GNInsertController insert commands into MIDI files

GNInsertController

by Günter Nagler

MIDI devices can be controlled by MIDI commands during playing a song. This tool helps inserting such commands at absolute or relative MIDI locations into one or more MIDI files.

Define actions that can be repeated to other songs in future.

Apply actions to a folder that contains MIDI files in its subdirectories.

GNInsertController

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: Juni 2023 in Austria, Graz

Programming

Günter Nagler, Austria

Table of Contents

1 Introduction

1.1 Insert controllers into MIDI files

This project inserts defined MIDI commands at given relative or absolute positions in one or more MIDI channels.
Optionally it replaces existing MIDI commands (e.g. MIDI control, system exclusive) that are near a given range from the position of the new inserted command.

The insert definition can be saved to a file and reused later for other MIDI files.

The insert operations can be applied to a single file or to a folder that contains MIDI files (batch operations).

1.2 License

GNInsertController software is commercial. It requires a license legally purchased from company Share-it to enable all features and batch operation.

Visit http://www.gnmidi.com/gnorderen.htm to purchase a software license.

The end user license agreement will be shown at start of the program and must be accepted that the software can be used.

The software might apply changes to your MIDI files if the user wants that instead of writing a new file.
In this case the software tries to copy the original file to a backup filename in same folder.
Keep original files for the case you want to reuse a MIDI file without the changes made by this software.

Purchased license must be installed to turn the demo into full version. After some days using the program with installed license the software might ask for registering the current computer.
This can be done online or by email to info@gnmidi.com (the email text will be generated by the dialog). After successful registration you get a code for a computer that must be entered into the dialog.
Keep the code and your license file (mail text) at a safe place. Using an other computer requires requesting another code, your license allows to use the software on max. 5 of your computers.
For using more of your computers you would need to purchase another license.
Each company employee needs an own license. Only the user with the name mentioned in the license text is allowed to use the software with the installed license.

(c) 2016 GN Midi Solutions

1.3 Updates

From time to time there might be an update of the software released that fixes problems of an earlier software version or improves the software or adds new functionality to the software.
Upgrades are free till version 1.99 for registered users.

Visit the product page http://www.gnmidi.com/gninsertcontrolleren.htm and download the newest demo version. Uninstall your previous software version and then install the newer version.
Your installed license and software settings and previous created insert control definition files (*.insdef) should be still remembered.

1.4 GNMIDI product as support

Our GNMIDI software product has many operations to analyse, convert, modify MIDI files. GNMIDI demo may be tried free for 14 days.

Following GNMIDI functions might be interesting for users of this software:

menu Analyse/compare MIDI file against a second one: compares two similiar MIDI files and displays their differences. After inserting and replacing commands you could compare the result MIDI file and original MIDI file and see that the changes were as expected.
http://www.gnmidi.com/handbook/english/comparemidifiles.htm

menu Convert/MIDI to ASCII Text: converts MIDI file to text so that the content of a MIDI file can be read by human. The text may be changed and can be converted back to MIDI file using GNMIDI.
http://www.gnmidi.com/handbook/english/midi2txt.htm

menu Convert/MIDI to csv table: converts MIDI file to a .csv spreadsheet so that the content can be viewed and modified as a table using a text editor or spreadsheet application (e.g. openoffice.org).
http://www.gnmidi.com/handbook/english/midi2csv.htm

menu Modify/synchronisation editor: insert and synchronise lyric lines into MIDI files and MP3 files
http://www.gnmidi.com/handbook/english/synchronize_editor.htm

menu Modify/karaoke editor: insert and lyrics into MIDI files and synchronise syllables to melody notes
http://www.gnmidi.com/handbook/english/karaoke_editor.htm

Maybe also many others that are not mentioned here:
http://www.gnmidi.com/handbook/english/gnmidien_content.htm

2 GNInsertController usage

2.1 General

The GNInsertController application is for Windows systems only. Supported Windows systems:
Windows 7 and Windows 8 and Windows 10 and Windows 11
If S-Mode is active in Windows Home then Windows does not allow to start this application.

The EULA (end user license agreement) must be accepted that the software can be used.

A registration reminder dialog appears sometimes when using the demo without installed license.
Using the demo longer than 14 days without purchasing a license is not allowed.

First create a new insert controller definition or open an existing insert controller definition (*.insdef).
A window will show tab panes for each defined action. At least one action must be defined that the insert operation can be run.
First action form will be automatically created. Later you can append more actions using menu Action/New action. The actions will be applied to the MIDI song in order as they are defined.
An action is usually inserting a command into given channels of a MIDI file and optionally delete previous similar command on the insert channel within a given range (replace command).

Each action has properties that can be set in following 4 tab panes:
Command
Channels

Replace
Position

If there is an error (syntax input error or missing input in any of the tabs) the status bar will display an error message.
As long as there are errors the actions can not be applied to MIDI files and the run operations are still disabled.

The filled definition can be saved to a new file (Save As) or to the existing defintion file (Save).
This is useful to redo the same operation at a later time using other MIDI files without need to redefine the operation every time.

A valid filled definition can be run as insert operation following ways:

single MIDI file (overwrite existing file and backup old file with other name)
single MIDI file to a new MIDI file
input folder into output folder (insert operation is done to each valid MIDI file and all other files are copied)

Close the window if the definition is not needed anymore in this session. If the definition is not saved then user will be asked before close is done.

2.2 Menu

2.2.1 File

File menu offers operations to create, open or save *.insdef files (called insert definition).
Here you also find the recently used definition files.

2.2.1.1 New definition

Creates a new definition to enter actions.

2.2.1.2 Open definition

Choose an existing definition file (*.insdef) and loads the contained actions. A new window will show the definition and allow to modify or run actions.

2.2.1.3 Save definition

Save current data to the already existing file (*.insdef) which will be overwritten.

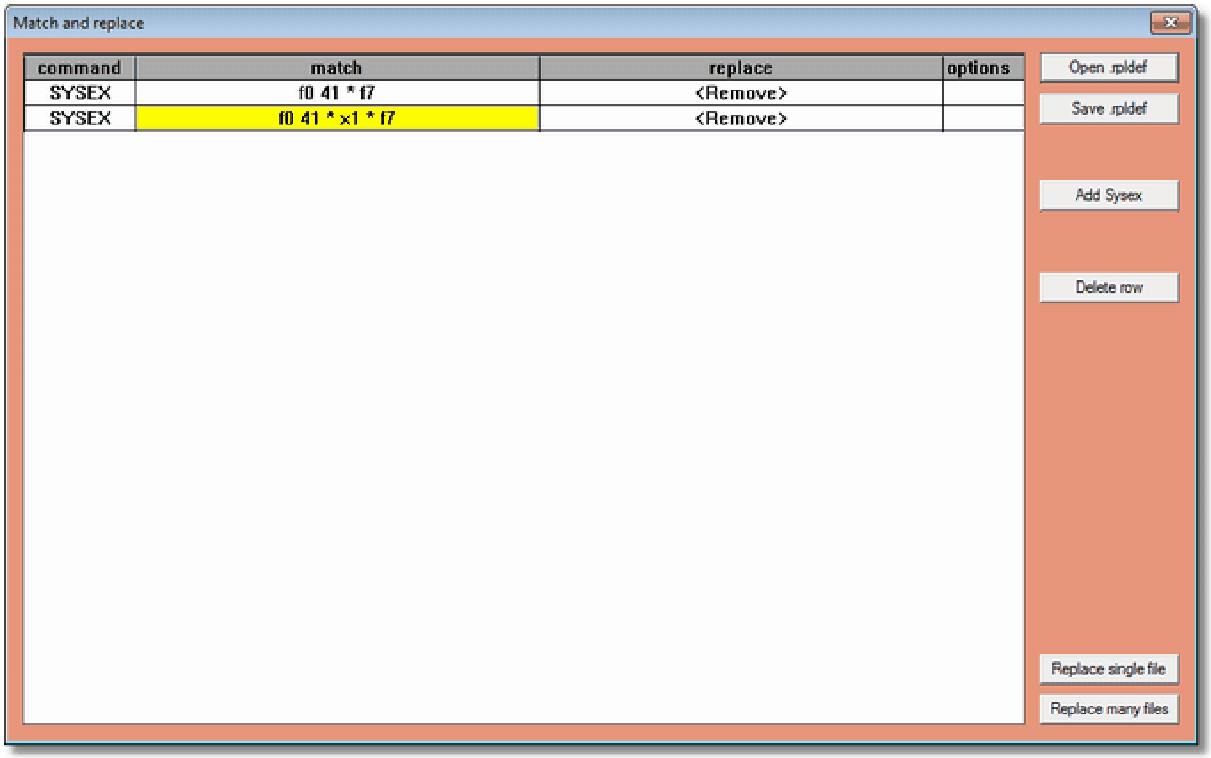
2.2.1.4 Save definition As

Save current loaded definition data to a new *.insdef file. Choose a new name or location for the definition.

2.2.1.5 Close definition

Close current definition. If it contains modified data then it asks if saving the definition is wanted.

2.2.1.6 Match and replace



command	match	replace	options
SYSEX	f0 41 * f7	<Remove>	
SYSEX	f0 41 * x1 * f7	<Remove>	

The match and replace dialog allows to define a list of replace or remove of matching commands.

new definition
when opening the dialog the definition list is empty and can be filled with new match and replace rules.

open definition (.rpldef)
choose a .rpldef text file that contains a definition. The file content is checked and loaded into the list. When list contains unsaved definitions then you will be asked to save the definitions.

save definition (.rpldef)
choose a .rpldef file and location where the definitions should be saved.
The file can be loaded later again and applied to other files or modified.

definition list box
the list box shows a match and replace rule in each row. Options are reserved for future commands (e.g. channel numbers).
Click with left mouse button into a row to select the row. Double click with left mouse button into a row to modify the row.

add sysex
opens a dialog where you can enter or modify a replace definition.
Leave the replace field empty if you want to delete commands matching the given pattern.

Delete row
select one or more rows in the list box and then use delete row button to remove these rows. The content of those rows will be lost.

Close dialog (right top in the dialog frame)
close the dialog. You will be asked to save unsaved definitions.

Replace single file
Choose a MIDI file that where you want to apply the loaded match and replace definitions.

Replace many files (batch conversion)
Opens a dialog where you must fill the input and output folder fields.
Then click button ok to start the conversion. The progress will be shown in the information fields below. Logfile will also contain information about the conversions.

Currently following types of replacing are supported:
Sysex (System exclusive commands)

Only the first matching action is applied
example
rule1: replace sysex F0 7E 7F 09 01 F7 against F0 7E 7F 09 00 F7
rule2: delete sysex F0 * F7
will only apply rule1 even if rule2 matches too (result is that all sysex except sysex F0 7E 7F 09 01 F7 will be removed).

2.2.1.7 MIDI summary

this operation lets choose a MIDI file and then creates a summary of the MIDI file content.
It shows some standard information (version, resolution, number of tracks) and some important commands sorted by tracks and channels and song position.
The result is temporary a **HTML document** that will be displayed with your standard web browser.
The text is written using **language English or German** depending on the selected program language.

This output should help you to get more content information about a MIDI file before and after you applied operations to a MIDI file.

Here you can find an **example** file.

The operation is in file menu and can be started using shortcut Alt-U (make sure that no edit field has the keyboard focus).

Following **commands** are displayed:

- `Metatext (all types of MIDI text)`
- `Tempo`
- `Sysex`
- `Controller`
- `Program`

The operation adds comments to show the meaning of controller, program and sysex commands
The operation uses *.sysexdef files from your application data folder and application program folder to describe standard and user defined sysex commands.

By **default** following **options** are used:

```
[Settings]
SummaryOutputFormat=HTML
SummaryTimeFormat=time
```

Following **options** are supported which can be added in your user settings file **gninsertcontroller.ini** in documents folder (edit using notepad editor):
[Settings]
SummaryOutputFormat=HTML
SummaryOutputFormat=text

SummaryTimeFormat=time
SummaryTimeFormat=midiunit
SummaryTimeFormat=millisecond
SummaryTimeFormat=bar

```
[sysexdefinitions]
sysexdef1=c:\sysexdefinitions
sysexdef2=c:\keyboards\korgi3.sysexdef

[ctrldefinitions]
ctrldef1=c:\users\username\documents\user.ctrldef

[progdefinitions]
progdef1=c:\users\username\documents\user.progdef
```

Hint:
time uses format mm:ss:milliseconds
midiunit is the number of midi units since song start
milliseconds is the number of milliseconds since song start
bar uses format measure.beat.unit where 1.1.000 is the beginning of song

Hint: a batch tool midisummary is available that can create MIDI summaries for all your MIDI files (updating summaries, summary per MIDI file or summary per folder).

Hint: a **.sysexdef** file is a text file with lines
sysexpattern = comment
e.g. F0 7E 7F 09 01 F7 = GM Reset
It supports the same **wildcards** (*, xx) as match and replace operation sysex patterns .
The program generates a user.sysexdef file at first program start in your own documents folder.

Hint: a **.ctrldef** file is a text file with lines
controllernumber = comment for this controller
e.g.

```
7 = volume level 0-127
```

Most important controllers automatically are commented by the program. Use this ctrldef to describe controllers that you use additionally.
The program generates a user.ctrldef file at first program start in your own documents folder.

Hint: by default only first **wheel controller** will be displayed in the summary because they are usually disturbing. Use following setting in gninsertcontroller.ini in your documents folder to show all wheel controllers:

```
[Settings]
SummaryHideWheel=0
```

Hint: a **.progdef** file is a text file with lines
prognr, bankmsb, banklsb = comment

e.g.
1 = Grand Piano
17,0,66 = XG Cheez Organ

prognr is 1-128
bank msb and bank lsb are **optional** and if specified must be 0-127

Hint: different kind of **drums** are currently not supported (e.g. power drums, electric drums)

Hint: By default **GM program names** are defined

Hint: comment of first matching program definition will be used

The program generates a user.progdef file at first program start in your own documents folder.

2.2.1.8 Exit

Closes the application. If an open definition contains modified data then it asks if saving the definition is wanted.

2.2.2 Action

The action menu offers functions to add or delete actions to a definition or apply actions of the definition to MIDI files,

2.2.2.1 Add new action

Creates a new action (yet incomplete) and adds it to the current definition. Fill the fields of the 4 categories with valid data that the action can be used to run.

2.2.2.2 Delete selected action

This deletes the current action tab including all its data. The change will be persistent when saving the definition.

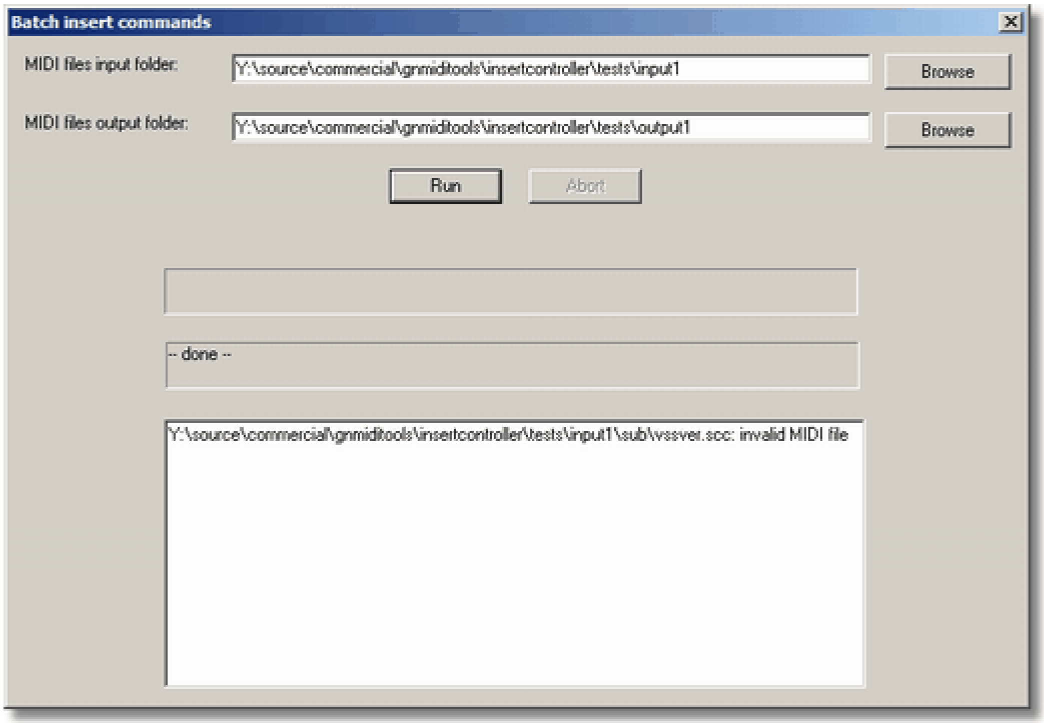
2.2.2.3 Run actions (overwrite single MIDI file)

All actions data fields must be valid that this operation is available.
Choose a MIDI file from a writable folder that will be modified.
A backup of the original file will be created with a versioned title.

2.2.2.4 Run actions (save result in new file)

All actions data fields must be valid that this operation is available.
Choose an input MIDI file and an output MIDI file (should not exist yet).
The input MIDI file will be read then the actions of the definition applied and then the result written to the output file.

2.2.2.5 Batch run actions (MIDI folder)



MIDI files input folder: Y:\source\commercial\gnmiditools\insertcontroller\tests\input1

MIDI files output folder: Y:\source\commercial\gnmiditools\insertcontroller\tests\output1

-- done --

Y:\source\commercial\gnmiditools\insertcontroller\tests\input1\sub\vssver.scc: invalid MIDI file

All actions data fields must be valid that this operation is available. Interactive positioning is not supported for batch run.
Choose an input folder and an output folder (should be empty).
By clicking the start button you can begin the batch conversion.
For every MIDI file in the input folder (or sub folders) the actions will be applied and result written to a corresponding output folder.
Non MIDI files are copied to the output folder.
Errors are logged to a text file and displayed in the errors box.

The batch processing can be aborted pressing the abort button.

2.2.2.6 open Logfile

Running actions will write messages into the GNInsertcontroller log file with infos and error messages.
This menu command opens the log file in Windows standard text editor (e.g. notepad).

2.2.3 Settings

Enter topic text here.

2.2.3.1 MIDI output

When using the option position interactive then the positions can be defined while playing the MIDI song.
This setting is used to determine the output MIDI device that is used to play the song.

2.2.3.2 Workaround for Windows 10 Microsoft GS Wavetable Synth incompatibility

In Windows 10 some computers have problems playing MIDI songs using the Windows 10 internal MIDI player device Microsoft GS Wavetable synth.
The bad behavior is often that a MIDI file can be opened and played once by the device and then for

any song that you would like to play it does not open the file.
It is a compatibility problem since it does not occur with any other Windows version and only with this MIDI device (e.g. playing through MIDI cable or using a soundfont MIDI device is not affected).

Enable the workaround only if you notice such problems playing MIDI.

2.2.4 View

Enter topic text here.

2.2.4.1 Status Bar

Status bar at bottom of the application shows information and error messages during filling a definition form.

It shows the registration status of current application:
not registered ..... Demo is running (14 days max. allowed)
registered by Max Mustermann ..... Max Mustermann has license to use the full version, only this person has permission to use the full software.

2.2.4.2 Languages

This application supports translation text used by user interface. Standard language is English. First supported translation is German language.

Switching to a new selected language will be done at next program start.

translation files are named gninsertcontroller_v1.0_XX.translation where XX is a language code (e.g. de for German). The file must be in the program folder of the application.
They contain the language caption and English text translated into an other language separated by " = "

e.g.

```
[INIT]
language=Deutsch

[menu]
&File = &Datei
```

2.2.5 Window

Enter topic text here.

2.2.5.1 Cascade

Order the opened windows overlapping so that the active window is in front.

2.2.5.2 Tile

Order the opened windows side by side.

2.2.5.3 Arrange Icons

Order the minimized windows (displayed as icons).

2.2.6 Help

2.2.6.1 About GNInsertController

Display application title, copyright and version information.

2.2.6.2 Homepage

Open the homepage in internet browser.

2.2.6.3 Online order

Open order page for GN MIDI Solution products in internet browser.

2.2.6.4 Documentation

Open help documents to this application.

Using F1 key during pressed mouse button you can open help page to most dialogs, menu items, toolbar items.

2.2.6.5 Request access code for this computer

The software license is for use on maximum of 5 of your own computers. After about 14 days the application will ask for requesting an access code that this computer will be used with your license in future.
This can be done online and getting directly a code that can be entered into the dialog or an email option for computers that have no internet.

The code usually only needs to be entered once. Only if you change the computer or install on another computer the code must be requested again.

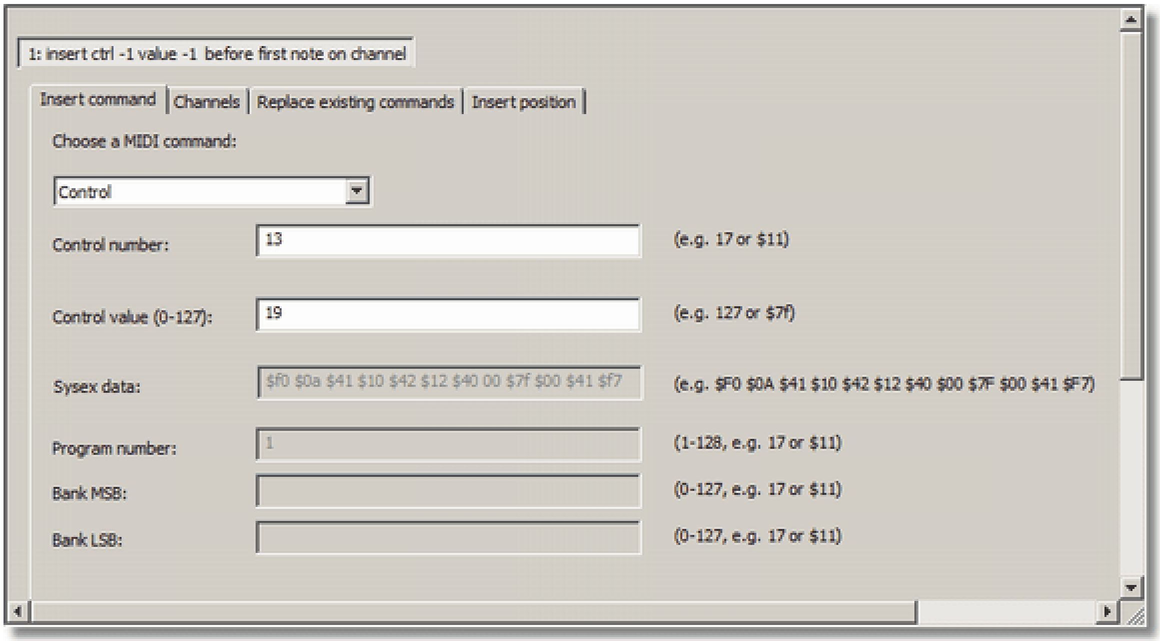
2.2.6.6 Check updates

opens a web page from http://www.gnmidi.com in your internet browser that shows if there is a newer version of this application available.

If you have a valid license the update is free till version 1.99

2.3 Insert controller definition (*.insdef)

2.3.1 Command



1: insert ctrl -1 value -1 before first note on channel

Choose a MIDI command: Control

Control number: 13 (e.g. 17 or $11)

Control value (0-127): 19 (e.g. 127 or $7f)

Sysex data: $f0 $0a $41 $10 $42 $12 $40 00 $7f $00 $41 $f7 (e.g. $F0 $0A $41 $10 $42 $12 $40 $00 $7F $00 $41 $F7)

Program number: 1 (1-128, e.g. 17 or $11)

Bank MSB: (0-127, e.g. 17 or $11)

Bank LSB: (0-127, e.g. 17 or $11)

Following MIDI commands can be inserted by action rules:

- Control
- System exclusive (Sysex)
- Program (sound)
- RPN (registered parameter control)
- NRPN (non-registered parameter control)

Control

using a control command settings of a MIDI device can be changed e.g. volume, effects

a control command requires a MIDI channel (choose one or more channels in next tab)
a control command has two parameters
MIDI control number 0 - 127 (e.g. 7 = volume)
MIDI control value   0 - 127 (e.g. volume 0 = off)

see https://www.midi.org/specifications/item/table-3-control-change-messages-data-bytes-2 for table of common control numbers

System exclusive (sysex f0 ... f7)

using a system exclusive non-standard MIDI settings or MIDI data can be set.

Usually sysex data is represented by hexadecimal numbers enclosed between f0 and f7.

Following syntax for GM Reset sysex data is possible:
F0 7E 7F 09 01 F7
$F0 $7E $7F $09 $01 $F7
$F0 $7E $7F 9 1 $F7
0xF0 0x7E 0x7F 0x09 0x01 0xF7

0x and $ start a hexadecimal value between 0 and $7f (and values $f0 and $f7).
The values must be separated by spaces.

e.g. see http://ascii.cl/conversion.htm for a table that shows decimal and hexadecimal values between 0 and 255

In some cases channel numbers are included in sysex data.
Use the placeholder {CHANNEL} instead of the channel number.
(for selected channel number 1-16 the data byte 0-15 will be inserted for this placeholder).

e.g. f0 7f {CHANNEL} 42 17 00 f7

when inserting into channel 1 the sysex data f0 7f 00 42 17 00 f7 will be inserted into the MIDI file.

Change: since program version 1.2 all sysex data values are hexadecimal values. Therefore used value 10 is same as $10 (= decimal 16).

In earlier versions decimal values were accepted which caused disadvantage that usual hexadecimal values required a $ sign in front.

Program

using a program a MIDI sound can be chosen

A program number is between 1 - 128. To reach more sounds of a device bank MSB and bank LSB numbers can be used (0-127).
Your MIDI device manual should list the available sound programs including optional bank numbers.

The values can be specified decimal (e.g. 127) or hexadecimal (e.g. $7f).

The bank numbers are optional. When omitted then they are not inserted. Existing bank numbers are removed if they are near the insert location and replace option is used.

General MIDI defines a sorted list of sounds, which are available on every General MIDI compatible MIDI device:
GM Level 1 Sound set

Drum channels use drum kit programs (each drum note refers to an other drum sound from the given drum kit).

In General MIDI compatible devices channel 10 is reserved for drums.

RPN (registered parameter control 101,100)

registered parameter controls use parameter MSB and parameter LSB to address a parameter followed by a control data that sets a value to this parameter. Optionally a LSB data value can be set too.

Currently only following RPN are defined by MIDI standard:
Pitch Bend Range (0, 0):   data sets number of semitones
Master fine tuning (0, 1):   sets the tuning displacement by fractionals of semitones
Master coarse tuning (0, 2): sets the tuning displacement by number in whole semitones (center value is 64), LSB data not used
Tuning Program Select (0, 3): rarely used
Tuning Bank Select (0, 4): rarely used
Null (127, 127): resets the parameter address so that data settings do not change unexpected parmeter.

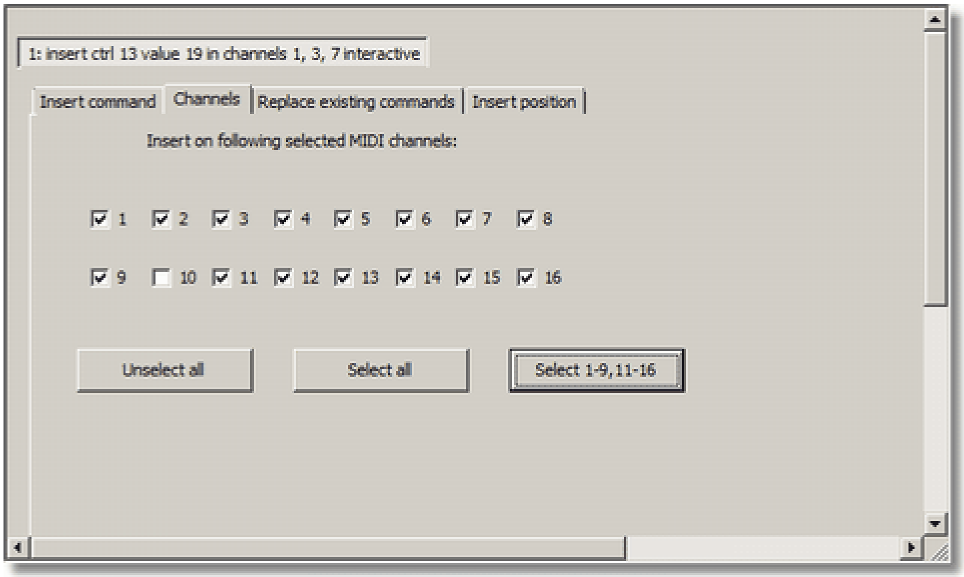
NRPN (non-registered parameter control 99,98)

non-registered parameter controls use parameter MSB and parameter LSB to address a parameter followed by a control data that sets a value to this parameter. Optionally a LSB data value can be set too.
The meaning of the parameters is defined by the MIDI device that is used.

Null parameter:
Null (127, 127): resets the parameter address so that unintended data settings do not change unexpected parmeter.

Hint: RPN and NRPN controller commands share the last used **parameter address** set by controls 98-101 and also share the data and LSB data controls.
A parameter controller is set by changing the parameter address first and then setting data LSB-data value.
This might cause troubles if inserting a new parameter controller into middle of an existing parameter controller since the address would be changed unintended and data sent to wrong parameter.
Usually parameter controllers should not contain pauses between the 3-4 controls that belong together and best practice is to append null parameter address after the data values are set.
This program tests if inserting parameter controller at a specified position is allowed. If there is no MIDI pause inside parameter controller command it should not occur that inserting is impossible but theoretically a MIDI author can do strange things.

2.3.2 Channels



1: insert ctrl 13 value 19 in channels 1, 3, 7 interactive

Insert on following selected MIDI channels:

☑ 1 ☑ 2 ☑ 3 ☑ 4 ☑ 5 ☑ 6 ☑ 7 ☑ 8

☑ 9 ☐ 10 ☑ 11 ☑ 12 ☑ 13 ☑ 14 ☑ 15 ☑ 16

Many MIDI commands need a channel number. Select one or more channels 1-16 where the chosen command should be inserted.

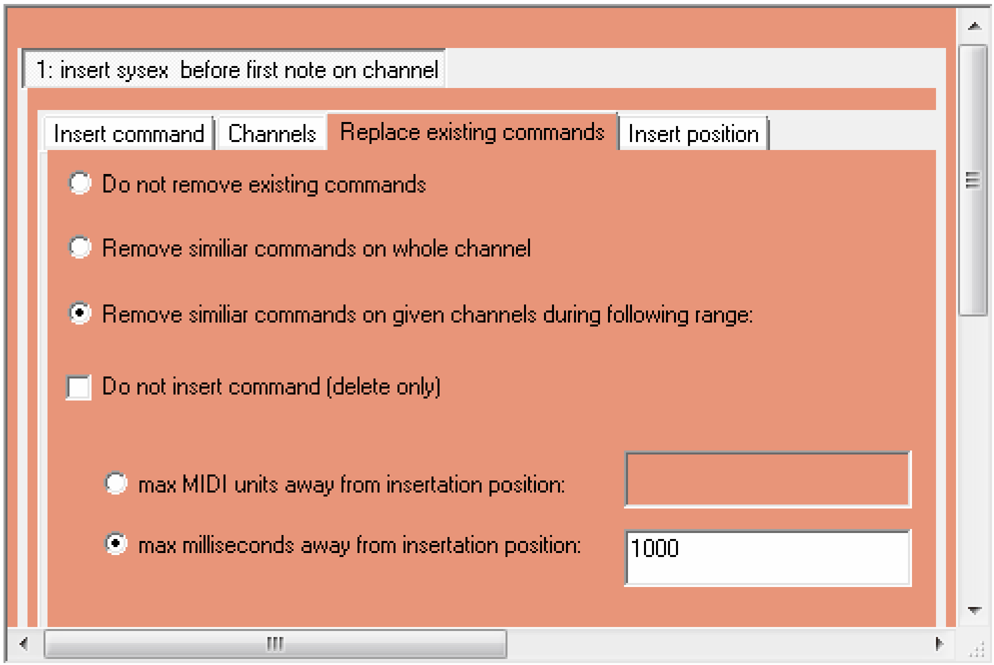
Use one of the buttons to quicklier unselect or select more channel numbers:

Unselect all all check boxes will be unselected so that no channel is selected

Select all all check boxes will be selected so that channels 1-16 are selected

Select 1-9,11-16 all check boxes except channel 10 are selected, all channels except drum channel 10 are selected

2.3.3 Replace



1: insert sysex before first note on channel

Insert command | Channels | Replace existing commands | Insert position

○ Do not remove existing commands

○ Remove similiar commands on whole channel

● Remove similiar commands on given channels during following range:

□ Do not insert command (delete only)

○ max MIDI units away from insertation position:

● max milliseconds away from insertation position: 1000

When inserting a certain command then in many cases it is not wanted that the command is not inserted again if a similar command with other parameter value on same channel near the new position already exists.
Instead optionally such an existing command can be replaced (old command removed and new command inserted).

Check the checkbox option that existing commands near the insert position are removed.

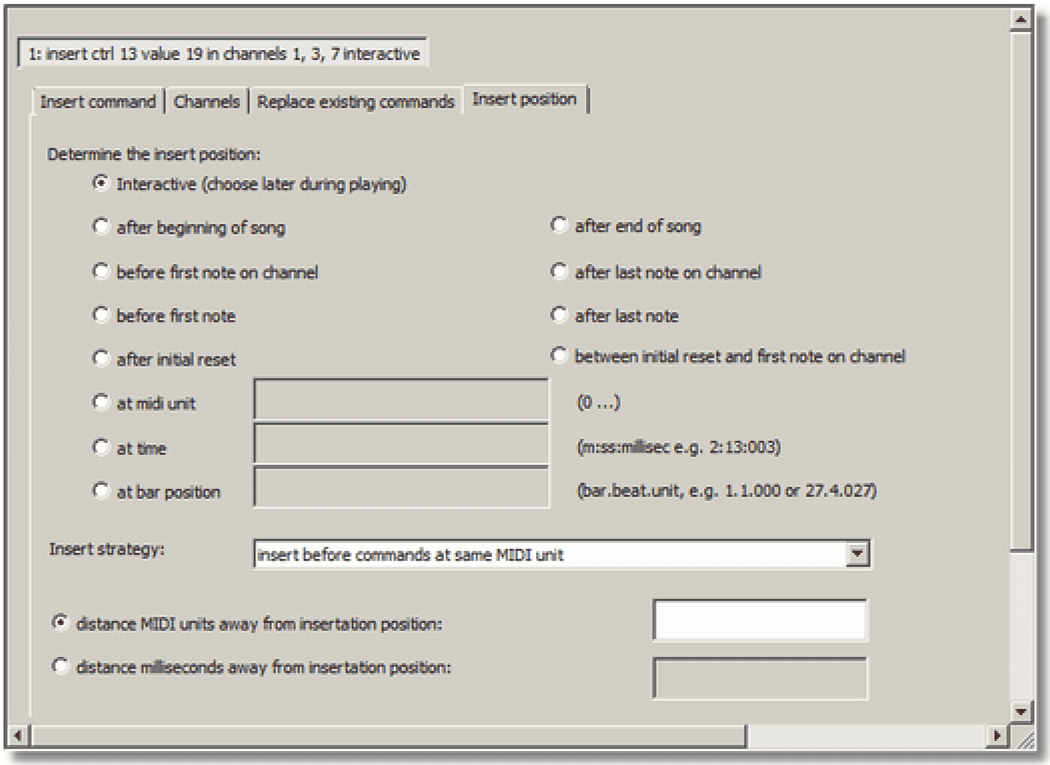
The maximum distance between inserting position and existing command position (same channel, same command type) can be chosen by type MIDI units or milliseconds:

MIDI units number of MIDI units

Millseconds number of milliseconds

Do not insert command (delete only)
check this checkbox if you want to delete a command (and not insert a new command)
(Warning: Using "Do not remove existing commands" together with "do not insert command" makes no sense)

2.3.4 Position



1: insert ctrl 13 value 19 in channels 1, 3, 7 interactive

Determine the insert position:

● Interactive (choose later during playing)

○ after beginning of song ○ after end of song

○ before first note on channel ○ after last note on channel

○ before first note ○ after last note

○ after initial reset ○ between initial reset and first note on channel

○ at midi unit (0 ...)

○ at time (m:ss:millisec e.g. 2:13:003)

○ at bar position (bar.beat.unit, e.g. 1.1.000 or 27.4.027)

Insert strategy: insert before commands at same MIDI unit

● distance MIDI units away from insertation position:

○ distance milliseconds away from insertation position:

For inserting commands a position must be defined where to insert the command.
The position can be defined **interactive, absolute or relative** to existing commands.

Interactive

Choosing position interactive will show a dialog to define time positions before running the action after the MIDI file is chosen and playing the MIDI file helps defining the absolute time positions.

Absolute

at MIDI unit

the MIDI unit number

at time

the time position can be entered in different ways:
minute:second:millisecond (e.g. 2:17:130) or second:millisecond (e.g. 137:130) or millisecond (e.g. 137130)

at bar

use the format barnumber:beatnumber:unit (e.g. 1:1:0 is beginning of song).

at beginning of song

before any MIDI command

at end of song

after latest end of track

Relative

after insert position of previous action

with this option it is possible to insert more than one commands at same position (sequence). A previous action (that does not use an other position) defines the location where the commands are to insert. The other options in this dialog are not used since all position settings are used from a previous action. The commands of a sequence are inserted without pauses.

before first note on channel

searches on the chosen channel first note and inserts the new command before the note

after last note on channel

searches on the chosen channel last note and inserts the new command after end of the note

before first note

searches first note on any channel and inserts the new command before the note

after last note

searches last note on any channel and inserts the new command after end of the note

after initial reset

searches a GM/GM2/GS/XG reset sysex before first note in song and inserts the new command after the sysex command

between inital reset and first note on channel

searches a GM/GM2/GS/XG reset sysex before first note in song and searches first note on the chosen channel.

if both exist then the new command will be inserted between both commands.

if only the first note on channel exists then the new command will be inserted before the first note.

if only the initial reset sysex exists then the new command will be inserted after the initial reset sysex.

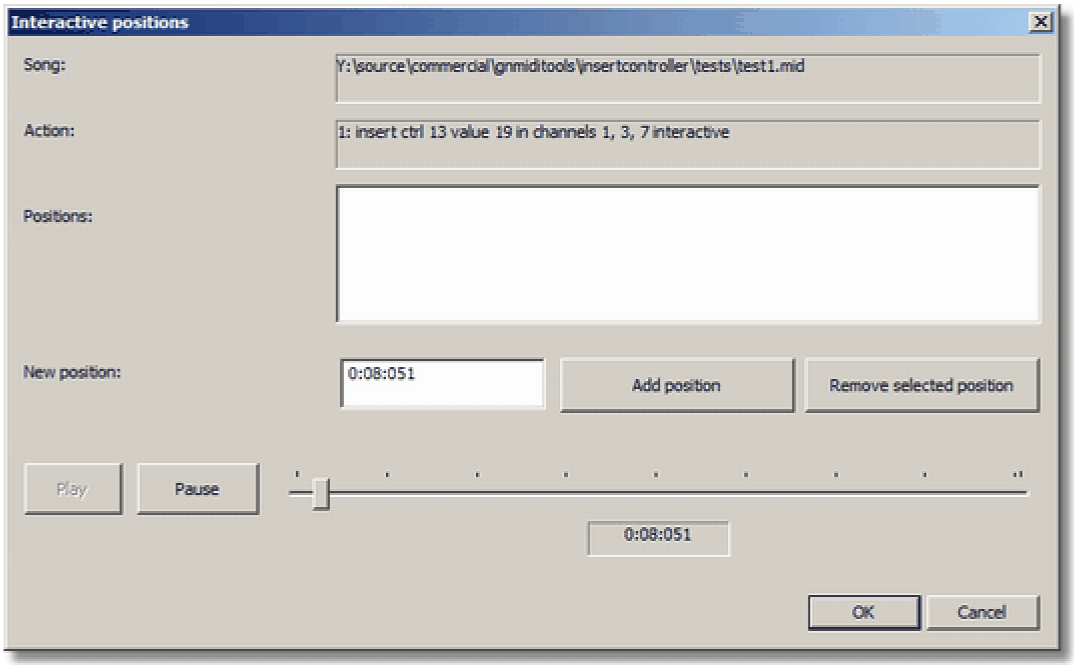
Distance

In all these cases it is possible to enter a distance that the insert position should be kept away from the above command.

The distance can be specified by type MIDI units or milliseconds.

This distance helps to keep the command be enough away from the related command (e.g. initial MIDI reset).

2.3.4.1 dialog Interactive



Song: Y:\source\commercial\gnmiditools\insertcontroller\tests\test1.mid

Action: 1: insert ctrl 13 value 19 in channels 1, 3, 7 interactive

New position: 0:08:051

0:08:051

The interactive position can be used for a single MIDI file.
Before the action can be started a MIDI file must be chosen and this dialog shows.
The valid MIDI file will be played and shows the current position as song time.

The slider can be used to move to an other position.
Pause button can be used to stop the playing.
Play button can be used to start the playing (e.g. after pause).

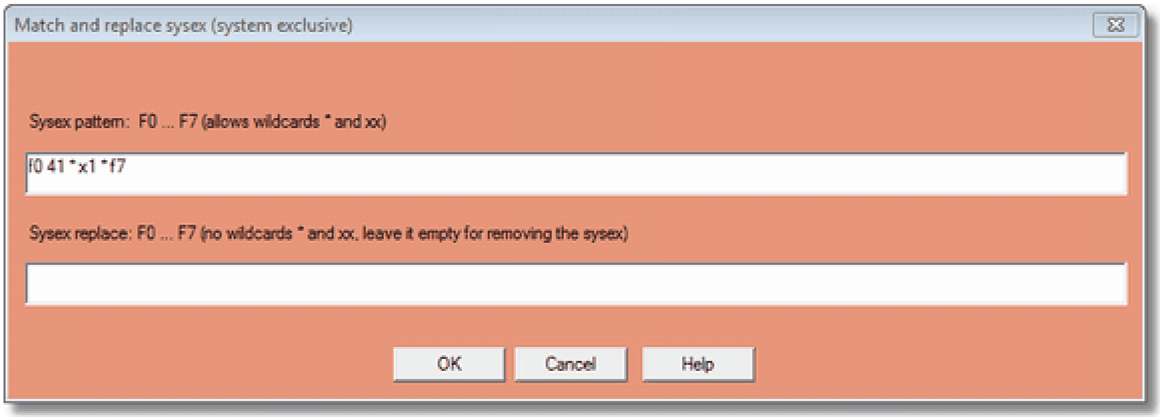
Use button add position to add the current position to the list of insert positions.

Select a position line from the list then it is possible to remove this position from the list if unwanted.

use OK button to continue with the action.
Use cancel button to abort the action.

The list of positions will be used by the action to insert a command at all these positions.

2.4 Sysex match and replace definition



Match and replace sysex (system exclusive)

Sysex pattern: F0 ... F7 (allows wildcards * and xx)

f0 41 *x1 *f7

Sysex replace: F0 ... F7 (no wildcards * and xx, leave it empty for removing the sysex)

A sysex match and replace definition requires a
sysex pattern
and a
sysex replace.

The sysex replace can be left empty to indicate that the **matching commands** should be **removed**. In the .rpldef file this is written by using the **keyword delete** instead of specifying the replace sysex data.

Constant sysex data

Sysex pattern can use constant sysex data and sysex replace must contain constant sysex data.

Sysex data is written as a list of **hexadecimal values** that must **begin with value F0 and end with F7**. The values can also be written with leading $ to indicate hexadecimal values. The values between must be in **range 00 - 7F** (is decimal 0-127).

Hint: The values are **not case sensitive** (F0 is same as f0).
Hint: Smaller values do not need 2 hexadecimal digits (00 is same as 0, 0A is same as A).
Hint: All values are treated as hexadecimal values. A **conversion table** can be found in internet (e.g. https://ascii.cl/conversion.htm)

examples:
F0 7E 7F 09 01 F7 (GM Reset sysex)
$f0 $7e $7f 9 1 $f7

ASCII Text inside sysex

In rare cases sysex might contain readable text in sysex data (e.g. chord names, lyrics). Such ASCII literals might be written as "text" inside the sysex data.
The text will be converted into a series of bytes 00-7F.

example:
F0 00 20 24 00 01 "D#" F7 (is same as F0 00 20 24 00 01 44 23 F7)

Wildcards (place holder) to match variable sysex data

Star * matches none or more bytes in sysex data, it might be used more than once in a pattern.

examples:
F0 * F7 matches any sysex
F0 * 7F * F7 matches any sysex that contains 7F somewhere in data (e.g. F0 7E 7F 09 01 F7)
F0 7E 7F * F7 matches any sysex that starts with F0 7E 7F (e.g. F0 7E 7F 09 01 F7)

xx matches exactly one sysex data byte (that has any value 00-7F)

examples:
F0 7E xx 09 01 F7   matches F0 7E 7F 09 01 F7   but also other sysex data with different value at 3rd position (e.g. F0 7E 00 09 01 F7)
F0 xx xx xx xx F7   matches all sysex data with this length

x can be used to match a nibble (half byte) of a hexadecimal sysex data.

examples:
F0 7E 7F 09 x1 F7   matches F0 7E 7F 09 01 F7, F0 7E 7F 09 11 F7, ..., F0 7E 7F 09 71 F7
F0 7E 7F 09 0x F7   matches F0 7E 7F 09 00 F7, F0 7E 7F 09 01 F7, ..., F0 7E 7F 09 0F F7

Hint: wildcard symbols never match F0 or F7

Hint: use wildcards with care

Index

The GNInsertController software may be tested free for 14 days. Download the demo from http://www.gnmidi.com . We hope that you will find GNInsertController useful for your musical work. If you like GNInsertController please register the software.

Günter Nagler,
GN MIDI Solutions

GNInsertController for entertainers