

MIDI Poet -- User's Manual  
1999-2007 Eugenio Tisselli  
<http://www.motorhueso.net>

## ***Introduction***

**MIDI**Poet**** is a software tool that allows the manipulation of text and image on a computer in real-time. It has two parts: Composer and Player, with which you can, respectively, compose and play pieces of interactive text and image on the computer. These pieces may or may not respond to external impulses, such as MIDI messages or the computer keyboard, and generate visual manifestations that involve the manipulation of the different attributes of text (content, font, position, size, etc), image (content, position, etc) and other elements and visual effects. Conceptually, MIDI**Poet** is based on the notion of fields of events: a set of potential behaviors of visual elements on the computer screen that happen or not depending on internal conditions or external manipulations.

## ***Logos***



MIDI**Poet** Composer



MIDI**Poet** Player

## ***Basic concepts***

**Composition:** This is the first step for the creation of a MIDIpoet piece. Fields of events are created by composing; they will be performed later to produce a visual piece. MIDIpoet Composer lets you generate these fields of events by the use of diagrams that are drawn directly on screen.

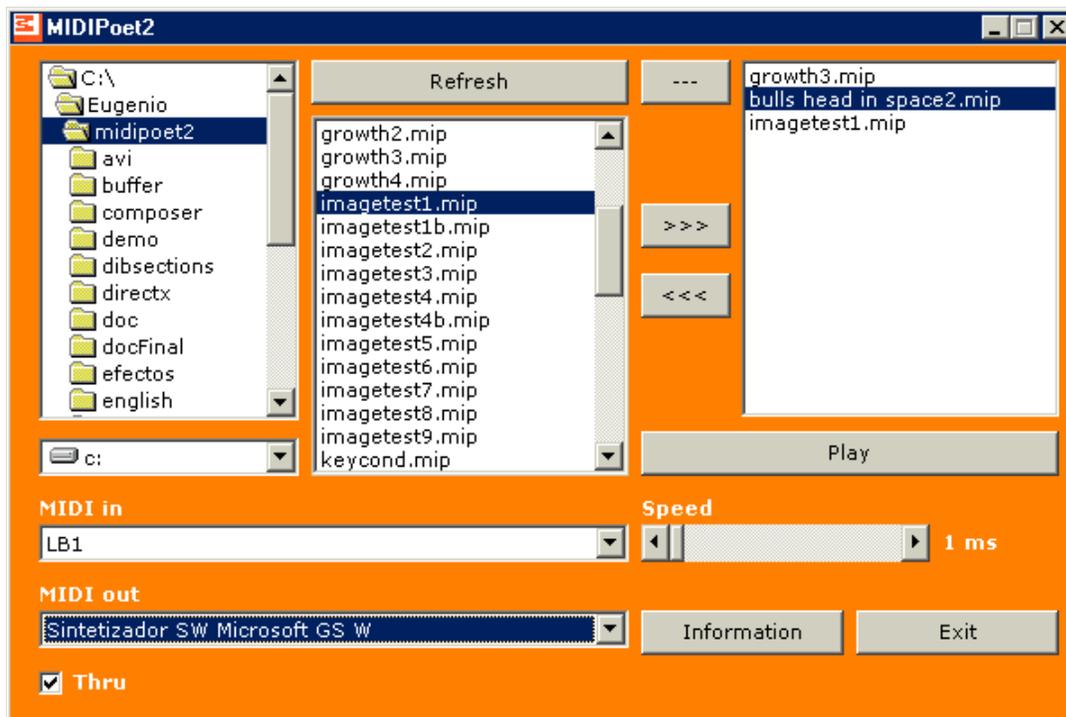
**Performance:** The visual result of a MIDIpoet composition is achieved through its performance. The MIDIpoet Player reads the fields of events of a composition and generates the visual elements (text and/or image) involved in these fields.

**Manipulation:** The concept of manipulation in MIDIpoet implies the generation of an impulse that activates one or more fields of events. MIDIpoet allows you to map different MIDI (or computer keyboard) messages to specific modifications of text and/or image.

## **The MIDI Poet Player**

The MIDI Poet player is a program that lets you play the image/text pieces previously composed using MIDI Poet composer, and saved as \*.mip files. Various pieces can be played sequentially by including them in a playlist.

1. Choose one or various compositions (\*.mip files) by specifying their path (left side of the screen), select them from the list at the center and include them in the playlist (right side of the screen) by using the >>> button.
2. If any of the chosen pieces uses MIDI, specify the input and/or output ports (bottom of the screen). Activate the "Thru" checkbox if you want MIDI Thru (MIDI input = MIDI output)
3. Regulate the speed at which the pieces will be played by using the "Speed" scroll bar.
4. Press the "Play" button to start playing the piece selected in the playlist.
5. To exit MIDI Poet player, press the "Exit" button.





## **The MIDI Poet Player**

**"Refresh" button:** MIDI Poet player and MIDI Poet composer are independent programs, that can be used simultaneously. While composing a piece, you can preview it with MIDI Poet player. The "Refresh" button updates the list of available pieces in the specified path. If a new composition has been saved using this path, it will not appear on the list until the "Refresh" button is pressed.

**">>>", "<<<" and "---" buttons:**

- >>> when a piece is selected from the central list, it can be included in the playlist by pressing the ">>>" button
- <<< a piece can be removed from the playlist by choosing it and pressing the "<<<" button
- --- to clear the playlist, use the "---" button

**MIDI Out and Thru:** MIDI Poet player can generate MIDI output; in the field labeled "MIDI out" you can choose the MIDI port to which the output messages will be sent during the performance. MIDI outputs can be generated in two ways: if the "Thru" checkbox is marked, all MIDI messages that arrive at the input port will be routed to the specified output port. If the "Thru" checkbox is not marked, MIDI outputs will only be the MIDI messages internally generated by a MIDI Poet piece.

**Speed:** The speed at which MIDI Poet pieces are played can be adjusted by varying the update frequency. By default, the speed is 1 millisecond (maximum speed)



## *The MIDI Poet Player*

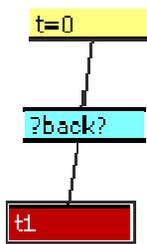
- When the "Play" button is pressed, a small "loading" screen will appear. Loading time depends on the complexity of the piece, and the number of images it includes.
- To start playing, press the "F5" key. This key also allows to pause/continue the playing of a piece at any moment.
- To increase playing speed, press the "Page up" key. To decrease it, press "Page down".
- To play the next piece in the playlist, press "F1" key. To play the previous piece, press "F2".
- To stop playing and return to the configuration screen, press "Esc" key.
- If a piece reacts to computer keyboard events, you must press "Ctrl" plus the desired key. For instance, if an event is activated by pressing the "a" key, you should press "Ctrl" + "a" simultaneously.
- If a piece accepts live text input, type it directly and press the "Enter" key when done.

Go to <http://www.motorhueso.net/midipeng> to find sample MIDI Poet pieces ready for download!

## **The MIDIcPoet Composer**

MIDIcPoet Composer lets you compose interactive pieces involving text and/or image. The composition is done by drawing logical diagrams that indicate the forms of occurrence of visual events, that is, they indicate what happens to text/image, and when and how it happens. Each of these diagrams is a tree formed by blocks and segments; blocks represent events, or conditions that must be fulfilled in order to make events happen. The diagrams or trees are hierarchical structures, with relationships established by segments that join the blocks. The final branches are the events (children); which are triggered when the conditions (parents) are fulfilled.

This is an example of a simple logical diagram, in which there are two conditions (yellow and blue blocks) that must be fulfilled so that the event (red block) can happen.



Diagrams are constructed in the following way:

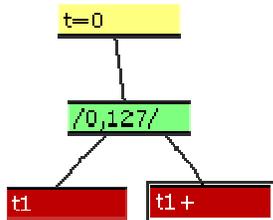
- A parent can have many children; a child cannot have many parents.
- A child can exist without parents. This means that the event represented by this block doesn't need any conditions to happen.
- Various separate diagrams can be drawn on the same composition. The composition will be the sum of all drawn diagrams.

# M<sup>i</sup>c The MIDI Poet Composer

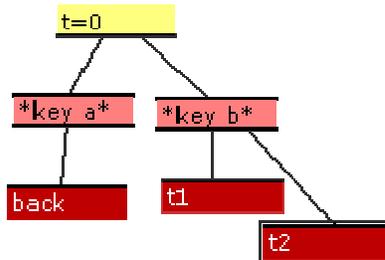
These are some examples of possible diagrams:



This is the simplest possible diagram, in which event "t1" will happen at the beginning of the piece, since it has no conditions.



When the yellow and green conditions are met, the two red events will happen simultaneously.



If the conditions "t=0" and "key a" are met, the "back" event happens. If the "t=0" and "key b" conditions are met, the events "t1" and "t2" happen simultaneously (first t1, then t2 -- left-right, top-down)

Blocks are distinguished by their color and label. The black stripes on top and bottom of blocks indicate whether a block can be a parent (black strip at the bottom) and/or a child (black strip on top)

## **The MIDI Poet Composer**

The following blocks, which represent conditions, can be parent blocks (most of them can also be child blocks)

- time** **Time:** Indicates the time at which the children blocks will be activated. Time cannot be a child block. If a diagram doesn't have this block at its root, its default activation time will be 0 (the start of the piece)
- ??** **Conditional:** Specifies internal conditions that have to be met in order to activate the children blocks. Can be parent and/or child.
- \*\*** **Keyboard Input:** Specifies which keyboard key has to be pressed in order to activate the children blocks. Can be parent and/or child.
- //** **MIDI Input:** Specifies a range of MIDI notes; when any of these notes is received, the children blocks are activated. Can be parent and/or child.

The following blocks represent MIDI Poet events. They can only be children blocks, or placed independently on the composition space.

- text** **Text Event:** Event that modifies the properties of one or many text areas.
- image** **Image Event:** Event that modifies the properties of one or many image areas.
- back** **Screen Background Event:** Event that modifies the properties of the screen's background.
- style** **Global Drawing Style Event:** Event that modifies the global drawing style.
- midiout** **MIDI Output Event:** Event that generates a MIDI out message.

Each text or image event is referred to as an "area", and identified by a unique area number. An unlimited number of independent text and image areas can exist. The areas can be defined as they are required. Each of these areas has attributes or properties (text: content, position, color, etc; image: content, position, etc) which can be modified using the corresponding event commands.

 **The MIDIcPoet Composer****Creating a new composition**

Select Composition --> New from the menu.

**Loading an existing composition**

Select Composition --> Open from the menu. A screen will appear to let you choose the composition you want to load. Compositions are files with the .mip extension.

**Saving a composition**

Select Composition --> Save (or Save As, for using a different name) from the menu. A screen will appear to let you choose the composition you want to save.

**General properties of a composition**

To specify the general properties of a composition, choose Composition --> Properties from the menu.

The general properties of a composition are:

- Title
- Area (X,Y) -- Indicates the X and Y measures (expressed in pixels) of the composition's area. This is the area inside which all the composition's events will occur.
- IMPORTANT NOTICE: Due to the algorithms used by MIDIcPoet, smaller areas will improve the composition's performance. Please experiment with different area sizes to achieve the desired results.
- Position -- If the composition's area is smaller than the screen's resolution (for example, if the area measures 640x480 pixels and the screen measures 1024x768), the position indicates the X,Y coordinates of the composition's area, relative to the top left corner of the screen. Thus, a position of 0,0 will place the composition's area at the top left corner of the display. Below the position fields, there are two checkboxes titled "center X" and "center Y". If these boxes are checked, the composition's area will automatically appear at the center of the screen.
- Images path -- If the checkbox labeled "Search images in same folder" is marked, all the images included in



## ***The MIDI Poet Composer***

the composition should be located in the same folder as the composition. If this checkbox is unmarked, MIDI Poet will search for the images at their original location.

# **The MIDI Poet Composer**

## **How to draw diagrams**

To start working on a composition, choose Composition --> New from the menu. An empty workspace will appear, along with the following toolbar.



The toolbar contains various buttons that allow you to draw and organize the blocks and segments of a diagram.



### **Edit diagram**

This button must be selected in order to create segments between blocks, and to move blocks around. To create a segment between two blocks:

- Place the mouse pointer over the lower black stripe of the parent block.
- While pressing the left mouse button, drag the pointer to the upper black stripe of the child block.
- Release the button. The parent-child relation is established between the two chosen blocks.
- Alternatively, the segment can be drawn in the inverse direction (from child to parent)

To move a block around:

- Place the mouse pointer over the block (not on the black stripes)
- While pressing the left mouse button, move the block to its new location.
- Release the button.

## **The MIDI Poet Composer**

The following buttons on the toolbar can be used to add the different types of blocks to the composition:



Add Time block



Add Conditional block



Add MIDI Input block



Add Keyboard Input block



Add Screen Event block



Add Global Drawing Style Event block



Add Text Event block



Add Image Event block



Add MIDI Output Event block

 **The MIDIcPoet Composer****Editing the diagrams**

The following diagram editing operations can be chosen from a pop-up menu that appears when the right mouse button is clicked and the mouse pointer is over a block:

- **Edit:** Add/modify the internal parameters of the selected block. A block's parameters can also be edited by double clicking on the desired block.
- **Copy:** Copy the desired block.
- **Paste:** Duplicate a previously copied block.
- **Cut:** Stores a block in memory and deletes it from screen. It can later be pasted.
- **Delete:** Delete the selected block. A block can also be deleted by selecting it and pressing the "Del" key. Segments are deleted in this same way.

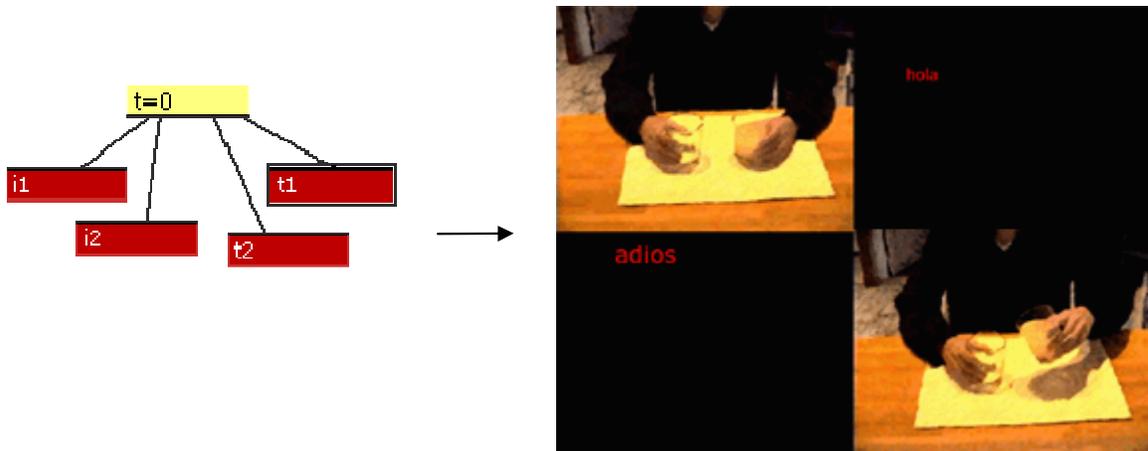
# The MIDIcPoet Composer

## Events: General concepts

The events (red blocks) in a MIDIcPoet piece modify the properties of an area or set of areas.

## About areas

We can think of areas as the characters in the piece. In MIDIcPoet pieces, the characters can be texts or images. Each of these characters appears (or disappears) in a specific moment, and has behaviors which can be either pre-defined or responsive to external impulses. Each character is independent from the rest, and has specific attributes which can be modified. The modification of these attributes can result in animations, changes in color, size, etc.



Two image events and two text events, and their corresponding image and text areas.

The text and image areas can contain any text or image respectively, and their content can vary through time. You can have as many text and image areas as you need, limited only by your machine's RAM.

## **The MIDIcPoet Composer**

While composing a MIDIcPoet piece, the text and image areas are defined as needed. Areas are defined using a unique number, which will be specified in the event properties screen. For instance, the first text area you use can be number 1, the next one number 2, and so on. You create new areas simply by defining new numbers. Each area is identified by its number. There can be more than one event modifying a single area; in such case, both events will make reference to this area's number.

Text events are represented in diagrams as red blocks labeled with the letter "t", followed by a number that represents the area or areas that are modified when the event occurs. If an event modifies text area '1', the event will be labeled "t1" in the diagram. If an event modifies various areas at the same time, for example, areas 1 to 10, the events label will be "t1-10". Events can affect multiple areas only if their numbers are contiguous. The same goes for image events, which are labeled with letter "i".

### **Cycles**

Text and image events can occur only once (they happen on a specific time and/or when a condition is met) or repeatedly (they happen in cyclic way and with a specific frequency, starting from a specific time and/or when a condition is met). This allows, for example, to activate the animation of an image sequence, or to modify progressively the X and Y coordinates of a text.

Cycles are a property of each event. One area can have an unlimited number of simultaneous cycles. For example, a text area can have its content changed, be moved and resized using independent cycles for each modification, each one with its own frequency of occurrence.

Cyclic events are distinguished in the diagrams with the "+" symbol.

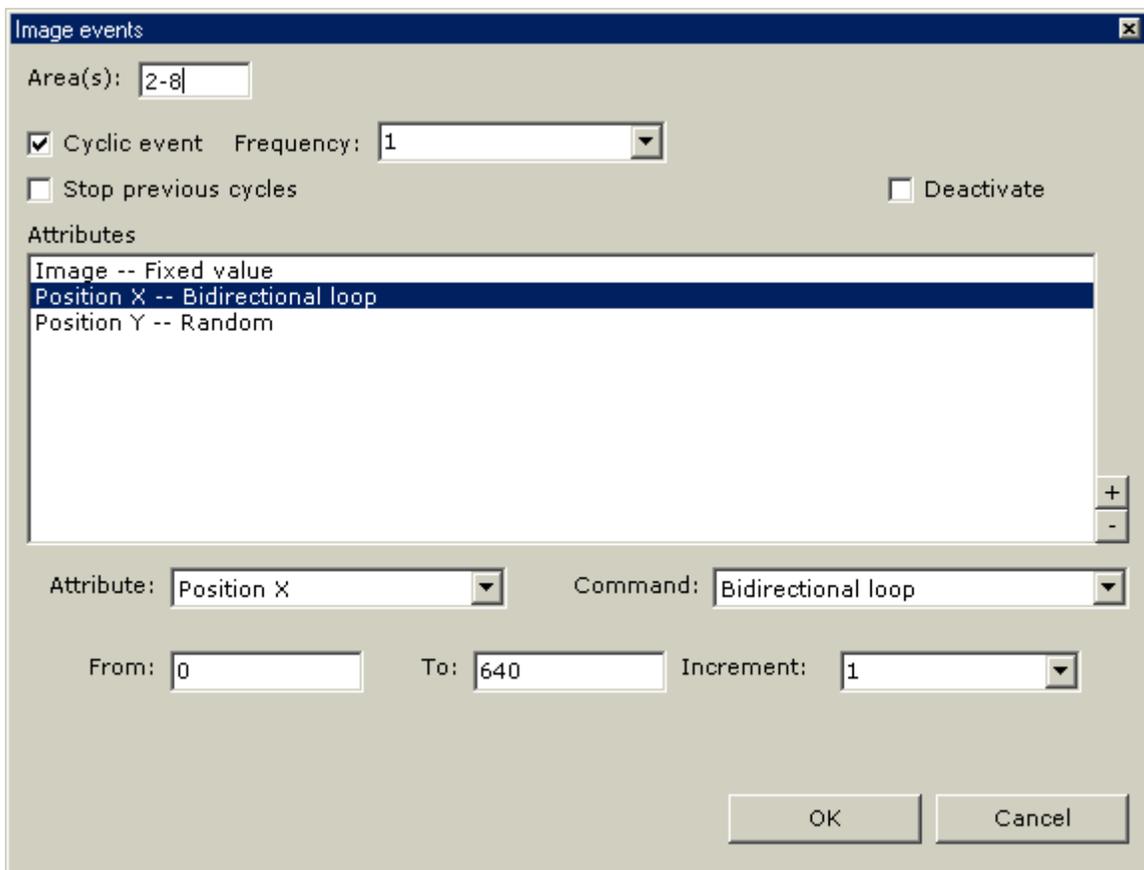
 i2-8+ This is a cyclic event that affects areas 2,3,4,5,6,7 and 8.

**Other types of events (all of which can be cyclic)**

- **Screen Background:** These events affect the properties of the screen background; the scenery on which text and image areas appear. The only property of the screen background which can be modified is its color.
- **Screen Drawing Style:** These events affect the way text and image areas are drawn on screen.
- **MIDI Outputs:** Events that generate a MIDI message that is sent to the chosen MIDI out port.

**Editing the properties of events**

The following screen appears when an event is chosen for editing:



 **The MIDI Poet Composer**

The editing screen's title indicates whether the current event modifies the properties of a text or image area, or the screen's background or drawing style.

If the current event modifies text or image, the modified areas must be specified. This is done by introducing the number of the area(s) in the field on top of the screen. If the event modifies a single area, its number must be entered here, whether it is a new area or a previously existing one. If the event modifies a range of areas, the lower and upper numbers of this range must be entered, separated by a hyphen. In the above sample screen, the event modifies image areas 2,3,4,5,6,7 and 8. Areas must be specified starting from number 1. The "Area(s)" field does not appear when modifying the properties of a screen background or drawing style event.

Below the "Area(s)" field, there is a checkbox labeled "Cyclic event". Putting a mark in this box will mean that the current event will occur repeatedly, and with a specific frequency. In this case, a field labeled "Frequency" appears to the right of the checkbox. This field can contain one of these values:

- A numeric value, expressed in milliseconds, that indicates the event's frequency.
- MIDI Tables. If the current event will be activated by a MIDI input, a resulting value containing a numeric value can be taken as the event's frequency.

If the "Cyclic event" checkbox is left unmarked, the event will occur only once.

There are two more checkboxes:

- "Stop previous cycles": If this box is marked, the event will only serve to stop all of the specified area or areas' previous cycles.
- "Deactivate": If this box is marked, the event will make the specified area or areas disappear from the screen.

If none of the above checkboxes is marked, the different attributes of the chosen areas can be modified using the list that appears below them.

 **The MIDI Poet Composer**

The attributes depend on the type of the area or areas modified by the event. An event can modify one or more attributes of the current area(s). The list that occupies the center of the screen specifies all the attributes that will be modified. To add or remove attributes from this list, use the "+" (add) or "-" (remove) buttons.

When an attribute is selected in this list, it can be changed using the "Attribute" and "Command" fields below the list. The "Attribute" field is a drop-down list that lets you choose the attribute which will be modified. In the "Command" field, you can specify the type of modification the current attribute will suffer. The available commands vary, depending on the current attribute.

Commands require different parameters that specify the values or ranges that will be used to modify the attribute of the area. Parameter specification varies according to the attribute and command, and can involve numeric values, numeric ranges (with or without an increment value), text values, names of images, fonts or drawing styles, or other sources from which a value will be taken (data lists, MIDI tables, MIDI ranges or other areas).

The MIDI output events have their own special editing screen.

## **Text events**

Text events modify the attributes of one or various text areas.

A text area's modifiable attributes are:

- **Text:** textual content of the area.
- **Font:** font used to display the text
- **Size:** size of the text, expressed in points.
- **Angle:** angle of the text. It is measured in relation to the upper left corner of the text area, and expressed in degrees.
- **Position X:** horizontal component of the text area's position. It is expressed in pixels, and ranges from 0 (left) to the value of the piece's area width.
- **Position Y:** vertical component of the text area's position. It is expressed in pixels, and ranges from 0 (top) to the value of the piece's area height.
- **Red:** Red component of the text's color. Ranges from 0 to 255.
- **Green:** Green component of the text's color. Ranges from 0 to 255.
- **Blue:** Blue component of the text's color. Ranges from 0 to 255.

Colors in MIDIcPoet are expressed as a combination of RGB (red, green, blue) values.

The attributes' values can be modified using the following commands:

Valid for all attributes:

- **Fixed value:** A unique value, numeric or textual.
- **Random:** A random value chosen from a range of values or from a data list.
- **Loop:** A value that increases from a starting point to an ending point, and then starts again. The values can be taken from a numeric range or a data list.
- **Bidirectional loop:** A value that increases/decreases inside a range. When it reaches the range's upper or lower limit, it changes direction. The values can be taken from a numeric range or a data list.

## **The MIDIcPoet Composer**

- **Travel:** A value that increases from a starting point to an ending point, and then stops. The values can be taken from a numeric range or a data list.
- **Drunk:** A value resulting from a random walk between the starting and ending points of a range. The values can be taken from a numeric range or a data list.
- **MIDI Table:** A value taken from a MIDI table. If the event is activated by a MIDI trigger, this value will correspond to the triggering message's specific note or velocity values.

The Loop, Bidirectional loop, Travel and Drunk commands need, besides a range of values, an increment value, which can be numeric or a resulting value of a MIDI table. For example, a bidirectional loop with a range of 0-255 and with an increment value of 2 will go from 0 to 255 and back, stepping 2 positions every time. Another example: a loop that goes through the values of the data list "data 1", and with an increment value of "vel table 1", will loop through the elements in the list "data 1", skipping "n" positions, where "n" is a resulting value coming from the MIDI table "vel table 1".

Valid for numeric attributes:

- **Increase:** Increases the value of the attribute by the specified quantity
- **Decrease:** Decreases the value of the attribute by the specified quantity.
- **MIDI Range:** A value coming from a MIDI range. If the event is activated by a MIDI trigger, this value will correspond to the triggering message's specific note or velocity values.

Valid for textual attributes:

- **Next:** The value will be the next element of a data list.
- **Previous:** The value will be the previous element of a data list.

These commands use an internal index to keep track of the current element in the specified list.

Valid only for the "text" attribute:

- **Mix letters:** Mixes the letters of the area's current text.



## ***The MIDI Poet Composer***

- **Mix words:** Mixes the words of the area's current text.
- **Shift left:** Shifts all letters of the area's current text to the left.
- **Shift right:** Shifts all letters of the area's current text to the right.

All modifications to an area's "text" attribute can be accumulative -- this means that every modification is added to the area's previous text value. This is done by marking the "Concatenate text" checkbox.

## *The MIDIcPoet Composer*

### **Image Events**

Image events modify the attributes of one or various image areas.

An image area's modifiable attributes are:

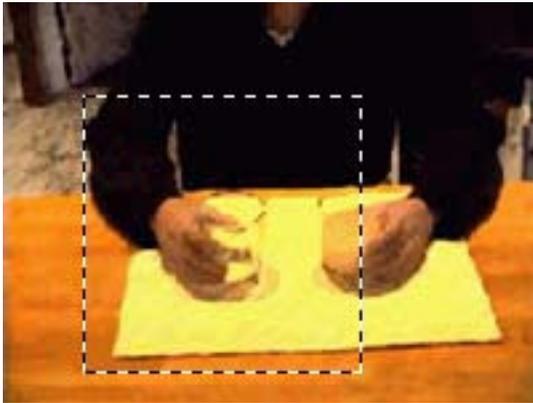
- **Image:** Image content of the area. This is the name of an image already included in the piece's image bank. See the images section.
- **Mirror X, Mirror Y:** Display the X or Y reflection of the image.
- **Style:** Drawing style used to draw the image on the screen. (see the drawing styles section)
- **Position X:** Horizontal component of the image area's position. It is expressed in pixels, and ranges from 0 (left) to the value of the piece's area width.
- **Position Y:** Vertical component of the image area's position. It is expressed in pixels, and ranges from 0 (top) to the value of the piece's area height.

The following attributes refer to the copying and pasting of the drawing area. By default, images are drawn on the screen using their original size. If the copying and pasting parameters are changed, the size and proportion of the image on screen will change accordingly, while the original image is not modified. For example, suppose the following image measures 640x480 pixels:



## *The MIDIcPoet Composer*

If we change the copying area of the image to a rectangle whose upper left corner's coordinates are 100,100 (X1, Y1), and whose width and height are 300 pixels, we will have the following copying area:



If we change the pasting measures to 640 (width) and 100 (height), this is the way the image will appear on the screen:



- Copy X1: X1 coordinate of the image's copying rectangle (0 by default)
- Copy Y1: Y1 coordinate of the image's copying rectangle (0 by default)
- Copy width: Width of the image's copying rectangle (by default it's equal to the image's width)
- Copy height: Height of the image's copying rectangle (by default it's equal to the image's height)
- Paste width: Width of the image's pasting rectangle (by default it's equal to the image's width)
- Paste height: Height of the image's pasting rectangle (by default it's equal to the image's height)

 **The MIDIc Poet Composer**

The values of the attributes can be modified by using the following commands:

Valid for all attributes:

- **Fixed value:** A unique value, numeric or textual.
- **Random:** A random value chosen from a range of values or from a data list.
- **Loop:** A value that increases from a starting point to an ending point, and then starts again. The values can be taken from a numeric range or a data list.
- **Bidirectional loop:** A value that increases/decreases inside a range. When it reaches the range's upper or lower limit, it changes direction. The values can be taken from a numeric range or a data list.
- **Travel:** A value that increases from a starting point to an ending point, and then stops. The values can be taken from a numeric range or a data list.
- **Drunk:** A value resulting from a random walk between the starting and ending points of a range. The values can be taken from a numeric range or a data list.
- **MIDI Table:** A value taken from a MIDI table. If the event is activated by a MIDI trigger, this value will correspond to the triggering message's specific note or velocity values.

The Loop, Bidirectional loop, Travel and Drunk commands need, besides a range of values, an increment value, which can be numeric or a resulting value of a MIDI table. For example, a bidirectional loop with a range of 0-255 and with an increment value of 2 will go from 0 to 255 and back, stepping 2 positions every time. Another example: a loop that goes through the values of the data list "data 1", and with an increment value of "vel table 1", will loop through the elements in the list "data 1", skipping "n" positions, where "n" is a resulting value coming from the MIDI table "vel table 1".

Valid for numeric attributes:

- **Increase:** Increases the value of the attribute by the specified quantity
- **Decrease:** Decreases the value of the attribute by the specified quantity.



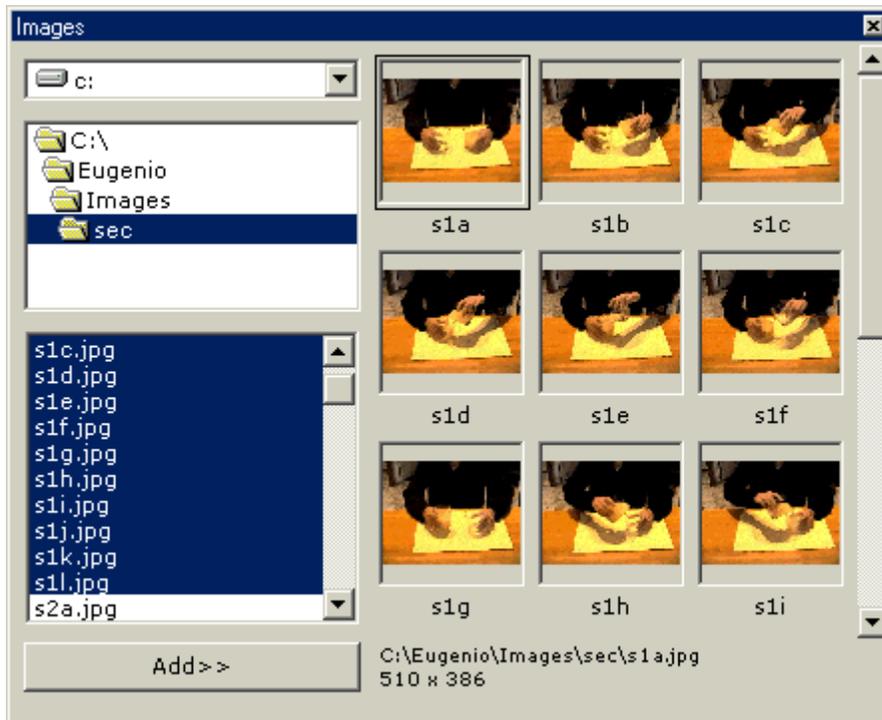
## ***The MIDI Poet Composer***

- **MIDI Range:** A value coming from a MIDI range. If the event is activated by a MIDI trigger, this value will correspond to the triggering message's specific note or velocity values.

# The MIDI Poet Composer

## Including images

The option "Window --> Images" on the menu will let you choose the images included in the composition. The following screen will appear:



The necessary fields for specifying the images' path are found on the left side of this screen. Once a path containing image files has been chosen, all the images found in the path will be shown in the list below. MIDI Poet can read bmp, jpg and gif images. The desired images can be selected from this list, and included in the composition's image bank by pressing the "Add >>" button.

On the right side of the screen, the images that form the composition's image bank are shown, but not in their actual size; they are scaled to fit the screen. Each image is shown along with its name; these names will be used later to identify them and use them in image events. The full image's name and path, and its X and Y measures (in pixels) will be shown on the lower side of the screen when an image is clicked. To see all the included images, use the scroll bar on the right side of the screen.

 **The MIDIcPoet Composer**

Images located in different paths can be used in the same composition, but bear in mind that:

- If the checkbox "Search in same path" on the general properties screen is checked, every selected image must be in the composition's path.
- If this box is not marked, the images will be searched for at their original paths.

To remove one or many images from the composition's bank, click on the first image you want to remove and press the "Del" key. A screen will appear, asking for the last image to delete. This allows you to delete a single image or a sequence of images.

## Screen background events

The screen background events modify the screen background's color.

The screen background's modifiable attributes are:

- **Red:** Red component of the background color. Ranges from 0 to 255.
- **Green:** Green component of the background color. Ranges from 0 to 255.
- **Blue:** Blue component of the background color. Ranges from 0 to 255.

NOTE: Colors in MIDIcPoet are expressed as a combination of RGB (red, green, blue) values.

The attributes' values can be modified using the following commands:

- **Fixed value:** A unique value, numeric or textual.
- **Random:** A random value chosen from a range of values or from a data list.
- **Loop:** A value that increases from a starting point to an ending point, and then starts again. The values can be taken from a numeric range or a data list.
- **Bidirectional loop:** A value that increases/decreases inside a range. When it reaches the range's upper or lower limit, it changes direction. The values can be taken from a numeric range or a data list.
- **Travel:** A value that increases from a starting point to an ending point, and then stops. The values can be taken from a numeric range or a data list.
- **Drunk:** A value resulting from a random walk between the starting and ending points of a range. The values can be taken from a numeric range or a data list.
- **MIDI Table:** A value taken from a MIDI table. If the event is activated by a MIDI trigger, this value will correspond to the triggering message's specific note or velocity values.
- **Increase:** Increases the value of the attribute by the specified quantity
- **Decrease:** Decreases the value of the attribute by the specified quantity.



## *The MIDIcPoet Composer*

- **MIDI Range:** A value coming from a MIDI range. If the event is activated by a MIDI trigger, this value will correspond to the triggering message's specific note or velocity values.

The Loop, Bidirectional loop, Travel and Drunk commands need, besides a range of values, an increment value, which can be numeric or a resulting value of a MIDI table. For example, a bidirectional loop with a range of 0-255 and with an increment value of 2 will go from 0 to 255 and back, stepping 2 positions every time. Another example: a loop that goes through the values of the data list "data 1", and with an increment value of "vel table 1", will loop through the elements in the list "data 1", skipping "n" positions, where "n" is a resulting value coming from the MIDI table "vel table 1".

## *The MIDI Poet Composer*

### **Global drawing style events**

The global drawing style events can modify the way the visual elements of a MIDI Poet piece are drawn on the screen. When a MIDI Poet piece is played, MIDI Poet redraws on each cycle all the active visual elements. By default, these elements are shown without any transformation, but there are different drawing styles that affect the way visual elements appear.

The screen's drawing style attributes are:

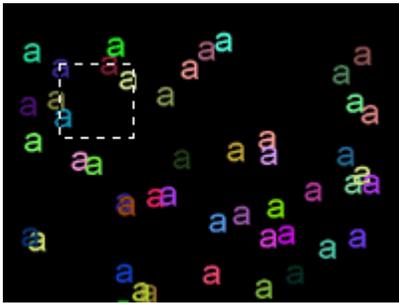
- Clear screen: This attribute specifies whether the screen is cleared on every cycle or not. If the screen is not cleared, text and image areas will leave a trail as they move or animate.
- Style. Drawing style used to draw the text and image areas on the screen.

The following attributes refer to the copying and pasting of the drawing area. By default, texts and images are drawn on the screen have their own specific measures. If the copying and pasting parameters are changed, the size and proportion of all visual elements change accordingly. For example, let's suppose that the following image shows a moment of a MIDI Poet piece, and that the piece's drawing area is 640x480:



Now, if we establish that the copying area of the piece will be a square which has the coordinates 100,100 (X1, Y1) for its upper left corner, and a width and height of 100, we would have the following copying area:

# The MIDI Poet Composer



If we say the pasting measures are 640x480, this is what we will see on the screen:



- Copy X1: X1 coordinate of the drawing area's copying rectangle (0 by default)
- Copy Y1: Y1 coordinate of the drawing area's copying rectangle (0 by default)
- Copy width: Width of the drawing area's copying rectangle (by default it's equal to the area's initial width)
- Copy height: Height of the drawing area's copying rectangle (by default it's equal to the area's initial height)
- Paste width: Width of the drawing area's pasting rectangle (by default it's equal to the area's initial width)
- Paste height: Height of the drawing area's pasting rectangle (by default it's equal to the area's initial height)

 **The MIDI Poet Composer**

The attributes' values can be modified using the following commands:

Valid for all attributes:

- **Fixed value:** A unique value, numeric or textual.
- **Random:** A random value chosen from a range of values or from a data list.
- **Loop:** A value that increases from a starting point to an ending point, and then starts again. The values can be taken from a numeric range or a data list.
- **Bidirectional loop:** A value that increases/decreases inside a range. When it reaches the range's upper or lower limit, it changes direction. The values can be taken from a numeric range or a data list.
- **Travel:** A value that increases from a starting point to an ending point, and then stops. The values can be taken from a numeric range or a data list.
- **Drunk:** A value resulting from a random walk between the starting and ending points of a range. The values can be taken from a numeric range or a data list.
- **MIDI Table:** A value taken from a MIDI table. If the event is activated by a MIDI trigger, this value will correspond to the triggering message's specific note or velocity values.

The Loop, Bidirectional loop, Travel and Drunk commands need, besides a range of values, an increment value, which can be numeric or a resulting value of a MIDI table. For example, a bidirectional loop with a range of 0-255 and with an increment value of 2 will go from 0 to 255 and back, stepping 2 positions every time. Another example: a loop that goes through the values of the data list "data 1", and with an increment value of "vel table 1", will loop through the elements in the list "data 1", skipping "n" positions, where "n" is a resulting value coming from the MIDI table "vel table 1".

Valid for numeric attributes:

- **Increase:** Increases the value of the attribute by the specified quantity
- **Decrease:** Decreases the value of the attribute by the specified quantity.
- **MIDI Range:** A value coming from a MIDI range. If the event is activated by a MIDI trigger, this value will



## ***The MIDIc Poet Composer***

correspond to the triggering message's specific note or velocity values.

Valid only for the "style" attribute:

- **Next:** The value will be the next element of a data list.
- **Previous:** The value will be the previous element of a data list.

### **About drawing styles**

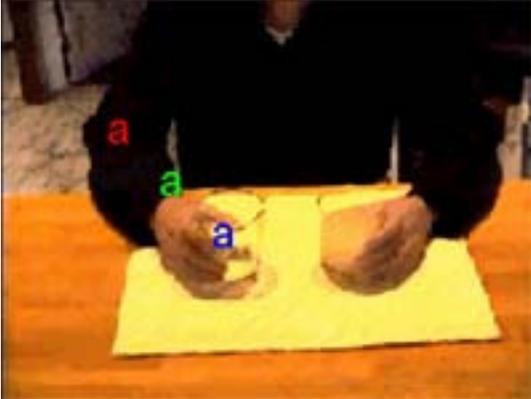
Drawing styles are the different ways the visual elements (images and text) are drawn on the screen. These styles can be applied to the whole screen, using the screen drawing events, or to individual images, by modifying the image area's "style" attribute. Some of these styles can give different results when combined with changes in the screen's background colors. Please experiment.

The drawing styles are:

- BLACKNESS Fills the drawing area with black color.
- DSTINVERT Inverts the drawing area's colors.
- MERGECOPY Combines the source (visual elements) and destination (drawing area) colors using the logical AND operator.
- MERGEPAIN T Combines the source (visual elements) and destination (drawing area) colors using the logical OR operator.
- NOTSRCOPY Inverts the original colors of the visual elements.
- NOTSRCERASE Combines and inverts the source (visual elements) and destination (drawing area) colors using the logical OR operator.
- SRCAND Combines the source (visual elements) and destination (drawing area) colors using the logical AND operator.
- SRCOPY Copies the visual elements to the drawing area without transforming them. This is the default value.
- SRCERASE Combines the source colors (visual elements) and the inverted destination colors (drawing area) using the logical AND operator.
- SRCINVERT Combines the source (visual elements) and destination (drawing area) colors using the logical XOR operator.
- SRCPAIN T Combines the source (visual elements) and destination (drawing area) colors using the logical OR operator.
- WHITENESS Fills the drawing area with white color.

**M<sup>i</sup>c** *The MIDIpoet Composer*

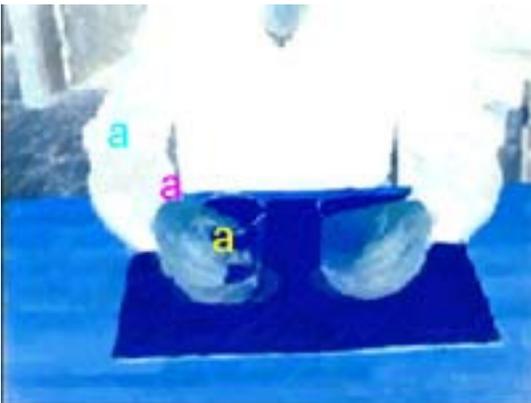
A few drawing style examples



SRCCOPY (original)



MERGECOPY

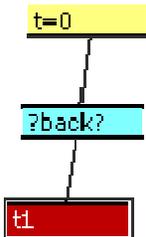


MERGEPAIN

# *The MIDIcPoet Composer*

## Time

In MIDIcPoet, time is measured in milliseconds. A starting time can be specified at the top of a diagram. This means every child of the time block will start to be active after the specified number of milliseconds. For instance:



Event t1 will start being active after 0 milliseconds (start of the performance), but only if the "back" condition is met.

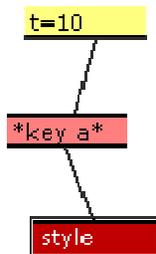
To edit the starting time of a diagram, double click the time block (yellow). The following screen will appear:



# *The MIDIc Poet Composer*

## Keyboard input

In MIDIc Poet, events can be triggered by pressing a key on the computer's keyboard.



In this diagram, the global drawing style event is triggered when the "a" key is pressed after time=10 ms.

To specify which key will be the trigger, double click the keyboard input block (pink). The following screen will appear:



When the MIDIc Poet piece is performed, the trigger keys must be pressed using the "Ctrl" key. For the above example, the keys "Ctrl" + "a" must be pressed simultaneously to trigger the event.

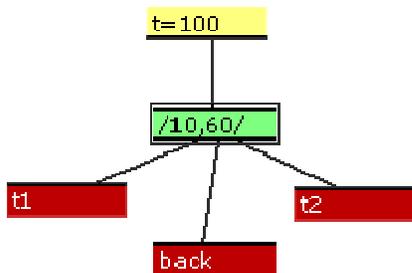
# *The MIDIpoet Composer*

## **MIDI input**

One of the most important features of MIDIpoet is its capability to receive MIDI input and activate text/image events according to the message received. Specifically, MIDIpoet can read Note On and Note Off messages, coming through any MIDI channel. The input MIDI port will be chosen prior the performance of a piece.

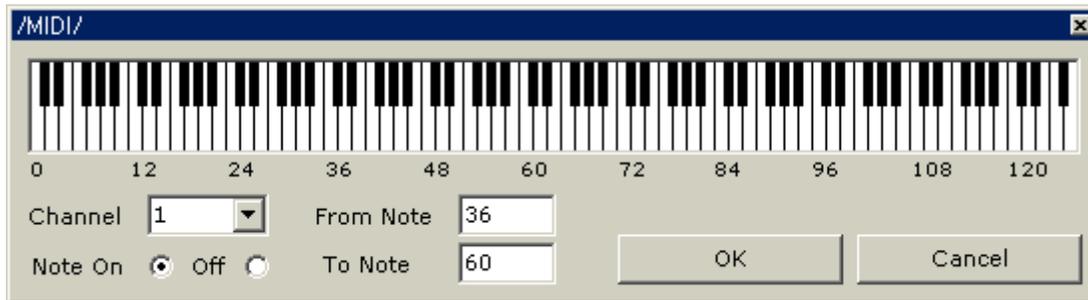
Ranges are used to specify the notes that activate events. In each range, lower and upper notes are specified; any note received inside the range will trigger the MIDI Input block's children. It is also necessary to specify a MIDI channel, and whether the received message should be Note On or Note Off.

In the following diagram, starting at time 100, notes between 10 and 60 will activate the events (t1,t2 and back).



To edit the MIDI input range, double click the MIDI input block (green). The following screen will appear.

 *The MIDIc Poet Composer*

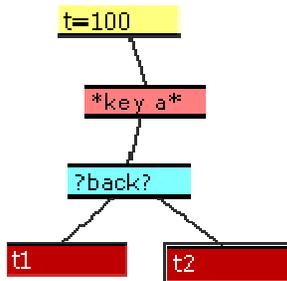


This screen allows the specification of all the information regarding the incoming MIDI messages: the channel (1-16), Lower note (0-127), Upper note(0-127) and message type (Note On or Note Off)

The keyboard in the upper part of the screen may be useful to choose a note and find its corresponding MIDI value.

## Conditionals

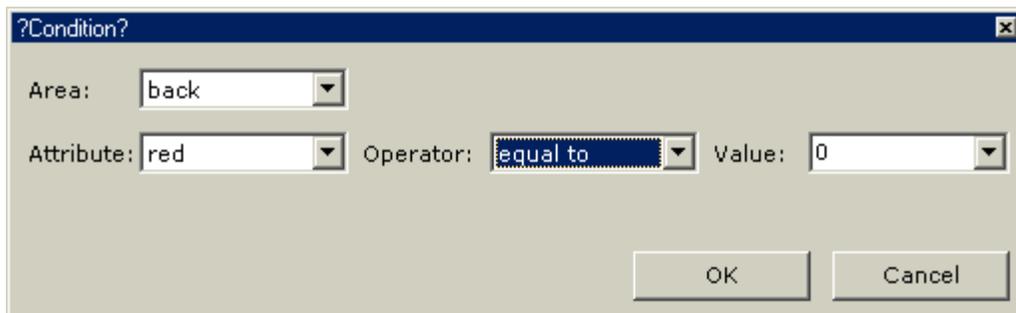
MIDI Poet events can be triggered not only by external impulses (keyboard or MIDI input), but also by internal conditions. Through the use of conditionals, MIDI Poet checks whether a specific attribute of a text or image area, or the screen background, meet certain conditions. For instance:



Starting at time 100 ( $t=100$ ), if the "a" key is pressed and the "back" condition is met, the events "t1" and "t2" occur.

The ?back? condition asks an internal question regarding a specific attribute of an element or area in the composition (the screen background in this case)

By double clicking the conditional block (light blue), the following screen will appear:



In this example, if the "red" attribute of the screen background is "equal to 0", the conditional block's children will become active. If this condition is not met, they remain inactive. In general, if a condition has a TRUE value, the block's children will be active; if it is FALSE they remain inactive.

 **The MIDIcPoet Composer**

Conditionals can be constructed by asking specific questions about the different attributes of the composition's elements: text areas, image areas and the screen background. The valid comparison operators are:

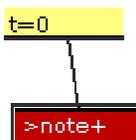
- Equal to
- Greater than (numeric values)
- Smaller than (numeric values)
- Different from
- Between (numeric values) -- This operator requires two values: the lower and upper limits to which the attribute's value will be compared.

# *The MIDIpoet Composer*

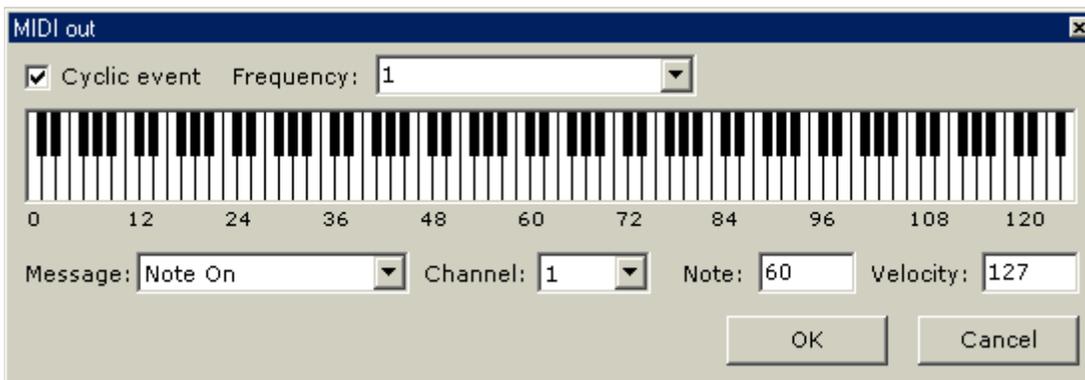
## MIDI output events

MIDIpoet can generate MIDI output messages, which can be useful to communicate with other programs or any device capable of reading MIDI messages. MIDI output events are used to generate Note or Control Change output messages during a piece's performance. The output port through which these events will be sent can be chosen before the piece is played.

In the following diagram, Note Out MIDI message will be generated after time 0.



The "+" sign indicates that the event is cyclic; this means the note will be generated repeatedly after time 0. This is the MIDI output events' screen:



The MIDI output message type (Note or Control Change) is specified here, along with its channel (1-16) and parameters. If the message type is note, a note (0-127) and its velocity (0-127) must be specified. If the message is a Control Change, the control (0-127) and its value (0-127) must be specified.

MIDI output events can be cyclic; this can be specified by marking the "Cyclic event" checkbox and specifying the frequency value: a numeric value, or a MIDI table from which a frequency value will be taken.

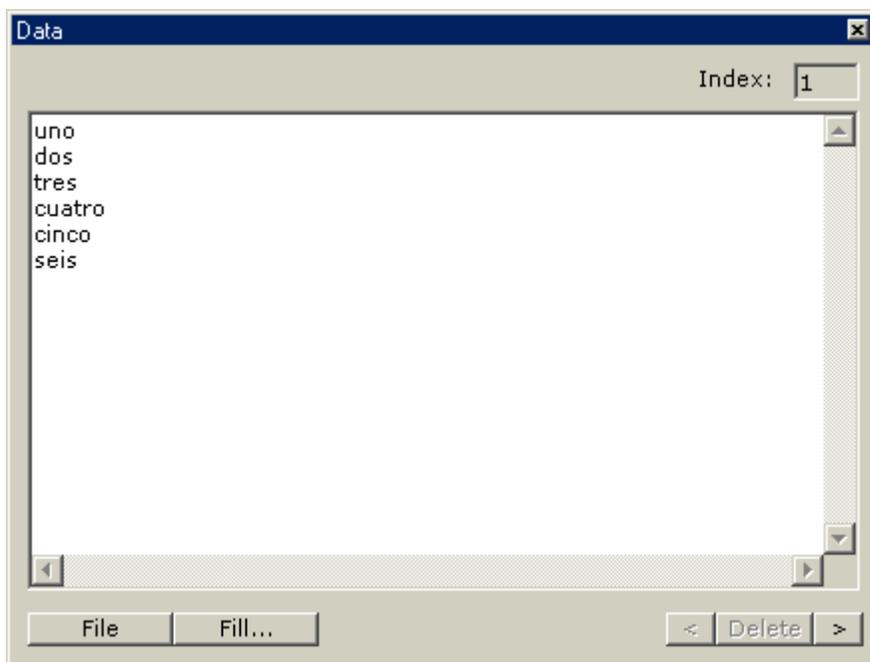
# **The MIDIcPoet Composer**

## **Data sources: lists, ranges and tables**

Text, image, screen background and global style events modify the attributes of the various components of a MIDIcPoet piece. These attributes can be modified in a different number of ways. For example, a random text can be chosen from a list of possible texts, or the screen's background color can vary according to the incoming MIDI notes. Also, the frequency with which a cyclic event occurs can vary according to a note's velocity. All these processes require the definition of data sources: data lists, MIDI ranges and MIDI tables.

### **Lists**

Data lists are lists of numeric or textual elements that can be used as resulting values for an area's modified attributes. There can be as many lists as needed, and each list can have an unlimited number of elements. Each list can be later recalled by specifying its index, a unique numeric identifier. When a new list is created, an index is automatically assigned to it. To create/modify/delete lists, choose the Window --> Data option from the menu. This is the data lists screen:



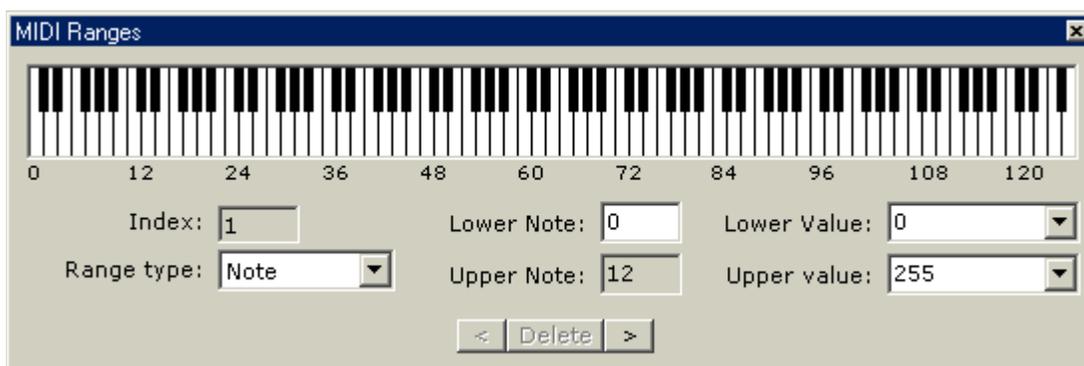
## **The MIDI Poet Composer**

The current list's index is at the top right corner of the screen, in this case the index is 1. The big field at the center of the screen is the list itself, with each of its elements occupying a single line. Elements of a list can be a word or words, or a number. Lists can be filled using the contents of a text (.txt) file. To do this, press the "File" button and choose the file. The list can also be filled with other elements: image names, fonts or drawing styles. To fill the list with one of these elements press the "Fill..." button. A screen that lets you choose which elements you want to use will appear.

Use the "<" and ">" buttons to move through existing lists; press the "Delete" button to delete the current list.

### **MIDI Ranges**

MIDI ranges let you modify the value of an area's numeric attribute according to a MIDI note input. Actually two different ranges must be specified in a MIDI range: a range of incoming note or velocity values, and a range of resulting values which will be applied to an attribute. When MIDI Poet receives a MIDI message that falls inside the first range, calculations will be performed to obtain the corresponding value from the second range, which will be relative to the MIDI input. An unlimited number of MIDI ranges can be defined; each range can respond to either note or velocity values. MIDI ranges are later recalled using their index, a unique numeric identifier that is automatically generated when a new MIDI range is created. To create/modify/delete MIDI ranges, choose the Window --> MIDI --> Ranges option from the menu. This is the MIDI ranges screen:



 **The MIDIpoet Composer**

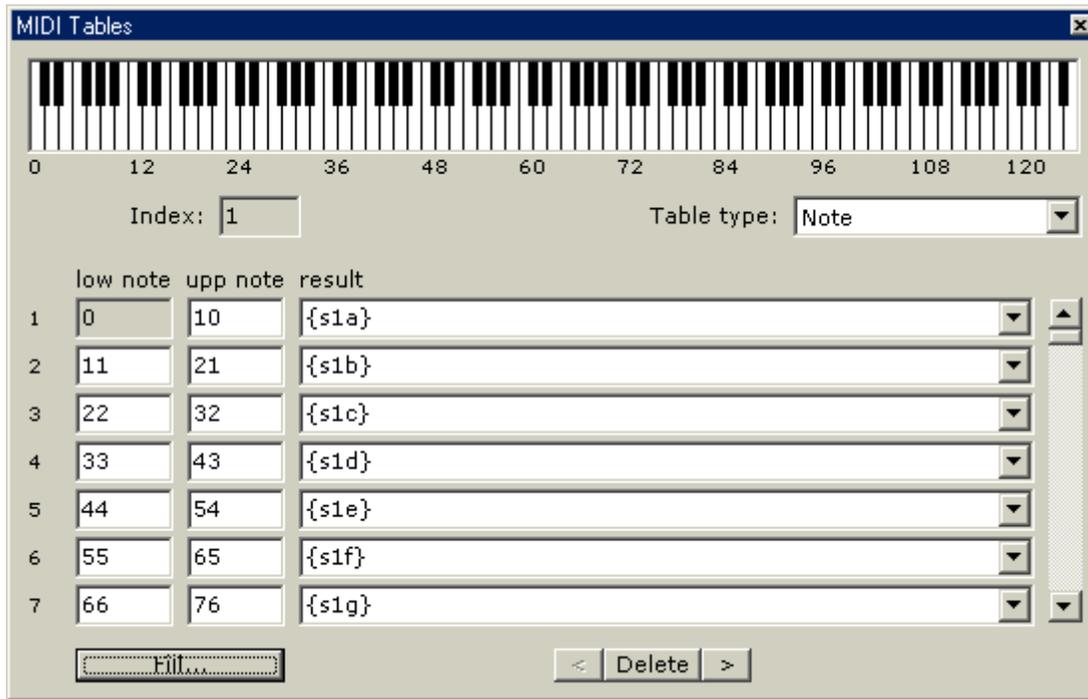
In this case, a MIDI note range is defined. If a note between 0 and 12 is received, values ranging from 0 to 255 will be generated. For example, note 6 results in a value of 127. Resulting values will always be rounded integers. The keyboard at the top of the screen can be useful to find a specific note's MIDI value. Previously included images can be used as resulting values of a MIDI range. In this case, a range of images is defined by choosing their names from the drop down lists labeled "Lower value" and "Upper value".

Use the "<" and ">" buttons to move through existing MIDI ranges; press the "Delete" button to delete the current MIDI range.

### **MIDI Tables**

MIDI tables let you modify an area's attribute according to incoming MIDI values. MIDI tables are lists of MIDI value ranges, which can be note or velocity values. For each of the ranges, there is a single resulting value, which can be numeric or textual. When MIDIpoet receives a MIDI value that is inside one of the ranges, the value of that range will be applied to the selected attribute. The resulting values of a MIDI table can also modify an event's frequency, or an increment used by one of the event's processes. MIDI tables can respond to note or velocity values. MIDI tables are later recalled using their index, a unique numeric identifier that is automatically generated when a new MIDI table is created. To create/modify/delete MIDI tables, choose the Window --> MIDI --> Tables option from the menu. This is the MIDI tables screen:

# The MIDIc Poet Composer



In this example there is a table with seven ranges of MIDI values. These ranges make reference to incoming MIDI note values. The first range comprises notes 0-10, and yields a value of {s1a} (an image name). The second range comprises notes 11-21, and yields a value of {s1b}, and so on. The keyboard at the top of the screen helps in finding a specific note's MIDI value. The resulting value of a range can be a text or a number; a previously included image can be chosen from the drop-down list. Use the scroll bar at the right of the screen to see all the ranges in a table. A table can have up to 128 ranges. Ranges, and their resulting values, can be automatically generated, by pressing the "Fill..." button. This button will show a screen that lets you specify the number of ranges, and/or a number of resulting values of different types (images, fonts or drawing styles).

Use the "<" and ">" buttons to move through existing MIDI tables; press the "Delete" button to delete the current MIDI table.

## ***Index***

<b>Topic</b>	<b>Page</b>
Introduction	2
Logos	2
Basic concepts	3
MIDIpoet Player	4
MIDIpoet Composer	7
Creating a new composition	10
Loading an existing composition	10
Saving a composition	10
General properties of a composition	10
How to draw diagrams	12
Editing the diagrams	14
Events: general concepts	15
Cyclic events	16
Editing the properties of events	17
Text events	20
Image events	23
Including images	27
Screen background events	29
Global drawing style events	31
About drawing styles	35
Time	37
Keyboard input	38
MIDI input	39
Conditionals	41
MIDI output events	43
Data sources: lists, ranges and tables	44