Denemo User Manual

Richard Shann

richard@rshann.plus.com

Updates for version 1.2.4

Contents

Part I

Getting Started

Introduction

Denemo lets you create musical scores. You can type music in using the keyboard or play it in using a MIDI contoller or the microphone input. You can edit your music - the input display window shows you what you are working on in music notation - and you can playback to check how it sounds. The Print View window shows the final printed score and typeset to the highest standards used in the music publishing industry - thanks to the Lilypond music typesetting program.

This separation of the final typesetting window from the input display avoids the constant dragging about of overlapping notation which is typical of music notation programs. It also means you are less likely to input a slur when you mean a tie, or a whole note rest when you mean a whole measure rest, for example - these often look similar in the typeset and are easily confused, but Denemo's input display makes clear the distinction. Nevertheless, if you do wish to tweak the appearance dragging and re-positioning of some notation is possible on the final typeset display.

1 Input Methods

Most people will start with playing around with Denemo via the pc keyboard. By default, keys `a-g` represent the note names and `0-6` the different durations (whole note, half note etc). When the cursor is appending (blue) pressing key `0` will insert a whole note at the cursor. Pressing the `a` key will append the nearest A. If the duration of the next note is the same you can simply type the note name, it will use the last entered duration. There are keypresses (+/-, and plus/minus on the numeric keypad) to set sharp/flat/double-sharp/double-flat for the next entered note and to sharpen or flatten. Use +/- with the shift key to sharpen or flatten an existing note. Likewise, Shift with a number key edits the duration of a note.

When the cursor is moved on to a note (with the arrow key or mouse) the cursor turns green and then `a-g` edits the note name. To insert a note before a note at the cursor use a double strike `A`,`A-G`,`G`. When the cursor is green the keys `0-6` change the duration of the note, while Shift-`0-6` insert a note before the cursor. (If you use the numeric keypad you can use `Shift-KP 0-6` for this). The arrow keys move the cursor around, the period key adds a dot while `Alt-a-g` add notes to the chord at the cursor. `Ins` inserts a note in a chord whether the cursor is on (green) or after (blue) a note, a double `Del`, `Del` removes it.

When in the appending position the cursor shows as a large blue or red rectangle on a note-position. Blue indicates a note can be appended at that point without overflowing the measure. Red indicates that the measure is already full. When the cursor is a smaller green rectangle it indicates that you are not in the appending position: you can edit the note/object at the cursor or insert before it. The vertical blue line indicates the insertion point.

For a more detailed description of Denemo's various input methods, see A Survey of the Input Methods. You can find more shortcuts by exploring the menu system - e.g. under the Notes/Rests menu the Select Duration submenu gives Remove Dot with the shortcut Control-period shown in blue next to the command.

Alternatively, a list is available in the Help menu. There are thousands of commands in Denemo, so get to know the Command Center where you can search for commands by key words such as slur, ossia, beam, Da Capo, cresc. etc.

Initially, Denemo starts with tooltips popping up almost everywhere. When they become too annoying you can tame them with Help->Turn Excessive Tooltips (Off/On), and you can further delay the remaining tooltips appearing via Edit->Change Preferences.

1.1 Some Common Keyboard Shortcuts

Here are a few of the keyboard shortcuts that are commonly used in Denemo.

- Letters `a-g` edit the note at the cursor to be A-G, if the cursor is in the appending position then notes are added. Letters `A-G` (either `CapsLock` or `Shift`) insert a note at the cursor.
- Numbers `0-6` are used to refer to the note durations Whole Note ... 64th Note. They insert a duration at the cursor, which you then give a pitch to with a note name. With the shift key held (or the CapsLock on) `0-6` edits the duration of the note at the cursor. The corresponding Numeric Keypad keys can be used instead.
- The period (`.`) dots a note, `Ctrl-.` removes a dot, on the numeric keypad Decimal (.) cycles through dotted, double-dotted, no-dot.
- `Alt-0-6` inserts a rest.
- `Alt-a-g` Add notes to a chord. Or position the cursor and use `Insert` to add a note, `Ctrl-Insert` to remove it.
- `7` starts a slur, `8` extends it, while `9` reduces it.
- `Shift-/` inserts a cautionary accidental.
- `Esc` switches between various views of the music which can allow more music on the screen at once.

- TAB alternately inserts a start or stop triplet marker.
- Multiply (* on the numeric keypad) ties/unties the note at the cursor.
- Divide (/ on the numeric keypad) sets/usets the note at the cursor as a grace note.
- Add/Subtract (+/- on the numeric keypad) sharpens/flattens the note at the cursor.

Two-key shortcuts are also available such as "B,s" for start repeat barline and "B,e" for end repeat barline.

1.2 Some Common Mouse Shortcuts

Here are a few of the mouse shortcuts that are commonly used in Denemo (keyboards may vary, for control, shift, alt etc modifiers). Note that under the Input menu is an option to turn on more mouse-friendly buttons, this is the default for Windows users.

- Scroll Wheel pans up/down to bring staffs out of view into the window.
- Shift Key and Scroll Wheel pans the score left/right.
- Control Key and Scroll Wheel zooms.
- Right-button click edits at the cursor.
- Shift-right-button click edits things attached to the object at the cursor.
- Double click left-button describes the object clicked on.
- Ctrl-Shift left-button drag allow you to move stuff in the display if it is cluttered. Typesettting is not affected.
- Ctrl-Shift Key and Right mouse button gives the menu of directives to insert at the cursor.
- Shift Key and Left mouse button drags notes up and down.

1.3 Some Common Uses of MIDI in

The MIDI keyboard too can be customized to perform different actions. Usually, playing notes adds or edits the score (like hitting note names at the pc-keyboard). With the sustain pedal pressed chords are generated (the Alt key can be used for this too). If the interval between the notes played is augmented or diminished it is played on a different channel, so that you are alerted to possible pitch spelling errors (e.g. inputting A-sharp for B-flat). By holding down the Ctrl key a score can be checked by playing the notes - the cursor only advances if the correct note is played, and the Shift key can be held down to route the MIDI keyboard straight to the output (e.g. to check a phrase before playing it in). There is a button in the MIDI-in Controls to do this too. The Pitch Bend controller can be used to set the range of sharps and flats to be used, and the modulation controller can be used to mark sections of the music. With the MIDI controller set as Input source (Input menu) the duration keys create pure durations (notes colored yellow/brown) - you can enter as much of the rhythm as you wish, and then play the notes on top.

Denemo is used via the computer keyboard, MIDI controller or playing in via a musical instrument. Clicking on a musical item in the score moves the cursor to it, and the status bar at the bottom will describe the item clicked. Double-clicking the item will describe it in more detail. Right-clicking on items in the score allows you perform the most common edits on it.Clicking on the initial clef, key or timesignature lets you edit these, clicking to the right or left of the measures visible moves you forwards and backwards in the score. Selections can be made by dragging with the mouse from one note to another or off to one side (the window then scrolls). And right-clicking any menu item gets help and the opportunity to set a shortcuts for the menu item.

1.4 Note and Rest Entry Palettes

On starting Denemo for the very first time a series of palettes are placed near the main window with buttons to insert and change notes. For more serious use these just waste space, so by right clicking on them and choosing Edit this Palette they can be hidden.

Part II
Using Denemo

2 General Concepts

The unit of work in Denemo is a musical score, which can be saved in a single file (with .denemo suffix). This is represented on the screen by a "tab". If you have several tabs open at once they appear just above the music in the main window, and you can switch between them by clicking on the tab.

One score may contain several movements, which you can move between (`PgUp`, `PgDown`), insert duplicate, merge and delete with the Movements menu. A movement is a continuous piece of music with titles etc.

> When you have more than one movement numbered buttons appear in the score title bar for you to navigate by. The first thing on the status bar after any pending accidental is the movement number.

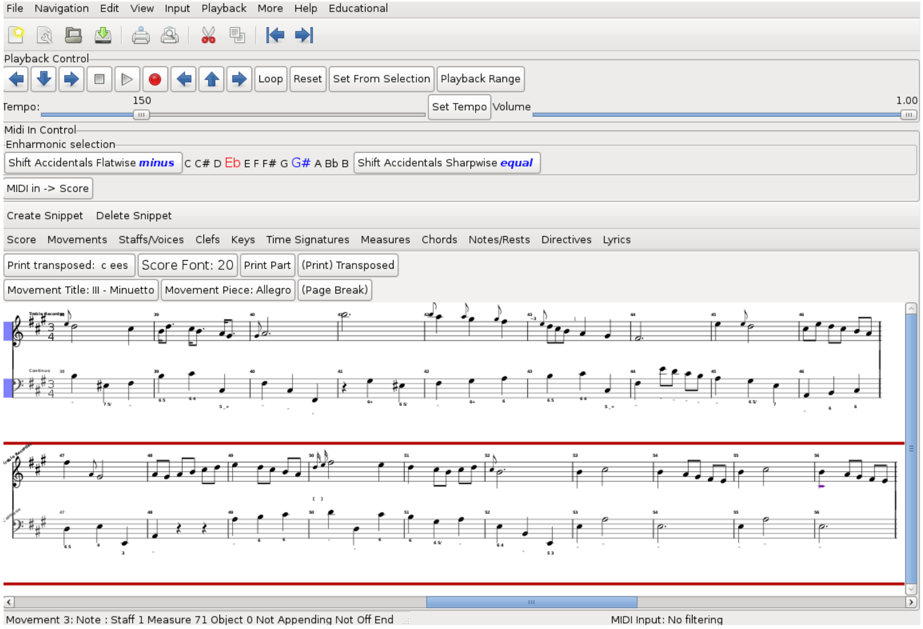
Within a movement there are staffs (arranged vertically) and within the staffs Denemo Objects. These can be notes, chords, key changes, time signature changes and Denemo Directives (see Denemo Directives). The notes and chords are displayed in conventional format (though not fully typeset). The Denemo Directives are used for most things that are not chords or notes etc: Metronome marks, repeat barlines etc are good examples. They can also be attached to chords, individual notes in a chord, to a staff as a whole and to the score as a whole; in this case the directive can be thought of as an attribute of the object it is attached to. Each Denemo Directive carries its own display method. For example the Close Repeat barline appears as a Denemo Object in the input display, while a Directive attached to a staff (e.g. Instrument Name, or Smaller Staff Size) may appear in a menu under a tools icon to the right of the staff. In the case of "Smaller Staff" the directive directs the LilyPond typesetter to make the staff smaller, and it can be edited from the menu under the tools icon to the left of the clef. Another tools icon appears if directives are attached to voices within a staff, again to the left of the clef, below the staff one.

If there is more than one voice on a staff it is best displayed on a separate staff in the Denemo Display, for ease of editing - the clef is drawn pale and there is no timesignature for such extra voices so it is easy to understand what is going on. The Print Preview window as usual shows the final typeset appearance.

In the Denemo display a cursor shows where the next note will be entered/edited. It is red for an over-full bar, blue for appending into an under-full one and green when editing (i.e. on an already entered object).

There are two sorts of titles. Book titles have a separate title page with titles for individual movements which can be listed in an automatically generated table of contents. Simple titles give a Title and (optionally) movement titles on the same page. Comments can be place on chords in the score and these can be automatically collected into an appendix.

Figure 1: Denemo Main Window


File Navigation Edit View Input Playback More Help Educational
Playback Control
Loop
Reset
Set From Selection
Playback Range
Tempo:
150
Set Tempo
Volume
1.00
Midi In Control
Enharmonic selection
Shift Accidentals Flatwise minus
C C# D Eb E F F# G G# A Bb B
Shift Accidentals Sharpwise equal
MIDI in -> Score
Create Snippet
Delete Snippet
Score Movements Staffs/Voices Clefs Keys Time Signatures Measures Chords Notes/Rests Directives Lyrics
Print transposed: c ees
Score Font: 20
Print Part
(Print) Transposed
Movement Title: III - Minuetto
Movement Piece: Allegro
(Page Break)
Movement 3: Note : Staff 1 Measure 71 Object 0 Not Appending Not Off End
MIDI Input: No filtering

When a file is loaded it opens at the point where you left off editing it. The position and size of the window is restored as well as the position and size of the source pdf you are transcribing from.

3 Denemo Input Window

The main window has menus and toolbars at the top, and palettes at top and right hand side. Which menus and palettes are shown are selected via the View menu. At the bottom is a status line showing which movement you are in and what sort of object the cursor is on. If there are any MIDI filters that active they will be noted at the right in the status bar. The small colored rectangle is the Denemo Cursor, which shows where notes etc will be added/edited next - the insertion point is marked as a blue vertical bar. Selected music is indicated by a green background.

In between is where the music input is displayed, the Denemo Display area. When zoomed out you see just the few measures you are working on. By dragging the red bar at the bottom of the score upwards you get space for more of the music. If you have many staffs they may not all fit: you can still drag the red bar upwards to see several lines of just one or two staffs for instance. If you need more room you can hide the menus (using the View Menu), and arrange a "page view" of the input music - useful when using playback.

The menus in Denemo are unusual: they not only let you do some particular action, but also each menu item lets add the command to a palette, enquire what the action does in more detail and set keyboard/mouse shortcuts for the action. A single keystroke can be set as a shortcut simply by pressing the key while the menu item is selected. All the extra functionality of menu items can be accessed by right-clicking the menu item, while the usual left click is for executing the action itself. The menus can always be torn off for working with particular items (e.g. working with different movements or with measures, dynamics etc.). When you place a commands in a palette, and you can choose or create any number of these, free-floating or docked in the main display. Also available by right-clicking is creating new actions - often by modifying ones that are already there - using the Scheme scripting window.

3.1 The Main Menubar

The Main Menubar has menus for overall control of the program. It contains the following submenus:

- File
- Navigation
- Edit
- View
- Input
- Playback
- More
- Educational
- Help

3.1.1 File

> This menu has been re-organized and much expanded, so the following is only a rough guide.

Use the **File** menu to perform global operations related to storing and retrieving from file systems. The File menu contains the following Menu Items:

Menu Item	**Description**
New	Create score from scratch.
Open	Open an existing Denemo file
AddăMovements	Open an existing Denemo file and append the movements from it to the current score

Menu Item	Description
AddăStaffs	Open an existing Denemo file and append the staffs from the (first) movement of it to the staffs in the current movement.
Open Recent	Open a file recently edited.
Open Standard Template	Create score from a stock template.
Open from Gallery	Create score from an example in the Gallery.
Open Custom Template	Create score from a template you have saved.
New Window	Open another instance of Denemo.
Open in New	Open an existing Denemo filein a new tab (or "page") so that both can be used at once.
Save	Save the current file.
Save As	Save the current file with a new name.
Save Parts	Saves a set of parts (staves) in LilyPond format.
Export PDF	Run LilyPond on the current file to create a PDF.
Print	Displays the score in a pdf view, from which it can be printed.
Print Excerpt	Displays only the selection in image viewer, from which it can be saved.
Print Current Part	Displays one part taken from the score in a pdf view, from which it can be printed.
Close	Close the current score but keep other Denemo scores open.
Quit	Close the current scores and exitDenemo.

3.1.2 Edit

This menu has been expanded, so the following only covers some of the possibilities.

Use the Edit menu to access common editing commands.

Menu Item	Description
Undo	Undo the pitch and rhythm just added.
Redo	Redo the pitch and rhythm just added.
Select	Menu items for selecting music in the score.
Cut	Copy the selected notation to the clipboard and deletethe selection.
Copy	Copy the selected notation to the clipboard.
Paste	Paste the copied notation at the cursor point.
Paste LilyPond Notes	Paste text as LilyPond notes. Open a .ly file in a text editor, copy some notes and then execute this command to capture them into Denemo.

Menu Item	Description
Edit Object	A general interface to editing whatever is at the cursor, especially Directives attached to notes/chords.
Change Preferences	Set external programs (LilyPond, Browser, Graphics Editor etc.), autosave interval etc. These values will be remembered when you re-start the program
Commands Behavior	
Quick Settings checkbox	One touch setting of keyboad shortcuts

3.1.3 View

Use the View menu to toggle toolbars and palettes used with the mouse.

Menu Item	Description
Hide/Show Menus	Three ways of showing the Denemo display area are supported. With/without the menus and as a multiline page. This cycles through the three displays.
Typeset Music	Shows the score as engraved by the LilyPond typesetter. You cannot edit in this window, but you can position the cursor in the display and drag some items to indicate how you want an item edited.
Command Center	Search for commands by entering likely words, set up one or two key shortcuts, load customized shortcuts or commands ...
Score Layout	Show the score layouts associated with this score. The score layout is the final section of the LilyPond syntax that describes how to layout the staffs, voices, lyrics, titles etc.
Lyrics	Shows any lyrics for the current staff/voice. Each verse has its own tab, when selected the lyric placement shows in the Denemo display, so you can adjust by typing in the lyrics view window
Snippets	Shows a menu bar with snippets - custom rhythm patterns. Enter snippets by selecting patterns and then use them as the an extended version of the prevailing duration, or insert them and overlay the inserted rhythms with pitches via keyboard, MIDI-keyboard or microphone
Tools	The conventional icons for Open, Print etc
Playback Controls	When checked a set of playback/record controls are placed above the display
Midi In Control	When checked a set of controls for a connected MIDI keyboard are placed above the display

Menu Item	Description
Score Titles, Controls etc	If this is checked any Titles, indent settings etc applying to the score can be shown as buttons above the display. They must be created with their graphic field set for this.
Object Menu	Menus of all the Denemo commands listed under type of object from Score down to individual notes and directives.
LilyPond	Pops up a window for customizing the syntax that Denemo generates for the LilyPond engraver to typeset
Scheme Script	Pops up a window for showing scripts written in Scheme. These can be executed or saved as new commands. Sequences of commands can be recorded here.
Score	This hides/shows the main Denemo display.
Palettes	Gives access to palettes of buttons that can be arraged to make commands available via mouse or keyboard (via the Activate Palette Button command).
Display Zoom	Zoom the main Denemo display (usually done with ctrl-mouse wheel).

3.1.4 Input

Choose whether to use external sources (Audio from the Mic input or MIDI) for Denemo. To use these you may need to review the settings in Edit->Prefs first.

Here too there are commands to change the way the inputs (keyboard, mouse and MID) behave.

3.1.5 More

Add commands to Denemo. Extra commands are available (those which not everyone will want) via this menu. More Commands gives commands shipped with Denemo, while My Commands gives ones that you have created locally. Note that the extra commands can also be loaded at the menu where you are looking for them (you right-click on a menu item, and if there are more commands for that menu the More Commands for this Menu item will show),

3.1.6 Playback

This menu has been expanded, so the following only covers some of the possibilities.

Use the Playback menu to listen to your score. Denemo lets you hear your score, using an internal synthesizer. Playing of sections of the music and looping is possible (including editing as the music loops, so that you can listen to different possibilities) as well as recording the audio output (mixed with anything you play on your MIDI keyboard if you have recording set in the MIDI controls)

Menu Item	Description
Play	Plays from the start marker to the end marker
Stop	Stops the playback

3.1.7 Help

> This menu has been expanded, so the following only covers some of the possibilities.

Use the Help menu to get help using Denemo.

Menu Item	Description
Help	Launch a browser to view this manual.
Chat to Other Users	Connects you to a site with other users to give advice.
About	View Denemo version information.

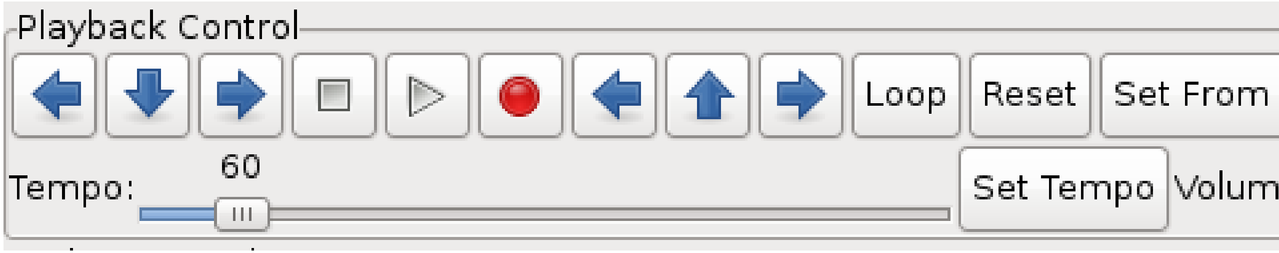
3.2 Toolbar

Use the Toolbar to access common Denemo commands via a mouse click. The Toolbar contains the following icons:

Icon	Description
🗋	Creates a new document.
Open	Opens the Open File dialog box.
Save	Saves the current file
🖨	Runs LilyPond to convert the current file to PDF and sends it to the printer for printing
↶	Undoes the last action.
↷	Redoes the previous undo action.
✂	Cuts the current selection to the clipboard.
⎘	Copies the selected notation to the clipboard.

Icon	Description
📋	Pastes the current clipboard item at the cursor position.
⇤	Jumps to the first measure of the movement.
⇥	Jumps to the last measure of the movement.

3.2.1 Playback Controls



This panel has been expanded, so the following only covers some of the possibilities.

Use this panel to control playback of the current movement. It has the following controls:

Icon	Description
	Runs DenemoFirst script.
	Runs DenemoGoBack script.
	Moves the start playback position backwards. Runs DenemoGoBack script.
	Sets start playback position to cursor. Runs DenemoStartToCursor script.
	Moves the start playback position forwards. Runs DenemoNext script.
	Runs DenemoRewind script.
	Runs DenemoStop script.
	Runs DenemoPlay script.
	Starts playing back while recording from MIDI-in.(Runs DenemoRecord scriptto do this).
	Moves the end playback position backwards.Runs DenemoGoForward script.
	Sets the end playback position to the cursor.Runs DenemoEndToCursor script.

	Moves the end playback position forwards. Runs DenemoLast script.
Loop	Runs DenemoLoop script.
Conductor	When you press Play the music advances past the cursor only while the mouse is moving.
Reset	Internal Synth only. Performs a reset on the synth. Also resets the start/end play positions to whole piece (the visual display shows just a red bar at the start)
Panic	Jack only. Emits jack panic.
Set from Selection	Sets the selection to be the start/end. Note first use in a session is buggy.
Tempo	For movements with a single tempo this tempo is changed for rest of the playback. If the movement has an embedded MIDI tempo change, the change will be altered proportionally. The movement tempo setting itself is not altered - you are just altering the playback.(See next button or Movements->MIDI Tempo for setting that)
Set Tempo	Sets the overall tempo of the movement to the value fixed by the Tempo slider
Volume	Reduces the current volume by the % set. Again, this applies only to playback and proportionally affects embedded dynamics.
Set	Calls the DenemoSetPlaybackIntervalToSelection script.

3.2.2 Midi In Controls

This panel has been extended, the following is a rough guide.

Use this panel to control MIDI input to the current movement. It has the following controls:

Icon	**Description**
	Sets the end sharps and flats to use.
MIDI->Score	Directs the MIDI in to the score or to a MIDI recorder. The MIDI recorder is active when Record button is pressed. When you wish to return to score editing, this button re-directs MIDI-in to the score.
Delete	Deletes your MIDI recording.
Convert	Converts your MIDI recording to Notation. (*Not yet at all practical!*)
MIDI Status	Tells you how MIDI in will be treated. Press Control/Shift/Alt keys to modify or click and select Listening to input MIDI, Checking notes in the score against input MIDI or Appending/Editing at the cursor.
Switch to Play Along Playback	When you press Play with this option set, the music will not advance past the cursor until you play the note. (Mute the current staff to prevent double sounding of notes)

3.3 Object Menu

The object menus give you actions (such as insert, delete ...) ordered by the objects (notes, staffs, measures ...) that they act on.

Lyrics	Create and delete lyrics.
Menu Item	**Description**
Score	Settings that apply to the whole score.
Movements	Insert/remove navigate, change the properties of movements.
Staffs/Voices	Insert/remove, navigate, swap, change the properties of staffs or voices
Clefs	Insert, change or set the clef.
Keys	Insert a change or set the key.
Time Signatures	Insert a change or set the time signature.
Measures	Insert/remove navigate measures.
Chords	Insert/remove notes from a chord. Note that Chord Symbols are markings on notes (see Notes/Rests menu)
Notes/Rests	Menu items for inserting/changing/deleting note and rests, beaming, ties, slurs, setting the notehead style, grace notes, triplets, transposing, markings (e.g. hairpins) attached to notes and more
Directives	These are objects inserted between the notes of a score to place marks or start/stop spanning effects such as textual cresc. dim., cadenzas, ottava, fret diagrams and more.

Score In the score properties dialog things that affect the whole piece can be set. In the score properties dialog you can include directives to apply to all the movements.

> When a file is loaded it opens with the cursor at the last saved position.

Adding Markings Dynamics, Slurs, Trills, Tempo Indications etc are in the Notes/Rests menu (if attached to a note or chord) or Directives menu (if standalone objects)

You can collect together your favorite ones, either by assigning shortcuts, or creating a palette of favorite commands which just invoke the action you use.

3.3.1 Palettes

Palettes are collections of commands which can be executed by clicking with the mouse or typing `p` followed by (part of) the label of the button followed by the `Enter` *or* `Return` key. (The palette must be selected - you can do this while typing the label or by using a button).

In the View menu you can select palettes of commands to use. Palette items can be invoked from the keyboard (see the Edit Menu) or by clicking on them. Right-clicking on a palette button allows you to edit the button or the entire palette. You can dock the palette in the main display, arrange it as a column, line or block and much more.

You should make palettes your best friend - when you find a command in the menu system that you need right click on it and choose Add to Palette. This way you will build up one or more collections of commands that you need. By right clicking on a palette button you can choose a name for the command that suits you, and even make special notes about its use in the tooltip.

3.3.2 The Command Center

In the View menu you can show the Command Center. This has a list of all the Denemo commands, hidden or not, with their shortcut(s). The box at the top left gives full details of the currently selected command and its location in the menu system. Below that is a search box for finding any command you need by matching words in its description or label. By choosing "Fuzzy" you will make the search less strict - one word could be missing.

The search starts as you type in the first letter of any key words you want, so check after each letter to see if you have found the command you need. To search for further commands matching your keywords press the arrow next to the search box.

The command center is also where you can set one-key shortcuts such as `Ctrl-x` and two-key shortcuts such as `Shift-a` followed by `Shift-a` (which is written as `A,A` next to the command). You can also lookup which command responds to a given shortcut, and execute the selected command here.

In addition you can save and load whole sets of shortcuts and commands.

4 Music Snippets

4.1 Overview

Music snippets are short selections from a voice which can be stored on the Snippet tool bar. They have two main uses: they can act as a repository of "motifs" which can be inserted (`Ctrl-Space`) at the cursor, and they can be used as an extension to the "prevailing duration" idea. In the second use, by selecting a snippet as you enter pitches the durations are assigned from the next step in the snippet. The first case is especially useful when entering a complex rhythm which is repeated many times in the piece. In all cases the snippet can contain all sorts of attributes and markings, slurs, beaming indications, ornaments etc, and these will then all be entered with a single keypress or as you add pitches.

4.2 Basic Use

The Snippet tool bar has only one button initially: "Create". This button lets you create a snippet from the current selection. The selection should be a contiguous selection of objects in a single voice. Once created the snippet appears on the snippet tool bar as a button - the label is generated to indicate the content, and this label can be further edited by clicking on the button and choosing "Edit Label". Other options on clicking a snippet button are to insert the snippet, to select the snippet (in which case as you enter pitches they follow the rhythm of the snippet, including any slurs, trills, or articulations that are in the snippet) etc. `Ctrl-Space` inserts the selected snippet or (if none is selected) inserts the snippet that you specify by number. Shift-Space selects the next snippet (or first if none selected), to unselect a snippet choose a duration (0, 1...) to be used instead for entering pitches.

In the insertion case, you may have two or three snippets that apply to a given score - special bits of text that appear over notes for example - and you can insert them using Ctrl-Space, 1 (or 2 or 3 ...) as needed.

In the pattern-following case you will be selecting snippets using Shift-Space and then playing in notes (and backing up using backspace in case of error - the pattern backs up for you).

5 The Print View Window

The print view window is shown by checking View->Print View. At the top are buttons to turn continuous typesetting off or on, and to select Full score or Parts. The mouse scroll buttons enable vertical panning and with Control pressed, zooming. Use Ctrl+Shft for finer control zoom. Right click for help with tweaking beam angles, slurs etc.

Right clicking in this pane also allows you to drag offsets to apply to items that LilyPond has positioned badly. Slurs and beaming can be altered here, and page or line breaks added. Or you can select an amount of padding to apply to some object. This is generally a better way of indicating that it is too close to something else, since other objects will move away from it as needed - that is, if you later edit something it will behave more intelligently.

6 The Score Layouts Window

The score layouts window is shown by checking View->Score Layout.

You can use layouts to print the music in different ways. For example, one layout may have the score transposed, or be for just two parts printed together.

You can customize any layout in this window - positioning lyrics above the staff rather than below for example, or setting the staff groupings.

You can make Denemo Directives conditional on the score layout, so that for example a page break only applies when printing the full score, or is just for printing one part. In this way if, for example, one part does not need separate first and second time bars, then these can be marked as not being for that layout. See Directives->Conditional Directives for setting this.

> The score layout can only be edited graphically when first created. Once reloaded from disk it is reduced to its LilyPond text form and can then only be edited in the LilyPond view.

6.1 Score and Movement Button Bars

The View->Titles, Buttons etc" checkbox makes two horizontal button bars visible (if they have buttons on them). What the buttons show are titles/composer etc for the score and for the current movement. The score button bar holds the movement indicator buttons for scores that have several movements - you can click on one of these to move quickly to a different movement. In addition other score-wide settings create buttons on the score titles bar and movement wide settings on the movement titles bar. The menu items that populate these are under Score->Titles etc., and Movement->Titles etc and other menus.

7 A Survey of the Input Methods

7.1 Introduction

Denemo allows you to explore all the actions (insert/delete/edit/navigate...) using the menus. So, for example, inserting notes can be found under Notes/Rests while inserting a Staff comes under Staffs/Voices. When you have found the action you need hovering over the item will give an explanation of what it does. The keyboard or mouse shortcut to use for the command is in blue on the right. You can set the keyboard shortcut just by pressing the desired key while the menu item is selected. You can right click on the menu item to set a a two-key keyboard shortcut or mouse shortcut to activate the action. In addition you can customize the mouse actions to initiate commands, and set the cursor that will show while certain mouse conditions apply.

Denemo offers several ways of entering music: typing at the keyboard, playing in via a MIDI keyboard (controller), playing in acoustically using a microphone or choosing items from the menu system.

Using the menus for everything would be very slow, but next to each menu item is the keypress that you can use instead. If there is no keypress shown, and you want to use that item often, then right click on the menu item and you can set a key combination for that menu action. Right clicking is also useful for getting a description of what the menu action is.

Typing at the keyboard can be very fast, if you forget which keypress you need you can consult the menu system to find out. This method is good for touch typists.

For entering a lot of already written-out music, playing the music in via a MIDI keyboard or microphone can offer the fastest and most musical method.

One way to do this is enter the music as pure rhythms, ignoring the note names. The durations show as yellow/brown notes. Each duration sounds its own pitch and length as you enter the durations. As you start each measure the bell will sound so that you can keep your eyes on the score you are copying without needing to look up and check that you haven't miss-typed. If you are a reading musician you will find that you tap in the rhythm rhythmically which helps to keep your place in the piece you are entering.

Once you have entered the rhythms for one or two measures you play the notes via MIDI - they automatically fill in the rhythm for you. When you play notes in via MIDI they give their pitches to the rhythm you have notated. By this means you can enter a piece of music in the time taken to play it twice - once to give the rhythm and once to give the pitches. If the piece modulates strongly you may need to shift the set of accidentals used to match the score. If entering pitches via a mic you need to navigate to the start point, but this isn't needed for MIDI in.

7.2 Playing Notes into Denemo - MIDI

Denemo can take pitches directly from MIDI input. Select Input->Midi Input. The control panel allows you to choose the enharmonic range, the default centers around the initial key signature (e.g. E-flat to G-sharp for C-major). If you enter a diminished or augmented interval the note will be played in a separate MIDI channel which can be used to alert you to pitch-spelling mistakes (e.g. entering A-flat when G-sharp was meant, you will get a lot more augmented and dimished intervals if you have the enharmonic range set wrongly for your piece). Under Input->MIDI are various "MIDI filters" that allow you to control how the MIDI input is used: with none active the behavior is as if you had entered the notes using the pc-keyboard, but with the following advantages

- The octave, accidental and note name are all entered in one press of the MIDI key. The cursor automatically advances so you can continue to play in notes.

- The duration keys enter the yellow/brown duration-only notes, when you play a MIDI key the cursor automatically moves to the first of these.
- Holding down the `Alt` key (or sustain pedal) lets you enter chords.
- Holding down the `Ctrl` key lets you check the pitches of a piece already entered. The cursor only advances if the note played is the one at the cursor.
- Holding down the `Shift` key lets you listen to the MIDI keyboard without affecting the score.
- The problem of entering the wrong enharmonic is largely avoided by Denemo's simple pitch-spelling strategy. The more extreme intervals are played in a separate MIDI channel so that if, for example you enter F - A-sharp it sounds quite distinct from F - B-flat.

The MIDI filters are scheme scripts, so they can be tailored to do whatever you wish. One example is a filter that enables you to enter Figured Bass figures by playing the notes corresponding to the figures while holding down the bass note. Another is the "AngryDelete" filter. With this filter on notes are entered normally with the cursor advancing automatically, but if you make a mistake and press the wrong note just hit the next one much louder and it will make the correction for you! MIDI filters are found under Input -> MIDI

7.3 Playing Notes into Denemo - Audio (Souncard Mic Input)

Denemo can listen for, and detect the pitch of notes on the mic input of the computer; it doesn't attempt to guess the rhythm - such systems do not work well - but you will find that playing the notes in time will help you to play them in, as well as make playing them in a musical experience rather than a chore.

Playing the notes in can be much quicker than using the keyboard since the note octave and accidental are all given just by playing the note. If you are able to play a musical instrument then this will probably be much faster for you than typing note names, octave shifts and accidentals at the computer keyboard. Using the headphones-out of an electronic keyboard avoids "noises-off" interfering with the pitch detection. Many microphones and pickups benefit from some pre-amplification - it is worth getting the level right before you begin.

When you select the Input->Audio on the Main Menu the Pitch Recognition window pops up. While the mouse pointer is inside the score drawing area the score is sensitive to pitches heard via the microphone input. The background colour of the score changes to show that the notes will be entered into the score.

There are two ways of using the pitch entry - Overlay mode (default) and Insert mode. The button marked Insert causes notes to be entered into the current measure in the prevailing rhythm - the mode is set to Insert for this.

The button marked Overlays overlays the notes already present with the pitches you sound. There is a third button, marked Tuning, which is a state-of-the art musical instrument tuner.

Music is entered into the measure which holds the cursor. If you are overlaying a rhythm already entered, then the first un-overlayed note in the measure is overlayed by the note detected. You can delete the overlay using the regular delete keys, or clear them altogether if you want to start over in a measure. If you are in Insert mode then the notes detected will be inserted at the cursor position.

Use the enharmonic shift to select whether Bb or A# should be entered when you enter the given pitch - you can usefully go as far as B# and Fb.

Use the transpose control to shift up or down by octaves.

Most of the other settings would require study of the Aubio documentation to understand, but the one marked threshold may be useful to make the detection less sensitive to ambient noises if using a microphone with an acoustic instrument.

The best set-up is to plug the headphones-out socket of an electronic keyboard into the mic input, and choose a piano setting on the electronic keyboard.

If you don't have any musical instrument that you can plug directly into the mic in, then you can use an acoustic instrument with a microphone, in which case move your microphone closer or further from your instrument to get reliable detection. Too close and you get double detections, too far and you get missed ones. To check for good detection open a piece of music, set Overlay mode and put the cursor in the first measure and play the piece in – the notes should all turn blue if you have perfect detection. It is worth while getting perfect detection – more than one or two miss-detects per piece of music and you may want to use the Insert rather than the Overlay method.

The Audio Input button introduces a special entry mode where the pitches you play in will overlay the rhythm, appearing as blue notes. You can delete any wrong pitches using the usual delete keys, without deleting the rhythm. In fact if you have an "interloper" (an extra spurious note) you can delete it and the other pitches will all move along to their correct places.

Another method of playing music in acoustically doesn't involve entering the rhythm separately. For this select Input->Audio and then choose Insert instead of Overlays on the Pitch Recognition Panel that pops up. With Insert the sounded notes are entered as in the prevailing rhythm. The same applies if you have MIDI - by choosing a rhythm (e.g. half-note, quarter-note, or a custom rhythm pattern) and playing in the pitches you can enter the music into a blank score.

7.4 Using the Denemo From the PC keyboard

7.4.1 Note Entry

Basic Note entry is to type a number key `0-6` for the duration of the note followed by a letter key `a-g` for the note name. (The shorter duration notes are available, but the keys `7,8,9` are used for starting and extending/reducing slurs). If the duration is the same as the last entered note then just the note name is needed. The duration can be dotted with the period key, and the octave adjusted with comma (for down) or apostrophe (for up).

When you append music by pressing a duration key with a MIDI controller active, you get a brown note, indicating a duration which has not been given a pitch. Once you have pressed a note-name it becomes that note printed in black. Prior to that it is a pure rhythm and will play back as a drum beat and will print as a space.

When not appending music, you need to use the `shift` key to insert a new note, or set the `Caps Lock`. So `Shift-a`,`Shift-a` inserts the note A etc. Plain `a` edits the note at the cursor to be an A. Likewise with the numeric keypad the `Shift` key can be used to edit a note duration, while the plain number keys insert the duration of the note at the cursor. (If you do not have a numeric keypad, you have to use the number keys - best to plug in an additional USB keyboard with numeric keypad, they are very cheap nowadays).

7.4.2 Chord Entry

The standard keybinding for entering a note in a chord is the `Insert` key, `Del`,`Del` deletes. This enters the note at the cursor position, use `-` and `+` to flatten/sharpen the note before you enter it (avoiding hearing both notes if you flatten/sharpen afterwards). There are also commands to enter a named note `Alt-a`, `Alt-b`... are the standard shortcuts for this. Spacer rests have shortcuts too, but on Windows machines you will need to reclaim the `Windows` Mod-4 key from Microsoft or re-define the shortcuts.

8 Transcribing from Facsimile or Hand-written Score

8.1 Opening Source PDF

In the File->Open menu is an option for opening a PDF containing music to be transcribed from. The PDF is opened in a Source View window. Right-click in this window at the point that your transcription will start. This will create a link in your score at the cursor, which can be clicked on to open the source view window with the blue highlight marker on that point in the score. (Depending on the position of the top of the page you may need to scroll the Source View window up or down to bring it into view). When you re-open the score to check or edit at some point you can click on a nearby link and the source file will be opened and the position highlighted once again.

8.2 Taking Snapshots of a Facsimile

This feature is now largely redundant as the above method will usually be easier

This is particularly useful for difficult-to-read scores. Each measure of the original can be captured as an image and stored along with the Denemo measure that is your transcription. This makes the transcribing process much easier as you can glance at your original without losing your place in a large messy manuscript. Capturing the images of the measures is done using a screenshot type of process - you need to have the original visible on the screen so you can indicate the extent of each measure.

First display the original score or part on the screen using your pdf viewer or other image display software. Then, in Denemo, choose one of the Capture Measures commands. For a Full score you can capture all the parts at once - choose Movement->Capture Score->Capture Measures (Score). If you have parts use Staff->Capture Score->Capture Measures (One Staff).

Now you can continuously enter measures by left clicking at each measure. You can adjust the start position of the measure (which is normally the end of the last one) by right-clicking. As you enter a measure image a Denemo measure is entered into the score and the image placed underneath it. To stop entering images press a key on the keyboard. There are also commands to delete an erroneous measure-image and to insert a new one - N.B. when you delete the image for a measure all the other measures shift along by one, so they are under the wrong bars until you insert a new screenshot for the measure.

Now you can transcribe the music straight from the images directly under the measure you have selected. LaManina.denemo is an example: the original print made in Venice is quite difficult to decipher and has a couple of errors. In this Denemo file each measure for each part can be seen directly beneath the transcription. You will notice that in places the original measure was split across two lines; at the moment there is no mechanism for joining up the two halves, so instead the following measure has an image containing the remainder of the measure plus the next measure.

8.3 Using a Proof-Read PDF

If you send the PDF output of your score to someone to proof-read they can make PDF annotations to it. In the File->Open menu choose Open Proof Read PDF to open a proof-read PDF. It will open at the first annotation, and, by clicking on the note the annotation applies to, the annotation is transferred as a comment into the score before the note chosen. When you have done all the annotations on the first page advance to the next page with annotations using the button "Next".

There are limitations - the PDF must be created from your score with point-and-click (the default) and you must not alter the score in a way that would invalidate the point-and-click. (Basically, don't edit again until the proof-reading is done, always good practice!) Also, enter the annotations in order from the start and don't edit before you have entered all the annotations, as this could invalidate the point-and-click.

9 Score Setup

Denemo provides different ways to set up scores:

- Create template with the special name "default.denemo". If this is placed at the top level of your templates directory (that is ~/.denemo/templates/default.denemo), then it will be opened each time you start a new score.
- Create an init.denemo score with everything you want to start up with, and any actions you want to be taken before you begin. This will be opened when you first start Denemo (that is once per Denemo session). See 10.11 on page 32 for details on what actions you can have Denemo perform before you start.
- Use a Template
- Use shortcut keys to add staves and set attributes
- Use Menus

9.1 Using Templates to Setup Scores

Denemo comes with a few preinstalled templates. You can also create your own templates as well, in fact you can use any Denemo file as a template, just open it using File->Open->Open Custom Template. (The only difference is that, opened this way, the score is a new i.e. untitled score)

1. Navigate to File, Open Template. The Open dialog box appears.
2. Double click on a template name from the list of templates. The dialog box closes and the template appears.
3. Adjust the clef, key and time signatures as needed.

Part III
Advanced Topics

10 Customizing Denemo

10.1 Adding and Editing Key/Mouse Shortcuts

Denemo allows you to choose which keys activate which commands (shortcuts). It also allows you to choose Mouse press/release gestures (in combination with keys such as `Shift`, `NumLock`, `CapsLock` etc).

See The Command Center for setting shortcuts. The following method can also be used. To choose a shortcut for a menu item, select the item (it becomes highlighted) and press the key that you want to become the shortcut. For mouse shortcuts, right-click the menu item and choose the Create Mouse Shortcut, to change the pointer shape during mouse operations invoke the Command Center with the Edit Shortcuts option. The Set Mouse Shortcut Dialog requires you to set which button and action you want to use and then to hold/lock the keyboard modifier while clicking on the big button. The setting you have chosen is shown, and you click ok to accept it.

> Mouse Shortcuts are tricky to set - you are able to control what happens on mouse button press and release, which can be used to do things like drag notes up and down. But they are tricky to set up!

If your choice of keyboard shortcut already belongs to another command, you are warned about this, and given the chance to change your mind or steal the shortcut. Also, you can choose whether to distinguish between keypresses with, say the `NumLock` down and those without or to ignore it. If the `NumLock` is set but no command for that key specifically requires it then the setting of `NumLock` is ignored and the keypress is treated as if the `NumLock` was off.

The menu item that activates the command shows the shortcuts that exist in bright blue lettering. PrsL-`CapsLoc`, for example, is the Press action on the Left mouse key while the `CapsLock` light is on. In the default command set this is linked to the BeginSlur command, so that together with the RlsL-`CapsLoc` for the release of the left mouse button, you can "draw" slurs by using the mouse. Similarly, `Shift` with left mouse button (MveL-`Shift`) is set to allow you to drag notes up and down the staff. (This is using Command MoveNoteToCursor).

10.2 More Commands

Adding More Features

It is possible to add more commands to the set that Denemo ships with. Some of these are shipped with Denemo, but not added automatically so as to keep the menu sizes manageable. Generally, you right-click to find more menu items to add; once added you can set a keyboard/mouse shortcut and keep the command in your default setup. Scheme to be executed at startup can be placed in the denemo.scm file.

An example - quicker dynamics selection It can be slow to choose dynamics (`Ctrl-D`) because the list is quite long to move down (or move to the mouse). You can define a list of the dynamics you actually want to be readily available as for example: (define DenemoDynamicList '(("f" "127" "Forte") ("p" "127" "Piano"))) This makes forte available just by pressing Return and piano available by pressing down arrow Return. (The others via More of course). If you place this in (home directory)/.denemo-x.x.x/actions/denemo.scm then it will be defined when you start Denemo.

10.2.1 Recording a Denemo Macro

You can also get Denemo to record a frequently used sequence of actions - it can be installed in the menu system and given a keyboard shortcut just like any other command.

To do this choose View->Scheme Script. (You do not have to understand what a Scheme Script is to use this!). In the window that pops up check Record Scheme Script and then do the set of steps you wish Denemo to record.

As a simple example, suppose you wanted a command to delete the next note (there are commands to delete the previous note and the current note, so why not?). You would enter some notes, put on Record Scheme Script and then move the cursor right two steps and delete the previous note (with Backspace if that is your shortcut). (Each time you use a command you will see the Scheme syntax for that command entered into the Scheme window). Now turn off Record Scheme Script and you can experiment with your new command by pressing Execute in the Scheme Script window. To save this new command for future use, you right click on a menu item (in the menu where you would like the command to appear) and choose "Save Script as New Menu Item", this will ask you to make up a name for your new command as well as a label for the menu item etc.

You can save the command in your default sequence of commands (via Edit->Customize Commands...->Manage Command Set->Save as Default Command Set). If not you will be asked if you want to save your new commands when you exit. Otherwise you can re-load via the More->My Commands menu item.

10.2.2 Editing a LilyPond Tweak

There is another way of adding your own favorite LilyPond tweaks, which is by modifying a tweak that has already been done. This can be done by using the text-edit dialog on an existing Denemo Directive (usually found under the Advanced button provided by an edit script). Here you can see the actual LilyPond text that will be inserted, and there is a button to create a script to generate whatever you choose to enter. You can also enter the name of graphic images (.png files) that are to be used to represent your item in the display, and say where it should be positioned.

Once you have created the script, it can be saved in the menu system by right clicking on an item in the menu where you want to place the command, and choosing "Save Script as New Menu Item" as above.

10.2.3 Writing Scheme Scripts

If you are a programmer you will have guessed that you can edit the Scheme Script window to create any command you want. Even if you are not familiar with Scheme you may find that you can adapt other scripts to do what you want.

A good example of this is a script to insert a particular LilyPond directive into the score. This is all that many scripts do: it is easy to see the piece of LilyPond in the Scheme Script window, and by changing it you can create a new command.

For example, from the More menu select the command /menus/ObjectMenu/Instruments/Orchestral/RehearsalMark (the files are laid out in folders/directories in the same way as the menu system itself). Then choose Get Script from the right click menu. With this command its script is appended to the Scheme Script window. Here you can see the part that says

```
(d-DirectivePut-standalone-postfix "RehearsalMark" " \\mark \\default" )
```

which is inserting the LilyPond directive "\mark \default" which inserts the default rehearsal mark. (The extra backslashes are needed to tell Scheme that you literally mean a \ sign). You can change this to insert any other LilyPond that you need - always doubling the \ signs. The you can save as a new menu item, or use Save Script to customize the script you started with.

More ambitious programmers will need to know all the commands available. Besides the complete Denemo command set (the list is given in the Command Center window) there are the following additional Scheme procedures defined.

Commands Getting User Input All these commands are invoked from scheme as (d-Command args...)

- GetChar returns a string containing a single character from the user (blocks waiting for a keypress)
- GetKeypress returns a string representing a keypress from the user (blocks waiting for a keypress) (e.g. Up for the up arrow key etc.)
- GetCommand returns a string containing the command name for the user's keypress (blocks waiting for a keypress)
- GetCommandFromUser

- GetUserInput takes three strings, displays and returns the user's response as a string.
- RadioBoxMenu takes an arbitrary number of arguments, each argument is a pair, presents a menu of the first items in the pairs and returns the second item as chosen.
- RadioBoxMenuList like RadioBoxMenu but takes a list
- GetUserInput takes three arguments (title, prompt, suggested value) and returns the string typed by the user in the pop-up dialog or #f if Cancelled.

Commands Getting Information about Object at Cursor

- GetType returns a scheme string indicating the type of the current object
- GetNoteName returns a scheme string, giving the note name a-g of the current note
- GetNote returns a scheme string, the note name, accidental and octave of the current note (LilyPond notation)
- GetNotes returns a scheme string, the notes of a chord separated by NULL characters
- d-InitializeScript deprecated - init.scm is now run automatically. (parameter: action-name) runs a script init.scm in the menupath attached to the action whose name is given. Used by commands that need a time-consuming initialization step which can be skipped on subsequent calls.
- d-PutNoteName takes a string argument, a note in LilyPond notation. Changes the note at the cursor. Not for use with multi-note chords.
- d-DiatonicShift Deprecated.
- d-NextObject moves cursor to next object, returning TRUE if current object has changed
- d-NextChord as d-NextObject, but skipping non-chord objects. chords includes rests and chords with 1 or more notes.
- d-NextNote as d-NextChord but skipping rests (i.e. chords with 0 notes).
- d-NextStandaloneDirective as NextObject, stopping on a standalone directive. Using d-DirectiveGet-standalone tag a directive of a particular tag can be found.
- (d-Directive-type? optional-tag) where type is one of score, scoreheader, movementcontrol, header, paper, layout, clef, timesig, keysig, staff, voice, standalone, chord or note.This returns #t if the cursor is on a directive of type (with tag optional-tag if optional-tag is present) else #f

- (d-DirectivePut-standalone? tag) inserts a standalone directive with the given tag at the current cursor position and places the cursor on it.
- (d-DirectiveGetTag-type) where type is one of score, scoreheader, movementcontrol, header, paper, layout, clef, timesig, keysig, staff, voice, standalone, chord or note. This returns the tag if the cursor is on a directive of type else #f
- (d-DirectiveGetTagForTag-type tag) returns the tag passed in if the cursor is on directive of type with that tag, else it returns the tag of the first directive of type that is present at the cursor, else #f
- d-WarningDialog Pass a string argument to pop up a warning.
- d-GetOption (parameter string of options). Takes a null separated set of options and pops up a dialog offering them to the user. Returns the one chosen or #f if the user cancels.
- d-GetMidi
- d-PutMidi
- d-PlayMidiKey
- d-BassFigure
- d-GetNoteAsMidi
- d-RefreshDisplay
- d-InputFilterNames sets the status bar
- d-Chordize Ensure that even a single note is treated as a chord - needed for some LilyPond constructs (e.g. fingerings)

Standard Denemo commands that pop up dialogs will work as usual if no argument is passed to them from Scheme. In general if a string consisting of strings of the form "name=value" is passed these will be used and no popup will occur. The field "name" will be given the value "value". The \0 is a NULL character that separates the assignment strings. As a shorthand if there is only one value being passed you can just pass the value. The names of the fields depend on the action being called. So for example:

```
(d-Open "filename=myfile.denemo")
```

will open the file "myfile.denemo".

This feature is being rolled out, and currently works for:

- d-InsertLilyDirective (directive, display. minpixels) (deprecated function)

- d-AttachLilyToChord (prefix, postfix, display) deprecated see d-DirectiveGet-chord-* and d-DirectivePut-chord-* below
- d-AttachLilyToNote (prefix, postfix, display) deprecated see d-DirectiveGet-note* and Put below
- d-StaffProperties understands a couple of property=value settings
- d-InitialClef
- d-InsertClef
- d-InitialKey
- d-InsertKey
- d-Open filename
- d-ScoreProperties (fontsize= size of font to be used for score)

10.2.4 Denemo Directives

Denemo directives are things like special barlines, rehearsal marks that appear amongst the notes, as well things like trills that are attached to notes and modifications to other things in the score to give them a different appearance or behavior. If they are objects in their own right ("Standalone Directives"), like the first time bar marker then they appear betwen notes and the cursor can be placed on them. If they are attached to other things (such as a page break attached to a movement) then they are modified with the command used to create them.

Double clicking on Standalone Directives, gives information about the directive, while right-clicking will edit it.

10.3 LilyPond Editing

Introduction From version 0.7.8 we have the ability to edit the LilyPond output within Denemo. This approach immediately makes Denemo able to do many more things (e.g. multiple verses for songs) with the music still editable from within Denemo. The gallery of examples and the standard templates contain examples which you can use.

These can be used without knowing the LilyPond language (provided a suitable template or example file exists). Alternatively, with a general idea of how a LilyPond file works tweaks from the LilyPond documentation can be inserted into the LilyPond output and stored with the Denemo, leaving open the possibility of further editing of the notes within Denemo without the need to re-apply tweaks or keep separate LilyPond files.

10.3.1 Using the LilyPond Window

Under the View menu is a Show LilyPond item which pops up a window with the LilyPond output in it. The text is interspersed with buttons which enable you to hide or show the various sections or to create custom versions. The text in bold can be altered and the alterations are kept in the Denemo file.

The two windows are kept in sync, so you can move back and forth between editing textually and editing in Denemo.

Right clicking on the text gives a menu for actions on the LilyPond text. LilyPond text can be inserted between notes, and the final section (the score layout) can be turned into editable text (see Score Layout). There is one for moving the cursor to the LilyPond text for the current Denemo object. Moving the cursor in the text window with the arrow keys causes the Denemo window cursor to move in synchronism.

This menu also includes a Print command that operates on the visible LilyPond text in the window. This means you can open specific custom score layouts and print from them, or even make a temporary edit for just one print. If you save the score with custom layouts then these are remembered. So when you reload and the custom score layout will be printed.

Detail Several custom layouts can be kept, selecting them in the Score Layout view enables the same Denemo file to print a variety of things from the same music input.

For example, a full score or a set of parts, or several voices on group of staffs or even a piano reduction.

The various Voices/Staffs for the different Movements are separated by buttons. These are labelled by enumerating the movements and staffs in order, so the first voice in the first movement has the music defined as "MvmntIVoiceIMusic" and so on. Within these music blocks you can insert arbitrary LilyPond text between notes, (the insertion points are marked by grey blocks) and the text inserted will appear as a LilyPond directives in the main Denemo window.

The main Denemo window also moves its cursor to correspond with where you are editing. (It should of course move the cursor immediately you click on a point in the music, but, as yet, it only moves the cursor when you press a key just before the note name - one of the arrow keys will do).

Custom layout blocks can be created by right-clicking on the Standard score layout and selecting create custom score layout. The Score Layout window allows creation of customized layouts via a GUI. If you print a single part from all movements, you get the standard scoreblock for this and can add it to your custom score block so that a single print command prints, e.g. both a full score and parts. Using the \book {} block you can put these into separate files (stored in the folder .denemo in your home folder).

The music defined by MvmntIVoiceIMusic is then used in the score blocks at the end of the LilyPond window, by the expression \MvmntIVoiceIMusic. This means that the same music can be output in several different ways, so that the same Denemo file can contain custom score blocks to output the music as a Piano Score with several voices or separate parts, for example.

The definitions for MvmntIVoiceIMusic actually look like this:

- `MvmntIVoiceIProlog = {\MvmntIVoiceITimeSig \MvmntIVoiceIKeySig \MvmntIVoiceIClef}`

- `MvmntIVoiceIMusic = {\MvmntIVoiceIProlog \MvmntIVoiceI}`

Where \MvmntIVoiceI is the actual block of notes you have written, while the other definitions hold the time signature, clef etc. By using these, you can print the same music with different clefs, still maintaining the ability to edit the notes in Denemo.

Note that the normal Denemo Print commands are still operate as they do if you never look at the LilyPond window. What happens is that if you select the Print Current Part menu item then this creates a tailored standard scoreblock for that part. However, if you use the File->Print command then it prints the first custom scoreblock that is open (visible) (or the standard scoreblock if you do not have any custom scoreblocks).

> Clicking (as opposed to using the arrow keys) in the LilyPond text does not move the cursor in the Denemo window. When you delete a LilyPond directive textually you have to move the cursor to start re-inserting it.

10.4 More Features

10.4.1 Piano Staffs, Orchestral Scores etc

A piano staff can be added using the Staff->Add Staff menu. In addition, piano staffs, and staff groups such as choir staff can be created using the Staff Groupings menu under the staff menu. You can set a piano staff within a staff group by setting successively the StartPiano and StartGroup contexts on a single staff, or more generally setting StartPiano, EndPiano on adjacent staffs within a staff group.

10.5 Single Staff Polyphony

Use the Staffs/Voices->Voices menu for placing more than one voice on a staff. You can set the initial voice number from this menu (voices 1 and 3 are stem up with slurs and ties etc adjusting to suit, voices 2 and 4 are stem down). Directives can be placed in the music to change voice, see Directives->Typesetter->Voices menu.

You will also need spacer rests for voices that are silent, and commands to displace rests vertically and horizontally (see Notes/Rests->Rest Insertion menu), and commands from the Voices menu under the Staffs/Voices and Directives->Typesetter menus.

10.6 Entering Vocal Music

Choose Lyrics->Add Verse. You need to have the Lyrics View visible (see View menu). Each verse has a separate tab, with the current verse for the current staff being visible at any one time. Stanza numbers can be entered as well as font-changing commands. Use -- to extend syllables over more notes (see LilyPond documentation).

10.7 Entering Figured Bass

To enter figures choose Notes/Rests->Markings->Figured Bass. There are some shortcuts that enable all the work to be done with the numeric keypad. A brief summary is given by right clicking the option. The conventions are described in the LilyPond docs.

10.8 Fret Diagrams

Fret Diagrams can be placed on the score using the command Fret Diagram in the ObjectMenu->Directives->Markings menu In addition by assigning a Denemo staff to display as fret diagrams, chords can be entered in standard notation which will then be dislayed as fret diagrams. See ObjectMenu->StaffMenu->StaffPropertiesMenu->FretDiagrams for this.

10.9 Tablature

Music can be displayed in tablature - the default is for standard guitar tuning but others can be set. The menus to explore are:

ObjectMenu->StaffMenu->StaffPropertiesMenu->Tablature

ObjectMenu->Directives->Typesetter->Tablature

It is possible to display the same music as both notation and tablature, and example of how to do this will be found under File->Open->Open Example.

10.10 Entering Chord Symbols

To enter chord symbols choose Notes/Rests->Markings->Chord Symbols. A brief summary is given by right clicking the option. The conventions are described in the LilyPond docs.

Chord Charts can also be created - there is a Chord Charts palette for these.

10.11 Musical Scores that Do Things!

By saving a Denemo score with a script defined (in the script window) you can create music lessons, automatic midi player... the possibilities are endless. When you open such a score, the script is run - it can take user input and manipulate the score, or do other actions as your fancy takes you.

There is a special score init.denemo that is run on startup. By editing this you can startup with whatever template and whatever actions you wish to be performed. If you set it to do something that quits Denemo, you may need to delete the file before using Denemo normally again. Your local init.denemo is stored in the directory .denemo/actions in your home directory.To create it put the script you want in the script window and use SaveAs selecting ~/.denemo/actions/init.denemo as your file to save to.

10.11.1 What Happens at Startup

On starting the scheme script ~/.denemo-(version number)/actions/denemo.scm is executed (where ~ means your home directory and version number is 1.1.2 or later).

A denemo file called init.denemo is/was loaded, but this is deprecated.

In addition, on startup a set of keyboard and mouse shortcuts, and a selection of optional menu items are loaded. Other sets are available via the Edit->Customize Commands ...->Manage Command Set dialog.

10.12 Starting Denemo - Command Line Options

Denemo –help shows the options at startup.

Part IV

Technical Reference - Denemo Directives

11 Denemo Directives

Denemo Directives give attributes to objects that are not built-in but can be changed by the user.

Denemo Directives can be attached at almost every level of a Denemo score and can modify the behavior of the element concerned. They contain fields to describe how the element's properties should be modified, either in the display or in the printing. Elements, such as clefs notes etc have their own built-in display and print properties; Denemo Directives allow you (or scripts you invoke) to modify them for many more purposes than the built-in set allows. This means Denemo can grow - you can add features - without getting a new version.

For example the drum clef is not built-in to Denemo. Instead a directive attached to the clef has a field (graphic) set to an image of the drum clef, and another field (postfix) set to the LilyPond syntax for a drum clef, while another field (override) is set to indicate that these values should replace the normal ones, rather than adding to them.

The elements that can be modified in this way are the following:

score: the LilyPond fields (prefix and postfix) are placed at the start of the score and just before each movement. The display field is shown at the top of the display.

scoreheader: Attached to the score. The postfix field is put inside a \header{} block at the start of the score.

movementcontrol: Attached to a movement. The prefix field is placed before the movements \score{} block, the postfix after it.

header: Attached to a movement. As scoreheader but for \header[] blocks inside the movement's score block.

paper: Attached to the score. The postfix is placed inside a \paper{} block.

layout: Attached to a movement. The postfix is placed inside a \layout{} block in the movement's scoreblock.

clef: Attached to a clef or clef change. The graphic holds the displayed icon, gx,gy its position. The postfix field is put into the music at the point where the clef is found, replacing the normal text if the override is set.

timesig: Attached to a time signature or time signature change. The graphic holds the displayed icon, gx,gy its position. The postfix field is put into the music at the point where the time signature is found, replacing the normal text if the override is set.

keysig: as timesig but for key signatures. (e.g. used to supress keysignatures in drum clef).

staff: The postfix field modifies the whole staff context, with the display field printed at the start of the staff

voice: The postfix field modifies the voice context, with the display field printed at the start of the staff containing the voice

standalone: A directive not attached to an music element - it comes with the music and is used for things like repeat bars etc.

chord: The prefix field is emitted before the LilyPond for the chord and the postfix after it.

note: The prefix field is emitted before the LilyPond for the note and the postfix after it. Examples are fingerings attached to notes etc. Again the display and graphic fields a placed in the display positioned relative to the note via the coordinate fields gx, gy (for the graphic) and tx, ty for the display text.

The Directives are sufficiently important to have their own commands.

d-DirectivePut-type-field where type is one of score, scoreheader, movementcontrol, header, paper, layout, clef, timesig, keysig, staff, voice, standalone, chord or note and field is one of display, tx, ty, gx, gy, graphic, prefix, postfix, override, midibytes. These commands take two arguments, a tag (string) and a value to set. For example:

```
(d-DirectivePut-note-postfix "LHFinger" "3")
```

will put the fingering 3 on (after) the note at the cursor.

d-DirectiveGet-type-field type is one of score, scoreheader, movementcontrol, header, paper, layout, clef, timesig, keysig, staff, voice, standalone, chord or note and field is one of display, tx, ty, gx, gy, graphic, prefix, postfix, override, midibytes. This function returns the value in the field or #f if there is no directive with the given tag at the cursor.

d-DirectiveDelete-type type is one of score, scoreheader, movementcontrol, header, paper, layout, clef, timesig, keysig, staff, voice, standalone, chord or note. This function returns #t or #f if a directive with the given tag was deleted.

11.1 The Directive Fields

The fields of the Denemo Directive can control the Denemo Display and the LilyPond output.

The fields in d-DirectiveGet/Put have the following meanings:

postfix - A fragment of LilyPond to be output (after the LilyPond for any object the directive is attached to).

prefix - A fragment of LilyPond to be output (before the LilyPond for any object the directive is attached to).

display - text to be shown in the Denemo Display

tx,ty - where to show the text in the Denemo Display

graphic - For directives that are in the music this is a .png image to be shown in the Denemo Display (the directory bitmaps holds these). The graphic can be saved for a command using the right-click -> Save Graphic command, after selecting a portion of the print preview as the image required. For directives attached to the score, movement etc the string set here will be displayed on the button in the button box for that sort of directive (see Show Score Titles etc in view menu for showing this button box).

gx,gy - where to show thegraphic in the Denemo Display

minpixels - how much space to leave for this item in the Denemo Display

override - Contains bits to determine whether the LilyPond contained in the Directive (postfix and/or prefix fields) should override the normal LilyPond output, and whether the Graphic should replace the normal Denemo display for the item. A further tranche of bits controls MIDI output for the directive, which can override the normal MIDI interpretation of the music and provide additional information not explicit in the music notation (e.g. the tempo of an Adagio marking).

midibytes - a string of numbers (in hexadecimal format) whose interpretation is given by the MIDI bits in the override field

The override field contains the following bits:

DENEMO_OVERRIDE_LILYPOND: override the LilyPond output normally used at this point, rather than adding to the normal output. The text used is in the prefix and postfix fields.

DENEMO_OVERRIDE_GRAPHIC: overrides what Denemo would normally show in the display with the image named in the graphic field of the directive

The MIDI bits in the override field are as follows

DENEMO_OVERRIDE_VOLUME: the MIDI velocity to use

DENEMO_OVERRIDE_DURATION: affects the duration of a note. not yet implemented

DENEMO_OVERRIDE_REPEAT: indicates that an earlier passage should be repeated (from a directive of the same tag). not yet implemented

DENEMO_OVERRIDE_CHANNEL: midibytes field gives the MIDI channel to use (? implemented?)

DENEMO_OVERRIDE_TEMPO: midibytes field gives the tempo to use.

The interpretation of these flags is modified by the following flags:

DENEMO_OVERRIDE_ONCE: the value in midibytes is to be used just for the item the directive is attached to. not yet implemented

DENEMO_OVERRIDE_STEP: the value in midibytes is to be used from this point on

DENEMO_OVERRIDE_RAMP: the value in midibytes is to be used as a starting value, a corresponding directive (i.e. with the same tag) gives the findal value, values are then interpolated between these. not yet implemented

DENEMO_OVERRIDE_RELATIVE: value in midibytes is used relative to the current value (otherwise it is an absolute value; e.g. an absolute velocity etc)

DENEMO_OVERRIDE_PERCENT: value in midibytes is interpreted as percentage value. not yet implemented

These flags are combined together to get the combination required for the directive using the scheme procedure called logior.

Here is an example, making a step-change in volume of 0x40 in the MIDI output, and printing "più mosso" in the output score

```
;;;;;;;;;; piu mosso
(d-DirectivePut-standalone-minpixels "StepTempo" 20)
(d-CursorLeft)
(d-DirectivePut-standalone-override "StepTempo" (logior DENEMO_OVERRIDE_TEMPO DENEMO_OVERRI
(d-DirectivePut-standalone-midibytes "StepTempo" "40")
(d-DirectivePut-standalone-display "StepTempo" "piu mosso")
(d-DirectivePut-standalone-postfix "StepTempo" "^\\markup {\\bold \\italic \"più mosso\"}")
(d-RefreshDisplay)
```

In this example the directive is a standalone directive. MIDI commands are being extended to apply to chords and notes. Implementation of the commands for Voices, Movements, Score etc are not yet done.

11.2 Directive Edit Scripts

Introduction Score and movement directives can define a value for their graphic - a button then appears at the top of the score which can be used to edit the directive. Likewise Staff and Voice directives show as a properties icon before the staff to which they apply (staff directives above, voice directives below). Clicking on these lets you edit the directive.

The command EditDirective can be used when the cursor is on a Denemo Directive object, or an object with a Denemo Directive attached to it. What happens then is determined by a script named after the "tag", or name of the directive. For example the command RehearsalMark creates a Directive with tag "RehearsalMark" and EditDirective runs a script called RehearsalMark.scm.

There is also a low-level editing dialog which is invoked if no editing script exists or directly from scheme using (d-DirectiveTextEdit-* tagname). This allows you to edit and delete a directive directly. The other directive editing commands are EditScoreDirective EditMovementDirective EditStaffDirective EditVoiceDirective EditClefDirective EditKeysigDirective EditTimesigDirective for directives attached to the relevant objects.

The low level edit of directives from scheme uses the following command:

```
(d-DirectiveTextEdit-<field> <tagname>)
```

Where <tagname> specifies the directive to be edited and <field> is one of score, scoreheader, movementcontrol, header, paper, layout, clef, timesig, keysig, staff, voice, standalone, chord or note.

For example,

```
(d-DirectiveTextEdit-paper "PrintAllHeaders")
```

gives low-level access to the directive setting the print all headers command in the paper block of the LilyPond output.

11.2.1 Initialization Scripts

Each menu can have an initialization script, which can contain procedures that would be too time consuming to define every time they are needed. These scripts are guaranteed to be run before any menu item is activated within the menu. They can be read/written using the right click menu on any scripted menu item.

11.2.2 Edit Scripts

Each directive has a tag field, so that it can be recognized by the scripts that manipulate it. In particular for each tag there can be an edit script, for editing a directive of that tag.

Edit scripts are kept in a directory actions/editscripts parallel with the directory actions/menus where the commands themselves are kept.

You can read and write edit scripts by using the low level editing dialog on a directive with the tag you are writing for.

> The low level editing dialog is the one that is presented if there is no edit script. Edit scripts can themselves give access to this dialog by including an option cue-Advanced which calls (d-DirectiveTextEdit-field tag) for the appropriate field and tag. For an example see the edit script for Instruments -> Orchestra -> RehearsalMark.

They have available functions to enable editing of directives which are defined in actions//denemo.scm executed at startup. The functions and variables for script editing are:

- d-GetOption followed by a nul separated list of options, offers the options to the user and returns one, or #f if the user cancels.
- Extra-Offset tag type context. Takes three string options: tag is the name of the directive to be edited, which must be the name of a LilyPond object, type is the type of directive (note, chord, standalone, staff, voice or score), context is the LilyPond context of the object. Only tag is required. Shifts the object in the LilyPond output.
- SetPadding tag type context. As Extra-Offset, it sets the space left around the item in the LilyPond engraving process.
- SetRelativeFontSize tag type context. As Extra-Offset, it shifts the font size of the following text in the LilyPond engraving process.
- CreateButton tag label this is just a convenience function to put a button with the passed in label onto a button box at the top of the screen. Scripts can attach actions to such tagged buttons.
- d-SetDirectiveTagActionScript tag scheme-actions. This command sets scheme-actions as the actions to be performed when the button of the given tag is clicked. The default action is to run any editscript associated with tag, and if none to run the d-DirectiveTextEdit-score on the directive that displays the button.
- d-DirectiveGetForTag-field tag. Useful variables defined:
  - (define stop "\0")
  - (define cue-Advanced "Advanced")
  - (define cue-PlaceAbove "Place above staff")
  - (define cue-PlaceBelow "Place below staff")
  - (define cue-SetRelativeFontSize "Set Relative Font Size")
  - (define cue-OffsetPositionAll "Offset Position (All)")
  - (define cue-OffsetPositionOne "Offset Position (One)")
  - (define cue-EditText "Edit Text")

  - (define cue-SetPadding "Set Padding")
  - (define cue-Delete "Delete")

Part V
Obtaining and Installing Denemo

Windows Mac OSX Linux

Denemo is available from the Downloads page of the Denemo website http://denemo.org, where more up-to-date information will be found.

Denemo is available from a variety of sources for different distributions. The latest stable release (tar.gz and .deb formats) is available for download from http://denemo.org/downloads-page/. You can install Denemo from the Debian unstable repositories, using the command apt-get install denemo. Builds for Macintosh are available from the Gnu-Darwin project. The Denemo development branch can be downloaded using anonymous CVS or Git.

Anonymous Git checkout:

```
git clone git://git.savannah.gnu.org/denemo.git
```

Anonymous CVS checkout:

```
cvs -d:pserver:anonymous@pserver.git.sv.gnu.org:/denemo.git co -d denemo master
```

Dependencies To build Denemo from a source package, pleas see the website for an up-to-date list of dependencies. Remember to install the development packages as well (check your distribution for the specific package name):

> For Debian Users: Type apt-cache showsrc denemo at a command line to determine what are the package names for Denemo dependencies. Type apt-get build-dep denemo to build the dependent files or use apt-get to install the packages individually by copying and pasting their names to the command line.

Installing Denemo from Source Code Denemo is available in a variety of formats. The current stable release is available either as source code or in binary format. The Development branch of Denemo is available as a GIT source tree.

.1 To install from source code:

1. Open a terminal window.
2. Change directory to the directory to which you downloaded the Denemo source package.
3. Uncompress the source package using standard Linux tools (tar and gunzip).
4. Change directory to the uncompressed source directory.

.2 Generating a Configure Script

Git does not come with a Configure script; generate one by typing and pressing Enter after the line:

```
./autogen.sh
```

To build from source, press Enter after each line:

```
./configure
make
make install
```

If you are not logged in as root user, for the last step type su and your root password, or alternately type sudo make install.