. As outgoing Key Stroke use "Key Up Event". Enter the letter to be "typed" in the text field (e.g. "X").
2. The second translator's incoming MIDI message is e.g. 90 45 qq (i.e. it will react on all Note On messages). As outgoing action, use "Key Down Event". Type the same letter in text field as for the Key Up event (e.g. "X"). Enable the Key Repeat box.

Now press the key on your MIDI Keyboard down. As long as you press it, the letter "X" will be typed. Once you release the key, the keystroke emulation stops, too.



***translator outgoing – keystroke emulation***



- **Mouse**

Included in Mouse-type outgoing actions are multiple types of actions: **Movement**, **Abs Position**, **Button Clicks** and **Wheel**.

Each of those type of mouse actions includes settings for a variety of different parameters that can be manipulated to control the system mouse.

- **Movement**

Movement events can be transmitted to the system mouse pointer using this mouse event type. Move movement is defined in 'Mickey's' which are the smallest measurement of mouse movement. Movement is defined using two text boxes, one for Up/Down movement and the other for Left/Right movement. For Up and Left movement, use negative numbers. For Down and Right movement, use positive numbers.

- **Abs Position**

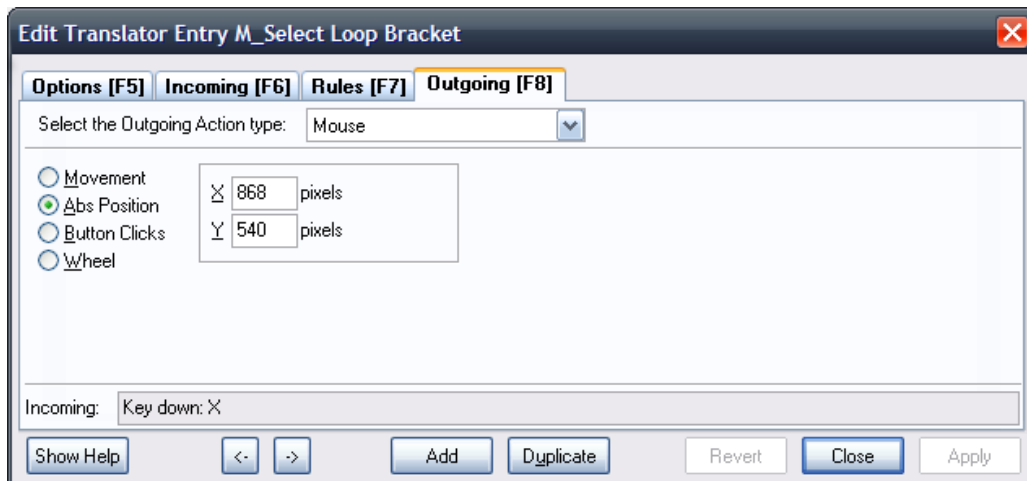
Absolute positioning type movement allows for the mouse to be moved to a pre-determined point on the screen. Absolute positioning is measured in pixels, similarly to how screen resolution is set. Use your system's screen resolution as a guide for setting absolute positioning (example: if you have a 1024x768 screen, the exact center of the screen would be absolute position 512x384).

- **Button Clicks**

Left, middle and right button clicks can be emulated using this mouse outgoing action. Button click events are usually positioned after mouse movement events, allowing the user to position the mouse properly before clicking or dragging, etc. A Button Click event is comprised of a complete Mouse Up/Mouse Down event, unless otherwise selected in the checkboxes provided.

- **Wheel**

Mouse wheel events can also be transmitted. Mouse wheel events can either be Forward (away from you) or Backward (towards you).

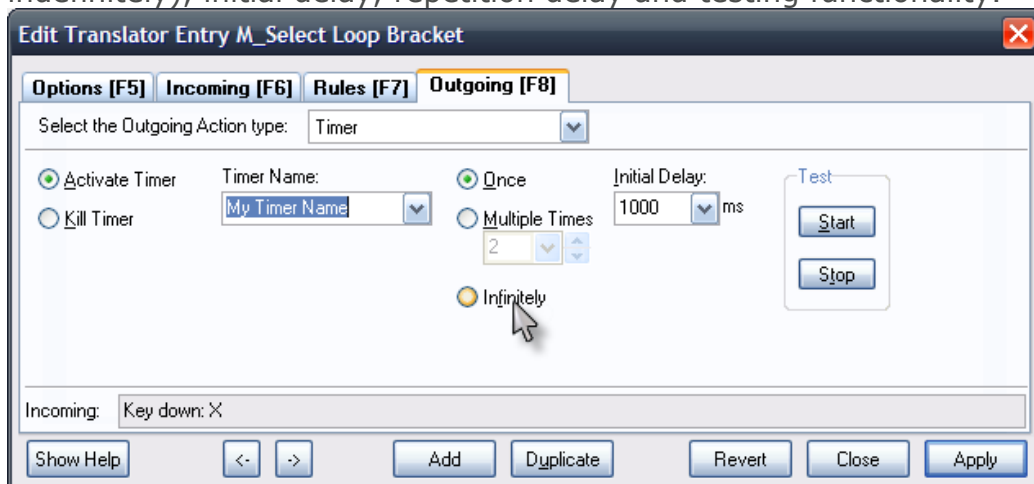


***translator outgoing - mouse***

### • **Timer**

There are two main types of Timer actions - Activate Timer and Kill Timer. Timers can be instantiated by selecting them as an outgoing action, and setting their appropriate repeat times and other options. Timers are usually associated with other translators that have the Timer name as their incoming action. For instance, if you wanted to repeat the 'Up' arrow key as long as a condition is met, you would first create an outgoing Timer action that set the repeat rate, then you would create a new translator which would output the keyboard emulation for the 'Up' arrow key - using your existing Timer translator as the incoming action.

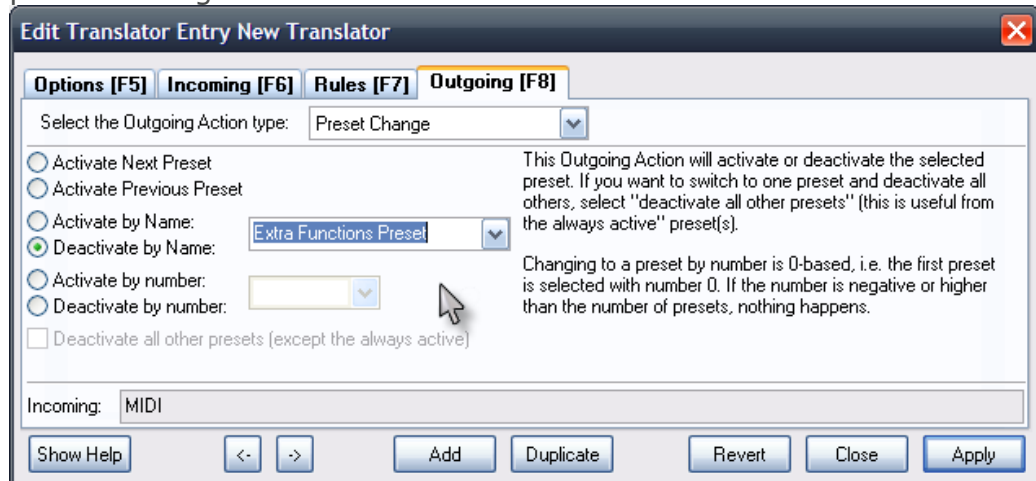
Timer parameters include repeat occurrence (once, multiple times, indefinitely), initial delay, repetition delay and testing functionality.



***translator outgoing - timer***

- **Preset Change**

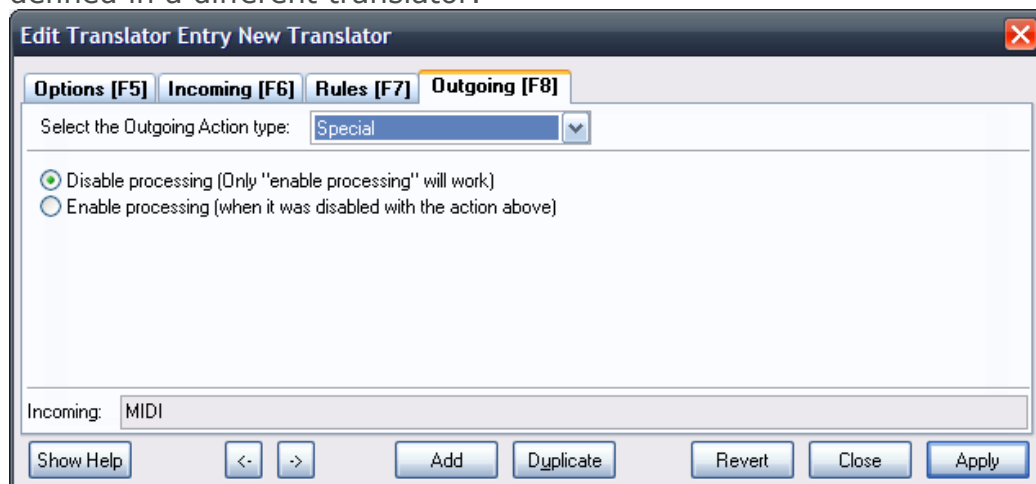
Preset change outgoing actions have the capability of managing project presets, either by activating certain ones, deactivating certain ones, or cycling through them all as a set. The '[always active]' preset is normally excluded from Preset Change outgoing action preset management.



***translator outgoing – preset change***

- **Special**

The Special type outgoing action has only two functions - disabling MIDI Translator Processing and enabling MIDI Translator Processing. The program as a whole can be bypassed using the 'Disable Processing' outgoing action, preventing any of the translators from activating. If MIDI Translator is disabled in this way, the only way to re-enable it is by using the 'Enable Processing' outgoing action defined in a different translator.



***translator outgoing - special***

- **WM\_Messages**

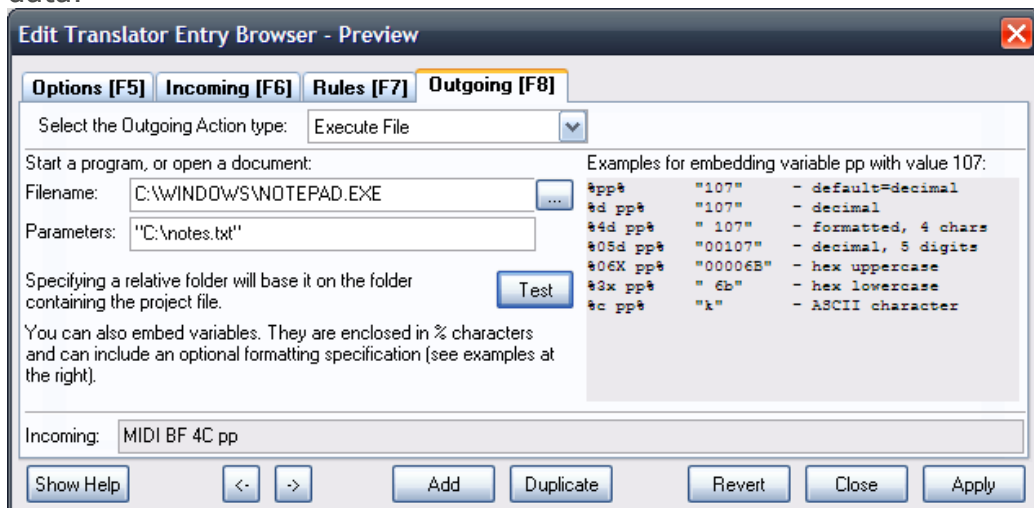
Bomes MIDI Translator is capable of outputting program-specific windows wm\_ messages. The breadth of wm\_ messages is beyond the scope of this manual. Please consult your program or systems programming manual for a list of appropriate wm\_ messages.



**translator outgoing - wm\_messages**

- **Execute File**

The Execute File outgoing action type lets you define an executable file with parameters to run as a resultant outgoing action. Enter the executable file name with path in the File Name text box, along with any parameters you would like to pass to it in the Parameters text box (usually surrounded by quotation marks). You can use the Test button to test execute your program after entering in the required data.



**translator outgoing - execute file**

## 6.3 Rules and Variables

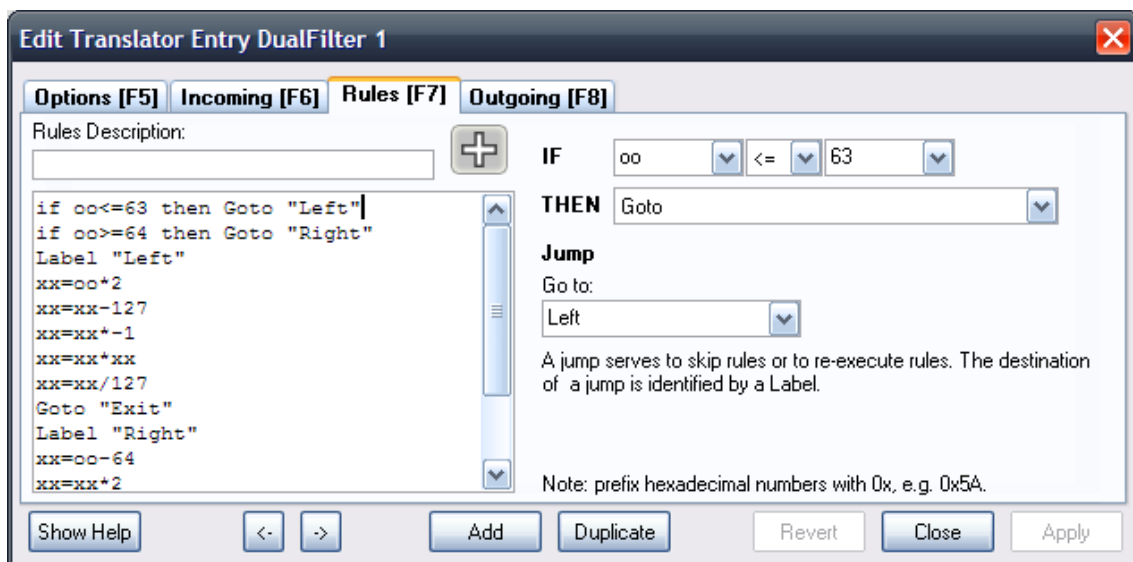
### > RULES AND VARIABLES OVERVIEW

Translators are comprised of three main sections: incoming actions, outgoing actions, and rules. This part of the documentation covers Rules, how to use and to be related to Variables, and what they can be used for.

Rules are basically simplified programming steps that take data from the incoming actions, or global variables, and can affect what happens with the translator's outgoing action. Rules use variables to pass data back and forth between the incoming action and the outgoing action of a translator.

Rules are normally processed from top to bottom: the rule on the first line is processed first, followed by the second and so on. Rules can also use Labels and Jumps to direct programming flow. Existing rules can be moved up and down in the rules box by first selecting the rule, then clicking the 'Up' or 'Down' buttons next to the Rules list.

Variables can either be defined in an incoming action, or through the Rules section of a translator. Incoming actions defined with a variable as part of the action will pass the variable on to the rules section to be processed and potentially used as a global variable or passed on to the outgoing action.



### ***translator rules***

## > RULE TYPES

There are seven types of rules in Bomes MIDI Translator. Three of these rules (Assignment, Expression and Conditional) deal directly with variables, changing their values and operating off of conditionals determined by variable values. Two of the rules (Jump and Label) are used for directing the flow of the rules programming, allowing you to make 'Functions' for complex rule sets. The other two rules (Exit Rules and Execute, Exit Rules and Ignore) are especially useful for conditionals, only enabling the outgoing action when a specific condition is met.

These seven rules are described in detail below:

- **Assignment** (*example: pp = 20, ga = oo*) - This rule type allows a straight assignment of a variable's value to a specific number or another variable's value. The variable you wish to assign is chosen on the left side of the equation from a drop-down box, while the source value or variable is selected or entered on the right side. Assignment rules are useful for assigning an input local variable to a global variable. They can also be used for assigning a specific value to an outgoing action depending on a conditional.
- **Expression** (*example: pp = 30 + qq, h0 = 128 / oo*) - Expression rules use basic arithmetic (plus:+,minus:-,multiply:\*,divide:/) or binary operators (AND:&,OR:|,XOR:^) to enter a value into a variable. A variable is selected from a drop-down box on the left side of the equation, while the two variables and/or numbers and operator are selected on the right side of the equation. Expression rules are useful for doing basic operators on incoming values to, for example, increase them or decrease them parametrically.
- **Jump** (*GOTO ...*) - Redirects the processing of the rules to a 'Label' point. A Label name may be typed in directly, or an existing label jump destination may be picked from the drop-down box.
- **Label** (*jump destination*) - This is the destination point in the rules processing that you would like a jump point to redirect to. Labels are useful for defining functions in your rules sets.
- **Exit Rules and execute Outgoing Action** - This is a direct action. Upon processing this rule, the Translator will immediately stop processing the rules and execute the Translator's Outgoing Action. These rules are commonly found coupled with Conditional rules and Labels to create complex processing statements.
- **Exit Rules and ignore Outgoing Action** - This is a direct action. This rule will immediately stop processing the rules set, but will NOT execute the outgoing action. This is useful for making Translators that ONLY execute when certain conditions are met.

- **Conditional** (*example: IF pp = 10 THEN ..., IF gc >= xx THEN ...*) - Conditional rules allow you to specify that a rule will ONLY execute if certain conditions are met. Conditional rules are constructed as follows:

**IF (value/variable) (==/!=/>=<=>/<) (value/variable) THEN**

If the preceding conditional is true, then one of the following actions is performed:

- **Assignment**
- **Expression**
- **Jump**
- **Skip Next Rule**
- **Skip Next 2 Rules**
- **Exit Rules, Execute Outgoing Action**
- **Exit Rules, Skip Outgoing Action**

Conditional Rules Operator Table

==	EQUALS	(true example: IF 10 = 10 THEN)
!=	DOES NOT EQUAL	(true example: IF 10 != 45 THEN)
>=	GREATER THAN OR EQUAL TO	(true example: IF 86 >= 45 THEN)
<=	LESS THAN OR EQUAL TO	(true example: IF 34 <= 34 THEN)
>	GREATER THAN	(true example: IF 10 > 4 THEN)
<	LESS THAN	(true example: IF 24 < 80 THEN)

## > VARIABLE TYPES

There are two main types of variables in Bomes MIDI Translator: Local variables and Global variables. Variables can be set either with incoming actions or with rules. Once set, a Local variable may be used for the duration of an active Translator, losing it's value once the Translator has finished processing. A Global variable, however, will retain it's value as long as Bomes MIDI Translator is running.

- **Local Variables** - Local Variables are defined by character combinations in the following range: **oo-xx** (*example: pp, ss, ww, etc...*). Local variables retain their value as long as the Translator is processing. Local variables are normally the most commonly used variables, and are useful for holding temporary values. Local variables can be used in incoming actions to pass, for example, a continuous controller value to the Rules section of a Translator, where it can then be processed and resent to the outgoing action.
- **Global Variables** - Global variables are defined by character combinations in the following ranges: **ga-gz/g0-g9, ha-hz/h0-h9, ..., na-nz/n0-n9** (*example: h4, kd, j0, nb, etc*). Global variables retain their value indefinitely. Global variables are useful for passing information between translators. One common use of global variables is to create a 'Shift' button on your controller, which can then control which translators are processed depending on the state of the shift control. Another common use of global variables is to 'Hold' a controller's value while a timer is running, allowing you to re-send that value when the timer is done processing.



## > USING RULES AND VARIABLES

One of the most useful way to use Rules and Variables in your Translator settings is in the translation of a velocity or cc value to another value. Variables may utilized in the mapping of an incoming MIDI action in a translator by changing the last value to a variable setting instead of a static value. Variables may be used in both incoming and outgoing translator actions, allowing values input into translators to be processed, and then sent on to the outgoing MIDI port while retaining full routing flexibility.

### Sample Incoming MIDI Actions

- **BF 29 oo** - This Incoming MIDI event will activate our translator on ANY value of the control **BF 29** (MIDI Channel 16, Control 41), and then pass the velocity or continuous control value on to the rules section of the translator for processing.
- **BF xx oo** - This Incoming MIDI event will activate the associated translator on ANY activity on MIDI channel 16, for ANY controller. Rules like this are useful for effecting a whole controller range on a MIDI channel, such as changing the velocity of a whole keyboard layout.

### Sample Outgoing MIDI Actions

- **ga 29 xx** - This Outgoing MIDI event will output the 'xx' value (most of the time, this will be a value calculated in the translator's rules section) on the 41 controller on MIDI channel 'ga'. In this example, the 'ga' variable can be predetermined by another translator, or tied to another MIDI control processed by Bomes MIDI Translator. For example, the 'ga' variable can be set by a knob on your MIDI controller, giving you the capability to select which MIDI channel the translator is actually transmitting to in real-time.

# 7 Settings

Settings for Bome's MIDI Translator allow the end user to modify the general behavior of the program.

Setup options for Bome's MIDI Translator are divided up into six general categories.

- Startup Options
- Appearance
- MIDI Ports / Aliases
- MIDI Router
- Virtual MIDI Ports
- Default MIDI Ports

Also, settings information can be imported and exported from Bome's MIDI Translator, allowing you to backup and restore your program settings, as well as allowing you to transport settings information to different computers running MIDI Translator.

## > STARTUP OPTIONS

The Startup section deals with how Bome's MIDI Translator initially starts. Many users may want Bome's to run seamlessly in the background of their computers without having to think about it. Having the program start up automatically with the operating system and then run hidden in the background can be preferable once you have all your Translators created.

- **Start Minimized:** If this option is enabled, Bome's MIDI Translator will automatically start in it's minimized state when run.
- **Autostart with Windows:** This option causes MIDI Translator to automatically start when Windows is booted.
- **Show Splash Screen:** Enable or disable the showing of the program splash screen when MIDI Translator is run.

## > APPEARANCE

The Appearance section of the settings window deals with how the program behaves in the system tray and how MIDI incoming and outgoing action messages are displayed in the Translator interface. Keep an eye on the program status icon in the system tray, it can be used to diagnose problems with your setup quickly.

- **Show Tray Icon:** This option turns the program tray icon on and off.
- **Animate Tray Icon:** This options enabled the program tray icon to display basic input/output routing information. Please refer to the Reference for more detailed information on this.
- **Minimize To Tray:** Enables the program to minimize to the system tray instead of the taskbar, freeing it from view on the system taskbar with any other programs.
- **Friendly Descriptions:** This enables or disables the display of 'friendly' messages in the Translator user interface. When Friendly Descriptions is disabled, only MIDI hexadecimal codes are displayed in the incoming and outgoing actions of translators. With this option enabled, the Incoming and Outgoing action fields of each Translator are converted to decimal, and have other helpful MIDI reference information inserted.

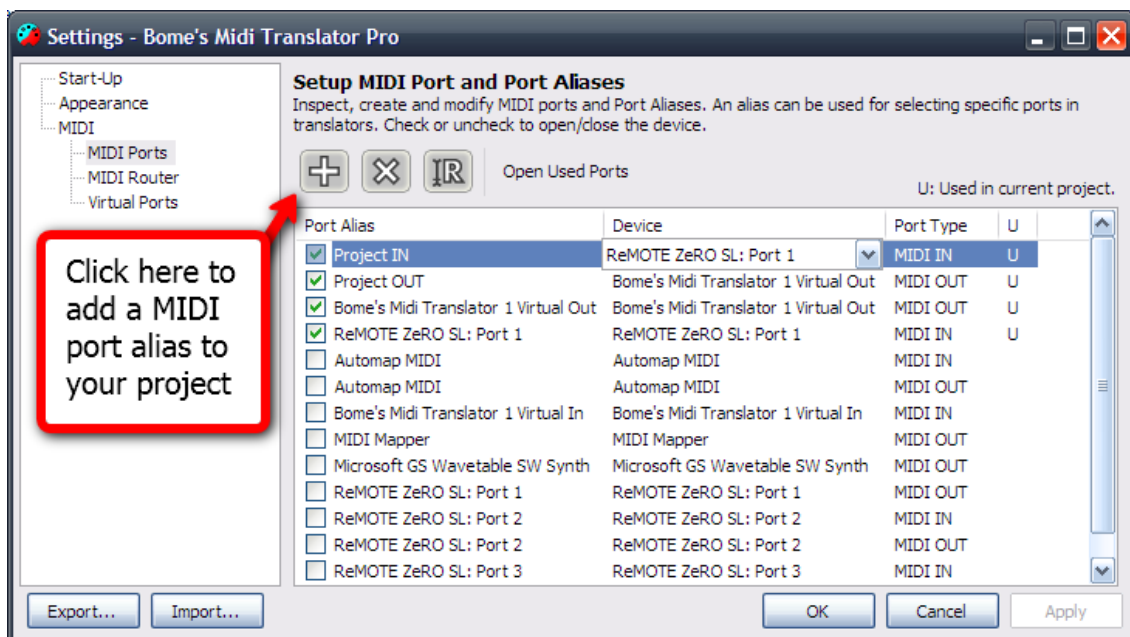
## > MIDI PORTS AND ALIASES

The MIDI Ports screen is where you set your incoming and outgoing MIDI ports, as well as where you specify project aliases. Simply mark the checkbox next to the MIDI IN and OUT ports you wish to use in your project and they will become available for use in translators.

All available MIDI ports are automatically assigned a same-named alias when MIDI Translator is started. Any project that is opened from another user with different MIDI hardware will STILL have entries for their unique MIDI incoming and outgoing settings. Using the MIDI ports screen, one can reassign the MIDI assignments of a project to point to any combination of hardware and/or software MIDI ports.

To create a new MIDI port alias, simply click on the Add Alias button above the MIDI port listing. A new alias will be created, which can be named anything and assigned to any hardware or software MIDI port.

MIDI Aliases are useful for working with multiple input and output MIDI sources, such as connecting multiple hardware MIDI devices, or connecting one or more hardware MIDI devices to multiple software inputs.

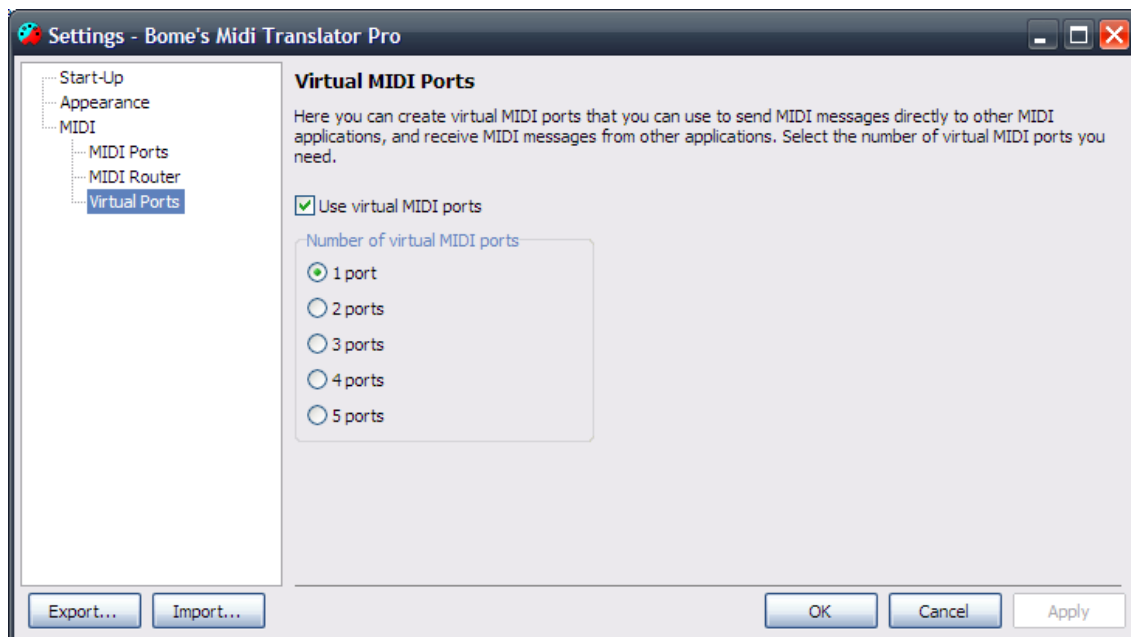


### ***MIDI port assignment***

## > MIDI ROUTER

The MIDI Translator Router is a powerful but simple way for MIDI Thru connection to be made between MIDI Interfaces. All detected MIDI IN interfaces are displayed on the left column of the screen, while detected MIDI OUT interfaces are displayed on the right side. Simply drag and drop a new line between the two interfaces you wish to create a THRU connection on and one will be created represented by a solid line between them.

MIDI THRU interface connections can span from one MIDI IN connection to many MIDI OUT connections.

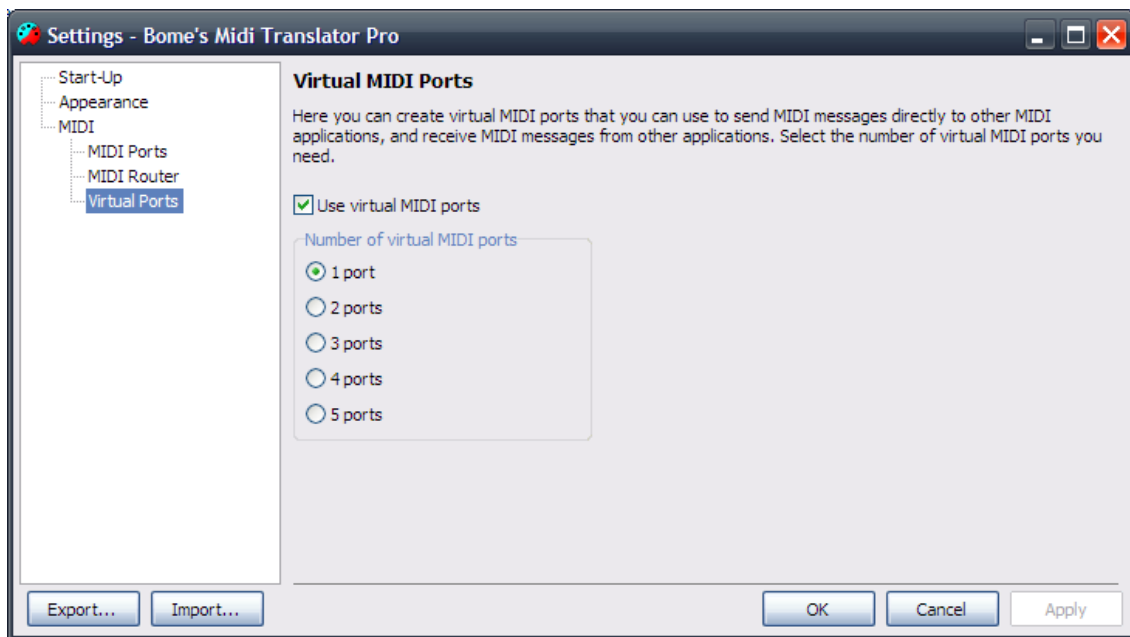


### ***MIDI router (MIDI THRU)***

## > VIRTUAL PORTS

Starting with version 1.7, Bome's MIDI Translator now includes built-in virtual MIDI port drivers that enable the end user to link up MIDI controllers and applications seamlessly without using any 3rd party software. Up to five sets of virtual MIDI ports may be installed at any one time, allowing expanded control and flexibility of your MIDI routing.

Simply choose how many pairs of virtual MIDI ports you require in your processing (1 virtual MIDI port = 1 MIDI IN, 1 MIDI OUT), select Apply and follow the on-screen hardware installation instructions similar to the product installation covered in the Quickstart guide.



### ***virtual MIDI ports***

## > AUTHOR INFO

The Author Info screen lets you attach information to your Bome's MIDI Translator project file that will travel with the file if you decide to redistribute it. This section is particularly of use if you are sharing project template files with other users.

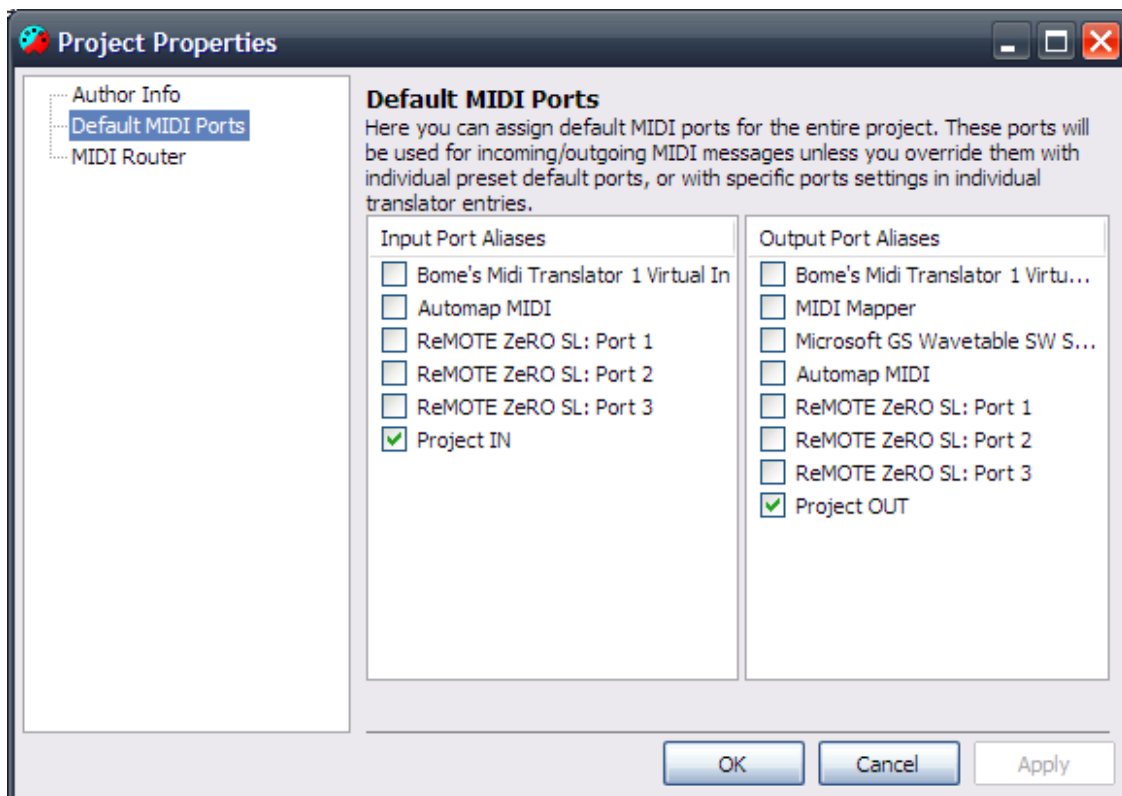
Information collected in the Author Info page includes;

- **Author Name**
- **Author Contact Info**
- **Comments**

## > DEFAULT MIDI PORTS

The Default MIDI Ports settings screen lets the user specify the incoming and outgoing MIDI ports that are used by the project when it is opened. This functionality is useful when transporting project files between computers that may have different MIDI controllers. Project MIDI port aliases can be created in the MIDI Ports / Aliases screen and selected in the Default MIDI Ports screen to ensure that rules created for one MIDI device can be linked to another easily.

The Default MIDI Ports screen can be accessed via either the program menu by navigating to File / Project Properties / Default MIDI Ports.



### ***default MIDI ports***

## > EXPORT / IMPORT SETTINGS

The MIDI Translator settings Export/Import feature allows a user to backup their settings to a .bmts file for restore at a later time.

- **Export:** With this function, you export your settings to a .bmts file. These settings include:
  - Window size and position
  - Selected translator preset
  - All program options
- **Import:** After saving your settings with the Export function, you can restore the settings again with the Import settings function in the Settings menu. This is particularly helpful if you want to transfer your settings to another computer, give them to a friend, or you have to reinstall your OS.



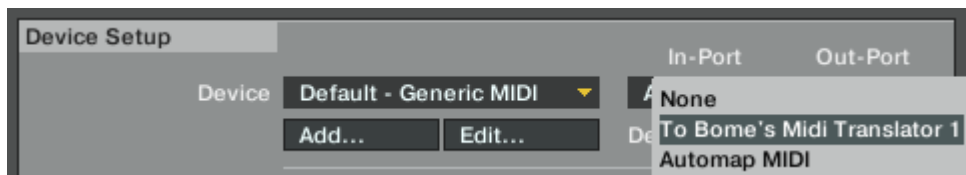
# 8 Usage Examples

## 8.1 Traktor / Ableton Live Sync

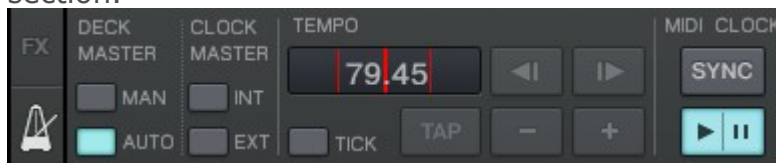
Beginning with Bome's MIDI Translator v1.7 and above, a virtual MIDI port system is included as an optional install with the application.

Virtual MIDI ports are available for use as unidirectional MIDI ports,

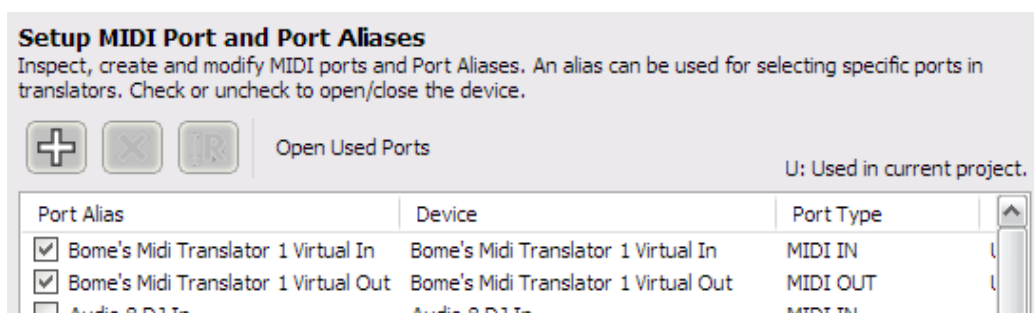
1. The first step is to select the Bome's virtual port as the MIDI output in your MIDI clock source application. In this example we're using Traktor to act as the MIDI clock master. Select "Bome's MIDI Translator 1" as the Out-Port in the program preferences.



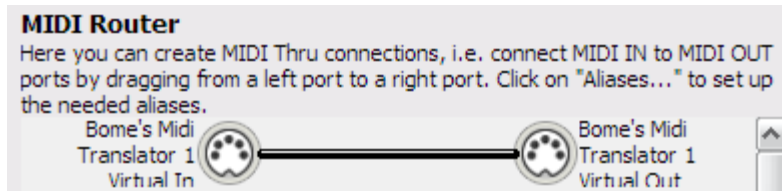
Make sure you have the program set to transmit the clock signal by going into the MIDI Clock category of the preferences dialog and making sure "Send MIDI Clock" is checkmarked. Also make sure that you have MIDI clock playing by accessing the metronome section of the user interface and clicking the "Play" button in the MIDI CLOCK section.



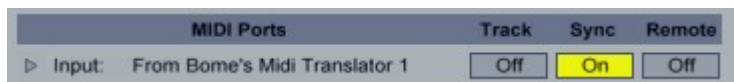
2. The next step is to configure Bome's MIDI Translator to activate the virtual ports we wish to use coming IN, going OUT and then to link them via the program MIDI Router. Activate the Virtual MIDI port OUT and IN devices by going into Settings / MIDI Ports and putting a checkmark next to each MIDI device.



Next you'll need to make a link between the IN device and the OUT device in the MIDI Router. Open the MIDI Router by clicking on Settings / MIDI Router / Project Properties. Drag a link between Bome's MIDI Translator 1 Virtual IN and Bome's MIDI Translator 1 Virtual OUT. This will tell MIDI Translator to pass all data from Traktor thru to our MIDI clock destination source Ableton Live.



3. The final step is to activate our MIDI clock control in our destination application. In Live's preferences dialog window, select the MIDI Sync category and turn Sync on for "Input: From Bome's MIDI Translator 1" to tell Live to receive MIDI clock from MIDI Translator.



# 9 Reference

## > TERMINOLOGY

- **Project:** A Project is a collection of Presets, grouped by a set of common attributes such as MIDI in/out/thru settings, appearance settings, and other common application-wide attributes.
- **Preset:** [ref:presets] A Preset is a collection of Translators. There are individually named Presets, which may be switched and activated separately, as well as an '[always active]' preset, which is perpetually active.
- **Translator:** [ref:translators] A Translator is an individual "rule" defining the translation of a single 'Incoming Action' event through to a single 'Outgoing Action' event.
- **Name:** The Name refers to the unique name given to an individual Translator.
- **Incoming Trigger:** [ref:translators:incoming] An Incoming Trigger event is an event that causes an associated Translator to start. Incoming Trigger events can be one of the following:
  - **MIDI message** [ref:translators:incoming:midi]
  - **Keystroke** [ref:translators:incoming:keys]
  - **Timer** [ref:translators:incoming:timer]
  - **Preset Change** [ref:translators:incoming:preset]
  - **Special** [ref:translators:incoming:special]
- **Outgoing Action:** [ref:translators:outgoing] An Outgoing Action event is an event associated with a translator that has been activated. Bomes MIDI Translator will generate an Outgoing Action event depending on the Translator Rules, and if certain criteria are met. An Outgoing Action event may be one of the following:
  - **MIDI message** [ref:translators:outgoing:midi]
  - **Key Stroke Emulation** [ref:translators:outgoing:keys]
  - **Mouse** [ref:translators:outgoing:mouse]
  - **Timer** [ref:translators:outgoing:timer]
  - **Preset Change** [ref:translators:outgoing:preset]
  - **Special** [ref:translators:outgoing:special]
  - **WM\_Messages** [ref:translators:outgoing:wm]
  - **Execute File** [ref:translators:outgoing:execute]
- **Timer:** A Timer is an internal function of Bome's MIDI Translator that will generate repeated Incoming Triggers depending on the Timer properties.
- **Options:** [ref:translators:options] The options available to change the behavior of a Translator object. Options pertaining to a Translator event are comprised of the following:
  - **Name** [ref:translators:options:name]
  - **Active** [ref:translators:options:active]
  - **Stop Processing** [ref:translators:options:stop]

## > KEYBOARD SHORTCUTS

### Global:

**Shift+Esc:** MIDI Panic - Stops all MIDI information immediately

**Ctrl-O:** Open Project - Opens an existing MIDI Translator project file

**Ctrl-S:** Save Project - Saves current project, prompting for a name if first save

**Ctrl-F12:** Save Project As... - Saves current project as a new file name

**Ctrl-F4:** Close Project - Closes current project, prompting for save if applicable

**Ctrl+E:** Show Edit Window - Shows the Translator edit window

### Preset/Translator Interface:

**Del:** Delete currently selected object

**F2:** Rename currently selected object

**Ctrl+A:** Select All objects in active viewing area

**Ctrl+D:** Duplicate selected object

**Ctrl+Up/Ctrl+Down:** move selected object up/down in list view

### Translator Interface only:

**Ctrl+C:** Copy Translator to clipboard

**Ctrl+V:** Paste Translator from clipboard

**Ctrl+X:** Cut Translator to clipboard

### Edit Translator Window:

**Ctrl+N:** Move to next Translator in preset

**Ctrl+P:** Move to previous Translator in preset

**Ctrl+A:** Apply edits to Translator

**Ctrl+R:** Revert edits to state prior to editing

**Alt+C:** Midi Capture (for MIDI as Incoming or Outgoing Action)

**Ctrl+PageDown:** Cycle to next tab

**Ctrl+PageUp:** Cycle to previous tab

**F5:** Jump directly to the General tab

**F6:** Jump directly to the Edit Incoming Action

**F7:** Jump directly to the Edit Rules (Pro+ version)

**F8:** Jump directly to the Edit Outgoing Action

## > COMMAND LINE SWITCHES

**/debug:** debugging output through OutputDebugString

**/nosilent:** more debugging output

**/nodebug /silent:** reverse the meaning

**/settings <filename>:** load .bmts file instead of loading settings from registry.

**/project <filename>:** load .bmtf file at startup

**/midiin <MIDI dev>:** select the named MIDI device on startup for input

**/midiout <MIDI dev>:** select the named MIDI device on startup for output

**/autostart:** add MT Player to hidden autostart group

**/noautostart:** remove from hidden autostart

**/bmidi <num ports>:** select the number of virtual MIDI ports you wish MIDI Translator to use

## > MENU REFERENCE

### o **File**

- o **Open:** Open Project - Opens an existing MIDI Translator project file
- o **Open Merge:** Merge Project - Opens an existing MIDI Translator project file and merges presets with open project
- o **Save:** Save Project - Saves current project, prompting for a name if first save
- o **Save As:** Save Project As... - Saves current project as a new file name
- o **Close:** Close Project - Closes current project, prompting for save if applicable
- o **Project Properties:** Opens the Project Properties screen
- o **Export Encrypted:** Export encrypted project file - only used in Author edition
- o **Exit:** Exit Bomes MIDI Translator, prompting for save if applicable

### o **Preset**

- o **New:** Create a new preset in currently open program template
- o **Rename:** Rename currently selected preset
- o **Delete:** Delete currently selected preset
- o **Duplicate:** Create an identical copy of currently selected preset
- o **Move:** Submenu - Move selected preset up/down/top/bottom
- o **Save As Text File:** Save currently selected preset as a readable text file for easy display
- o **Properties:** Enter preset properties screen to specify always on, default port properties

### o **Translator**

- o **Add:** Add new blank translator to currently active preset
- o **Edit:** Edit currently selected translator
- o **Rename:** Rename currently selected translator
- o **Duplicate:** Create an identical copy of currently selected translator in current preset
- o **Activate:** Enable currently selected translator to process incoming and outgoing events
- o **Deactivate:** Disable currently selected translator from process incoming and outgoing events
- o **Move**
  - o **Up:** Move currently selected translator up one entry in preset list
  - o **Down:** Move currently selected translator down one entry in preset list
  - o **Top:** Move currently selected translator to top of preset list
  - o **Bottom:** Move currently selected translator to bottom of preset list
- o **Cut:** Remove currently selected translator and place in program clipboard

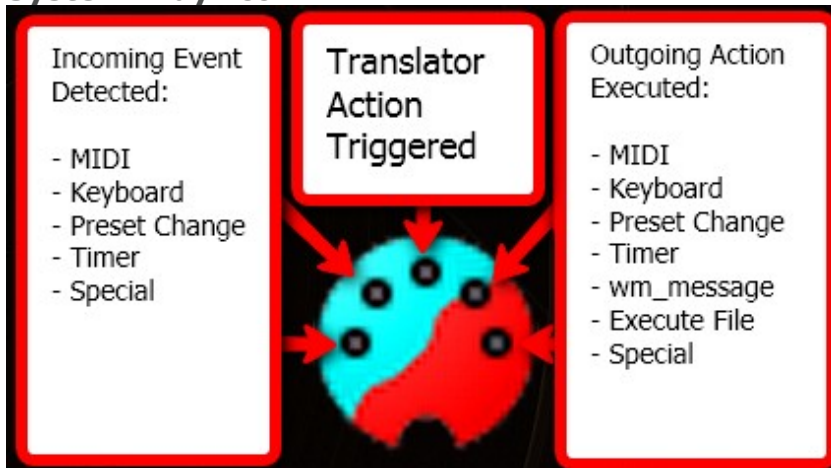
- **Copy:** Copy currently selected translator to program clipboard
- **Paste:** Create a new translator identical to translator previously 'cut' or 'copied' to program clipboard
- **Delete:** Remove currently selected translator from preset
- **MIDI**
  - **MIDI Ports:** Enable/disable MIDI ports for your project
  - **Project Default Ports:** Select the default MIDI ports for your project
  - **Routing (MIDI Thru):** Create MIDI Thru connections between MIDI interfaces
  - **Preset Default Ports:** Select distinct default ports for a specific preset
  - **Open Used Devices:** Open all MIDI devices that are used in the project and close all other ports
  - **Rescan MIDI Devices:** Rescan for open and closed MIDI ports
- **View**
  - **Log Window:** Show program log windows, which displays detailed information about incoming and outgoing actions, as well as letting you display all global system variables
  - **Event Monitor:** Display Activity Monitor on the bottom of the program window to monitor MIDI and program activity
- **Options**
  - **Supress outgoing keystroke when active:** Do not transmit keystroke messages when activated
  - **Ignore keystroke as Incoming Action when active:** Do not recognize keystroke input when activated
  - **Settings:** Open the Settings window for program configuration
- **Help**
  - **Help Overview:** Index of help resources available for Bome's MIDI Translator
  - **Purchase Midi Translator:** Open web browser page detailing purchasing information
  - **Other Bome Products:** Open web browser page showing other products from Bome.com
  - **About:** Show program license, version and copyright information

## > SYSTEM TRAY ICON

### ◦ System Tray Right-Click Menu

- **Exit:** Exit the program
- **MIDI**
  - **MIDI Ports:** Shortcut to MIDI Ports/Aliases screen
  - **Project Default Ports:** Shortcut to Project Default MIDI port assignment screen
  - **Routing (MIDI Thru):** Shortcut to MT Router
  - **Open Used Devices:** This command tells Bome's MIDI Translator to attempt to open the MIDI devices used in the project for exclusive access
  - **Rescan MIDI Devices:** Rescan your system for new/changed MIDI devices (use if you plug in a new MIDI device while Bome's MIDI Translator is running)
- **Presets:** Shows the current Activated/Deactivated state of the presets available in your Project, and allows you to selectively enable or disable them
- **Show/Minimize:** Minimizes or restores the main program interface

### ◦ System Tray Icon



*system tray icon status*