Sequencer64 (seq64) Developer Reference Manual 0.90.2

Generated by Doxygen 1.8.13

Contents

1	Seq	uencer6	64		1
	1.1	Introdu	uction		1
2	MID	l File Pa	ırsing in S	Sequencer64	3
	2.1	Introdu	uction		3
	2.2	SMF 1	Parsing .		3
		2.2.1	MIDI File	Header, MThd	3
		2.2.2	MIDI Tra	ck, MTrk	4
			2.2.2.1	Channel Events	4
			2.2.2.2	Meta Events	5
		2.2.3	Meta Eve	ents Summary	6
			2.2.3.1	Sequence Number (0x00)	6
			2.2.3.2	Track/Sequence Name (0x03)	7
			2.2.3.3	End of Track (0x2F)	7
			2.2.3.4	Set Tempo Event (0x51)	7
			2.2.3.5	Time Signature Event (0x58)	8
			2.2.3.6	SysEx Event (0xF0)	9
			2.2.3.7	Sequencer Specific (0x7F)	10
			2.2.3.8	Non-Specific End of Sequence	10
	2.3	SMF 0	Parsing .		10
	24	Runnir	na Status		11

ii CONTENTS

3	JAC	CK, Live, and Song Modes in Sequencer64	13
	3.1	Introduction	13
	3.2	JACK Functions	13
		3.2.1 jack_client_open()	14
		3.2.2 jack_on_shutdown()	14
		3.2.3 jack_set_sync_callback()	14
		3.2.4 jack_set_process_callback()	15
		3.2.5 jack_set_session_callback()	15
		3.2.6 jack_activate()	15
		3.2.7 jack_release_timebase()	15
		3.2.8 jack_client_close()	15
		3.2.9 jack_transport_start()	15
		3.2.10 jack_transport_stop()	15
		3.2.11 jack_transport_locate()	15
		3.2.12 jack_transport_reposition()	16
		3.2.13 jack_transport_query()	16
	3.3	Modes Operation	16
		3.3.1 No JACK, Live Mode	16
		3.3.2 No JACK, Song Mode	16
		3.3.3 JACK Transport	17
	3.4	Breakage	17
	3.5	JACK References	18
4	Heer	r Testing of Sequencer64 with Yoshimi	19
•	4.1	Introduction	19
	4.1	Smoke Test	19
	4.3	Tests in the Patterns Window	20
		4.3.1 Button Clicks on a Pattern	21
		· · · · · · · · · · · · · · · · · · ·	
	4.4	4.3.3 The Sequencer64 User File	21
	4.4	Tests Using Valgrind	21
		4.4.1 Valgrind Suppressions	22
		4.4.2 Full Valgrind Leak-Checking	22
		4.4.2.1 Leak-Checking Basic Operation	23
	4.5	Specific Fault Debugging	23
	4.6	Snipping of a MIDI file.	23

5	Spee	ed Issue of Sequencer64	25
	5.1	Introduction	25
	5.2	Initial Change of Containers	25
	5.3	Back to the Original Container	26
	5.4	What is Next, the Vector?	26
6	Lice	nses	27
	6.1	License Terms for the This Project	27
	6.2	XPC Application License	27
	6.3	XPC Library License	28
	6.4	XPC Documentation License	28
	6.5	XPC Affero License	29
	6.6	XPC License Summary	29
7	Todo	o List	31
8	Depr	recated List	35
9	Nam	espace Index	37
	9.1	Namespace List	37
10	Hiera	archical Index	39
	10.1	Class Hierarchy	39
11	Data	Structure Index	43
	11.1	Data Structures	43

iv CONTENTS

12	Nam	espace	Documen	itation	49
	12.1	Gtk Na	mespace I	Reference	49
	12.2	seq64 l	Namespac	ee Reference	49
		12.2.1	Detailed I	Description	64
		12.2.2	Typedef [Documentation	64
			12.2.2.1	midibyte	64
			12.2.2.2	bussbyte	64
			12.2.2.3	midishort	65
			12.2.2.4	midilong	65
			12.2.2.5	midipulse	65
			12.2.2.6	midibpm	65
			12.2.2.7	rterror_callback	65
			12.2.2.8	rtmidi_callback_t	65
		12.2.3	Enumera	tion Type Documentation	66
			12.2.3.1	wave_type_t	66
			12.2.3.2	seq_modifier_t	66
			12.2.3.3	seq_event_type_t	67
			12.2.3.4	seq_scroll_direction_t	67
			12.2.3.5	clock_e	68
			12.2.3.6	interaction_method_t	68
			12.2.3.7	c_music_scales	68
			12.2.3.8	draw_type_t	69
			12.2.3.9	mouse_action_e	69
			12.2.3.10	edit_action_t	69
			12.2.3.11	rtmidi_api	70
		12.2.4	Function	Documentation	70
			12.2.4.1	swap()	70
			12.2.4.2	wave_type_name()	71
			12.2.4.3	extract_timing_numbers()	71
			12.2.4.4	pulses_to_string()	72

12.2.4.5 pulses_to_measurestring()
12.2.4.6 pulses_to_midi_measures()
12.2.4.7 pulses_to_timestring() [1/2]
12.2.4.8 pulses_to_timestring() [2/2]
12.2.4.9 measurestring_to_pulses()
12.2.4.10 midi_measures_to_pulses()
12.2.4.11 timestring_to_pulses() [1/2] 7.
12.2.4.12 string_to_pulses()
12.2.4.13 string_to_midibyte()
12.2.4.14 shorten_file_spec()
12.2.4.15 string_not_void()
12.2.4.16 string_is_void()
12.2.4.17 strings_match()
12.2.4.18 log2_time_sig_value()
12.2.4.19 tempo_us_to_bytes()
12.2.4.20 zoom_power_of_2()
12.2.4.21 bpm_from_tempo_us()
12.2.4.22 tempo_us_from_bpm()
12.2.4.23 pulse_length_us()
12.2.4.24 delta_time_us_to_ticks()
12.2.4.25 ticks_to_delta_time_us()
12.2.4.26 clock_tick_duration_bogus()
12.2.4.27 clock_ticks_from_ppqn()
12.2.4.28 double_ticks_from_ppqn()
12.2.4.29 pulses_per_measure()
12.2.4.30 measures_to_ticks()
12.2.4.31 wave_func()
12.2.4.32 extract_port_names()
12.2.4.33 extract_bus_name()
12.2.4.34 extract_port_name()

vi

12.2.4.35 help_check()
12.2.4.36 parse_options_files()
12.2.4.37 parse_command_line_options()
12.2.4.38 write_options_files()
12.2.4.39 build_details()
12.2.4.40 message_concatenate()
12.2.4.41 info_message()
12.2.4.42 error_message()
12.2.4.43 to_string()
12.2.4.44 file_access()
12.2.4.45 file_exists()
12.2.4.46 file_readable()
12.2.4.47 file_writable()
12.2.4.48 file_accessible()
12.2.4.49 file_executable()
12.2.4.50 file_is_directory()
12.2.4.51 make_directory()
12.2.4.52 ppqn_is_valid()
12.2.4.53 jack_sync_callback()
12.2.4.54 jack_shutdown_callback()
12.2.4.55 jack_timebase_callback()
12.2.4.56 jack_transport_callback()
12.2.4.57 create_jack_client()
12.2.4.58 show_jack_statuses()
12.2.4.59 get_current_jack_position()
12.2.4.60 jack_session_callback()
12.2.4.61 invalid_key()
12.2.4.62 keyval_name()
12.2.4.63 keyval_normalize()
12.2.4.64 create_lash_driver()

CONTENTS vii

12.2.4.65 lash_driver()
12.2.4.66 delete_lash_driver()
12.2.4.67 millisleep()
12.2.4.68 is_null_midipulse()
12.2.4.69 output_thread_func()
12.2.4.70 input_thread_func()
12.2.4.71 rc()
12.2.4.72 usr()
12.2.4.73 choose_ppqn()
12.2.4.74 timestring_to_pulses() [2/2]
12.2.4.75 jack_dummy_callback()
12.2.4.76 seq_app_name()
12.2.4.77 seq_client_name()
12.2.4.78 seq_version()
12.2.4.79 make_section_name()
12.2.4.80 font_render()
12.2.4.81 adjustment_dummy()
12.2.4.82 is_ctrl_key() [1/3]
12.2.4.83 is_shift_key() [1/3] 103
12.2.4.84 is_no_modifier()
12.2.4.85 is_ctrl_key() [2/3]
12.2.4.86 is_shift_key() [2/3] 103
12.2.4.87 is_ctrl_key() [3/3]
12.2.4.88 is_shift_key() [3/3]
12.2.4.89 is_ctrl_shift_key()
12.2.4.90 is_super_key()
12.2.4.91 test_widget_click()
12.2.4.92 update_mainwid_sequences()
12.2.4.93 update_perfedit_sequences()
12.2.4.94 FF_RW_timeout()

viii CONTENTS

	12.2.4.95 clamp() [1/2]	105
	12.2.4.96 clamp() [2/2]	106
	12.2.4.97 silence_jack_errors()	106
	12.2.4.98 silence_jack_info()	106
	12.2.4.99 midi_api_name()	106
	12.2.4.100midi_probe()	106
	12.2.4.101midi_input_test()	107
	12.2.4.102min()	107
	12.2.4.103jack_process_rtmidi_input()	107
	12.2.4.104jack_process_rtmidi_output()	108
	12.2.4.105ack_process_io()	109
	12.2.4.106jack_message_bit_bucket()	109
	12.2.4.107midi_input_callback()	109
12.2.5	Variable Documentation	109
	12.2.5.1 c_controller_names	109
	12.2.5.2 EVENT_STATUS_BIT	109
	12.2.5.3 EVENT_ANY	110
	12.2.5.4 EVENT_NOTE_OFF	110
	12.2.5.5 EVENT_NOTE_ON	110
	12.2.5.6 EVENT_AFTERTOUCH	110
	12.2.5.7 EVENT_CONTROL_CHANGE	110
	12.2.5.8 EVENT_PROGRAM_CHANGE	110
	12.2.5.9 EVENT_CHANNEL_PRESSURE	110
	12.2.5.10 EVENT_PITCH_WHEEL	111
	12.2.5.11 EVENT_MIDI_SYSEX	111
	12.2.5.12 EVENT_MIDI_QUARTER_FRAME	111
	12.2.5.13 EVENT_MIDI_SONG_POS	111
	12.2.5.14 EVENT_MIDI_SONG_SELECT	111
	12.2.5.15 EVENT_MIDI_SONG_F4	111
	12.2.5.16 EVENT_MIDI_SONG_F5	112

12.2.5.17 EVENT_MIDI_TUNE_SELECT
12.2.5.18 EVENT_MIDI_SYSEX_END
12.2.5.19 EVENT_MIDI_SYSEX_CONTINUE
12.2.5.20 EVENT_MIDI_CLOCK
12.2.5.21 EVENT_MIDI_SONG_F9
12.2.5.22 EVENT_MIDI_START
12.2.5.23 EVENT_MIDI_CONTINUE
12.2.5.24 EVENT_MIDI_STOP
12.2.5.25 EVENT_MIDI_SONG_FD
12.2.5.26 EVENT_MIDI_ACTIVE_SENS
12.2.5.27 EVENT_MIDI_RESET
12.2.5.28 EVENT_MIDI_META
12.2.5.29 EVENT_NULL_CHANNEL
12.2.5.30 EVENT_GET_CHAN_MASK
12.2.5.31 EVENT_CLEAR_CHAN_MASK
12.2.5.32 EVENTS_ALL
12.2.5.33 EVENTS_UNSELECTED
12.2.5.34 c_midibus_output_size
12.2.5.35 c_midibus_input_size
12.2.5.36 c_midibus_sysex_chunk
12.2.5.37 c_midibus
12.2.5.38 c_midich
12.2.5.39 c_midiclocks
12.2.5.40 c_triggers
12.2.5.41 c_notes
12.2.5.42 c_timesig
12.2.5.43 c_bpmtag
12.2.5.44 c_triggers_new
12.2.5.45 c_mutegroups
12.2.5.46 c_midictrl

12.2.5.47 c_musickey
12.2.5.48 c_musicscale
12.2.5.49 c_backsequence
12.2.5.50 c_transpose
12.2.5.51 c_perf_bp_mes
12.2.5.52 c_perf_bw
12.2.5.53 c_midi_track_ctrl
12.2.5.54 c_midi_control_bpm_up
12.2.5.55 c_midi_control_bpm_dn
12.2.5.56 c_midi_control_ss_up
12.2.5.57 c_midi_control_ss_dn
12.2.5.58 c_midi_control_mod_replace
12.2.5.59 c_midi_control_mod_snapshot
12.2.5.60 c_midi_control_mod_queue
12.2.5.61 c_midi_control_mod_gmute
12.2.5.62 c_midi_control_mod_glearn
12.2.5.63 c_midi_control_play_ss
12.2.5.64 c_midi_controls
12.2.5.65 c_midi_control_playback
12.2.5.66 c_midi_control_record
12.2.5.67 c_midi_control_solo
12.2.5.68 c_midi_control_thru
12.2.5.69 c_midi_control_bpm_page_up
12.2.5.70 c_midi_control_bpm_page_dn
12.2.5.71 c_midi_control_16
12.2.5.72 c_midi_control_17
12.2.5.73 c_midi_control_18
12.2.5.74 c_midi_control_19
12.2.5.75 c_midi_controls_extended
12.2.5.76 g_midi_control_limit

CONTENTS xi

12.2.5.77 c_scales_policy
12.2.5.78 c_scales_transpose_up
12.2.5.79 c_scales_transpose_dn
12.2.5.80 c_scales_text
12.2.5.81 c_key_text
12.2.5.82 c_interval_text
12.2.5.83 c_chord_text
12.2.5.84 c_chord_number
12.2.5.85 c_chord_table_text
12.2.5.86 c_chord_size
12.2.5.87 c_chord_table
12.2.5.88 c_max_instruments
12.2.5.89 c_max_busses
12.2.5.90 versiontext
12.2.5.91 long_options
12.2.5.92 s_arg_list
12.2.5.93 s_help_1a
12.2.5.94 s_help_1b
12.2.5.95 s_help_2
12.2.5.96 s_help_3
12.2.5.97 s_help_4
12.2.5.98 s_build_alsamidi_support
12.2.5.99 s_build_portmidi_support
12.2.5.100s_build_rtmidi_support
12.2.5.101s_build_highlight_empty
12.2.5.102s_build_lash_support
12.2.5.103s_build_jack_support
12.2.5.104s_build_jack_session
12.2.5.105s_event_editor
12.2.5.106s_build_pause_support

xii CONTENTS

12.2.5.107s_build_use_event_map
12.2.5.108s_build_chord_generator
12.2.5.10%_build_edit_highlight
12.2.5.110s_build_timesig_tempo
12.2.5.111s_build_midi_vector
12.2.5.112s_build_solid_grid
12.2.5.113s_build_follow_progress
12.2.5.114s_statistics_support
12.2.5.115s_strip_empty_mutes
12.2.5.116s_seq32_jack_support
12.2.5.117s_seq32_transport
12.2.5.118s_seq32_transpose
12.2.5.119s_seq32_menu_buttons
12.2.5.120s_seq32_lfo_support
12.2.5.121s_debug_mode
12.2.5.122s_status_pairs
12.2.5.123s_character_mapping
12.2.5.124s_global_lash_driver
12.2.5.125c_status_replace
12.2.5.126c_status_snapshot
12.2.5.127c_status_queue
12.2.5.128g_rc_settings
12.2.5.129g_user_settings
12.2.5.130s_handlesize [1/2]
12.2.5.131s_jitter_amount
12.2.5.132gs_mainwid_pointer
12.2.5.133c_mainwid_x
12.2.5.134c_mainwid_y
12.2.5.135gs_perfedit_pointer_0
12.2.5.136gs_perfedit_pointer_1
12.2.5.137s_handlesize [2/2]

CONTENTS xiii

13	Data	Structu	ure Documentation	133
	13.1	seq64:	:AbstractPerfInput Class Reference	133
		13.1.1	Constructor & Destructor Documentation	134
			13.1.1.1 AbstractPerfInput()	134
			13.1.1.2 ~AbstractPerfInput()	134
		13.1.2	Member Function Documentation	135
			13.1.2.1 on_button_press_event()	135
			13.1.2.2 on_button_release_event()	135
			13.1.2.3 on_motion_notify_event()	135
			13.1.2.4 activate_adding()	135
			13.1.2.5 handle_motion_key()	135
			13.1.2.6 is_adding()	136
			13.1.2.7 set_adding()	136
			13.1.2.8 is_adding_pressed()	136
			13.1.2.9 set_adding_pressed()	136
		13.1.3	Friends And Related Function Documentation	136
			13.1.3.1 perfroll	136
		13.1.4	Field Documentation	136
			13.1.4.1 m_adding	136
			13.1.4.2 m_adding_pressed	137
	13.2	seq64:	automutex Class Reference	137
		13.2.1	Detailed Description	137
		13.2.2	Constructor & Destructor Documentation	137
			13.2.2.1 automutex() [1/3]	137
			13.2.2.2 automutex() [2/3]	137
			13.2.2.3 automutex() [3/3]	137
			13.2.2.4 ~automutex()	138
		13.2.3	Member Function Documentation	138
			13.2.3.1 operator=()	138
		13.2.4	Field Documentation	138

xiv CONTENTS

	13.2.4.1 m_safety_mutex
13.3 seq64	::busarray Class Reference
13.3.1	Constructor & Destructor Documentation
	13.3.1.1 busarray()
	13.3.1.2 ~busarray()
13.3.2	Member Function Documentation
	13.3.2.1 add() [1/2]
	13.3.2.2 add() [2/2]
	13.3.2.3 initialize()
	13.3.2.4 count()
	13.3.2.5 bus()
	13.3.2.6 start()
	13.3.2.7 stop()
	13.3.2.8 continue_from()
	13.3.2.9 init_clock()
	13.3.2.10 clock()
	13.3.2.11 sysex()
	13.3.2.12 play()
	13.3.2.13 set_clock()
	13.3.2.14 set_all_clocks()
	13.3.2.15 get_clock()
	13.3.2.16 get_midi_bus_name()
	13.3.2.17 print()
	13.3.2.18 port_exit()
	13.3.2.19 set_input()
	13.3.2.20 set_all_inputs()
	13.3.2.21 get_input()
	13.3.2.22 is_system_port()
	13.3.2.23 poll_for_midi()
	13.3.2.24 get_midi_event()

CONTENTS xv

	13.3.2.25 replacement_port()
13.3.3	Field Documentation
	13.3.3.1 m_container
13.4 seq64	::businfo Class Reference
13.4.1	Detailed Description
13.4.2	Constructor & Destructor Documentation
	13.4.2.1 businfo() [1/3]
	13.4.2.2 businfo() [2/3]
	13.4.2.3 businfo() [3/3]
	13.4.2.4 ~businfo()
13.4.3	Member Function Documentation
	13.4.3.1 remove()
	13.4.3.2 bus() [1/3]
	13.4.3.3 bus() [2/3]
	13.4.3.4 active()
	13.4.3.5 initialize()
	13.4.3.6 initialized()
	13.4.3.7 init_clock() [1/3]
	13.4.3.8 init_input() [1/2]
	13.4.3.9 bus() [3/3]
	13.4.3.10 activate()
	13.4.3.11 deactivate()
	13.4.3.12 init_clock() [2/3]
	13.4.3.13 init_input() [2/2]
	13.4.3.14 start()
	13.4.3.15 stop()
	13.4.3.16 continue_from()
	13.4.3.17 init_clock() [3/3]
	13.4.3.18 clock()
	13.4.3.19 sysex()

xvi CONTENTS

13.4.4	Friends And Related Function Documentation
	13.4.4.1 busarray
13.4.5	Field Documentation
	13.4.5.1 m_bus
	13.4.5.2 m_active
	13.4.5.3 m_initialized
	13.4.5.4 m_init_clock
	13.4.5.5 m_init_input
13.5 seq64	::click Class Reference
13.5.1	Detailed Description
13.5.2	Constructor & Destructor Documentation
	13.5.2.1 click() [1/3]
	13.5.2.2 click() [2/3]
	13.5.2.3 click() [3/3]
13.5.3	Member Function Documentation
	13.5.3.1 operator=()
	13.5.3.2 is_press()
	13.5.3.3 is_left()
	13.5.3.4 is_middle()
	13.5.3.5 is_right()
	13.5.3.6 x()
	13.5.3.7 y()
	13.5.3.8 button()
	13.5.3.9 modifier()
	13.5.3.10 mod_control()
	13.5.3.11 mod_control_shift()
	13.5.3.12 mod_super()
13.5.4	Field Documentation
	13.5.4.1 m_is_press
	13.5.4.2 m_x

CONTENTS xvii

		13.5.4.3 m_y	59
		13.5.4.4 m_button	59
		13.5.4.5 m_modifier	59
13.6	seq64::	ondition_var Class Reference	59
	13.6.1	Detailed Description	30
	13.6.2	Constructor & Destructor Documentation	30
		13.6.2.1 condition_var()	30
	13.6.3	Member Function Documentation	30
		13.6.3.1 wait()	30
		13.6.3.2 signal()	31
	13.6.4	Field Documentation	31
		13.6.4.1 sm_cond	31
		13.6.4.2 m_cond	31
13.7	seq64::	onfigfile Class Reference	31
	13.7.1	Constructor & Destructor Documentation	33
		13.7.1.1 configfile()	33
		13.7.1.2 ~configfile()	33
	13.7.2	Member Function Documentation	33
		13.7.2.1 next_data_line()	33
		13.7.2.2 line_after()	34
		13.7.2.3 parse()	34
		13.7.2.4 write()	34
		13.7.2.5 get_error_message()	35
		13.7.2.6 set_error_message()	35
	13.7.3	Field Documentation	35
		13.7.3.1 m_error_message	35
		13.7.3.2 m_name	35
		13.7.3.3 m_d	35
		13.7.3.4 m_line	35
13.8	seq64::	ditable_event Class Reference	36

xviii CONTENTS

13.8.1	Detailed Description	39
13.8.2	Member Enumeration Documentation	39
	13.8.2.1 category_t	39
	13.8.2.2 timestamp_format_t	70
13.8.3	Constructor & Destructor Documentation	70
	13.8.3.1 editable_event() [1/4]	70
	13.8.3.2 editable_event() [2/4]	71
	13.8.3.3 editable_event() [3/4]	71
	13.8.3.4 editable_event() [4/4]	71
	13.8.3.5 ~editable_event()	71
13.8.4	Member Function Documentation	72
	13.8.4.1 value_to_name()	72
	13.8.4.2 name_to_value()	72
	13.8.4.3 operator=()	72
	13.8.4.4 parent()	73
	13.8.4.5 category() [1/3]	73
	13.8.4.6 category() [2/3]	73
	13.8.4.7 category_string()	73
	13.8.4.8 category() [3/3]	73
	13.8.4.9 timestamp_string()	73
	13.8.4.10 timestamp() [1/3]	74
	13.8.4.11 timestamp() [2/3]	74
	13.8.4.12 timestamp() [3/3]	74
	13.8.4.13 time_as_pulses()	74
	13.8.4.14 time_as_measures()	74
	13.8.4.15 time_as_minutes()	75
	13.8.4.16 set_status_from_string()	75
	13.8.4.17 format_timestamp()	75
	13.8.4.18 stock_event_string()	75
	13.8.4.19 status_string()	76

CONTENTS xix

		13.8.4.20	meta_st	ring() .				 	 	 	 	 	176
		13.8.4.21	seqspec	_string	ı()			 	 	 	 	 	176
		13.8.4.22	channel	_string(()			 	 	 	 	 	176
		13.8.4.23	data_str	ing() .				 	 	 	 	 	 176
		13.8.4.24	analyze	()				 	 	 	 	 	176
	13.8.5	Field Doc	umentati	on				 	 	 	 	 	177
		13.8.5.1	sm_cate	egory_r	names .			 	 	 	 	 	 177
		13.8.5.2	sm_cha	nnel_e\	vent_na	mes		 	 	 	 	 	177
		13.8.5.3	sm_syst	tem_ev	ent_nar	mes .		 	 	 	 	 	177
		13.8.5.4	sm_met	a_even	ıt_name	s		 	 	 	 	 	 177
		13.8.5.5	sm_prop	p_event	t_name:	s		 	 	 	 	 	177
		13.8.5.6	sm_cate	egory_a	arrays .			 	 	 	 	 	178
		13.8.5.7	m_pare	nt				 	 	 	 	 	 178
		13.8.5.8	m_categ	gory .				 	 	 	 	 	178
		13.8.5.9	m_name	e_cateç	jory			 	 	 	 	 	 178
		13.8.5.10	m_forma	at_time	stamp			 	 	 	 	 	 178
		13.8.5.11	m_name	e_times	stamp .			 	 	 	 	 	 178
		13.8.5.12	m_name	e_statu	s			 	 	 	 	 	 178
		13.8.5.13	m_name	e_meta				 	 	 	 	 	 179
		13.8.5.14	m_name	e_seqs	pec			 	 	 	 	 	 179
		13.8.5.15	m_name	e_chan	nel			 	 	 	 	 	 179
		13.8.5.16	m_name	e_data				 	 	 	 	 	 179
13.9	seq64::	editable_e	vents Cla	ass Ref	erence			 	 	 	 	 	 179
	13.9.1	Member T	Typedef [Jocume	entation			 	 	 	 	 	 181
		13.9.1.1	Key					 	 	 	 	 	 181
		13.9.1.2	EventsF	air				 	 	 	 	 	 181
		13.9.1.3	Events					 	 	 	 	 	 181
		13.9.1.4	iterator					 	 	 	 	 	 181
		13.9.1.5	const_it	erator				 	 	 	 	 	 182
	13.9.2	Construct	or & Des	structor	Docum	entatio	on	 	 	 	 	 	 182

	13.9.2.1	e	ditabl	e_ev	ents(() [1	/3]			٠.		 	 		 		 	 182
	13.9.2.2	е	ditabl	e_ev	ents(() [2	/3]					 	 		 		 	 182
	13.9.2.3	е	ditabl	e_ev	ents(() [3	/3]					 	 		 		 	 182
	13.9.2.4	~	√edita	ıble_e	event	ts()							 			 		 182
13.9.3	Member F	Fui	nctior	n Doc	ume	ntat	ion					 	 		 		 	 183
	13.9.3.1	O	perat	or=()									 			 	 	 183
	13.9.3.2	tir	ming()									 			 	 	 183
	13.9.3.3	st	tring_	to_pu	ulses	s()						 	 		 	 		 183
	13.9.3.4	lo	ad_e	vents	s() .							 	 		 		 	 183
	13.9.3.5	Sã	ave_e	ents	s()							 	 		 		 	 184
	13.9.3.6	eı	vents	() .								 	 		 		 	 184
	13.9.3.7	be	egin()) [1/:	2]							 	 		 		 	 184
	13.9.3.8	be	egin()) [2/:	2]							 	 		 		 	 184
	13.9.3.9	eı	nd()	[1/2]								 	 		 		 	 184
	13.9.3.10	0 ei	nd()	[2/2]									 		 	 		 184
	13.9.3.11	1 dı	ref()	[1/2]] .							 	 		 		 	 184
	13.9.3.12	2 dı	ref()	[2/2]	l -								 		 	 		 185
	13.9.3.13	3 c	ount())								 	 		 		 	 185
	13.9.3.14	4 a	dd()	[1/2]								 	 		 		 	 185
	13.9.3.15	5 a	dd()	[2/2]								 	 		 		 	 185
	13.9.3.16	6 re	place	э() .								 	 		 		 	 186
	13.9.3.17	7 re	emove	э() .								 	 		 		 	 186
	13.9.3.18	8 cl	ear()									 	 		 		 	 186
	13.9.3.19	9 cı	urren	t_eve	ent()	[1/2	2].					 	 		 	 	 	 186
	13.9.3.20	0 cı	urren	t_eve	ent()	[2/2	2] .					 	 		 		 	 186
13.9.4	Friends A	And	l Rela	ated F	=unct	tion	Doc	ume	entat	tion		 	 		 	 	 	 187
	13.9.4.1	eı	vents	lots								 	 		 	 	 	 187
13.9.5	Field Doc	cun	nenta	tion								 	 		 	 	 	 187
	13.9.5.1	m	n_eve	nts .								 	 		 	 	 	 187
	13.9.5.2	m	n_curi	rent_	even	t						 	 		 	 	 	 187

CONTENTS xxi

13.9.5.3 m_sequence
13.9.5.4 m_midi_parameters
13.10seq64::event Class Reference
13.10.1 Detailed Description
13.10.2 Member Typedef Documentation
13.10.2.1 SysexContainer
13.10.3 Constructor & Destructor Documentation
13.10.3.1 event() [1/2]19
13.10.3.2 event() [2/2]19
13.10.3.3 ~event()
13.10.4 Member Function Documentation
13.10.4.1 operator=()
13.10.4.2 operator<()
13.10.4.3 set_timestamp()
13.10.4.4 get_timestamp()
13.10.4.5 get_channel()
13.10.4.6 check_channel()
13.10.4.7 is_channel_msg()
13.10.4.8 is_one_byte_msg()
13.10.4.9 is_two_byte_msg()
13.10.4.10s_note_msg()
13.10.4.11is_strict_note_msg()
13.10.4.12s_desired_cc_or_not_cc()
13.10.4.13mod_timestamp()
13.10.4.14set_status() [1/2]
13.10.4.15set_status() [2/2]19
13.10.4.16set_status_keep_channel()
13.10.4.17set_channel()
13.10.4.18get_status()
13.10.4.19non_cc_match()

xxii CONTENTS

13.10.4.20cc_match()
13.10.4.21set_data() [1/2]
13.10.4.22set_data() [2/2]
13.10.4.23get_data()
13.10.4.24ncrement_data1()
13.10.4.25decrement_data1()
13.10.4.26ncrement_data2()
13.10.4.27decrement_data2()
13.10.4.2&append_sysex() [1/2]
13.10.4.29append_sysex() [2/2]
13.10.4.30restart_sysex()
13.10.4.31get_sysex() [1/2]
13.10.4.32get_sysex() [2/2]
13.10.4.33set_sysex_size()
13.10.4.34get_sysex_size()
13.10.4.35ink()
13.10.4.36get_linked()
13.10.4.37s_linked()
13.10.4.3&lear_link()
13.10.4.39paint()
13.10.4.40unpaint()
13.10.4.41is_painted()
13.10.4.42mark()
13.10.4.43unmark()
13.10.4.44is_marked()
13.10.4.45select()
13.10.4.46unselect()
13.10.4.47/s_selected()
13.10.4.48make_clock()
13.10.4.49data()

CONTENTS xxiii

13.	10.4.50get_note()	 	205
13.	10.4.51set_note()	 	205
13.	10.4.52\text{ranspose_note()}	 	205
13.	10.4.53get_note_velocity()	 	206
13.	10.4.54set_note_velocity()	 	206
13.	10.4.55s_note_on()	 	206
13.	10.4.56s_note_off()	 	206
13.	10.4.57/s_note()	 	207
13.	10.4.58s_note_off_recorded()	 	207
13.	10.4.59print()	 	207
13.	10.4.60get_rank()	 	207
13.10.5 Fie	ld Documentation	 	207
13.	10.5.1 m_timestamp	 	208
13.	10.5.2 m_status	 	208
13.	10.5.3 m_channel	 	208
13.	10.5.4 m_data	 	208
13.	10.5.5 m_sysex	 	208
13.	10.5.6 m_sysex_size	 	208
13.	10.5.7 m_linked	 	208
13.	10.5.8 m_has_link	 	209
13.	10.5.9 m_selected	 	209
13.	10.5.10m_marked	 	209
13.	10.5.11m_painted	 	209
13.11seq64::eve	nt_list::event_key Class Reference	 	209
13.11.1 De	tailed Description	 	209
13.11.2 Co	nstructor & Destructor Documentation	 	210
13.	11.2.1 event_key() [1/2]	 	210
13.	11.2.2 event_key() [2/2]	 	210
13.11.3 Me	mber Function Documentation	 	210
13.	11.3.1 operator<()	 	210

xxiv CONTENTS

13.11.4 Field Documentation
13.11.4.1 m_timestamp
13.11.4.2 m_rank
13.12seq64::event_list Class Reference
13.12.1 Detailed Description
13.12.2 Member Typedef Documentation
13.12.2.1 Events
13.12.2.2 EventsPair
13.12.2.3 iterator
13.12.2.4 const_iterator
13.12.3 Constructor & Destructor Documentation
13.12.3.1 event_list() [1/2]
13.12.3.2 event_list() [2/2]
13.12.3.3 ~event_list()
13.12.4 Member Function Documentation
13.12.4.1 operator=()
13.12.4.2 begin() [1/2]
13.12.4.3 begin() [2/2]
13.12.4.4 end() [1/2]
13.12.4.5 end() [2/2]
13.12.4.6 count()
13.12.4.7 empty()
13.12.4.8 add()
13.12.4.9 append()
13.12.4.10push_back()
13.12.4.11is_modified()
13.12.4.12unmodify()
13.12.4.13remove()
13.12.4.14clear()
13.12.4.15merge()

CONTENTS xxv

13.12.4.16sort()	218
13.12.4.17dref() [1/2]	218
13.12.4.18dref() [2/2]	218
13.12.4.19ink_new()	219
13.12.4.20clear_links()	219
13.12.4.21verify_and_link()	219
13.12.4.22mark_selected()	219
13.12.4.23mark_out_of_range()	219
13.12.4.24mark_all()	220
13.12.4.25unmark_all()	220
13.12.4.26remove_marked()	220
13.12.4.27unpaint_all()	220
13.12.4.2&ount_selected_notes()	220
13.12.4.29any_selected_notes()	221
13.12.4.30count_selected_events()	221
13.12.4.31select_all()	221
13.12.4.32unselect_all()	221
13.12.4.33print()	221
13.12.4.34events()	222
13.12.5 Friends And Related Function Documentation	222
13.12.5.1 editable_events	222
13.12.5.2 midifile	222
13.12.5.3 midi_container	222
13.12.5.4 midi_splitter	222
13.12.5.5 sequence	222
13.12.6 Field Documentation	222
13.12.6.1 m_events	222
13.12.6.2 m_is_modified	223
13.13seq64::eventedit Class Reference	223
13.13.1 Constructor & Destructor Documentation	227

xxvi CONTENTS

13.13.1.1 eventedit()	. 227
13.13.1.2 ∼eventedit()	. 228
13.13.2 Member Function Documentation	. 229
13.13.2.1 enqueue_draw()	. 229
13.13.2.2 set_seq_title()	. 229
13.13.2.3 set_seq_time_sig()	. 229
13.13.2.4 set_seq_ppqn()	. 229
13.13.2.5 set_seq_count()	. 229
13.13.2.6 set_event_category()	. 230
13.13.2.7 set_event_timestamp()	. 230
13.13.2.8 set_event_name()	. 230
13.13.2.9 set_event_data_0()	. 230
13.13.2.10set_event_data_1()	. 231
13.13.2.11perf_modify()	. 231
13.13.2.12set_dirty()	. 231
13.13.2.13v_adjustment() [1/2]	. 231
13.13.2.14v_adjustment() [2/2]	. 231
13.13.2.15change_focus()	. 232
13.13.2.16close_out()	. 232
13.13.2.17handle_close()	. 232
13.13.2.18handle_delete()	. 232
13.13.2.19handle_insert()	. 233
13.13.2.20handle_modify()	. 233
13.13.2.21handle_save()	. 233
13.13.2.22handle_cancel()	. 233
13.13.2.23on_realize()	. 233
13.13.2.24on_set_focus()	. 233
13.13.2.25on_focus_in_event()	. 234
13.13.2.26on_focus_out_event()	. 234
13.13.2.27on_key_press_event()	. 234

CONTENTS xxvii

13.13.2.28on_delete_event()
13.13.3 Friends And Related Function Documentation
13.13.3.1 eventslots
13.13.4 Field Documentation
13.13.4.1 m_table
13.13.4.2 m_vadjust
13.13.4.3 m_vscroll
13.13.4.4 m_eventslots
13.13.4.5 m_htopbox
13.13.4.6 m_showbox
13.13.4.7 m_editbox
13.13.4.8 m_optsbox
13.13.4.9 m_bottbox
13.13.4.10m_rightbox
13.13.4.11m_button_del
13.13.4.12m_button_ins
13.13.4.13m_button_modify
13.13.4.14m_button_save
13.13.4.15m_button_cancel
13.13.4.16m_label_seq_name
13.13.4.17m_label_time_sig
13.13.4.18m_label_ppqn
13.13.4.19m_label_channel
13.13.4.20m_label_ev_count
13.13.4.21m_label_spacer
13.13.4.22m_label_modified
13.13.4.23m_label_category
13.13.4.24m_entry_ev_timestamp
13.13.4.25m_entry_ev_name
13.13.4.26m_entry_ev_data_0

xxviii CONTENTS

13.13.4.27m_entry_ev_data_1
13.13.4.28m_label_time_fmt
13.13.4.29m_label_right
13.13.4.30m_seq
13.13.4.31m_have_focus
13.14seq64::eventslots Class Reference
13.14.1 Constructor & Destructor Documentation
13.14.1.1 eventslots()
13.14.1.2 ~eventslots()
13.14.2 Member Function Documentation
13.14.2.1 event_count()
13.14.2.2 line_count()
13.14.2.3 line_maximum()
13.14.2.4 line_increment()
13.14.2.5 top_index()
13.14.2.6 current_index()
13.14.2.7 pager_index()
13.14.2.8 load_events()
13.14.2.9 set_current_event()
13.14.2.1űnsert_event() [1/2]
13.14.2.11insert_event() [2/2]
13.14.2.12delete_current_event()
13.14.2.13modify_current_event()
13.14.2.14save_events()
13.14.2.15select_event()
13.14.2.16set_text()
13.14.2.17enqueue_draw()
13.14.2.1&convert_y()
13.14.2.19draw_event()
13.14.2.20draw_events()

CONTENTS xxix

13.14.2.21change_vert()	251
13.14.2.22page_movement()	251
13.14.2.23page_topper()	251
13.14.2.24decrement_top()	252
13.14.2.25ncrement_top()	252
13.14.2.26decrement_current()	252
13.14.2.27increment_current()	252
13.14.2.2&decrement_bottom()	253
13.14.2.29ncrement_bottom()	253
13.14.2.30on_realize()	253
13.14.2.31on_expose_event()	253
13.14.2.32on_button_press_event()	253
13.14.2.33on_button_release_event()	253
13.14.2.34on_focus_in_event()	254
13.14.2.35on_focus_out_event()	254
13.14.2.36on_scroll_event()	254
13.14.2.37on_size_allocate()	254
13.14.2.3&n_move_up()	254
13.14.2.39on_move_down()	254
13.14.2.40on_frame_up()	254
13.14.2.41on_frame_down()	255
13.14.2.42on_frame_home()	255
13.14.2.43on_frame_end()	255
13.14.3 Friends And Related Function Documentation	255
13.14.3.1 eventedit	255
13.14.4 Field Documentation	255
13.14.4.1 m_parent	255
13.14.4.2 m_seq	255
13.14.4.3 m_event_container	255
13.14.4.4 m_slots_chars	256

13.14.4.5 m_char_w	 256
13.14.4.6 m_setbox_w	 256
13.14.4.7 m_slots_x	 256
13.14.4.8 m_slots_y	 256
13.14.4.9 m_event_count	 256
13.14.4.10m_line_count	 256
13.14.4.11m_line_maximum	 257
13.14.4.12m_line_overlap	 257
13.14.4.13m_top_index	 257
13.14.4.14m_current_index	 257
13.14.4.15m_top_iterator	 257
13.14.4.16m_bottom_iterator	 257
13.14.4.17m_current_iterator	 257
13.14.4.18m_pager_index	 258
13.15seq64::font Class Reference	 258
13.15.1 Member Enumeration Documentation	 259
13.15.1.1 Color	 259
13.15.2 Constructor & Destructor Documentation	 259
13.15.2.1 font()	 260
13.15.3 Member Function Documentation	 260
13.15.3.1 init()	 260
13.15.3.2 render_string_on_drawable()	 260
13.15.3.3 char_width()	 261
13.15.3.4 char_height()	 261
13.15.3.5 padded_height()	 261
13.15.4 Field Documentation	 261
13.15.4.1 m_use_new_font	 261
13.15.4.2 m_cell_w	 261
13.15.4.3 m_cell_h	 261
13.15.4.4 m_font_w	 261

CONTENTS xxxi

13.15.4.5 m_font_h	 262
13.15.4.6 m_offset	 262
13.15.4.7 m_padded_h	 262
13.15.4.8 m_pixmap	 262
13.15.4.9 m_black_pixmap	 262
13.15.4.10m_white_pixmap	 262
13.15.4.11m_b_on_y_pixmap	 262
13.15.4.12m_y_on_b_pixmap	 263
13.15.4.13m_b_on_c_pixmap	 263
13.15.4.14m_c_on_b_pixmap	 263
13.15.4.15m_clip_mask	 263
13.16seq64::FruityPerfInput Class Reference	 263
13.16.1 Constructor & Destructor Documentation	 265
13.16.1.1 FruityPerfInput()	 265
13.16.2 Member Function Documentation	 265
13.16.2.1 on_button_press_event()	 265
13.16.2.2 on_button_release_event()	 266
13.16.2.3 on_motion_notify_event()	 266
13.16.2.4 update_mouse_pointer()	 267
13.16.2.5 on_left_button_pressed()	 267
13.16.2.6 on_right_button_pressed()	 267
13.16.2.7 activate_adding()	 268
13.16.2.8 handle_motion_key()	 268
13.16.3 Friends And Related Function Documentation	 268
13.16.3.1 perfroll	 268
13.16.4 Field Documentation	 268
13.16.4.1 m_current_x	 268
13.16.4.2 m_current_y	 269
13.17seq64::FruitySeqEventInput Struct Reference	 269
13.17.1 Constructor & Destructor Documentation	 269

xxxii CONTENTS

13.17.1.1 FruitySeqEventInput()	69
13.17.2 Member Function Documentation	69
13.17.2.1 update_mouse_pointer()	69
13.17.2.2 on_button_press_event()	70
13.17.2.3 on_button_release_event()	71
13.17.2.4 on_motion_notify_event()	71
13.17.3 Field Documentation	71
13.17.3.1 m_justselected_one	71
13.17.3.2 m_is_drag_pasting_start	71
13.17.3.3 m_is_drag_pasting	72
13.18seq64::FruitySeqRollInput Class Reference	72
13.18.1 Constructor & Destructor Documentation	72
13.18.1.1 FruitySeqRollInput()	72
13.18.2 Member Function Documentation	72
13.18.2.1 update_mouse_pointer()	72
13.18.2.2 on_button_press_event()	73
13.18.2.3 on_button_release_event()	73
13.18.2.4 on_motion_notify_event()	73
13.18.3 Field Documentation	74
13.18.3.1 m_erase_painting	74
13.18.3.2 m_drag_paste_start_pos	74
13.19seq64::gui_assistant Class Reference	74
13.19.1 Detailed Description	76
13.19.2 Constructor & Destructor Documentation	:76
13.19.2.1 gui_assistant()	76
13.19.2.2 ∼gui_assistant()	:76
13.19.3 Member Function Documentation	76
13.19.3.1 quit()	76
13.19.3.2 jack_idle_connect()	76
13.19.3.3 lash_timeout_connect()	77

CONTENTS xxxiii

13.19.3.4 keys() [1/2]	77
13.19.3.5 keys() [2/2]	77
13.19.4 Field Documentation	77
13.19.4.1 m_keys_perform	77
13.20seq64::gui_assistant_gtk2 Class Reference	77
13.20.1 Constructor & Destructor Documentation	78
13.20.1.1 gui_assistant_gtk2()	79
13.20.1.2 ~gui_assistant_gtk2()	79
13.20.2 Member Function Documentation	79
13.20.2.1 quit()	79
13.20.2.2 lash_timeout_connect()	79
13.20.2.3 jack_idle_connect()	79
13.20.3 Field Documentation	79
13.20.3.1 sm_internal_keys	30
13.21seq64::gui_drawingarea_gtk2 Class Reference	30
13.21.1 Detailed Description	33
13.21.2 Constructor & Destructor Documentation	33
13.21.2.1 gui_drawingarea_gtk2() [1/3]	33
13.21.2.2 gui_drawingarea_gtk2() [2/3]	33
13.21.2.3 gui_drawingarea_gtk2() [3/3]	34
13.21.2.4 ∼gui_drawingarea_gtk2()	34
13.21.3 Member Function Documentation	34
13.21.3.1 operator=()	34
13.21.3.2 window_x()	34
13.21.3.3 window_y()	34
13.21.3.4 current_x()	34
13.21.3.5 current_y()	35
13.21.3.6 drop_x()	35
13.21.3.7 drop_y()	35
13.21.3.8 force_draw()	35

13.21.3.9 perf()
13.21.3.10clear_window()
13.21.3.11set_line()
13.21.3.12draw_line() [1/6]
13.21.3.13draw_line() [2/6]
13.21.3.14draw_line_on_pixmap() [1/2]
13.21.3.15draw_line_on_pixmap() [2/2]
13.21.3.16draw_line() [3/6]
13.21.3.17draw_line() [4/6]
13.21.3.18draw_line() [5/6]
13.21.3.19draw_line() [6/6]
13.21.3.20render_string()
13.21.3.21render_string_on_pixmap()
13.21.3.22draw_rectangle() [1/6]
13.21.3.23draw_rectangle() [2/6]
13.21.3.24draw_rectangle() [3/6]
13.21.3.25draw_rectangle() [4/6]
13.21.3.26draw_rectangle() [5/6]
13.21.3.27draw_rectangle() [6/6]
13.21.3.28draw_rectangle_on_pixmap() [1/2]
13.21.3.29draw_rectangle_on_pixmap() [2/2]
13.21.3.30draw_normal_rectangle_on_pixmap()
13.21.3.31draw_drawable()
13.21.3.32scroll_hadjust()
13.21.3.33scroll_vadjust()
13.21.3.34scroll_hset()
13.21.3.35scroll_vset()
13.21.3.36set_current_drop_x()
13.21.3.37set_current_drop_y()
13.21.3.38gtk_drawarea_init()

CONTENTS XXXV

13.21.3.39on_realize()	96
13.21.4 Field Documentation	96
13.21.4.1 m_gc	96
13.21.4.2 m_window	.96
13.21.4.3 m_vadjust	96
13.21.4.4 m_hadjust	.96
13.21.4.5 m_pixmap	96
13.21.4.6 m_background	97
13.21.4.7 m_foreground	97
13.21.4.8 m_mainperf	97
13.21.4.9 m_window_x	:97
13.21.4.10m_window_y	:97
13.21.4.11m_current_x	97
13.21.4.12m_current_y	97
13.21.4.13m_drop_x	98
13.21.4.14m_drop_y	:98
13.22seq64::gui_palette_gtk2 Class Reference	98
13.22.1 Detailed Description	01
13.22.2 Member Typedef Documentation	01
13.22.2.1 Color	01
13.22.3 Constructor & Destructor Documentation	02
13.22.3.1 gui_palette_gtk2()	02
13.22.3.2 ~gui_palette_gtk2()	02
13.22.4 Member Function Documentation	02
13.22.4.1 load_inverse_palette()	02
13.22.4.2 is_inverse()	03
13.22.4.3 line_color()	03
13.22.4.4 progress_color()	03
13.22.4.5 black()	03
13.22.4.6 dark red()	03

xxxvi CONTENTS

	13.22.4.7 dark_green()	03
	13.22.4.8 dark_orange()	03
	13.22.4.9 dark_blue()	04
	13.22.4.10dark_magenta()	04
	13.22.4.11dark_cyan()	04
	13.22.4.12white()	04
	13.22.4.13grey()	04
	13.22.4.14dark_grey()	04
	13.22.4.15ight_grey()	04
	13.22.4.16red()	04
	13.22.4.17orange()	05
	13.22.4.18yellow()	05
	13.22.4.19green()	05
	13.22.4.20blue()	05
	13.22.4.21black_paint()	05
	13.22.4.22white_paint()	05
	13.22.4.23black_key()	05
	13.22.4.24white_key()	05
	13.22.4.25bg_color() [1/2]	06
	13.22.4.26bg_color() [2/2]	06
	13.22.4.27fg_color() [1/2]	06
	13.22.4.28g_color() [2/2]	06
3.22.5	Field Documentation	06
	13.22.5.1 m_is_inverse	06
	13.22.5.2 m_black	06
	13.22.5.3 m_dk_red	06
	13.22.5.4 m_dk_green	07
	13.22.5.5 m_dk_orange	07
	13.22.5.6 m_dk_blue	07
	13.22.5.7 m_dk_magenta	07

CONTENTS xxxvii

	13.22.5.8 m_dk_cyan	. 307
	13.22.5.9 m_red	. 307
	13.22.5.10m_white	. 307
	13.22.5.11m_orange	. 307
	13.22.5.12m_yellow	. 308
	13.22.5.13m_green	. 308
	13.22.5.14m_blue	. 308
	13.22.5.15m_grey	. 308
	13.22.5.16m_dk_grey	. 308
	13.22.5.17m_lt_grey	. 308
	13.22.5.18m_blk_paint	. 308
	13.22.5.19m_wht_paint	. 308
	13.22.5.20m_blk_key	. 309
	13.22.5.21m_wht_key	. 309
	13.22.5.22m_line_color	. 309
	13.22.5.23m_progress_color	. 309
	13.22.5.24m_bg_color	. 309
	13.22.5.25m_fg_color	. 309
13.23seq64:	gui_window_gtk2 Class Reference	. 310
13.23.1	Constructor & Destructor Documentation	. 311
	13.23.1.1 gui_window_gtk2()	. 311
	13.23.1.2 ~gui_window_gtk2()	. 312
13.23.2	Member Function Documentation	. 312
	13.23.2.1 perf()	. 312
	13.23.2.2 quit()	. 312
	13.23.2.3 redraw_period_ms()	. 312
	13.23.2.4 is_realized()	. 312
	13.23.2.5 scroll_hadjust()	. 312
	13.23.2.6 scroll_vadjust()	. 313
	13.23.2.7 scroll_hset()	. 313

xxxviii CONTENTS

13.23.2.8 scroll_vset()	13
13.23.2.9 on_realize()	13
13.23.3 Field Documentation	14
13.23.3.1 m_mainperf	14
13.23.3.2 m_window_x	14
13.23.3.3 m_window_y	14
13.23.3.4 m_redraw_period_ms	14
13.23.3.5 m_is_realized	14
13.24seq64::jack_assistant Class Reference	14
13.24.1 Constructor & Destructor Documentation	18
13.24.1.1 jack_assistant()	18
13.24.1.2 ~jack_assistant()	18
13.24.2 Member Function Documentation	19
13.24.2.1 parent() [1/2]	19
13.24.2.2 parent() [2/2]	19
13.24.2.3 is_running()	19
13.24.2.4 is_master()	19
13.24.2.5 get_ppqn()	19
13.24.2.6 get_beat_width()	19
13.24.2.7 set_beat_width()	19
13.24.2.8 get_beats_per_measure()	20
13.24.2.9 set_beats_per_measure()	20
13.24.2.10get_beats_per_minute()	20
13.24.2.11set_beats_per_minute()	20
13.24.2.12\text{ransport_state()}	20
13.24.2.13(ransport_not_starting()	21
13.24.2.14init()	21
13.24.2.15deinit()	22
13.24.2.16session_event()	22
13.24.2.17activate()	22

CONTENTS xxxix

13.24.2.18start()
13.24.2.19stop()
13.24.2.20position()
13.24.2.21output()
13.24.2.2&et_ppqn()
13.24.2.23get_jack_tick()
13.24.2.24get_jack_pos()
13.24.2.25toggle_jack_mode()
13.24.2.26set_jack_mode()
13.24.2.27get_jack_mode()
13.24.2.28get_jack_stop_tick()
13.24.2.29set_jack_stop_tick()
13.24.2.3 gack_frame_rate()
13.24.2.31get_follow_transport()
13.24.2.32set_follow_transport()
13.24.2.33toggle_follow_transport()
13.24.2.34toggle_song_start_mode()
13.24.2.35song_start_mode()
13.24.2.36set_start_from_perfedit()
13.24.2.37client()
13.24.2.3&lient_name()
13.24.2.39client_uuid()
13.24.2.40set_jack_running()
13.24.2.41tick_multiplier()
13.24.2.42client_open()
13.24.2.43get_jack_client_info()
13.24.2.44show_position()
13.24.2.45sync()
13.24.2.46set_position()
13.24.2.47info_message()

xI CONTENTS

13.24.2.4&error_message()	0
13.24.3 Friends And Related Function Documentation	2
13.24.3.1 jack_transport_callback	2
13.24.3.2 jack_shutdown_callback	2
13.24.3.3 jack_sync_callback	2
13.24.3.4 jack_timebase_callback	3
13.24.3.5 get_current_jack_position	4
13.24.3.6 jack_session_callback	4
13.24.4 Field Documentation	5
13.24.4.1 sm_status_pairs	5
13.24.4.2 m_jack_parent	5
13.24.4.3 m_jack_client	5
13.24.4.4 m_jack_client_name	5
13.24.4.5 m_jack_client_uuid	5
13.24.4.6 m_jack_frame_current	5
13.24.4.7 m_jack_frame_last	6
13.24.4.8 m_jack_pos	6
13.24.4.9 m_jack_transport_state	6
13.24.4.10m_jack_transport_state_last	6
13.24.4.11m_jack_tick	6
13.24.4.12m_jsession_ev	6
13.24.4.13m_jack_running	6
13.24.4.14m_jack_master	17
13.24.4.15m_jack_frame_rate	17
13.24.4.16m_toggle_jack	17
13.24.4.17m_jack_stop_tick	17
13.24.4.18m_follow_transport	17
13.24.4.19m_ppqn	17
13.24.4.20m_beats_per_measure	17
13.24.4.21m_beat_width	8

CONTENTS xli

13.24.4.22m_beats_per_minute	338
13.25seq64::jack_scratchpad Class Reference	338
13.25.1 Detailed Description	338
13.25.2 Field Documentation	339
13.25.2.1 js_current_tick	339
13.25.2.2 js_total_tick	339
13.25.2.3 js_clock_tick	339
13.25.2.4 js_jack_stopped	339
13.25.2.5 js_dumping	339
13.25.2.6 js_init_clock	339
13.25.2.7 js_looping	339
13.25.2.8 js_playback_mode	340
13.25.2.9 js_ticks_converted	340
13.25.2.10s_ticks_delta	340
13.25.2.11js_ticks_converted_last	340
13.25.2.12/s_delta_tick_frac	340
13.26seq64::jack_status_pair_t Struct Reference	340
13.26.1 Field Documentation	340
13.26.1.1 jf_bit	341
13.26.1.2 jf_meaning	341
13.27seq64::keybindentry Class Reference	341
13.27.1 Member Enumeration Documentation	342
13.27.1.1 type	342
13.27.2 Constructor & Destructor Documentation	342
13.27.2.1 keybindentry()	342
13.27.3 Member Function Documentation	343
13.27.3.1 set()	343
13.27.3.2 on_key_press_event()	343
13.27.4 Friends And Related Function Documentation	343
13.27.4.1 options	343

xlii CONTENTS

13.27.5 Field Documentation	343
13.27.5.1 m_key	343
13.27.5.2 m_type	344
13.27.5.3 m_perf	344
13.27.5.4 m_slot	344
13.28seq64::keys_perform Class Reference	344
13.28.1 Detailed Description	351
13.28.2 Member Typedef Documentation	351
13.28.2.1 SlotMap	351
13.28.2.2 RevSlotMap	351
13.28.3 Constructor & Destructor Documentation	351
13.28.3.1 keys_perform()	352
13.28.3.2 ~keys_perform()	352
13.28.4 Member Function Documentation	352
13.28.4.1 set_keys()	352
13.28.4.2 get_keys()	352
13.28.4.3 bpm_up() [1/2]	352
13.28.4.4 bpm_up() [2/2]	353
13.28.4.5 bpm_dn() [1/2]	353
13.28.4.6 bpm_dn() [2/2]	353
13.28.4.7 replace() [1/2]	353
13.28.4.8 replace() [2/2]	353
13.28.4.9 queue() [1/2]	354
13.28.4.10queue() [2/2]	354
13.28.4.11keep_queue() [1/2]	354
13.28.4.12keep_queue() [2/2]	354
13.28.4.13snapshot_1() [1/2]	354
13.28.4.14snapshot_1() [2/2]	354
13.28.4.15snapshot_2() [1/2]	355
13.28.4.16snapshot_2() [2/2]	355

CONTENTS xliii

13.28.4.17screenset_up() [1/2]
13.28.4.18screenset_up() [2/2]
13.28.4.19screenset_dn() [1/2]
13.28.4.20screenset_dn() [2/2]
13.28.4.21set_playing_screenset() [1/2]
13.28.4.22set_playing_screenset() [2/2]
13.28.4.23group_on() [1/2]
13.28.4.24group_on() [2/2]
13.28.4.25group_off() [1/2]
13.28.4.26group_off() [2/2]
13.28.4.27group_learn() [1/2] 357
13.28.4.2&group_learn() [2/2] 357
13.28.4.29start() [1/2]
13.28.4.30start() [2/2]
13.28.4.31pause() [1/2]
13.28.4.32pause() [2/2]
13.28.4.33pattern_edit() [1/2]
13.28.4.34pattern_edit() [2/2]
13.28.4.35event_edit() [1/2]
13.28.4.36event_edit() [2/2]
13.28.4.37stop() [1/2]
13.28.4.38stop() [2/2]
13.28.4.39song_mode() [1/2]
13.28.4.40song_mode() [2/2]
13.28.4.41menu_mode() [1/2]
13.28.4.42menu_mode() [2/2]
13.28.4.43follow_transport() [1/2]
13.28.4.44follow_transport() [2/2]
13.28.4.45fast_forward() [1/2]
13.28.4.46fast_forward() [2/2]

XIIV CONTENTS

13.28.4.47rewind() [1/2]
13.28.4.48rewind() [2/2]
13.28.4.49pointer_position() [1/2]
13.28.4.5\(\text{Opointer_position()}\) [2/2]
13.28.4.51toggle_mutes() [1/2]
13.28.4.52loggle_mutes() [2/2]
13.28.4.53toggle_jack() [1/2]
13.28.4.54toggle_jack() [2/2]
13.28.4.55tap_bpm() [1/2]
13.28.4.5@tap_bpm() [2/2]
13.28.4.57show_ui_sequence_key() [1/2]
13.28.4.58show_ui_sequence_key() [2/2]
13.28.4.59show_ui_sequence_number() [1/2]
13.28.4.60show_ui_sequence_number() [2/2]
13.28.4.61get_key_events()
13.28.4.62get_key_groups()
13.28.4.63get_key_events_rev()
13.28.4.64get_key_groups_rev()
13.28.4.65ookup_keyevent_key()
13.28.4.6dookup_keyevent_seq()
13.28.4.67lookup_keygroup_key()
13.28.4.68ookup_keygroup_group()
13.28.4.69key_name()
13.28.4.70set_all_key_events()
13.28.4.71set_all_key_groups()
13.28.4.72set_key_event()
13.28.4.73set_key_group()
13.28.4.74at_bpm_up()
13.28.4.75at_bpm_dn()
13.28.4.76at_replace()

CONTENTS xiv

1	3.28.4.77at_queue()	366
1	3.28.4.78at_keep_queue()	366
1	3.28.4.79at_snapshot_1()	366
1	3.28.4.80at_snapshot_2()	366
1	3.28.4.81at_screenset_up()	366
1	3.28.4.82at_screenset_dn()	366
1	3.28.4.83at_set_playing_screenset()	367
1	3.28.4.84at_group_on()	367
1	3.28.4.85at_group_off()	367
1	3.28.4.86at_group_learn()	367
1	3.28.4.87at_start()	367
1	3.28.4.88at_pause()	367
1	3.28.4.89at_song_mode()	367
1	3.28.4.90at_toggle_jack()	367
1	3.28.4.91at_menu_mode()	368
1	3.28.4.92at_follow_transport()	368
1	3.28.4.93at_fast_forward()	368
1	3.28.4.94at_rewind()	368
1	3.28.4.95at_pointer_position()	368
1	3.28.4.96at_toggle_mutes()	368
1	3.28.4.97at_tap_bpm()	368
1	3.28.4.9&at_pattern_edit()	368
1	3.28.4.99at_event_edit()	369
1	3.28.4.10@t_stop()	369
1	3.28.4.10at_show_ui_sequence_key()	369
1	3.28.4.102£_show_ui_sequence_number()	369
13.28.5 F	Friends And Related Function Documentation	369
1	3.28.5.1 options	369
1	3.28.5.2 perform	369
1	3.28.5.3 optionsfile	369

XIVI CONTENTS

3.28.6	6 Field Documentation	370
	13.28.6.1 m_key_show_ui_sequence_key	370
	13.28.6.2 m_key_show_ui_sequence_number	370
	13.28.6.3 m_key_events	370
	13.28.6.4 m_key_groups	370
	13.28.6.5 m_key_events_rev	370
	13.28.6.6 m_key_groups_rev	370
	13.28.6.7 m_key_bpm_up	370
	13.28.6.8 m_key_bpm_dn	371
	13.28.6.9 m_key_replace	371
	13.28.6.10m_key_queue	371
	13.28.6.11m_key_keep_queue	371
	13.28.6.12m_key_snapshot_1	371
	13.28.6.13m_key_snapshot_2	371
	13.28.6.14m_key_screenset_up	371
	13.28.6.15m_key_screenset_dn	371
	13.28.6.16m_key_set_playing_screenset	372
	13.28.6.17m_key_group_on	372
	13.28.6.18m_key_group_off	372
	13.28.6.19m_key_group_learn	372
	13.28.6.20m_key_start	372
	13.28.6.21m_key_pause	372
	13.28.6.22m_key_song_mode	372
	13.28.6.23m_key_toggle_jack	372
	13.28.6.24m_key_menu_mode	373
	13.28.6.25m_key_follow_transport	373
	13.28.6.26m_key_rewind	373
	13.28.6.27m_key_fast_forward	373
	13.28.6.28m_key_pointer_position	373
	13.28.6.29m_key_toggle_mutes	373

CONTENTS xlvii

13.28.6.30m_key_tap_bpm
13.28.6.31m_key_pattern_edit
13.28.6.32m_key_event_edit
13.28.6.33m_key_stop
13.29seq64::keys_perform_gtk2 Class Reference
13.29.1 Detailed Description
13.29.2 Constructor & Destructor Documentation
13.29.2.1 keys_perform_gtk2()
13.29.2.2 ~keys_perform_gtk2()
13.29.3 Member Function Documentation
13.29.3.1 key_name()
13.29.3.2 set_all_key_events()
13.29.3.3 set_all_key_groups()
13.30seq64::keys_perform_transfer Struct Reference
13.30.1 Field Documentation
13.30.1.1 kpt_bpm_up
13.30.1.2 kpt_bpm_dn
13.30.1.3 kpt_screenset_up
13.30.1.4 kpt_screenset_dn
13.30.1.5 kpt_set_playing_screenset
13.30.1.6 kpt_group_on
13.30.1.7 kpt_group_off
13.30.1.8 kpt_group_learn
13.30.1.9 kpt_replace
13.30.1.10kpt_queue
13.30.1.11kpt_keep_queue
13.30.1.12kpt_snapshot_1
13.30.1.13kpt_snapshot_2
13.30.1.14kpt_start
13.30.1.15kpt_stop

xlviii CONTENTS

13.30.1.16kpt_show_ui_sequence_key	38	30
13.30.1.17kpt_show_ui_sequence_number	38	30
13.30.1.18kpt_pattern_edit	38	30
13.30.1.19kpt_event_edit	38	30
13.30.1.20kpt_tap_bpm	38	30
13.30.1.21kpt_pause	38	30
13.30.1.22kpt_song_mode	38	30
13.30.1.23kpt_toggle_jack	38	30
13.30.1.24kpt_menu_mode	38	31
13.30.1.25kpt_follow_transport	38	31
13.30.1.26kpt_fast_forward	38	31
13.30.1.27kpt_rewind	38	31
13.30.1.28kpt_pointer_position	38	31
13.30.1.29kpt_toggle_mutes	38	31
13.31seq64::keystroke Class Reference	38	31
13.31.1 Detailed Description	38	32
13.31.2 Constructor & Destructor Documentation	38	32
13.31.2.1 keystroke() [1/3]	38	33
13.31.2.2 keystroke() [2/3]	38	33
13.31.2.3 keystroke() [3/3]	38	33
13.31.3 Member Function Documentation	38	33
13.31.3.1 operator=()	38	33
13.31.3.2 is_press()	38	34
13.31.3.3 is_letter()	38	34
13.31.3.4 is()	38	34
13.31.3.5 is_delete()	38	34
13.31.3.6 key()	38	35
13.31.3.7 shift_lock()	38	35
13.31.3.8 modifier()	38	35
13.31.3.9 mod_control()	38	35

CONTENTS xlix

13.31.3.10mod_control_shift()	35
13.31.3.11mod_super()	35
13.31.4 Field Documentation	36
13.31.4.1 m_is_press	36
13.31.4.2 m_key	36
13.31.4.3 m_modifier	36
13.32seq64::lash Class Reference	36
13.32.1 Detailed Description	37
13.32.2 Constructor & Destructor Documentation	37
13.32.2.1 lash()	37
13.32.3 Member Function Documentation	37
13.32.3.1 set_alsa_client_id()	37
13.32.3.2 start()	38
13.32.3.3 process_events()	38
13.32.3.4 init()	38
13.32.3.5 handle_event()	38
13.32.3.6 handle_config()	38
13.32.4 Field Documentation	39
13.32.4.1 m_perform	39
13.32.4.2 m_client	39
13.32.4.3 m_lash_args	39
13.32.4.4 m_is_lash_supported	39
13.33seq64::Ifownd Class Reference	90
13.33.1 Detailed Description	91
13.33.2 Constructor & Destructor Documentation	92
13.33.2.1 Ifownd()	92
13.33.2.2 ~lfownd()	92
13.33.3 Member Function Documentation	92
13.33.3.1 toggle_visible()	92
13.33.3.2 scale_lfo_change()	92

I CONTENTS

13.33.3.3 on_focus_out_event()	392
13.33.4 Field Documentation	393
13.33.4.1 m_seq	393
13.33.4.2 m_seqdata	393
13.33.4.3 m_hbox	393
13.33.4.4 m_scale_value	393
13.33.4.5 m_scale_range	393
13.33.4.6 m_scale_speed	393
13.33.4.7 m_scale_phase	393
13.33.4.8 m_scale_wave	394
13.33.4.9 m_wave_name	394
13.33.4.10m_value	394
13.33.4.11m_range	394
13.33.4.12m_speed	394
13.33.4.13m_phase	394
13.33.4.14m_wave	394
13.34seq64::maintime Class Reference	395
13.34.1 Detailed Description	397
13.34.2 Constructor & Destructor Documentation	397
13.34.2.1 maintime() [1/2]	397
13.34.2.2 maintime() [2/2]	397
13.34.2.3 ~maintime()	397
13.34.3 Member Function Documentation	397
13.34.3.1 operator=()	397
13.34.3.2 idle_progress()	397
13.34.3.3 on_realize()	398
13.34.3.4 on_expose_event()	398
13.34.4 Friends And Related Function Documentation	398
13.34.4.1 mainwnd	398
13.34.5 Field Documentation	398

13.34.5.1 m_beat_width	398
13.34.5.2 m_bar_width	399
13.34.5.3 m_pill_width	399
13.34.5.4 m_box_width	399
13.34.5.5 m_box_height	399
13.34.5.6 m_flash_width	399
13.34.5.7 m_flash_height	399
13.34.5.8 m_flash_x	399
13.34.5.9 m_box_less_pill	400
13.34.5.10m_tick	400
13.34.5.11m_ppqn	400
13.35seq64::mainwid Class Reference	400
13.35.1 Detailed Description	404
13.35.2 Constructor & Destructor Documentation	404
13.35.2.1 mainwid()	404
13.35.2.2 ~mainwid()	404
13.35.3 Member Function Documentation	405
13.35.3.1 set_screenset()	405
13.35.3.2 reset()	405
13.35.3.3 update_sequences_on_window()	405
13.35.3.4 draw_pixmap_on_window()	405
13.35.3.5 fill_background_window()	406
13.35.3.6 redraw()	406
13.35.3.7 seq_set_and_edit()	406
13.35.3.8 seq_set_and_eventedit()	406
13.35.3.9 draw_marker_on_sequence()	406
13.35.3.10update_markers()	407
13.35.3.11valid_sequence()	407
13.35.3.12draw_sequence_on_pixmap()	407
13.35.3.13draw_sequences_on_pixmap()	408

lii CONTENTS

13.35.3.14draw_sequence_pixmap_on_window()	 408
13.35.3.15seq_from_xy()	 408
13.35.3.1@imeout()	 409
13.35.3.17calculate_base_sizes()	 409
13.35.3.18select_fg_bg_colors()	 409
13.35.3.19on_realize()	 409
13.35.3.20on_expose_event()	 409
13.35.3.21on_button_press_event()	 410
13.35.3.22on_button_release_event()	 410
13.35.3.23on_motion_notify_event()	 411
13.35.3.24on_focus_in_event()	 411
13.35.3.25on_focus_out_event()	 411
13.35.4 Friends And Related Function Documentation	 412
13.35.4.1 mainwnd	 412
13.35.4.2 update_mainwid_sequences	 412
13.35.5 Field Documentation	 412
13.35.5.1 m_armed_progress_color	 412
13.35.5.2 m_moving_seq	 412
13.35.5.3 m_button_down	 412
13.35.5.4 m_moving	 412
13.35.5.5 m_old_seq	 413
13.35.5.6 m_screenset	 413
13.35.5.7 m_last_tick_x	 413
13.35.5.8 m_last_playing	 413
13.35.5.9 m_mainwnd_rows	 413
13.35.5.10m_mainwnd_cols	 413
13.35.5.11m_seqarea_x	 413
13.35.5.12m_seqarea_y	 413
13.35.5.13m_seqarea_seq_x	 414
13.35.5.14m_seqarea_seq_y	 414

13.3	35.5.15m_mainwid_x	 	414
13.3	35.5.16m_mainwid_y	 	414
13.3	35.5.17m_mainwid_border	 	414
13.3	35.5.18m_mainwid_spacing	 	414
13.3	35.5.19m_text_size_x	 	414
13.3	35.5.20m_text_size_y	 	414
13.3	35.5.21m_max_sets	 	415
13.3	35.5.22m_screenset_slots	 	415
13.3	35.5.23m_screenset_offset	 	415
13.3	35.5.24m_progress_height	 	415
13.36seq64::mair	wnd Class Reference	 	415
13.36.1 Con	structor & Destructor Documentation	 	420
13.3	36.1.1 mainwnd()	 	420
13.5	86.1.2 ∼mainwnd()	 	421
13.36.2 Mer	nber Function Documentation	 	421
13.5	36.2.1 open_file()	 	421
13.3	36.2.2 rc_error_dialog()	 	421
13.3	36.2.3 ppqn() [1/2]	 	422
13.3	36.2.4 ppqn() [2/2]	 	422
13.3	36.2.5 handle_signal()	 	422
13.3	36.2.6 adj_callback_ss()	 	422
13.3	36.2.7 adj_callback_bpm()	 	422
13.3	36.2.8 edit_callback_notepad()	 	422
13.3	36.2.9 timer_callback()	 	423
13.3	36.2.10set_play_image()	 	423
13.3	36.2.11set_songlive_image()	 	423
13.3	36.2.12start_playing()	 	423
13.3	36.2.13pause_playing()	 	424
13.3	36.2.14stop_playing()	 	424
13.3	36.2.15toggle_playing()	 	424

liv CONTENTS

13.36.2.16earn_toggle()
13.36.2.17open_performance_edit()
13.36.2.1&pen_performance_edit_2()
13.36.2.19enregister_perfedits()
13.36.2.20sequence_key()
13.36.2.21apply_song_transpose()
13.36.2.22update_window_title()
13.36.2.23toLower()
13.36.2.24file_new()
13.36.2.25ile_open()
13.36.2.26 ile_save()
13.36.2.27set_song_mute()
13.36.2.28file_import_dialog()
13.36.2.29options_dialog()
13.36.2.3Qack_dialog()
13.36.2.31about_dialog()
13.36.2.32build_info_dialog()
13.36.2.33query_save_changes()
13.36.2.34new_open_error_dialog()
13.36.2.35ile_save_as()
13.36.2.36 ile_exit()
13.36.2.37new_file()
13.36.2.38save_file()
13.36.2.39choose_file()
13.36.2.40s_save()
13.36.2.41install_signal_handlers()
13.36.2.42signal_action()
13.36.2.43edit_field_has_focus()
13.36.2.44populate_menu_file()
13.36.2.45populate_menu_edit()

13.36.2.4\(\phi\)opulate_menu_help()	 429
13.36.2.47populate_menu_view()	 429
13.36.2.48on_delete_event()	 430
13.36.2.49on_key_press_event()	 430
13.36.2.50on_key_release_event()	 430
13.36.2.51on_realize()	 430
13.36.2.52on_grouplearnchange()	 430
13.36.3 Field Documentation	 431
13.36.3.1 m_sigpipe	 431
13.36.3.2 m_tooltips	 431
13.36.3.3 m_menubar	 431
13.36.3.4 m_menu_file	 431
13.36.3.5 m_menu_edit	 431
13.36.3.6 m_menu_view	 431
13.36.3.7 m_menu_help	 431
13.36.3.8 m_ppqn	 432
13.36.3.9 m_main_wid	 432
13.36.3.10m_main_time	 432
13.36.3.11m_perf_edit	 432
13.36.3.12m_perf_edit_2	 432
13.36.3.13m_options	 432
13.36.3.14m_main_cursor	 432
13.36.3.15m_image_play	 433
13.36.3.16m_button_learn	 433
13.36.3.17m_button_stop	 433
13.36.3.18m_button_play	 433
13.36.3.19m_button_perfedit	 433
13.36.3.20m_button_jack	 433
13.36.3.21m_adjust_bpm	 433
13.36.3.22m_spinbutton_bpm	 433

Ivi CONTENTS

	13.36.3.23m_adjust_ss	. 434
	13.36.3.24m_spinbutton_ss	. 434
	13.36.3.25m_adjust_load_offset	. 434
	13.36.3.26m_spinbutton_load_offset	. 434
	13.36.3.27m_entry_notes	. 434
	13.36.3.28m_is_running	. 434
	13.36.3.29m_timeout_connect	. 434
	13.36.3.30m_menu_mode	. 435
	13.36.3.31m_call_seq_edit	. 435
	13.36.3.32m_call_seq_eventedit	. 435
13.37seq64::r	mastermidibase Class Reference	. 435
13.37.1	Constructor & Destructor Documentation	. 439
	13.37.1.1 mastermidibase()	. 439
	13.37.1.2 ~mastermidibase()	. 440
13.37.2	Member Function Documentation	. 440
	13.37.2.1 init()	. 440
	13.37.2.2 get_num_out_buses()	. 440
	13.37.2.3 get_num_in_buses()	. 440
	13.37.2.4 filter_by_channel() [1/2]	. 441
	13.37.2.5 filter_by_channel() [2/2]	. 441
	13.37.2.6 get_beats_per_minute()	. 441
	13.37.2.7 get_bpm()	. 441
	13.37.2.8 get_ppqn()	. 441
	13.37.2.9 is_dumping()	. 441
	13.37.2.10get_sequence()	. 441
	13.37.2.11start()	. 442
	13.37.2.12stop()	. 442
	13.37.2.13port_start()	. 442
	13.37.2.14port_exit()	. 442
	13.37.2.15play()	. 443

13.37.2.16continue_from()
13.37.2.17init_clock()
13.37.2.18set_clock() [1/2]
13.37.2.19sysex()
13.37.2.2\(\text{Qrint}()\)
13.37.2.21flush()
13.37.2.22set_sequence_input()
13.37.2.23dump_midi_input()
13.37.2.24nitialize_buses()
13.37.2.25get_midi_out_bus_name()
13.37.2.26get_midi_in_bus_name()
13.37.2.27poll_for_midi()
13.37.2.28s_more_input()
13.37.2.29get_midi_event()
13.37.2.30set_clock() [2/2]
13.37.2.31set_input()
13.37.2.32get_input()
13.37.2.33s_input_system_port()
13.37.2.34get_clock()
13.37.2.35set_ppqn()
13.37.2.36set_beats_per_minute()
13.37.2.37port_settings()
13.37.2.3&lock()
13.37.2.39nput()
13.37.2.40activate()
13.37.2.41api_init()
13.37.2.42api_start()
13.37.2.43api_continue_from()
13.37.2.44api_init_clock()
13.37.2.45api_stop()

Iviii CONTENTS

13.37.2.46api_set_ppqn()	451
13.37.2.47api_set_beats_per_minute()	451
13.37.2.48api_flush()	452
13.37.2.49api_clock()	452
13.37.2.50api_port_start()	452
13.37.2.51api_is_more_input()	452
13.37.2.52api_get_midi_event()	452
13.37.2.53api_poll_for_midi()	452
13.37.2.54save_clock()	453
13.37.2.55save_input()	453
13.37.3 Friends And Related Function Documentation	453
13.37.3.1 perform	453
13.37.3.2 midi_alsa_info	453
13.37.4 Field Documentation	453
13.37.4.1 m_max_busses	454
13.37.4.2 m_bus_announce	454
13.37.4.3 m_inbus_array	454
13.37.4.4 m_outbus_array	454
13.37.4.5 m_master_clocks	454
13.37.4.6 m_master_inputs	454
13.37.4.7 m_queue	454
13.37.4.8 m_ppqn	455
13.37.4.9 m_beats_per_minute	455
13.37.4.10m_dumping_input	455
13.37.4.11m_vector_sequence	455
13.37.4.12m_filter_by_channel	455
13.37.4.13m_seq	455
13.37.4.14m_mutex	455
13.38seq64::mastermidibus Class Reference	456
13.38.1 Detailed Description	458

13.38.2 Constructor & Destructor Documentation	58
13.38.2.1 mastermidibus() [1/2]	58
13.38.2.2 ~mastermidibus() [1/2]	58
13.38.2.3 mastermidibus() [2/2]	58
13.38.2.4 ~mastermidibus() [2/2]	58
13.38.3 Member Function Documentation	59
13.38.3.1 api_is_more_input() [1/2]	59
13.38.3.2 api_get_midi_event() [1/2]	59
13.38.3.3 api_poll_for_midi() [1/2]	59
13.38.3.4 api_init() [1/2]	30
13.38.3.5 api_set_ppqn() [1/2]	31
13.38.3.6 api_set_beats_per_minute() [1/2]	31
13.38.3.7 api_flush() [1/2]	31
13.38.3.8 api_start()	31
13.38.3.9 api_stop()	62
13.38.3.10api_continue_from()	62
13.38.3.11api_port_start() [1/2]	62
13.38.3.12activate()	62
13.38.3.13api_is_more_input() [2/2]	62
13.38.3.14api_get_midi_event() [2/2]	62
13.38.3.15api_poll_for_midi() [2/2]	63
13.38.3.16api_init() [2/2]	63
13.38.3.17api_set_ppqn() [2/2]	63
13.38.3.1&api_set_beats_per_minute() [2/2]	63
13.38.3.19api_flush() [2/2]	63
13.38.3.20api_port_start() [2/2] 46	63
13.38.3.21port_list()	63
13.38.4 Friends And Related Function Documentation	64
13.38.4.1 midi_alsa_info	64
13.38.4.2 midi_jack_info	64

IX

13.38.5 Field Documentation
13.38.5.1 m_alsa_seq
13.38.5.2 m_num_poll_descriptors
13.38.5.3 m_poll_descriptors
13.38.5.4 m_midi_master
13.38.5.5 m_use_jack_polling
13.39seq64::midi_alsa Class Reference
13.39.1 Constructor & Destructor Documentation
13.39.1.1 midi_alsa()
13.39.1.2 ~midi_alsa()
13.39.2 Member Function Documentation
13.39.2.1 get_client()
13.39.2.2 get_port()
13.39.2.3 api_init_out()
13.39.2.4 api_init_in()
13.39.2.5 api_init_out_sub()
13.39.2.6 api_init_in_sub()
13.39.2.7 api_deinit_in()
13.39.2.8 api_get_midi_event()
13.39.2.9 api_poll_for_midi()
13.39.2.10api_play()
13.39.2.11api_sysex()
13.39.2.12api_flush()
13.39.2.13api_continue_from()
13.39.2.14api_start()
13.39.2.15api_stop()
13.39.2.16api_clock()
13.39.2.17api_set_ppqn()
13.39.2.18api_set_beats_per_minute()
13.39.2.19set_virtual_name()

13.39.3 Field Documentation	⁷ 4
13.39.3.1 m_seq	74
13.39.3.2 m_dest_addr_client	74
13.39.3.3 m_dest_addr_port	74
13.39.3.4 m_local_addr_client	74
13.39.3.5 m_local_addr_port	74
13.39.3.6 m_input_port_name	74
13.40seq64::midi_alsa_info Class Reference	⁷ 5
13.40.1 Constructor & Destructor Documentation	⁷ 6
13.40.1.1 midi_alsa_info()	⁷ 6
13.40.1.2 ~midi_alsa_info()	77
13.40.2 Member Function Documentation	77
13.40.2.1 seq()	77
13.40.2.2 api_get_midi_event()	77
13.40.2.3 api_poll_for_midi()	78
13.40.2.4 api_set_ppqn()	78
13.40.2.5 api_set_beats_per_minute()	78
13.40.2.6 api_port_start()	78
13.40.2.7 api_flush()	79
13.40.2.8 get_all_port_info()	79
13.40.3 Field Documentation	30
13.40.3.1 sm_input_caps	30
13.40.3.2 sm_output_caps	30
13.40.3.3 m_alsa_seq	30
13.40.3.4 m_num_poll_descriptors	30
13.40.3.5 m_poll_descriptors	30
13.41seq64::midi_api Class Reference	31
13.41.1 Detailed Description	33
13.41.2 Constructor & Destructor Documentation	33
13.41.2.1 midi_api()	33

lxii CONTENTS

13.41.2.2 ∼midi_api()
13.41.3 Member Function Documentation
13.41.3.1 is_input_port()
13.41.3.2 is_virtual_port()
13.41.3.3 is_system_port()
13.41.3.4 api_connect()
13.41.3.5 api_init_out()
13.41.3.6 api_init_out_sub()
13.41.3.7 api_init_in()
13.41.3.8 api_init_in_sub()
13.41.3.9 api_deinit_in()
13.41.3.10api_get_midi_event()
13.41.3.11api_poll_for_midi()
13.41.3.12api_play()
13.41.3.13api_sysex()
13.41.3.14api_continue_from()
13.41.3.15api_start()
13.41.3.16api_stop()
13.41.3.17api_flush()
13.41.3.18api_clock()
13.41.3.19api_set_ppqn()
13.41.3.20api_set_beats_per_minute()
13.41.3.21api_get_bus_name()
13.41.3.22api_get_port_name()
13.41.3.23s_port_open()
13.41.3.24master_info() [1/2] 488
13.41.3.25master_info() [2/2]
13.41.3.2@parent_bus() [1/2]
13.41.3.27parent_bus() [2/2]
13.41.3.28master_midi_mode()

13.41.3.29error()	488
13.41.3.30user_callback()	489
13.41.3.31cancel_callback()	489
13.41.3.32set_port_open()	489
13.41.3.33nput_data()	489
13.41.4 Field Documentation	489
13.41.4.1 m_master_info	489
13.41.4.2 m_parent_bus	490
13.41.4.3 m_input_data	490
13.41.4.4 m_connected	490
13.41.4.5 m_error_string	490
13.41.4.6 m_error_callback	490
13.41.4.7 m_first_error_occurred	490
13.41.4.8 m_error_callback_user_data	490
13.42seq64::midi_container Class Reference	491
13.42.1 Detailed Description	493
13.42.2 Constructor & Destructor Documentation	493
13.42.2.1 midi_container()	493
13.42.2.2 ~midi_container()	493
13.42.3 Member Function Documentation	493
13.42.3.1 fill()	493
13.42.3.2 size()	494
13.42.3.3 done()	494
13.42.3.4 put()	495
13.42.3.5 get()	495
13.42.3.6 clear()	495
13.42.3.7 position_reset()	495
13.42.3.8 position()	495
13.42.3.9 position_increment()	495
13.42.3.10add_variable()	495

lxiv CONTENTS

13.42.3.11add_long()	496
13.42.3.12add_short()	496
13.42.3.13add_event()	496
13.42.3.14fill_seq_number()	496
13.42.3.15fill_seq_name()	497
13.42.3.16 ill_meta_track_end()	497
13.42.3.17/ill_proprietary()	497
13.42.3.18 ill_time_sig_and_tempo()	498
13.42.3.19song_fill_seq_event()	499
13.42.3.20song_fill_seq_trigger()	499
13.42.4 Friends And Related Function Documentation	499
13.42.4.1 midifile	499
13.42.5 Field Documentation	500
13.42.5.1 m_sequence	500
13.42.5.2 m_position_for_get	500
13.43seq64::midi_control Class Reference	500
13.43.1 Detailed Description	501
13.43.2 Member Enumeration Documentation	501
13.43.2.1 action	501
13.43.3 Constructor & Destructor Documentation	502
13.43.3.1 midi_control()	502
13.43.4 Member Function Documentation	502
13.43.4.1 active()	502
13.43.4.2 inverse_active()	
13.43.4.2 inverse_active()	502
	502 502
13.43.4.3 status()	502 502 503
13.43.4.3 status()	502503503
13.43.4.3 status()	502503503503

13.43.4.9 match()	504
13.43.4.10n_range()	504
13.43.5 Field Documentation	504
13.43.5.1 m_active	504
13.43.5.2 m_inverse_active	504
13.43.5.3 m_status	504
13.43.5.4 m_data	504
13.43.5.5 m_min_value	505
13.43.5.6 m_max_value	505
13.44seq64::midi_in_alsa Class Reference	505
13.44.1 Constructor & Destructor Documentation	507
13.44.1.1 midi_in_alsa()	507
13.45seq64::midi_in_jack Class Reference	507
13.45.1 Constructor & Destructor Documentation	509
13.45.1.1 midi_in_jack()	509
13.45.1.2 ~midi_in_jack()	509
13.45.2 Member Function Documentation	509
13.45.2.1 api_poll_for_midi()	510
13.45.2.2 api_get_midi_event()	510
13.45.2.3 open_client()	510
13.45.3 Field Documentation	510
13.45.3.1 m_client_name	511
13.46seq64::midi_info Class Reference	511
13.46.1 Constructor & Destructor Documentation	514
13.46.1.1 midi_info()	514
13.46.1.2 ~midi_info()	514
13.46.2 Member Function Documentation	514
13.46.2.1 midi_mode() [1/2]	514
13.46.2.2 midi_mode() [2/2]	514
13.46.2.3 midi_handle() [1/2]	514

lxvi CONTENTS

13.46.2.4 input_ports()
13.46.2.5 output_ports()
13.46.2.6 full_port_count()
13.46.2.7 clear()
13.46.2.8 app_name()
13.46.2.9 ppqn()
13.46.2.10bpm()
13.46.2.11api_set_ppqn()
13.46.2.12api_set_beats_per_minute()
13.46.2.13api_port_start()
13.46.2.14api_get_midi_event()
13.46.2.15api_poll_for_midi()
13.46.2.16api_flush()
13.46.2.17api_connect()
13.46.2.1&get_port_count()
13.46.2.19get_bus_id()
13.46.2.20get_bus_name()
13.46.2.21get_port_id()
13.46.2.22get_port_name()
13.46.2.23get_input()
13.46.2.24get_virtual()
13.46.2.25get_system()
13.46.2.26queue_number()
13.46.2.27connect_name()
13.46.2.2&port_list()
13.46.2.29global_queue() [1/2] 518
13.46.2.30error()
13.46.2.31get_all_port_info()
13.46.2.32add_bus()
13.46.2.33global_queue() [2/2] 519

13.46.2.34 midi_handle() [2/2]	519
13.46.2.35bus_container()	519
13.46.2.36nc_midi_port_info()	520
13.46.2.37ref_midi_port_info()	520
13.46.3 Friends And Related Function Documentation	520
13.46.3.1 rtmidi_info	520
13.46.4 Field Documentation	520
13.46.4.1 m_midi_mode_input	520
13.46.4.2 m_input	520
13.46.4.3 m_output	520
13.46.4.4 m_bus_container	521
13.46.4.5 m_global_queue	521
13.46.4.6 m_midi_handle	521
13.46.4.7 m_app_name	521
13.46.4.8 m_ppqn	521
13.46.4.9 m_bpm	521
13.46.4.10m_error_string	521
13.47seq64::midi_jack Class Reference	522
13.47.1 Constructor & Destructor Documentation	524
13.47.1.1 midi_jack()	525
13.47.1.2 ∼midi_jack()	526
13.47.2 Member Function Documentation	526
13.47.2.1 multi_client()	526
13.47.2.2 client_handle() [1/2]	526
13.47.2.3 jack_data()	526
13.47.2.4 remote_port_name() [1/2]	526
13.47.2.5 remote_port_name() [2/2]	527
13.47.2.6 port_handle() [1/2]	527
13.47.2.7 client_handle() [2/2]	527
13.47.2.8 port_handle() [2/2]	527

Ixviii CONTENTS

13.47.2.9 open_client_impl()	. !	527
13.47.2.1@lose_client()	. !	528
13.47.2.11close_port()	. !	528
13.47.2.12create_ringbuffer()	. !	528
13.47.2.13connect_port()	. !	528
13.47.2.14register_port()	. !	529
13.47.2.15open_client()	. !	529
13.47.2.16api_connect()	. !	529
13.47.2.17api_init_out()	. !	530
13.47.2.1&api_init_in()	. !	530
13.47.2.19api_init_out_sub()	. !	531
13.47.2.20api_init_in_sub()	. !	531
13.47.2.21api_deinit_in()	. !	531
13.47.2.22api_get_midi_event()	. !	531
13.47.2.23api_poll_for_midi()	. !	532
13.47.2.24api_play()	. !	532
13.47.2.25api_sysex()	. !	532
13.47.2.26api_flush()	. !	532
13.47.2.27api_continue_from()	. !	533
13.47.2.2&api_start()	. !	533
13.47.2.29api_stop()	. !	533
13.47.2.30api_clock()	. !	533
13.47.2.31api_set_ppqn()	. !	533
13.47.2.32api_set_beats_per_minute()	. !	534
13.47.2.33api_get_port_name()	. !	534
13.47.2.34set_virtual_name()	. !	534
13.47.3 Friends And Related Function Documentation	. !	534
13.47.3.1 midi_jack_info	. !	534
13.47.4 Field Documentation	. !	535
13.47.4.1 m_multi_client	. !	535

13.47.4.2 m_remote_port_name	35
13.47.4.3 m_jack_info	35
13.47.4.4 m_jack_data	35
13.48seq64::midi_jack_data Struct Reference	35
13.48.1 Detailed Description	36
13.48.2 Constructor & Destructor Documentation	36
13.48.2.1 midi_jack_data()	36
13.48.2.2 ~midi_jack_data()	36
13.48.3 Member Function Documentation	36
13.48.3.1 valid_buffer()	36
13.48.4 Field Documentation	37
13.48.4.1 m_jack_client	37
13.48.4.2 m_jack_port	37
13.48.4.3 m_jack_buffsize	37
13.48.4.4 m_jack_buffmessage	37
13.48.4.5 m_jack_lasttime	37
13.48.4.6 m_jack_rtmidiin	37
13.49seq64::midi_jack_info Class Reference	38
13.49.1 Constructor & Destructor Documentation	1 0
13.49.1.1 midi_jack_info()	1 0
13.49.1.2 ~midi_jack_info()	1 0
13.49.2 Member Function Documentation	1 0
13.49.2.1 multi_client()	1 0
13.49.2.2 client_handle() [1/2] 54	41
13.49.2.3 api_get_midi_event()	41
13.49.2.4 api_connect()	41
13.49.2.5 api_poll_for_midi()	41
13.49.2.6 api_set_ppqn()	41
13.49.2.7 api_set_beats_per_minute()	1 2
13.49.2.8 api_port_start()	12

IXX CONTENTS

13.49.2.9 api_flush()	543
13.49.2.10get_all_port_info()	543
13.49.2.11client_handle() [2/2]	544
13.49.2.12connect()	544
13.49.2.13disconnect()	544
13.49.2.14extract_names()	544
13.49.2.15add()	544
13.49.3 Friends And Related Function Documentation	544
13.49.3.1 midi_jack	544
13.49.3.2 jack_process_io	545
13.49.4 Field Documentation	545
13.49.4.1 m_multi_client	545
13.49.4.2 m_jack_ports	545
13.49.4.3 m_jack_client	545
13.50seq64::midi_list Class Reference	545
13.50.1 Member Typedef Documentation	547
13.50.1.1 CharList	547
13.50.2 Constructor & Destructor Documentation	547
13.50.2.1 midi_list()	547
13.50.2.2 ~midi_list()	548
13.50.3 Member Function Documentation	548
13.50.3.1 size()	548
13.50.3.2 done()	548
13.50.3.3 put()	548
13.50.3.4 get()	548
13.50.3.5 clear()	549
13.50.4 Field Documentation	549
13.50.4.1 m_char_list	549
13.51seq64::midi_measures Class Reference	549
13.51.1 Detailed Description	550

13.51.2 Constructor & Destructor Documentation	50
13.51.2.1 midi_measures() [1/2] 5.	50
13.51.2.2 midi_measures() [2/2] 5.	50
13.51.3 Member Function Documentation	50
13.51.3.1 measures() [1/2] 5	50
13.51.3.2 measures() [2/2] 5	50
13.51.3.3 beats() [1/2] 5	51
13.51.3.4 beats() [2/2] 5	51
13.51.3.5 divisions() [1/2]	51
13.51.3.6 divisions() [2/2]	51
13.51.4 Field Documentation	51
13.51.4.1 m_measures	51
13.51.4.2 m_beats	52
13.51.4.3 m_divisions	52
13.52seq64::midi_message Class Reference	52
13.52.1 Detailed Description	53
13.52.2 Member Typedef Documentation	53
13.52.2.1 container	53
13.52.3 Constructor & Destructor Documentation	53
13.52.3.1 midi_message()	53
13.52.4 Member Function Documentation	53
13.52.4.1 operator[]()	53
13.52.4.2 at() [1/2] 5	53
13.52.4.3 at() [2/2] 5	554
13.52.4.4 array()	554
13.52.4.5 count()	554
13.52.4.6 empty()	554
13.52.4.7 push()	554
13.52.4.8 timestamp() [1/2]	554
13.52.4.9 timestamp() [2/2]	554

Ixxii CONTENTS

13.52.5 Field Documentation
13.52.5.1 m_bytes
13.52.5.2 m_timestamp
13.53seq64::midi_out_alsa Class Reference
13.53.1 Constructor & Destructor Documentation
13.53.1.1 midi_out_alsa()
13.54seq64::midi_out_jack Class Reference
13.54.1 Constructor & Destructor Documentation
13.54.1.1 midi_out_jack()
13.54.1.2 ~midi_out_jack()
13.54.2 Member Function Documentation
13.54.2.1 send_message()
13.54.2.2 open_client()
13.55seq64::midi_port_info Class Reference
13.55.1 Constructor & Destructor Documentation
13.55.1.1 midi_port_info()
13.55.2 Member Function Documentation
13.55.2.1 add() [1/2]
13.55.2.2 add() [2/2]
13.55.2.3 clear()
13.55.2.4 get_port_count()
13.55.2.5 get_bus_id()
13.55.2.6 get_bus_name()
13.55.2.7 get_port_id()
13.55.2.8 get_port_name()
13.55.2.9 get_input()
13.55.2.10get_virtual()
13.55.2.11get_system()
13.55.2.12get_queue_number()
13.55.2.13connect_name()

13.55.3 Field Documentation	3
13.55.3.1 m_port_count	3
13.55.3.2 m_port_container	64
13.56seq64::midi_queue Class Reference	64
13.56.1 Detailed Description	64
13.56.2 Constructor & Destructor Documentation	5
13.56.2.1 midi_queue()	5
13.56.2.2 ~midi_queue()	5
13.56.3 Member Function Documentation	i5
13.56.3.1 empty()	i5
13.56.3.2 count()	5
13.56.3.3 full()	5
13.56.3.4 add()	5
13.56.3.5 pop()	6
13.56.3.6 pop_front()	6
13.56.3.7 allocate()	6
13.56.3.8 deallocate()	6
13.56.3.9 front()	6
13.56.4 Field Documentation	57
13.56.4.1 m_front	57
13.56.4.2 m_back	67
13.56.4.3 m_size	57
13.56.4.4 m_ring_size	57
13.56.4.5 m_ring	57
13.57seq64::midi_splitter Class Reference	57
13.57.1 Detailed Description	8
13.57.2 Constructor & Destructor Documentation	9
13.57.2.1 midi_splitter()	9
13.57.2.2 ~midi_splitter()	′0
13.57.3 Member Function Documentation	'0

Ixxiv

13.57.3.1 log_main_sequence()	570
13.57.3.2 initialize()	570
13.57.3.3 increment()	570
13.57.3.4 split()	571
13.57.3.5 ppqn()	571
13.57.3.6 count()	571
13.57.3.7 split_channel()	572
13.57.4 Field Documentation	572
13.57.4.1 m_ppqn	572
13.57.4.2 m_use_default_ppqn	572
13.57.4.3 m_smf0_channels_count	573
13.57.4.4 m_smf0_channels	573
13.57.4.5 m_smf0_main_sequence	573
13.57.4.6 m_smf0_seq_number	573
13.58seq64::midi_timing Class Reference	573
13.58.1 Detailed Description	574
13.58.2 Constructor & Destructor Documentation	574
13.58.2.1 midi_timing() [1/2]	574
13.58.2.2 midi_timing() [2/2]	574
13.58.3 Member Function Documentation	574
13.58.3.1 beats_per_minute() [1/2]	575
13.58.3.2 beats_per_minute() [2/2]	575
13.58.3.3 beats_per_measure() [1/2]	575
13.58.3.4 beats_per_measure() [2/2]	575
13.58.3.5 beat_width() [1/2]	575
13.58.3.6 beat_width() [2/2]	575
13.58.3.7 ppqn() [1/2]	576
13.58.3.8 ppqn() [2/2]	576
13.58.4 Field Documentation	576
13.58.4.1 m_beats_per_minute	576

13.58.4.2 m_beats_per_measure	76
13.58.4.3 m_beat_width	76
13.58.4.4 m_ppqn	77
13.59seq64::midi_vector Class Reference	77
13.59.1 Member Typedef Documentation	79
13.59.1.1 CharVector	79
13.59.2 Constructor & Destructor Documentation	79
13.59.2.1 midi_vector()	79
13.59.2.2 ~midi_vector()	80
13.59.3 Member Function Documentation	80
13.59.3.1 size()	80
13.59.3.2 done()	80
13.59.3.3 put()	80
13.59.3.4 get()	81
13.59.3.5 clear()	81
13.59.4 Field Documentation	81
13.59.4.1 m_char_vector	81
13.60seq64::midibase Class Reference	81
13.60.1 Constructor & Destructor Documentation	86
13.60.1.1 midibase()	87
13.60.1.2 ~midibase()	88
13.60.2 Member Function Documentation	88
13.60.2.1 show_bus_values()	88
13.60.2.2 display_name() [1/2]	88
13.60.2.3 bus_name() [1/2]	88
13.60.2.4 port_name() [1/2]	88
13.60.2.5 connect_name()	88
13.60.2.6 get_bus_index()	88
13.60.2.7 get_bus_id()	89
13.60.2.8 get_port_id()	89

lxxvi CONTENTS

13.60.2.9 ppqn()
13.60.2.10bpm()
13.60.2.11match()
13.60.2.12/s_virtual_port() [1/2]
13.60.2.13s_virtual_port() [2/2]
13.60.2.14s_input_port() [1/2]
13.60.2.15s_output_port()
13.60.2.16s_input_port() [2/2] 590
13.60.2.17s_system_port()
13.60.2.1&set_system_port_flag()
13.60.2.19set_clock()
13.60.2.20get_clock()
13.60.2.21set_clock_status()
13.60.2.22get_input()
13.60.2.23set_input_status()
13.60.2.24queue_number()
13.60.2.25set_bus_id()
13.60.2.26set_name()
13.60.2.27set_alt_name()
13.60.2.28set_multi_name()
13.60.2.29set_clock_mod()
13.60.2.30get_clock_mod()
13.60.2.31poll_for_midi()
13.60.2.32get_midi_event()
13.60.2.33nit_out()
13.60.2.34nit_in()
13.60.2.35deinit_in()
13.60.2.36nit_out_sub()
13.60.2.37init_in_sub()
13.60.2.3&play()

13.60.2.39sysex()	 595
13.60.2.40flush()	 595
13.60.2.41start()	 595
13.60.2.42stop()	 596
13.60.2.43clock()	 596
13.60.2.44continue_from()	 596
13.60.2.45nit_clock()	 596
13.60.2.4@rint()	 597
13.60.2.47set_input()	 597
13.60.2.4&display_name() [2/2]	 597
13.60.2.49bus_name() [2/2]	 597
13.60.2.50port_name() [2/2]	 597
13.60.2.51set_port_id()	 597
13.60.2.52api_poll_for_midi()	 598
13.60.2.53api_get_midi_event()	 598
13.60.2.54api_init_in_sub()	 598
13.60.2.55api_init_out_sub()	 598
13.60.2.56api_deinit_in()	 598
13.60.2.57api_play()	 599
13.60.2.5&api_sysex()	 599
13.60.2.59api_flush()	 599
13.60.2.60api_init_in()	 599
13.60.2.61api_init_out()	 599
13.60.2.62api_continue_from()	 600
13.60.2.63api_start()	 600
13.60.2.64api_stop()	 600
13.60.2.65api_clock()	 600
13.60.3 Friends And Related Function Documentation	 600
13.60.3.1 mastermidibus	 600
13.60.4 Field Documentation	 600

Ixxviii CONTENTS

13.60.4.1 m_clock_mod	601
13.60.4.2 m_bus_index	601
13.60.4.3 m_bus_id	601
13.60.4.4 m_port_id	601
13.60.4.5 m_clock_type	601
13.60.4.6 m_inputing	601
13.60.4.7 m_ppqn	601
13.60.4.8 m_bpm	601
13.60.4.9 m_queue	602
13.60.4.10m_display_name	602
13.60.4.11m_bus_name	602
13.60.4.12m_port_name	602
13.60.4.13m_lasttick	602
13.60.4.14m_is_virtual_port	602
13.60.4.15m_is_input_port	602
13.60.4.16m_is_system_port	603
13.60.4.17m_mutex	603
13.61 seq64::midibus Class Reference	603
13.61.1 Detailed Description	606
13.61.2 Constructor & Destructor Documentation	606
13.61.2.1 midibus() [1/3]	606
13.61.2.2 midibus() [2/3]	607
13.61.2.3 ~midibus() [1/2]	607
13.61.2.4 midibus() [3/3]	607
13.61.2.5 ~midibus() [2/2]	607
13.61.3 Member Function Documentation	608
13.61.3.1 get_client()	608
13.61.3.2 get_port()	608
13.61.3.3 api_init_out() [1/2]	608
13.61.3.4 api_init_in() [1/2]	608

	13.61.3.5 api_init_out_sub() [1/2]	609
	13.61.3.6 api_init_in_sub() [1/2]	609
	13.61.3.7 api_deinit_in() [1/2]	609
	13.61.3.8 api_play() [1/2]	609
	13.61.3.9 api_sysex()	610
	13.61.3.10api_flush()	610
	13.61.3.11api_continue_from() [1/2]	610
	13.61.3.12api_start() [1/2]	610
	13.61.3.13api_stop() [1/2]	611
	13.61.3.14api_clock() [1/2]	611
	13.61.3.15set_virtual_name()	611
	13.61.3.16api_connect()	611
	13.61.3.17api_init_in() [2/2]	612
	13.61.3.18api_init_in_sub() [2/2]	612
	13.61.3.19api_init_out() [2/2]	612
	13.61.3.20api_init_out_sub() [2/2]	612
	13.61.3.21api_deinit_in() [2/2]	612
	13.61.3.22api_get_midi_event()	612
	13.61.3.23api_poll_for_midi()	613
	13.61.3.24api_continue_from() [2/2]	613
	13.61.3.25api_start() [2/2]	613
	13.61.3.26api_stop() [2/2]	613
	13.61.3.27api_clock() [2/2]	614
	13.61.3.28api_play() [2/2]	614
13.61.4	Friends And Related Function Documentation	614
	13.61.4.1 mastermidibus	614
13.61.5	Field Documentation	614
	13.61.5.1 m_seq	614
	13.61.5.2 m_dest_addr_client	614
	13.61.5.3 m_dest_addr_port	615

IXXX

13.61.5.4 m_local_addr_client	15
13.61.5.5 m_local_addr_port	15
13.61.5.6 m_input_port_name	15
13.61.5.7 m_rt_midi	15
13.61.5.8 m_master_info	15
13.62seq64::midifile Class Reference	15
13.62.1 Detailed Description	18
13.62.2 Constructor & Destructor Documentation	18
13.62.2.1 midifile()	18
13.62.2.2 ~midifile()	19
13.62.3 Member Function Documentation	19
13.62.3.1 parse()	19
13.62.3.2 write()	20
13.62.3.3 write_song()	21
13.62.3.4 error_message()	21
13.62.3.5 error_is_fatal()	21
13.62.3.6 ppqn()	21
13.62.3.7 parse_smf_0()	21
13.62.3.8 parse_smf_1()	22
13.62.3.9 parse_prop_header()	22
13.62.3.1@parse_proprietary_track()	23
13.62.3.11pow2()	24
13.62.3.12checklen()	24
13.62.3.13add_trigger()	25
13.62.3.14read_long()	25
13.62.3.15read_short()	25
13.62.3.16read_byte()	26
13.62.3.17read_varinum()	26
13.62.3.18write_long()	26
13.62.3.19write_short()	26

	3.62.3.20read_byte_array()	27
	3.62.3.21write_byte()	27
	3.62.3.22write_varinum()	27
	3.62.3.23write_track_name()	28
	3.62.3.24read_track_name()	28
	3.62.3.25write_seq_number()	28
	3.62.3.26read_seq_number()	29
	3.62.3.27write_track_end()	29
	3.62.3.28write_header()	29
	3.62.3.29write_prop_header()	30
	3.62.3.30write_proprietary_track()	30
	3.62.3.31varinum_size()	31
	3.62.3.32prop_item_size()	31
	3.62.3.33track_name_size()	32
	3.62.3.34errdump() [1/2]	32
	3.62.3.35errdump() [2/2]	32
	3.62.3.36write_track()	33
	3.62.3.37seq_number_size()	33
	3.62.3.3&track_end_size()	33
	3.62.3.39s_sysex_special_id()	33
3.62.4	Field Documentation	34
	3.62.4.1 m_mutex	34
	3.62.4.2 m_file_size	34
	3.62.4.3 m_error_message	34
	3.62.4.4 m_error_is_fatal	34
	3.62.4.5 m_disable_reported	34
	3.62.4.6 m_pos	34
	3.62.4.7 m_name	34
	3.62.4.8 m_data	35
	3.62.4.9 m_char_list	35

Ixxxii CONTENTS

13.62.4.10m_new_format	635
13.62.4.11m_global_bgsequence	635
13.62.4.12m_ppqn	635
13.62.4.13m_use_default_ppqn	635
13.62.4.14m_smf0_splitter	635
13.63seq64::mutex Class Reference	636
13.63.1 Constructor & Destructor Documentation	637
13.63.1.1 mutex()	637
13.63.2 Member Function Documentation	637
13.63.2.1 lock()	637
13.63.2.2 unlock()	637
13.63.3 Field Documentation	637
13.63.3.1 sm_recursive_mutex	637
13.63.3.2 m_mutex_lock	637
13.64seq64::editable_event::name_value_t Struct Reference	637
13.64.1 Field Documentation	638
13.64.1.1 event_value	638
13.64.1.2 event_name	638
13.65seq64::options Class Reference	638
13.65.1 Member Enumeration Documentation	639
13.65.1.1 button	640
13.65.2 Constructor & Destructor Documentation	640
13.65.2.1 options()	640
13.65.3 Member Function Documentation	640
13.65.3.1 perf()	640
13.65.3.2 clock_callback_off()	641
13.65.3.3 clock_callback_on()	641
13.65.3.4 clock_callback_mod()	641
13.65.3.5 clock_mod_callback()	641
13.65.3.6 input_callback()	641

13.65.3.7 filter_callback()	641
13.65.3.8 transport_callback()	642
13.65.3.9 mouse_seq24_callback()	642
13.65.3.10mouse_fruity_callback()	642
13.65.3.11mouse_mod4_callback()	642
13.65.3.12mouse_snap_split_callback()	642
13.65.3.13mouse_click_edit_callback()	642
13.65.3.14ash_support_callback()	642
13.65.3.15add_midi_clock_page()	643
13.65.3.16add_midi_input_page()	643
13.65.3.17add_keyboard_page()	643
13.65.3.18add_extended_keys_page()	643
13.65.3.19add_mouse_page()	643
13.65.3.20add_jack_sync_page()	643
13.65.4 Field Documentation	643
13.65.4.1 m_tooltips	643
13.65.4.2 m_mainperf	644
13.65.4.3 m_button_ok	644
13.65.4.4 m_button_jack_transport	644
13.65.4.5 m_button_jack_master	644
13.65.4.6 m_button_jack_master_cond	644
13.65.4.7 m_button_jack_connect	644
13.65.4.8 m_button_jack_disconnect	644
13.65.4.9 m_notebook	644
13.66seq64::optionsfile Class Reference	645
13.66.1 Detailed Description	646
13.66.2 Constructor & Destructor Documentation	646
13.66.2.1 optionsfile()	646
13.66.2.2 ~optionsfile()	646
13.66.3 Member Function Documentation	646

IXXXIV

13.66.3.1 parse()	46
13.66.3.2 write()	48
13.66.3.3 error_message()	48
13.67seq64::perfedit Class Reference	49
13.67.1 Detailed Description	53
13.67.2 Constructor & Destructor Documentation	53
13.67.2.1 perfedit()	53
13.67.2.2 ~perfedit()	54
13.67.3 Member Function Documentation	54
13.67.3.1 init_before_show()	54
13.67.3.2 enqueue_draw()	54
13.67.3.3 zoom_check()	54
13.67.3.4 enregister_peer()	55
13.67.3.5 set_zoom()	55
13.67.3.6 get_toggle_jack()	55
13.67.3.7 toggle_jack()	55
13.67.3.8 rewind()	56
13.67.3.9 fast_forward()	56
13.67.3.10set_follow_transport()	56
13.67.3.11toggle_follow_transport()	56
13.67.3.12set_jack_mode()	56
13.67.3.13set_transpose()	56
13.67.3.14transpose_button_callback()	57
13.67.3.15set_beats_per_bar()	57
13.67.3.16set_beat_width()	57
13.67.3.17set_snap()	58
13.67.3.18set_guides()	58
13.67.3.19grow()	58
13.67.3.20set_looped()	58
13.67.3.21expand()	58

13.67.3.22collapse()	358
13.67.3.23copy()	359
13.67.3.24undo()	359
13.67.3.25redo()	359
13.67.3.2φορυp_menu()	359
13.67.3.27draw_sequences()	359
13.67.3.28imeout()	359
13.67.3.29set_image()	359
13.67.3.30start_playing()	360
13.67.3.31pause_playing()	360
13.67.3.3≿top_playing()	360
13.67.3.33toggle_playing()	360
13.67.3.34on_realize()	360
13.67.3.35on_key_press_event()	360
13.67.3.36on_key_release_event()	361
13.67.3.37on_delete_event()	361
13.67.4 Friends And Related Function Documentation	361
13.67.4.1 update_perfedit_sequences	361
13.67.5 Field Documentation	361
13.67.5 Field Documentation	
13.67.5.1 m_peer_perfedit	
13.67.5.1 m_peer_perfedit	361
13.67.5.1 m_peer_perfedit 6 13.67.5.2 m_table 6 13.67.5.3 m_vadjust 6	661 662
13.67.5.1 m_peer_perfedit 6 13.67.5.2 m_table 6 13.67.5.3 m_vadjust 6 13.67.5.4 m_hadjust 6	661 662 662
13.67.5.1 m_peer_perfedit 6 13.67.5.2 m_table 6 13.67.5.3 m_vadjust 6 13.67.5.4 m_hadjust 6 13.67.5.5 m_vscroll 6	661 662 662
13.67.5.1 m_peer_perfedit 6 13.67.5.2 m_table 6 13.67.5.3 m_vadjust 6 13.67.5.4 m_hadjust 6 13.67.5.5 m_vscroll 6 13.67.5.6 m_hscroll 6	661 662 662 662
13.67.5.1 m_peer_perfedit 6 13.67.5.2 m_table 6 13.67.5.3 m_vadjust 6 13.67.5.4 m_hadjust 6 13.67.5.5 m_vscroll 6 13.67.5.6 m_hscroll 6 13.67.5.7 m_perfnames 6	661 662 662 662 662
13.67.5.1 m_peer_perfedit 6 13.67.5.2 m_table 6 13.67.5.3 m_vadjust 6 13.67.5.4 m_hadjust 6 13.67.5.5 m_vscroll 6 13.67.5.6 m_hscroll 6 13.67.5.7 m_perfnames 6 13.67.5.8 m_perfroll 6	661 662 662 662 662 662
13.67.5.1 m_peer_perfedit 6 13.67.5.2 m_table 6 13.67.5.3 m_vadjust 6 13.67.5.4 m_hadjust 6 13.67.5.5 m_vscroll 6 13.67.5.6 m_hscroll 6 13.67.5.7 m_perfnames 6 13.67.5.8 m_perfroll 6 13.67.5.9 m_perftime 6	661 662 662 662 662 662
13.67.5.1 m_peer_perfedit 6 13.67.5.2 m_table 6 13.67.5.3 m_vadjust 6 13.67.5.4 m_hadjust 6 13.67.5.5 m_vscroll 6 13.67.5.6 m_hscroll 6 13.67.5.7 m_perfnames 6 13.67.5.8 m_perfroll 6 13.67.5.9 m_perftime 6 13.67.5.10m_menu_snap 6	661 662 662 662 662 662 662

IXXXVI

13.67.5.12m_button_xpose	33
13.67.5.13m_entry_xpose	33
13.67.5.14m_image_play	33
13.67.5.15m_button_snap	33
13.67.5.16m_entry_snap	33
13.67.5.17m_button_stop	33
13.67.5.18m_button_play	34
13.67.5.19m_button_loop	34
13.67.5.20m_button_expand	34
13.67.5.21m_button_collapse	34
13.67.5.22m_button_copy	34
13.67.5.23m_button_grow	34
13.67.5.24m_button_undo	34
13.67.5.25m_button_redo	34
13.67.5.26m_button_jack	35
13.67.5.27m_button_follow	35
13.67.5.28m_button_bpm	35
13.67.5.29m_entry_bpm	35
13.67.5.30m_button_bw	35
13.67.5.31m_entry_bw	35
13.67.5.32m_hbox	35
13.67.5.33m_hlbox	35
13.67.5.34m_tooltips	36
13.67.5.35m_menu_bpm	36
13.67.5.36m_menu_bw	36
13.67.5.37m_snap	36
13.67.5.38m_bpm	36
13.67.5.39m_bw	36
13.67.5.40m_ppqn	36
13.67.5.41m_is_running	36

13.67.5.42m_standard_bpm	67
13.68seq64::perfnames Class Reference	67
13.68.1 Detailed Description	69
13.68.2 Constructor & Destructor Documentation	69
13.68.2.1 perfnames()	69
13.68.2.2 ~perfnames()	70
13.68.3 Member Function Documentation	70
13.68.3.1 redraw_dirty_sequences()	70
13.68.3.2 enqueue_draw()	70
13.68.3.3 convert_y()	70
13.68.3.4 draw_sequences()	70
13.68.3.5 draw_sequence()	71
13.68.3.6 change_vert()	71
13.68.3.7 redraw()	71
13.68.3.8 on_realize()	71
13.68.3.9 on_expose_event()	72
13.68.3.10on_button_press_event()	72
13.68.3.11on_button_release_event()	72
13.68.3.12on_size_allocate()	73
13.68.3.13on_scroll_event()	73
13.68.4 Friends And Related Function Documentation	73
13.68.4.1 perfedit	73
13.68.5 Field Documentation	73
13.68.5.1 m_parent	74
13.68.5.2 m_names_chars	74
13.68.5.3 m_char_w	74
13.68.5.4 m_setbox_w	74
13.68.5.5 m_namebox_w	74
13.68.5.6 m_names_x	74
13.68.5.7 m_names_y	74

Ixxxviii CONTENTS

13.68.5.8 m_xy_offset	′5
13.68.5.9 m_seqs_in_set	' 5
13.68.5.10m_sequence_max	'5
13.68.5.11m_sequence_offset	'5
13.68.5.12m_sequence_active	' 5
13.69seq64::perform Class Reference	'5
13.69.1 Detailed Description	38
13.69.2 Member Enumeration Documentation	39
13.69.2.1 mute_op_t	39
13.69.2.2 ff_rw_button_t	90
13.69.3 Constructor & Destructor Documentation	90
13.69.3.1 perform()	90
13.69.3.2 ~perform())1
13.69.4 Member Function Documentation) 1
13.69.4.1 is_modified() [1/2])1
13.69.4.2 modify()) 1
13.69.4.3 ppqn())1
13.69.4.4 sequence_count()) 1
13.69.4.5 sequence_max())1
13.69.4.6 is_control_status())1
13.69.4.7 set_edit_sequence())1
13.69.4.8 unset_edit_sequence())2
13.69.4.9 is_edit_sequence())2
13.69.4.10get_beats_per_bar())2
13.69.4.11set_beats_per_bar())2
13.69.4.12get_beat_width())3
13.69.4.13set_beat_width())3
13.69.4.14clocks_per_metronome() [1/2])3
13.69.4.15clocks_per_metronome() [2/2])3
13.69.4.16set_32nds_per_quarter())3

13.69.4.17get_32nds_per_quarter()	}3
13.69.4.18us_per_quarter_note() [1/2])4
13.69.4.19us_per_quarter_note() [2/2]) 4
13.69.4.20gui() [1/2])4
13.69.4.21gui() [2/2])4
13.69.4.22keys() [1/2]) 4
13.69.4.23keys() [2/2]) 4
13.69.4.24master_bus())4
13.69.4.25 ilter_by_channel()) 5
13.69.4.26s_running()) 5
13.69.4.27s_pattern_playing() [1/2]) 5
13.69.4.28toggle_song_start_mode()) 5
13.69.4.29song_start_mode() [1/2]) 5
13.69.4.30song_start_mode() [2/2]) 5
13.69.4.31is_jack_running()) 5
13.69.4.32s_jack_master()) 5
13.69.4.33enregister()) 5
13.69.4.34toggle_jack_mode()	96
13.69.4.35set_jack_mode()	96
13.69.4.36get_toggle_jack()	96
13.69.4.37set_jack_stop_tick()) 6
13.69.4.3&ombine_bytes())7
13.69.4.39FF_rewind()) 7
13.69.4.40FF_RW_timeout())7
13.69.4.41start_from_perfedit() [1/2])7
13.69.4.42start_from_perfedit() [2/2]	98
13.69.4.43set_follow_transport()	98
13.69.4.44get_follow_transport()	98
13.69.4.45toggle_follow_transport()	98
13.69.4.46set_reposition()	98

xc CONTENTS

13.69.4.47f_rw_type() [1/2]
13.69.4.48f_rw_type() [2/2]
13.69.4.49rewind()
13.69.4.50fast_forward()
13.69.4.51reposition()
13.69.4.52clear_all()
13.69.4.53aunch()
13.69.4.54new_sequence()
13.69.4.55add_sequence()
13.69.4.5@delete_sequence()
13.69.4.57s_sequence_in_edit()
13.69.4.5&lear_sequence_triggers()
13.69.4.59print_triggers()
13.69.4.60finish()
13.69.4.61get_tick()
13.69.4.62set_tick()
13.69.4.63get_jack_tick()
13.69.4.64set_jack_tick()
13.69.4.65set_left_tick()
13.69.4.66get_left_tick()
13.69.4.67set_start_tick()
13.69.4.6&get_start_tick()
13.69.4.69set_right_tick()
13.69.4.70get_right_tick()
13.69.4.71left_right_size()
13.69.4.72s_active()
13.69.4.73apply_song_transpose()
13.69.4.74set_transpose()
13.69.4.75get_transpose()
13.69.4.76get_beats_per_minute()

CONTENTS xci

13.69.4.77set_sequence_control_status()
13.69.4.7&unset_sequence_control_status()
13.69.4.79sequence_playing_toggle()
13.69.4.80sequence_playing_change()
13.69.4.81sequence_playing_on()
13.69.4.82sequence_playing_off()
13.69.4.83mute_all_tracks()
13.69.4.84toggle_all_tracks()
13.69.4.85armed_saved()
13.69.4.8@toggle_playing_tracks()
13.69.4.87mute_screenset()
13.69.4.8&utput_func()
13.69.4.89nput_func()
13.69.4.90set_group_mute_state()
13.69.4.91get_group_mute_state()
13.69.4.92set_offset()
13.69.4.93get_offset()
13.69.4.94save_playing_state()
13.69.4.95restore_playing_state()
13.69.4.96key_name()
13.69.4.97get_key_events()
13.69.4.9&get_key_groups()
13.69.4.99get_key_events_rev()
13.69.4.10@jet_key_groups_rev()
13.69.4.10dhow_ui_sequence_key() [1/2]
13.69.4.102how_ui_sequence_key() [2/2]
13.69.4.108how_ui_sequence_number() [1/2]
13.69.4.10show_ui_sequence_number() [2/2]
13.69.4.1 05 okup_keyevent_key()
13.69.4.1 06 okup_keyevent_seq()

xcii CONTENTS

13.69.4.10\(\overline{0}\)okup_keygroup_key()
13.69.4.108okup_keygroup_group()
13.69.4.10start_playing()
13.69.4.11 Nause_playing()
13.69.4.11stop_playing()
13.69.4.11s2art_key()
13.69.4.11pause_key()
13.69.4.11stop_key()
13.69.4.11 5 arn_toggle()
13.69.4.11decrement_beats_per_minute()
13.69.4.11in crement_beats_per_minute()
13.69.4.11 Mage_decrement_beats_per_minute()
13.69.4.11 Spage_increment_beats_per_minute()
13.69.4.12decrement_screenset()
13.69.4.12\(\text{mcrement_screenset}()\)
13.69.4.12/2ighlight()
13.69.4.128_smf_0()
13.69.4.12glet_sequence() [1/2]
13.69.4.125et_sequence() [2/2]
13.69.4.12 Sequence_key()
13.69.4.125 equence_label()
13.69.4.128et_input_bus()
13.69.4.129et_clock_bus()
13.69.4.13@nainwnd_key_event()
13.69.4.13pderfroll_key_event()
13.69.4.13p2layback_key_event()
13.69.4.1300 ove_triggers()
13.69.4.13stopy_triggers()
13.69.4.13 5 ush_trigger_undo()
13.69.4.1360p_trigger_undo()

CONTENTS xciii

13.69.4.13p7op_trigger_redo()
13.69.4.138_dirty_main()
13.69.4.139_dirty_edit()
13.69.4.146_dirty_perf()
13.69.4.1 4\$ _dirty_names()
13.69.4.148_exportable()
13.69.4.14 Set_screenset()
13.69.4.14glet_screenset()
13.69.4.145et_playing_screenset()
13.69.4.1466ggle_other_seqs()
13.69.4.14toggle_other_names()
13.69.4.1416ave_undo()
13.69.4.149et_have_undo()
13.69.4.150ave_redo()
13.69.4.15det_have_redo()
13.69.4.154plit_trigger()
13.69.4.15@et_max_trigger()
13.69.4.15\(\text{sollapse}()\)
13.69.4.15 £ opy()
13.69.4.15@xpand()
13.69.4.1577idi_control_toggle()
13.69.4.158bidi_control_on()
13.69.4.159idi_control_off()
13.69.4.160nidi_control_event()
13.69.4.161andle_midi_control()
13.69.4.1622andle_midi_control_ex()
13.69.4.16@et_screen_set_notepad()
13.69.4.16durrent_screen_set_notepad()
13.69.4.165et_screen_set_notepad() [1/2]
13.69.4.166et screen set notepad() [2/2]

xciv CONTENTS

13.69.4.16 s et_playing_screenset()
13.69.4.16abny_group_unmutes()
13.69.4.1699ute_group_tracks()
13.69.4.17 Select_and_mute_group()
13.69.4.17slet_song_mute()
13.69.4.17%et_mode_group_mute()
13.69.4.17@nset_mode_group_mute()
13.69.4.17 st elect_group_mute()
13.69.4.175et_mode_group_learn()
13.69.4.17@inset_mode_group_learn()
13.69.4.17ថ្ងៃ_group_learning()
13.69.4.17 % et_and_copy_mute_group()
13.69.4.17activate()
13.69.4.18start()
13.69.4.18stop()
13.69.4.18 Paart_jack()
13.69.4.18 atop_jack()
13.69.4.18 dosition_jack()
13.69.4.185ff_sequences()
13.69.4.186I_notes_off()
13.69.4.185et_active()
13.69.4.188et_was_active()
13.69.4.18@set_sequences()
13.69.4.19 ø lay()
13.69.4.19slet_orig_ticks()
13.69.4.19 (a) beats_per_minute()
13.69.4.19&et_looping()
13.69.4.19 max_active_set()
13.69.4.19 ā unch_input_thread()
13.69.4.19@unch_output_thread()

CONTENTS xcv

	13.69.4.197it_jack_transport()	739
	13.69.4.19 Seinit_jack_transport()	739
	13.69.4.199eq_in_playing_screen()	739
	13.69.4.206_modified() [2/2]	739
	13.69.4.20//alid_midi_control_seq()	740
	13.69.4.2082_screenset_valid()	740
	13.69.4.20 s et_running()	740
	13.69.4.20 6_ pattern_playing() [2/2]	741
	13.69.4.20 5 et_playback_mode()	741
	13.69.4.206 ute_group_offset()	741
	13.69.4.20 g _seq_valid()	741
	13.69.4.20 8 _mseq_valid()	742
	13.69.4.20@stall_sequence()	742
	13.69.4.21i@ner_start()	743
	13.69.4.21im/ner_stop()	743
	13.69.4.21@amp_track()	743
	13.69.4.21set_key_event()	744
	13.69.4.21s/et_key_group()	744
	13.69.4.21@freate_master_bus()	744
	13.69.4.21add_clock()	745
	13.69.4.21 s et_clock()	745
	13.69.4.21 a dd_input()	745
	13.69.4.21 s et_input()	745
	13.69.4.22@et_input()	746
	13.69.4.2 2 \$_input_system_port()	746
13.69.5	5 Friends And Related Function Documentation	746
	13.69.5.1 jack_assistant	746
	13.69.5.2 keybindentry	746
	13.69.5.3 mainwnd	746
	13.69.5.4 midifile	746

xcvi CONTENTS

	13.69.5.5 optionsfile	747
	13.69.5.6 options	747
	13.69.5.7 perfedit	747
	13.69.5.8 perfroll	747
	13.69.5.9 input_thread_func	747
	13.69.5.10output_thread_func	747
	13.69.5.11jack_sync_callback	748
	13.69.5.12ack_transport_callback	748
	13.69.5.13ack_shutdown	749
	13.69.5.14ack_timebase_callback	749
	13.69.5.15get_current_jack_position	750
13.69.6	Field Documentation	750
	13.69.6.1 sm_mc_dummy	750
	13.69.6.2 m_song_start_mode	750
	13.69.6.3 m_start_from_perfedit	751
	13.69.6.4 m_reposition	751
	13.69.6.5 m_excell_FF_RW	751
	13.69.6.6 m_FF_RW_button_type	751
	13.69.6.7 m_mute_group	751
	13.69.6.8 m_armed_saved	751
	13.69.6.9 m_armed_statuses	751
	13.69.6.10m_tracks_mute_state	752
	13.69.6.11m_mode_group	752
	13.69.6.12m_mode_group_learn	752
	13.69.6.13m_mute_group_selected	752
	13.69.6.14m_playing_screen	752
	13.69.6.15m_playscreen_offset	752
	13.69.6.16m_seqs	752
	13.69.6.17m_seqs_active	753
	13.69.6.18m_was_active_main	753

CONTENTS xcvii

13.69.6.19m_was_active_edit	'53
13.69.6.20m_was_active_perf	'53
13.69.6.21m_was_active_names	'53
13.69.6.22m_sequence_state	'53
13.69.6.23m_transpose	'53
13.69.6.24m_out_thread	'54
13.69.6.25m_in_thread	'54
13.69.6.26m_out_thread_launched	'54
13.69.6.27m_in_thread_launched	'54
13.69.6.28m_running	'54
13.69.6.29m_is_pattern_playing	'54
13.69.6.30m_inputing	'54
13.69.6.31m_outputing	'55
13.69.6.32m_looping	'55
13.69.6.33m_playback_mode	'55
13.69.6.34m_ppqn	'55
13.69.6.35m_bpm	'55
13.69.6.36m_beats_per_bar	'55
13.69.6.37m_beat_width	'55
13.69.6.38m_clocks_per_metronome	'56
13.69.6.39m_32nds_per_quarter	'56
13.69.6.40m_us_per_quarter_note	'56
13.69.6.41m_master_bus	'56
13.69.6.42m_master_clocks	'56
13.69.6.43m_master_inputs	'56
13.69.6.44m_one_measure	'56
13.69.6.45m_left_tick	'57
13.69.6.46m_right_tick	'57
13.69.6.47m_starting_tick	'57
13.69.6.48m_tick	'57

xcviii CONTENTS

13.69.6.49m_jack_tick	757
13.69.6.50m_usemidiclock	757
13.69.6.51m_midiclockrunning	757
13.69.6.52m_midiclocktick	758
13.69.6.53m_midiclockpos	758
13.69.6.54m_dont_reset_ticks	758
13.69.6.55m_screen_set_notepad	758
13.69.6.56m_midi_cc_toggle	758
13.69.6.57m_midi_cc_on	758
13.69.6.58m_midi_cc_off	758
13.69.6.59m_offset	758
13.69.6.60m_control_status	759
13.69.6.61m_screenset	759
13.69.6.62m_seqs_in_set	759
13.69.6.63m_max_sets	759
13.69.6.64m_sequence_count	759
13.69.6.65m_sequence_max	759
13.69.6.66m_sequence_high	759
13.69.6.67m_edit_sequence	760
13.69.6.68m_is_modified	760
13.69.6.69m_condition_var	760
13.69.6.70m_jack_asst	760
13.69.6.71m_have_undo	760
13.69.6.72m_undo_vect	760
13.69.6.73m_have_redo	760
13.69.6.74m_redo_vect	761
13.69.6.75m_notify	761
13.69.6.76m_gui_support	761
13.70seq64::performcallback Struct Reference	761
13.70.1 Detailed Description	762

CONTENTS xcix

13.70.2 Member Function Documentation	3
13.70.2.1 on_grouplearnchange()	3
13.71 seq64::perfroll Class Reference	3
13.71.1 Constructor & Destructor Documentation	8
13.71.1.1 perfroll()	38
13.71.1.2 ~perfroll()	8
13.71.2 Member Function Documentation	8
13.71.2.1 set_guides()	38
13.71.2.2 update_sizes()	39
13.71.2.3 init_before_show()	39
13.71.2.4 fill_background_pixmap()	39
13.71.2.5 increment_size()	7 0
13.71.2.6 draw_all()	⁷ 0
13.71.2.7 follow_progress()	⁷ 0
13.71.2.8 redraw_progress()	⁷ 0
13.71.2.9 draw_progress()	⁷ 0
13.71.2.10redraw_dirty_sequences()	⁷ 0
13.71.2.11set_ppqn()	⁷ 1
13.71.2.12convert_xy()	7 1
13.71.2.13convert_x()	7 1
13.71.2.14snap_x()	⁷ 2
13.71.2.15draw_sequence_on()	'2
13.71.2.16draw_background_on()	'2
13.71.2.17draw_drawable_row()	'2
13.71.2.1&hange_horz()	'2
13.71.2.19change_vert()	73
13.71.2.20split_trigger()	73
13.71.2.21enqueue_draw()	′3
13.71.2.2&et_zoom()	7 3
13.71.2.23convert_drop_xy()	7 3

73
74
74
74
75
75
75
76
76
76
76
76
77
77
77
77
77
78
78
78
78
78
78
78
78
79
79
79
79
79

13.71.4.11m_names_y	779
13.71.4.12m_background_x	779
13.71.4.13m_size_box_w	⁷ 80
13.71.4.14m_measure_length	⁷ 80
13.71.4.15m_beat_length	⁷ 80
13.71.4.16m_old_progress_ticks	⁷ 80
13.71.4.17m_have_button_press	⁷ 80
13.71.4.18m_transport_follow	⁷ 80
13.71.4.19m_trans_button_press	⁷ 80
13.71.4.20m_4bar_offset	⁷ 81
13.71.4.21m_sequence_offset	⁷ 81
13.71.4.22m_roll_length_ticks	⁷ 81
13.71.4.23m_drop_tick	⁷ 81
13.71.4.24m_drop_tick_trigger_offset	⁷ 81
13.71.4.25m_drop_sequence	⁷ 81
13.71.4.26m_sequence_max	⁷ 81
13.71.4.27m_sequence_active	782
13.71.4.28m_fruity_interaction	782
13.71.4.29m_seq24_interaction	782
13.71.4.30m_interaction	782
13.71.4.31m_moving	782
13.71.4.32m_growing	782
13.71.4.33m_grow_direction	782
13.72seq64::perftime Class Reference	783
13.72.1 Constructor & Destructor Documentation	⁷ 85
13.72.1.1 perftime()	⁷ 85
13.72.1.2 ~perftime()	⁷ 86
13.72.2 Member Function Documentation	⁷ 86
13.72.2.1 reset()	⁷ 86
13.72.2.2 set_scale()	⁷ 86

13.72.2.3 set_guides()	786
13.72.2.4 increment_size()	787
13.72.2.5 enqueue_draw()	787
13.72.2.6 set_zoom()	787
13.72.2.7 draw_background()	787
13.72.2.8 draw_progress_on_window()	787
13.72.2.9 change_horz()	787
13.72.2.10set_ppqn()	788
13.72.2.11tick_to_pixel()	788
13.72.2.12pixel_to_tick()	788
13.72.2.13tick_offset()	789
13.72.2.14update_sizes()	789
13.72.2.15dle_progress()	789
13.72.2.1θupdate_pixmap()	789
13.72.2.17draw_pixmap_on_window()	789
13.72.2.18on_realize()	789
13.72.2.19on_expose_event()	789
13.72.2.20on_button_press_event()	790
13.72.2.21on_size_allocate()	790
13.72.2.22on_button_release_event()	790
13.72.2.23key_press_event()	791
13.72.3 Friends And Related Function Documentation	791
13.72.3.1 perfedit	791
13.72.4 Field Documentation	791
13.72.4.1 m_parent	791
13.72.4.2 m_4bar_offset	791
13.72.4.3 m_tick_offset	791
13.72.4.4 m_ppqn	792
13.72.4.5 m_snap	792
13.72.4.6 m_measure_length	792

13.72.4.7 m_left_marker_tick	 792
13.72.4.8 m_right_marker_tick	 792
13.72.4.9 m_perf_scale_x	 792
13.72.4.10m_timearea_y	 792
13.73seq64::midi_port_info::port_info_t Struct Reference	 792
13.73.1 Detailed Description	 793
13.73.2 Field Documentation	 793
13.73.2.1 m_client_number	 793
13.73.2.2 m_client_name	 793
13.73.2.3 m_port_number	 793
13.73.2.4 m_port_name	 794
13.73.2.5 m_queue_number	 794
13.73.2.6 m_is_input	 794
13.73.2.7 m_is_virtual	 794
13.73.2.8 m_is_system	 794
13.74seq64::rc_settings Class Reference	 794
13.74.1 Detailed Description	 799
13.74.2 Constructor & Destructor Documentation	 799
13.74.2.1 rc_settings() [1/2]	 799
13.74.2.2 rc_settings() [2/2]	 799
13.74.3 Member Function Documentation	 800
13.74.3.1 operator=()	 800
13.74.3.2 config_filespec()	 800
13.74.3.3 user_filespec()	 800
13.74.3.4 set_defaults()	 801
13.74.3.5 auto_option_save() [1/2]	 801
13.74.3.6 legacy_format() [1/2]	 801
13.74.3.7 lash_support() [1/2]	 801
13.74.3.8 allow_mod4_mode() [1/2]	 801
13.74.3.9 allow_snap_split() [1/2]	 801

13.74.3.10allow_click_edit() [1/2]
13.74.3.11show_midi() [1/2]
13.74.3.12priority() [1/2]
13.74.3.13stats() [1/2]
13.74.3.14pass_sysex() [1/2]
13.74.3.15with_jack_transport() [1/2]
13.74.3.16with_jack_master() [1/2]
13.74.3.17with_jack_master_cond() [1/2]
13.74.3.18with_jack_midi() [1/2]
13.74.3.19with_jack()
13.74.3.20filter_by_channel() [1/2]
13.74.3.21manual_alsa_ports() [1/2]
13.74.3.22reveal_alsa_ports() [1/2]
13.74.3.23print_keys() [1/2]
13.74.3.24device_ignore() [1/2]
13.74.3.25device_ignore_num() [1/2]
13.74.3.26nteraction_method() [1/2]
13.74.3.27ilename() [1/2]
13.74.3.28ack_session_uuid() [1/2]
13.74.3.29ast_used_dir() [1/2]
13.74.3.30config_directory() [1/2]
13.74.3.31config_filename() [1/2]
13.74.3.32user_filename() [1/2] 804
13.74.3.33config_filename_alt() [1/2]
13.74.3.34user_filename_alt() [1/2]
13.74.3.35application_name()
13.74.3.36app_client_name()
13.74.3.37auto_option_save() [2/2]
13.74.3.38egacy_format() [2/2]
13.74.3.39ash_support() [2/2]

	13./4.3.4@llow_mod4_mode() [2/2]	05
	13.74.3.41allow_snap_split() [2/2]	05
	13.74.3.42allow_click_edit() [2/2]	05
	13.74.3.43show_midi() [2/2]	06
	13.74.3.44priority() [2/2]	06
	13.74.3.45stats() [2/2]	06
	13.74.3.4\text{\text{\text{pass_sysex()}}} [2/2] 8	06
	13.74.3.47with_jack_transport() [2/2]	06
	13.74.3.48with_jack_master() [2/2]	06
	13.74.3.49with_jack_master_cond() [2/2]	06
	13.74.3.50with_jack_midi() [2/2]	07
	13.74.3.51filter_by_channel() [2/2]	07
	13.74.3.52manual_alsa_ports() [2/2]	07
	13.74.3.53reveal_alsa_ports() [2/2]	07
	13.74.3.54print_keys() [2/2]	07
	13.74.3.55device_ignore() [2/2]	07
	13.74.3.56device_ignore_num() [2/2]	07
	13.74.3.57interaction_method() [2/2]	08
	13.74.3.58ilename() [2/2]	808
	13.74.3.59ack_session_uuid() [2/2]	80
	13.74.3.60ast_used_dir() [2/2]	808
	13.74.3.61config_directory() [2/2]	09
	13.74.3.62set_config_files()	09
	13.74.3.63config_filename() [2/2]	09
	13.74.3.64user_filename() [2/2]	09
	13.74.3.65config_filename_alt() [2/2]	10
	13.74.3.6@user_filename_alt() [2/2]	10
	13.74.3.67home_config_directory()	10
13.74.4	Friends And Related Function Documentation	10
	13.74.4.1 optionsfile	11

	13.74.4.2 options	811
	13.74.4.3 mainwnd	811
	13.74.4.4 rtmidi_info	811
	13.74.4.5 parse_command_line_options	811
	13.74.4.6 help_check	812
13.74.5	Field Documentation	812
	13.74.5.1 m_auto_option_save	812
	13.74.5.2 m_legacy_format	812
	13.74.5.3 m_lash_support	812
	13.74.5.4 m_allow_mod4_mode	812
	13.74.5.5 m_allow_snap_split	813
	13.74.5.6 m_allow_click_edit	813
	13.74.5.7 m_show_midi	813
	13.74.5.8 m_priority	813
	13.74.5.9 m_stats	813
	13.74.5.10m_pass_sysex	813
	13.74.5.11m_with_jack_transport	813
	13.74.5.12m_with_jack_master	813
	13.74.5.13m_with_jack_master_cond	814
	13.74.5.14m_with_jack_midi	814
	13.74.5.15m_filter_by_channel	814
	13.74.5.16m_manual_alsa_ports	814
	13.74.5.17m_reveal_alsa_ports	814
	13.74.5.18m_print_keys	814
	13.74.5.19m_device_ignore	814
	13.74.5.20m_device_ignore_num	814
	13.74.5.21m_interaction_method	815
	13.74.5.22m_filename	815
	13.74.5.23m_jack_session_uuid	815
	13.74.5.24m_last_used_dir	815

CONTENTS cvii

13.74.5.25m_config_directory	315
13.74.5.26m_config_filename	315
13.74.5.27m_user_filename	315
13.74.5.28m_config_filename_alt	315
13.74.5.29m_user_filename_alt	316
13.74.5.30m_application_name	316
13.74.5.31m_app_client_name	316
13.75seq64::rect Class Reference	316
13.75.1 Field Documentation	316
13.75.1.1 x	316
13.75.1.2 y	317
13.75.1.3 height	317
13.75.1.4 width	317
13.76seq64::gui_drawingarea_gtk2::rect Struct Reference	317
13.76.1 Field Documentation	317
13.76.1.1 x	317
13.76.1.2 y	317
13.76.1.3 height	318
13.76.1.4 width	318
13.77seq64::rterror Class Reference	318
13.77.1 Detailed Description	319
13.77.2 Member Enumeration Documentation	319
13.77.2.1 Type	319
13.77.3 Constructor & Destructor Documentation	319
13.77.3.1 rterror()	319
13.77.3.2 ~rterror()	319
13.77.4 Member Function Documentation	320
13.77.4.1 print_message()	320
13.77.4.2 getType()	320
13.77.4.3 get_message()	320

cviii CONTENTS

13.77.4.4 what()	20
13.77.5 Field Documentation	20
13.77.5.1 m_message	20
13.77.5.2 m_type	20
13.78seq64::rtmidi Class Reference	!1
13.78.1 Detailed Description	23
13.78.2 Constructor & Destructor Documentation	23
13.78.2.1 rtmidi()	23
13.78.2.2 ∼rtmidi()	23
13.78.3 Member Function Documentation	:3
13.78.3.1 api_connect()	23
13.78.3.2 api_play()	<u>'</u> 4
13.78.3.3 api_continue_from()	<u>'</u> 4
13.78.3.4 api_start()	<u>'</u> 4
13.78.3.5 api_stop()	<u>'</u> 4
13.78.3.6 api_clock()	<u>'</u> 4
13.78.3.7 api_set_ppqn()	<u>'</u> 4
13.78.3.8 api_set_beats_per_minute()	25
13.78.3.9 api_init_out()	:5
13.78.3.10api_init_out_sub()	:5
13.78.3.11api_init_in()	:5
13.78.3.12api_init_in_sub()	25
13.78.3.13api_deinit_in()	25
13.78.3.14api_get_midi_event()	26
13.78.3.15api_poll_for_midi()	26
13.78.3.16api_sysex()	26
13.78.3.17api_flush()	26
13.78.3.18s_port_open()	26
13.78.3.19get_bus_id()	26
13.78.3.20get_bus_name()	27

13.78.3.21get_port_id()	827
13.78.3.22get_port_name()	827
13.78.3.23get_port_count()	827
13.78.3.24full_port_count()	827
13.78.3.25get_api() [1/2]	828
13.78.3.26get_api() [2/2]	828
13.78.3.27set_api()	828
13.78.3.28delete_api()	828
13.78.4 Friends And Related Function Documentation	828
13.78.4.1 midibus	828
13.78.5 Field Documentation	828
13.78.5.1 m_midi_info	828
13.78.5.2 m_midi_api	829
13.79seq64::rtmidi_in Class Reference	829
13.79.1 Detailed Description	831
13.79.2 Constructor & Destructor Documentation	831
13.79.2.1 rtmidi_in()	831
13.79.2.2 ~rtmidi_in()	832
13.79.3 Member Function Documentation	832
13.79.3.1 user_callback()	832
13.79.3.2 cancel_callback()	832
13.79.3.3 openmidi_api()	832
13.80seq64::rtmidi_in_data Class Reference	833
13.80.1 Detailed Description	834
13.80.2 Constructor & Destructor Documentation	834
13.80.2.1 rtmidi_in_data()	834
13.80.3 Member Function Documentation	834
13.80.3.1 queue() [1/2]	834
13.80.3.2 queue() [2/2]	834
13.80.3.3 message() [1/2]	834

13.80.3.4 message() [2/2]	 834
13.80.3.5 ignore_flags() [1/2]	 834
13.80.3.6 test_ignore_flags()	 835
13.80.3.7 ignore_flags() [2/2]	 835
13.80.3.8 do_input() [1/2]	 835
13.80.3.9 do_input() [2/2]	 835
13.80.3.10first_message() [1/2]	 835
13.80.3.11first_message() [2/2]	 835
13.80.3.12continue_sysex() [1/2]	 835
13.80.3.13continue_sysex() [2/2]	 836
13.80.3.14using_callback() [1/2]	 836
13.80.3.15using_callback() [2/2]	 836
13.80.3.16api_data() [1/3]	 836
13.80.3.17api_data() [2/3]	 836
13.80.3.18api_data() [3/3]	 836
13.80.3.19user_data() [1/3]	 836
13.80.3.20user_data() [2/3]	 837
13.80.3.21user_data() [3/3]	 837
13.80.3.22user_callback() [1/2]	 837
13.80.3.23user_callback() [2/2]	 837
13.80.4 Field Documentation	 837
13.80.4.1 m_queue	 837
13.80.4.2 m_message	 837
13.80.4.3 m_ignore_flags	 837
13.80.4.4 m_do_input	 838
13.80.4.5 m_first_message	 838
13.80.4.6 m_api_data	 838
13.80.4.7 m_using_callback	 838
13.80.4.8 m_user_callback	 838
13.80.4.9 m_user_data	 838

13.80.4.10m_continue_sysex	338
13.81seq64::rtmidi_info Class Reference	338
13.81.1 Detailed Description	340
13.81.2 Constructor & Destructor Documentation	341
13.81.2.1 rtmidi_info()	341
13.81.2.2 ~rtmidi_info()	341
13.81.3 Member Function Documentation	341
13.81.3.1 get_version()	341
13.81.3.2 get_compiled_api()	341
13.81.3.3 midi_mode() [1/2] 8	341
13.81.3.4 midi_mode() [2/2] 8	342
13.81.3.5 clear()	342
13.81.3.6 add_input()	342
13.81.3.7 add_output()	342
13.81.3.8 add_bus()	342
13.81.3.9 get_bus_id()	342
13.81.3.10get_bus_name()	343
13.81.3.11get_port_count()	343
13.81.3.12full_port_count()	343
13.81.3.13get_port_id()	343
13.81.3.14get_port_name()	343
13.81.3.15get_input()	344
13.81.3.16get_virtual()	344
13.81.3.17get_system()	344
13.81.3.18get_all_port_info()	344
13.81.3.19queue_number()	344
13.81.3.20app_name()	344
13.81.3.21global_queue()	344
13.81.3.22ppqn()	345
13.81.3.23api_set_ppqn()	345

13.81.3.24opm()	345
13.81.3.25api_set_beats_per_minute()	345
13.81.3.26api_port_start()	345
13.81.3.27api_get_midi_event()	345
13.81.3.28api_flush()	345
13.81.3.29api_poll_for_midi()	346
13.81.3.30port_list()	346
13.81.3.31selected_api() [1/2]	346
13.81.3.32get_api_info() [1/2]	346
13.81.3.33get_api_info() [2/2] 8	346
13.81.3.34api_connect()	346
13.81.3.35selected_api() [2/2]	346
13.81.3.36set_api_info()	347
13.81.3.37delete_api()	347
13.81.3.3&penmidi_api()	347
13.81.4 Friends And Related Function Documentation	347
13.81.4.1 mastermidibus	347
13.81.4.2 midibus	348
13.81.4.3 rtmidi_in	348
13.81.4.4 rtmidi_out	348
13.81.5 Field Documentation	348
13.81.5.1 m_info_api	348
13.81.5.2 sm_selected_api	348
13.82seq64::rtmidi_out Class Reference	349
13.82.1 Detailed Description	350
13.82.2 Constructor & Destructor Documentation	350
13.82.2.1 rtmidi_out()	350
13.82.2.2 ~rtmidi_out()	350
13.82.3 Member Function Documentation	351
13.82.3.1 openmidi_api()	351

CONTENTS cxiii

13.83seq64::Seq24PerfInput Class Reference	351
13.83.1 Constructor & Destructor Documentation	52
13.83.1.1 Seq24PerfInput()	52
13.83.2 Member Function Documentation	52
13.83.2.1 on_button_press_event()	53
13.83.2.2 on_button_release_event()	53
13.83.2.3 on_motion_notify_event()	53
13.83.2.4 activate_adding()	354
13.83.2.5 handle_motion_key()	354
13.83.3 Friends And Related Function Documentation	54
13.83.3.1 perfroll	54
13.83.4 Field Documentation	55
13.83.4.1 m_effective_tick	155
13.84seq64::Seq24SeqEventInput Struct Reference	155
13.84.1 Constructor & Destructor Documentation	55
13.84.1.1 Seq24SeqEventInput()	55
13.84.2 Member Function Documentation	55
13.84.2.1 set_adding()	55
13.84.2.2 on_button_press_event()	356
13.84.2.3 on_button_release_event()	356
13.84.2.4 on_motion_notify_event()	356
13.84.3 Field Documentation	357
13.84.3.1 m_adding	357
13.85seq64::seqdata Class Reference	357
13.85.1 Constructor & Destructor Documentation	360
13.85.1.1 seqdata()	360
13.85.1.2 ~seqdata()	361
13.85.2 Member Function Documentation	361
13.85.2.1 reset()	361
13.85.2.2 redraw()	361

13.85.2.4 set_data_type() 86 13.85.2.5 idle_redraw() 86 13.85.2.6 update_sizes() 86 13.85.2.7 update_pixmap() 86 13.85.2.8 draw_line_on_window() 86 13.85.2.9 xy_to_rect() 86 13.85.2.10draw_events_on() 86	62 62 62 63 64 64
13.85.2.6 update_sizes() 86 13.85.2.7 update_pixmap() 86 13.85.2.8 draw_line_on_window() 86 13.85.2.9 xy_to_rect() 86	62 62 63 64
13.85.2.7 update_pixmap() 86 13.85.2.8 draw_line_on_window() 86 13.85.2.9 xy_to_rect() 86	62 63 64 64
13.85.2.8 draw_line_on_window()	62 63 64
13.85.2.9 xy_to_rect()	63 64 64
	64 64
13.85.2.10draw_events_on()	64
13.85.2.11change_horz()	
13.85.2.12convert_x()	35
13.85.2.13render_number()	35
13.85.2.14draw_events_on_pixmap()	35
13.85.2.15draw_pixmap_on_window()	35
13.85.2.16on_realize()	35
13.85.2.17on_expose_event()	35
13.85.2.18on_button_press_event()	36
13.85.2.19on_button_release_event()	36
13.85.2.20on_motion_notify_event()	37
13.85.2.21on_leave_notify_event()	37
13.85.2.22on_scroll_event()	37
13.85.2.23on_size_allocate()	38
13.85.3 Friends And Related Function Documentation	38
13.85.3.1 Ifownd	38
13.85.3.2 seqevent	38
13.85.3.3 seqroll	38
13.85.4 Field Documentation	38
13.85.4.1 m_seq	38
13.85.4.2 m_zoom	38
13.85.4.3 m_scroll_offset_ticks	39
13.85.4.4 m_scroll_offset_x	39

13.85.4.5 m_number_w	 869
13.85.4.6 m_number_h	 869
13.85.4.7 m_number_offset_y	 869
13.85.4.8 m_status	 869
13.85.4.9 m_cc	 869
13.85.4.10m_numbers	 869
13.85.4.11m_old	 870
13.85.4.12m_drag_handle	 870
13.85.4.13m_dragging	 870
13.86seq64::seqedit Class Reference	 870
13.86.1 Detailed Description	 877
13.86.2 Constructor & Destructor Documentation	 878
13.86.2.1 seqedit()	 878
13.86.2.2 ~seqedit()	 878
13.86.3 Member Function Documentation	 878
13.86.3.1 set_zoom()	 878
13.86.3.2 set_snap()	 879
13.86.3.3 set_note_length()	 879
13.86.3.4 set_beats_per_bar()	 879
13.86.3.5 set_beat_width()	 880
13.86.3.6 set_transpose_image()	 880
13.86.3.7 set_rec_vol()	 880
13.86.3.8 horizontal_adjust()	 881
13.86.3.9 vertical_adjust()	 881
13.86.3.10horizontal_set()	 881
13.86.3.11vertical_set()	 881
13.86.3.12set_measures()	 882
13.86.3.13apply_length()	 882
13.86.3.14get_measures()	 882
13.86.3.15set_midi_channel()	 882

13.86.3.16set_midi_bus()
13.86.3.17set_scale()
13.86.3.18set_chord()
13.86.3.19set_key()
13.86.3.20set_background_sequence()
13.86.3.21transpose_change_callback()
13.86.3.22name_change_callback()
13.86.3.23play_change_callback()
13.86.3.24record_change_callback()
13.86.3.25q_rec_change_callback()
13.86.3.26thru_change_callback()
13.86.3.27undo_callback()
13.86.3.28redo_callback()
13.86.3.29set_data_type()
13.86.3.30update_all_windows()
13.86.3.31fill_top_bar()
13.86.3.32create_menus()
13.86.3.33popup_menu()
13.86.3.34popup_event_menu()
13.86.3.35popup_midibus_menu()
13.86.3.3@popup_sequence_menu()
13.86.3.37popup_tool_menu()
13.86.3.3&popup_midich_menu()
13.86.3.39create_menu_image()
13.86.3.4@imeout()
13.86.3.41do_action()
13.86.3.42mouse_action()
13.86.3.43start_playing()
13.86.3.44stop_playing()
13.86.3.45change_focus()

CONTENTS cxvii

13.86.3.46handle_close()	 888
13.86.3.47on_realize()	 889
13.86.3.48on_set_focus()	 889
13.86.3.49on_focus_in_event()	 889
13.86.3.50on_focus_out_event()	 889
13.86.3.51on_delete_event()	 889
13.86.3.52on_scroll_event()	 890
13.86.3.53on_key_press_event()	 890
13.86.4 Field Documentation	 891
13.86.4.1 seqmenu	 891
13.86.4.2 m_initial_snap	 891
13.86.4.3 m_initial_note_length	 891
13.86.4.4 m_initial_chord	 891
13.86.4.5 m_initial_zoom	 892
13.86.4.6 m_zoom	 892
13.86.4.7 m_snap	 892
13.86.4.8 m_note_length	 892
13.86.4.9 m_scale	 892
13.86.4.10m_chord	 892
13.86.4.11m_key	 892
13.86.4.12m_bgsequence	 892
13.86.4.13m_measures	 893
13.86.4.14m_ppqn	 893
13.86.4.15m_pp_whole	 893
13.86.4.16m_pp_eighth	 893
13.86.4.17m_pp_sixteenth	 893
13.86.4.18m_seq	 893
13.86.4.19m_menubar	 893
13.86.4.20m_menu_tools	 894
13.86.4.21m_menu_zoom	 894

13.86.4.22m_menu_snap
13.86.4.23m_menu_note_length
13.86.4.24m_menu_length
13.86.4.25m_toggle_transpose
13.86.4.26m_image_transpose
13.86.4.27m_menu_midich
13.86.4.28m_menu_midibus
13.86.4.29m_menu_data
13.86.4.30m_menu_key
13.86.4.31m_menu_scale
13.86.4.32m_menu_chords
13.86.4.33m_menu_sequences
13.86.4.34m_menu_bpm
13.86.4.35m_menu_bw
13.86.4.36m_menu_rec_vol
13.86.4.37m_vadjust
13.86.4.38m_hadjust
13.86.4.39m_vscroll_new
13.86.4.40m_hscroll_new
13.86.4.41m_seqkeys_wid
13.86.4.42m_seqtime_wid
13.86.4.43m_seqdata_wid
13.86.4.44m_seqevent_wid
13.86.4.45m_seqroll_wid
13.86.4.46m_button_lfo
13.86.4.47m_lfo_wnd
13.86.4.48m_table
13.86.4.49m_vbox
13.86.4.50m_hbox
13.86.4.51m_hbox2

13.86.4.52m_button_undo
13.86.4.53m_button_redo
13.86.4.54m_button_quantize
13.86.4.55m_button_tools
13.86.4.56m_button_sequence
13.86.4.57m_entry_sequence
13.86.4.58m_button_bus
13.86.4.59m_entry_bus
13.86.4.60m_button_channel
13.86.4.61m_entry_channel
13.86.4.62m_button_snap
13.86.4.63m_entry_snap
13.86.4.64m_button_note_length
13.86.4.65m_entry_note_length
13.86.4.66m_button_zoom
13.86.4.67m_entry_zoom
13.86.4.68m_button_length
13.86.4.69m_entry_length
13.86.4.70m_button_key
13.86.4.71m_entry_key
13.86.4.72m_button_scale
13.86.4.73m_entry_scale
13.86.4.74m_button_chord
13.86.4.75m_entry_chord
13.86.4.76m_tooltips
13.86.4.77m_button_data
13.86.4.78m_entry_data
13.86.4.79m_button_bpm
13.86.4.80m_entry_bpm
13.86.4.81m_button_bw

13.86.4.82m_entry_bw	901
13.86.4.83m_button_rec_vol	901
13.86.4.84m_toggle_play	902
13.86.4.85m_toggle_record	902
13.86.4.86m_toggle_q_rec	902
13.86.4.87m_toggle_thru	902
13.86.4.88m_entry_name	902
13.86.4.89m_editing_status	902
13.86.4.90m_editing_cc	902
13.86.4.91m_have_focus	902
13.87seq64::seqevent Class Reference	903
13.87.1 Constructor & Destructor Documentation	906
13.87.1.1 seqevent()	906
13.87.1.2 ~seqevent()	907
13.87.2 Member Function Documentation	907
13.87.2.1 reset()	907
13.87.2.2 redraw()	907
13.87.2.3 set_zoom()	907
13.87.2.4 set_snap()	907
13.87.2.5 set_data_type()	907
13.87.2.6 update_sizes()	908
13.87.2.7 draw_background()	908
13.87.2.8 draw_events_on_pixmap()	908
13.87.2.9 draw_pixmap_on_window()	908
13.87.2.10draw_selection_on_window()	908
13.87.2.11update_pixmap()	909
13.87.2.12force_draw()	909
13.87.2.13dle_redraw()	909
13.87.2.14x_to_w()	909
13.87.2.15drop_event()	909

13.87.2.16draw_events_on()	110
13.87.2.17start_paste()	10
13.87.2.18change_horz()	10
13.87.2.19convert_x()	10
13.87.2.20convert_t())11
13.87.2.21snap_y())11
13.87.2.22snap_x())11
13.87.2.23on_realize()	12
13.87.2.24on_expose_event()	12
13.87.2.25on_button_press_event()	12
13.87.2.26on_button_release_event()	12
13.87.2.27on_motion_notify_event()	13
13.87.2.28on_focus_in_event()	13
13.87.2.29on_focus_out_event()	13
13.87.2.30on_key_press_event())14
13.87.2.31on_size_allocate())14
13.87.3 Friends And Related Function Documentation)14
13.87.3.1 FruitySeqEventInput)14
13.87.3.2 Seq24SeqEventInput)14
13.87.4 Field Documentation	15
13.87.4.1 m_fruity_interaction	15
13.87.4.2 m_seq24_interaction	15
13.87.4.3 m_seq	15
13.87.4.4 m_zoom	15
13.87.4.5 m_snap	15
13.87.4.6 m_ppqn	15
13.87.4.7 m_old	15
13.87.4.8 m_selected	16
13.87.4.9 m_scroll_offset_ticks	16
13.87.4.10m_scroll_offset_x	16

13.87.4.11m_seqdata_wid	116
13.87.4.12m_selecting	116
13.87.4.13m_moving_init	116
13.87.4.14m_moving	116
13.87.4.15m_growing	116
13.87.4.16m_painting	17
13.87.4.17m_paste	17
13.87.4.18m_move_snap_offset_x	117
13.87.4.19m_status	117
13.87.4.20m_cc	17
13.88seq64::seqkeys Class Reference	17
13.88.1 Detailed Description	20
13.88.2 Constructor & Destructor Documentation	20
13.88.2.1 seqkeys()	20
13.88.2.2 ~seqkeys()	21
13.88.3 Member Function Documentation	21
13.88.3.1 set_scale()	21
13.88.3.2 set_key()	21
13.88.3.3 set_hint_key()	21
13.88.3.4 set_hint_state()	122
13.88.3.5 force_draw()	122
13.88.3.6 set_listen_button_press()	122
13.88.3.7 set_listen_button_release()	122
13.88.3.8 set_listen_motion_notify()	23
13.88.3.9 draw_area()	23
13.88.3.1@pdate_pixmap()	23
13.88.3.11convert_y()	23
13.88.3.12draw_key()	123
13.88.3.13change_vert()	24
13.88.3.14update_sizes()	124

CONTENTS cxxiii

	13.88.3.15reset()	924
	13.88.3.16s_black_key()	924
	13.88.3.17on_realize()	925
	13.88.3.18on_expose_event()	925
	13.88.3.19on_button_press_event()	925
	13.88.3.20on_button_release_event()	925
	13.88.3.21on_motion_notify_event()	926
	13.88.3.22on_enter_notify_event()	926
	13.88.3.23on_leave_notify_event()	926
	13.88.3.24on_scroll_event()	926
	13.88.3.25on_size_allocate()	927
13.88.4	Friends And Related Function Documentation	927
	13.88.4.1 seqroll	927
	13.88.4.2 FruitySeqRollInput	927
13.88.5	Field Documentation	927
	13.88.5.1 m_seq	927
	13.88.5.2 m_scroll_offset_key	928
	13.88.5.3 m_scroll_offset_y	928
	13.88.5.4 m_hint_state	928
	13.88.5.5 m_hint_key	928
	13.88.5.6 m_keying	928
	13.88.5.7 m_keying_note	928
	13.88.5.8 m_scale	928
	13.88.5.9 m_key	928
	13.88.5.10m_show_octave_letters	929
13.89seq64::s	seqmenu Class Reference	929
13.89.1	Detailed Description	933
13.89.2	Member Typedef Documentation	933
	13.89.2.1 SeqeditMap	933
	13.89.2.2 SeqeditPair	933

13.89.2.3 iterator	933
13.89.2.4 const_iterator	933
13.89.3 Constructor & Destructor Documentation	933
13.89.3.1 seqmenu()	933
13.89.3.2 ∼seqmenu()	934
13.89.4 Member Function Documentation	934
13.89.4.1 current_seq() [1/2]	934
13.89.4.2 is_modified() [1/2]	934
13.89.4.3 current_seq() [2/2]	934
13.89.4.4 set_edit_sequence()	934
13.89.4.5 unset_edit_sequence()	935
13.89.4.6 is_edit_sequence()	935
13.89.4.7 is_modified() [2/2]	935
13.89.4.8 get_current_sequence()	935
13.89.4.9 get_sequence()	935
13.89.4.10s_current_seq_active()	935
13.89.4.11is_current_seq_in_edit()	935
13.89.4.12new_current_sequence()	936
13.89.4.13new_sequence()	936
13.89.4.14delete_current_sequence()	936
13.89.4.15toggle_current_sequence()	936
13.89.4.1@opup_menu()	936
13.89.4.17seq_edit()	936
13.89.4.18seq_event_edit()	937
13.89.4.19create_seqedit()	937
13.89.4.20remove_seqedit()	937
13.89.4.21seq_set_and_edit()	937
13.89.4.22seq_set_and_eventedit()	938
13.89.4.23redraw()	938
13.89.4.24seq_new()	938

1	13.89.4.25seq_copy()	 938
1	13.89.4.26seq_cut()	 939
1	13.89.4.27seq_paste()	 939
1	13.89.4.2&eq_clear_perf()	 939
1	13.89.4.29set_bus_and_midi_channel()	 939
1	13.89.4.30set_transposable()	 939
1	13.89.4.31mute_all_tracks()	 940
1	13.89.4.32unmute_all_tracks()	 940
1	13.89.4.33oggle_all_tracks()	 940
1	13.89.4.34on_realize()	 940
13.89.5 F	Friends And Related Function Documentation	 940
1	13.89.5.1 mainwnd	 940
1	13.89.5.2 seqedit	 940
13.89.6 F	Field Documentation	 941
1	13.89.6.1 sm_seqedit_list	 941
1	13.89.6.2 m_menu	 941
1	13.89.6.3 m_mainperf	 941
1	13.89.6.4 m_clipboard	 941
1	13.89.6.5 m_seqedit	 941
1	13.89.6.6 m_eventedit	 941
1	13.89.6.7 m_current_seq	 942
1	13.89.6.8 m_modified	 942
13.90seq64::se	eqroll Class Reference	 942
13.90.1 (Constructor & Destructor Documentation	 948
1	13.90.1.1 seqroll()	 948
1	13.90.1.2 ~seqroll()	 949
13.90.2 N	Member Function Documentation	 949
1	13.90.2.1 set_snap()	 949
1	13.90.2.2 set_zoom()	 949
1	13.90.2.3 set_note_length()	 950

13.90.2.4 note_off_length()
13.90.2.5 add_note()
13.90.2.6 add_chord()
13.90.2.7 set_key()
13.90.2.8 set_scale()
13.90.2.9 set_chord()
13.90.2.10set_data_type()
13.90.2.11set_background_sequence()
13.90.2.12update_pixmap()
13.90.2.13update_sizes()
13.90.2.14update_background()
13.90.2.15draw_background_on_pixmap()
13.90.2.16draw_events_on_pixmap()
13.90.2.17draw_selection_on_window()
13.90.2.1&draw_progress_on_window()
13.90.2.19reset()
13.90.2.20update_and_draw()
13.90.2.21redraw()
13.90.2.22redraw_events()
13.90.2.23start_paste()
13.90.2.24complete_paste() [1/2] 954
13.90.2.25complete_paste() [2/2] 954
13.90.2.26follow_progress()
13.90.2.27force_draw()
13.90.2.28horizontal_adjust()
13.90.2.29vertical_adjust()
13.90.2.30snap_y()
13.90.2.31snap_x()
13.90.2.32convert_xy()
13.90.2.3&onvert_tn()

13.90.2.34xy_to_rect()
13.90.2.35convert_tn_box_to_rect()
13.90.2.3@convert_sel_box_to_rect()
13.90.2.37get_selected_box()
13.90.2.3&draw_events_on()
13.90.2.39dle_redraw()
13.90.2.40dle_progress()
13.90.2.41change_horz()
13.90.2.42change_vert()
13.90.2.43move_selection_box()
13.90.2.44move_selected_notes()
13.90.2.45grow_selected_notes()
13.90.2.46set_adding()
13.90.2.47update_mouse_pointer()
13.90.2.48button_press_initial()
13.90.2.49align_selection()
13.90.2.5@button_press()
13.90.2.51button_release()
13.90.2.52motion_notify()
13.90.2.5&lear_selected()
13.90.2.54clear_old()
13.90.2.55clear_flags()
13.90.2.56scroll_offset_x()
13.90.2.57scroll_offset_y()
13.90.2.58set_current_offset_x_y()
13.90.2.59adding()
13.90.2.60selecting()
13.90.2.61growing()
13.90.2.62normal_action()
13.90.2.63select_action()

13.90.2.64drop_action()
13.90.2.65moving()
13.90.2.66on_realize()
13.90.2.67on_expose_event()
13.90.2.68on_button_press_event()
13.90.2.69on_button_release_event()
13.90.2.70on_motion_notify_event()
13.90.2.71on_focus_in_event()
13.90.2.72on_focus_out_event()
13.90.2.73on_key_press_event()
13.90.2.74on_scroll_event()
13.90.2.75on_size_allocate()
13.90.2.76on_leave_notify_event()
13.90.2.77on_enter_notify_event()
13.90.3 Friends And Related Function Documentation
13.90.3.1 FruitySeqRollInput
13.90.4 Field Documentation
13.90.4.1 m_horizontal_adjust
13.90.4.2 m_vertical_adjust
13.90.4.3 m_old
13.90.4.4 m_selected
13.90.4.5 m_seq
13.90.4.6 m_seqkeys_wid
13.90.4.7 m_fruity_interaction
13.90.4.8 m_pos
13.90.4.9 m_zoom
13.90.4.10m_snap
13.90.4.11m_ppqn
13.90.4.12m_note_length
13.90.4.13m_scale

	13.90.4.14m_chord	972
	13.90.4.15m_key	972
	13.90.4.16m_adding	972
	13.90.4.17m_selecting	972
	13.90.4.18m_moving	972
	13.90.4.19m_moving_init	972
	13.90.4.20m_growing	972
	13.90.4.21m_painting	973
	13.90.4.22m_paste	973
	13.90.4.23m_is_drag_pasting	973
	13.90.4.24m_is_drag_pasting_start	973
	13.90.4.25m_justselected_one	973
	13.90.4.26m_move_delta_x	973
	13.90.4.27m_move_delta_y	973
	13.90.4.28m_move_snap_offset_x	973
	13.90.4.29m_progress_x	974
	13.90.4.30m_scroll_offset_ticks	974
	13.90.4.31m_scroll_offset_key	974
	13.90.4.32m_scroll_offset_x	974
	13.90.4.33m_scroll_offset_y	974
	13.90.4.34m_transport_follow	974
	13.90.4.35m_trans_button_press	974
	13.90.4.36m_background_sequence	974
	13.90.4.37m_drawing_background_seq	975
	13.90.4.38m_status	975
	13.90.4.39m_cc	975
13.91 seq64:	seqtime Class Reference	975
13.91.1	Constructor & Destructor Documentation	977
	13.91.1.1 seqtime()	978
	13.91.1.2 ~seqtime()	978

13.91.2 Member Function Documentation	78
13.91.2.1 reset()	78
13.91.2.2 redraw()	78
13.91.2.3 set_zoom()	78
13.91.2.4 draw_pixmap_on_window()	78
13.91.2.5 draw_progress_on_window()	79
13.91.2.6 update_pixmap()	79
13.91.2.7 change_horz()	79
13.91.2.8 update_sizes()	80
13.91.2.9 idle_progress()	80
13.91.2.10on_realize()	80
13.91.2.11on_expose_event()	80
13.91.2.12on_size_allocate()	80
13.91.2.13on_button_press_event()	80
13.91.2.14on_button_release_event()	80
13.91.3 Field Documentation	81
13.91.3.1 m_seq	81
13.91.3.2 m_scroll_offset_ticks	81
13.91.3.3 m_scroll_offset_x	81
13.91.3.4 m_zoom	81
13.91.3.5 m_ppqn	81
13.92seq64::sequence Class Reference	81
13.92.1 Detailed Description	92
13.92.2 Member Typedef Documentation	92
13.92.2.1 EventStack	92
13.92.3 Member Enumeration Documentation	92
13.92.3.1 select_action_e	92
13.92.4 Constructor & Destructor Documentation	93
13.92.4.1 sequence()	93
13.92.4.2 ~sequence()	93

13.92.5 Member Function Documentation	 993
13.92.5.1 operator=()	 993
13.92.5.2 partial_assign()	 993
13.92.5.3 events() [1/2]	 994
13.92.5.4 events() [2/2]	 994
13.92.5.5 any_selected_notes()	 994
13.92.5.6 triggerlist() [1/2]	 994
13.92.5.7 triggerlist() [2/2]	 994
13.92.5.8 get_trigger_count()	 994
13.92.5.9 set_trigger_paste_tick()	 994
13.92.5.10get_trigger_paste_tick()	 995
13.92.5.11number() [1/2]	 995
13.92.5.12number() [2/2]	 995
13.92.5.13modify()	 995
13.92.5.14event_count()	 995
13.92.5.15set_hold_undo()	 995
13.92.5.1&get_hold_undo()	 996
13.92.5.17set_have_undo()	 996
13.92.5.18have_undo()	 996
13.92.5.1%set_have_redo()	 996
13.92.5.20have_redo()	 996
13.92.5.21push_undo()	 996
13.92.5.22pop_undo()	 997
13.92.5.23pop_redo()	 997
13.92.5.24push_trigger_undo()	 997
13.92.5.25pop_trigger_undo()	 997
13.92.5.2&pop_trigger_redo()	 997
13.92.5.27set_name() [1/2]	 997
13.92.5.28set_name() [2/2]	 998
13.92.5.29set_measures()	 998

13.92.5.30get_measures()
13.92.5.31get_ppqn()
13.92.5.32set_beats_per_bar()
13.92.5.33get_beats_per_bar()
13.92.5.34set_beat_width()
13.92.5.35get_beat_width()
13.92.5.36measures_to_ticks()
13.92.5.37clocks_per_metronome() [1/2]
13.92.5.3&locks_per_metronome() [2/2] 999
13.92.5.39set_32nds_per_quarter()
13.92.5.40get_32nds_per_quarter()
13.92.5.41us_per_quarter_note() [1/2]
13.92.5.42us_per_quarter_note() [2/2]
13.92.5.43set_rec_vol()
13.92.5.44set_song_mute()
13.92.5.45toggle_song_mute()
13.92.5.46get_song_mute()
13.92.5.47apply_song_transpose()
13.92.5.48set_transposable()
13.92.5.49get_transposable()
13.92.5.50get_name()
13.92.5.51name()
13.92.5.52set_editing()
13.92.5.53get_editing()
13.92.5.54set_raise()
13.92.5.55get_raise()
13.92.5.56set_length()
13.92.5.57get_length()
13.92.5.5&get_last_tick()
13.92.5.59set_last_tick()

CONTENTS cxxxiii

13.92.5.60mod_last_tick()
13.92.5.61set_playing()
13.92.5.62get_playing()
13.92.5.63toggle_playing()
13.92.5.64toggle_queued()
13.92.5.65off_queued()
13.92.5.66on_queued()
13.92.5.67get_queued()
13.92.5.68get_queued_tick()
13.92.5.69check_queued_tick()
13.92.5.70set_recording()
13.92.5.71get_recording()
13.92.5.72set_snap_tick()
13.92.5.73set_quantized_rec()
13.92.5.74get_quantized_rec()
13.92.5.75set_thru()
13.92.5.76get_thru()
13.92.5.77s_dirty_main()
13.92.5.78s_dirty_edit()
13.92.5.79s_dirty_perf()
13.92.5.80s_dirty_names()
13.92.5.81set_dirty_mp()
13.92.5.82set_dirty()
13.92.5.83get_midi_channel()
13.92.5.84s_smf_0()
13.92.5.85set_midi_channel()
13.92.5.8@print()
13.92.5.87print_triggers()
13.92.5.8&play()
13.92.5.89play_queue()

13.92.5.90add_note()
13.92.5.91add_event() [1/2]
13.92.5.92add_chord()
13.92.5.93add_event() [2/2]
13.92.5.94append_event()
13.92.5.95sort_events()
13.92.5.96add_trigger()
13.92.5.97split_trigger()
13.92.5.9&grow_trigger()
13.92.5.99del_trigger()
13.92.5.10@et_trigger_state()
13.92.5.10stelect_trigger()
13.92.5.10@et_triggers()
13.92.5.10@nselect_triggers()
13.92.5.104 tersect_triggers()
13.92.5.1065tersect_notes()
13.92.5.106tersect_events()
13.92.5.10d/el_selected_trigger()
13.92.5.10&ut_selected_trigger()
13.92.5.109py_selected_trigger()
13.92.5.11plaste_trigger()
13.92.5.11rhove_selected_triggers_to()
13.92.5.11&elected_trigger_start()
13.92.5.11&elected_trigger_end()
13.92.5.11glet_max_trigger()
13.92.5.11move_triggers()
13.92.5.11@copy_triggers()
13.92.5.11@lear_triggers()
13.92.5.11@et_trigger_offset()
13.92.5.119et_midi_bus()

13.92.5.12@et_midi_bus()
13.92.5.12slet_master_midi_bus()
13.92.5.12% elect_note_events()
13.92.5.12%elect_events() [1/3]1020
13.92.5.124elect_events() [2/3]
13.92.5.125elect_events() [3/3]
13.92.5.126elect_event_handle()
13.92.5.12 s elect_linked()
13.92.5.128elect_even_or_odd_notes()
13.92.5.129elect_all_notes()
13.92.5.13@et_num_selected_notes()
13.92.5.13get_num_selected_events()
13.92.5.13 Select_all()
13.92.5.132ppy_selected()
13.92.5.134ut_selected()
13.92.5.13 <u>Saste_selected()</u>
13.92.5.136et_selected_box()
13.92.5.13get_clipboard_box()
13.92.5.13adjust_timestamp()
13.92.5.1369im_timestamp()
13.92.5.14dip_timestamp()
13.92.5.14rhove_selected_notes()
13.92.5.14a2ream_event()
13.92.5.146 hange_event_data_range()
13.92.5.14@nange_event_data_lfo()
13.92.5.1 45 crement_selected()
13.92.5.1466ecrement_selected()
13.92.5.14grow_selected()
13.92.5.14&tretch_selected()
13.92.5.14@move_marked()

13.92.5.15@nark_selected()
13.92.5.15remove_selected()
13.92.5.152npaint_all()
13.92.5.15@nselect()
13.92.5.15/derify_and_link()
13.92.5.156nk_new()
13.92.5.156ero_markers()
13.92.5.15play_note_on()
13.92.5.158day_note_off()
13.92.5.159ff_playing_notes()
13.92.5.16⊤()
13.92.5.1@fause()
13.92.5.162eset_draw_marker()
13.92.5.10@set_draw_trigger_marker()
13.92.5.16glet_next_note_event()
13.92.5.165et_minmax_note_events()
13.92.5.16@et_next_event() [1/2]
13.92.5.16get_next_event() [2/2]
13.92.5.16@et_next_trigger()
13.92.5.169uantize_events()
13.92.5.17@ush_quantize()
13.92.5.17/tanspose_notes()
13.92.5.17\(\frac{1}{2}\) hift_notes()
13.92.5.1769ultiply_pattern()
13.92.5.17\textit{musical_key()} [1/2]
13.92.5.17/5 usical_key() [2/2]
13.92.5.17@usical_scale() [1/2]
13.92.5.17musical_scale() [2/2]
13.92.5.17@ackground_sequence() [1/2]
13.92.5.179ackground_sequence() [2/2]

13.92.5.18£how_events()
13.92.5.18dopy_events()
13.92.5.182bte_off_margin()
13.92.5.18@vent_in_range()
13.92.5.184et_parent()
13.92.5.18 fut_event_on_bus()
13.92.5.186et_trigger_offset()
13.92.5.187djust_trigger_offsets_to_length()
13.92.5.188djust_offset()
13.92.5.18@move() [1/2]1043
13.92.5.19@move() [2/2]1044
13.92.5.19remove_all()
13.92.5.1922hannel_match()
13.92.6 Friends And Related Function Documentation
13.92.6.1 perform
13.92.6.2 triggers
13.92.7 Field Documentation
13.92.7.1 m_events_clipboard
13.92.7.2 m_parent
13.92.7.3 m_events
13.92.7.4 m_triggers
13.92.7.5 m_events_undo_hold
13.92.7.6 m_have_undo
13.92.7.7 m_have_redo
13.92.7.8 m_events_undo
13.92.7.9 m_events_redo
13.92.7.10m_iterator_draw
13.92.7.11m_channel_match
13.92.7.12m_midi_channel
13.92.7.13m_bus

13.92.7.14m_song_mute
13.92.7.15m_transposable
13.92.7.16m_notes_on
13.92.7.17m_masterbus
13.92.7.18m_playing_notes
13.92.7.19m_was_playing
13.92.7.20m_playing
13.92.7.21m_recording
13.92.7.22m_quantized_rec
13.92.7.23m_thru
13.92.7.24m_queued
13.92.7.25m_dirty_main
13.92.7.26m_dirty_edit
13.92.7.27m_dirty_perf
13.92.7.28m_dirty_names
13.92.7.29m_editing
13.92.7.30m_raise
13.92.7.31m_name
13.92.7.32m_last_tick
13.92.7.33m_queued_tick
13.92.7.34m_trigger_offset
13.92.7.35m_maxbeats
13.92.7.36m_ppqn
13.92.7.37m_seq_number
13.92.7.38m_length
13.92.7.39m_snap_tick
13.92.7.40m_time_beats_per_measure
13.92.7.41m_time_beat_width
13.92.7.42m_clocks_per_metronome
13.92.7.43m_32nds_per_quarter

13.92.7.44m_us_per_quarter_note	
13.92.7.45m_rec_vol	
13.92.7.46m_note_on_velocity	
13.92.7.47m_note_off_velocity	
13.92.7.48m_musical_key	
13.92.7.49m_musical_scale	
13.92.7.50m_background_sequence	
13.92.7.51m_mutex	
13.92.7.52m_note_off_margin	
13.93seq64::trigger Class Reference	
13.93.1 Detailed Description	
13.93.2 Constructor & Destructor Documentation	
13.93.2.1 trigger()	
13.93.3 Member Function Documentation	
13.93.3.1 operator<()	
13.93.3.2 length()	
13.93.3.3 tick_start() [1/2]	
13.93.3.4 tick_start() [2/2]	
13.93.3.5 increment_tick_start()	
13.93.3.6 decrement_tick_start()	
13.93.3.7 tick_end() [1/2]	
13.93.3.8 tick_end() [2/2]	
13.93.3.9 increment_tick_end()	
13.93.3.1@ecrement_tick_end()	
13.93.3.11offset() [1/2]	
13.93.3.12offset() [2/2]	
13.93.3.13ncrement_offset()	
13.93.3.14decrement_offset()	
13.93.3.15selected() [1/2]	
13.93.3.16selected() [2/2]	

13.93.4 Field Documentation
13.93.4.1 m_tick_start
13.93.4.2 m_tick_end
13.93.4.3 m_offset
13.93.4.4 m_selected
13.94seq64::triggers Class Reference
13.94.1 Member Typedef Documentation
13.94.1.1 List
13.94.1.2 Stack
13.94.2 Member Enumeration Documentation
13.94.2.1 grow_edit_t
13.94.3 Constructor & Destructor Documentation
13.94.3.1 triggers()
13.94.3.2 ~triggers()
13.94.4 Member Function Documentation
13.94.4.1 operator=()
13.94.4.2 set_ppqn()
13.94.4.3 set_length()
13.94.4.4 triggerlist() [1/2]
13.94.4.5 triggerlist() [2/2]
13.94.4.6 push_undo()
13.94.4.7 pop_undo()
13.94.4.8 pop_redo()
13.94.4.9 print()
13.94.4.1@lay()
13.94.4.11add()
13.94.4.12adjust_offsets_to_length()
13.94.4.13split() [1/2]
13.94.4.14grow()
13.94.4.15remove()

CONTENTS cxli

13	94.4.16get_state()	64
13	94.4.17select()	64
13	94.4.1&unselect()	65
13	94.4.19ntersect()	65
13	94.4.20remove_selected()	65
13	94.4.21copy_selected()	65
13	94.4.22paste()	66
13	94.4.23move_selected()	66
13	94.4.24get_selected_start()	66
13	94.4.25get_selected_end()	67
13	94.4.26get_maximum()	67
13	94.4.27move()	67
13	94.4.2©()	67
13	94.4.29clear()	69
13	94.4.30next()	69
13	94.4.31next_trigger()	69
13	94.4.32reset_draw_trigger_marker()	69
13	94.4.33set_trigger_paste_tick()	70
13	94.4.34get_trigger_paste_tick()	70
13	94.4.35adjust_offset()	70
13	94.4.36split() [2/2]	70
13.94.5 Fri	ends And Related Function Documentation	70
13	94.5.1 midi_container	71
13	94.5.2 midifile	71
13	94.5.3 sequence	71
13	94.5.4 Seq24PerfInput	71
13	94.5.5 FruityPerfInput	71
13.94.6 Fie	ld Documentation	71
13	94.6.1 m_parent	71
13	94.6.2 m_triggers	71

13.94.6.3 m_clipboard
13.94.6.4 m_undo_stack
13.94.6.5 m_redo_stack
13.94.6.6 m_iterator_play_trigger
13.94.6.7 m_iterator_draw_trigger
13.94.6.8 m_trigger_copied
13.94.6.9 m_paste_tick
13.94.6.10m_ppqn
13.94.6.11m_length
13.95seq64::user_instrument Class Reference
13.95.1 Detailed Description
13.95.2 Constructor & Destructor Documentation
13.95.2.1 user_instrument() [1/2]
13.95.2.2 user_instrument() [2/2]
13.95.3 Member Function Documentation
13.95.3.1 operator=()
13.95.3.2 is_valid()
13.95.3.3 set_defaults()
13.95.3.4 name()
13.95.3.5 controller_count()
13.95.3.6 controller_max()
13.95.3.7 controller_name()
13.95.3.8 controller_active()
13.95.3.9 set_controller()
13.95.3.10set_name()
13.95.3.11copy_definitions()
13.95.4 Field Documentation
13.95.4.1 m_is_valid
13.95.4.2 m_controller_count
13.95.4.3 m_instrument_def

CONTENTS cxliii

13.96seq64::user_instrument_t Struct Reference
13.96.1 Field Documentation
13.96.1.1 instrument
13.96.1.2 controllers
13.96.1.3 controllers_active
13.97seq64::user_midi_bus Class Reference
13.97.1 Detailed Description
13.97.2 Constructor & Destructor Documentation
13.97.2.1 user_midi_bus() [1/2]
13.97.2.2 user_midi_bus() [2/2]
13.97.3 Member Function Documentation
13.97.3.1 operator=()
13.97.3.2 is_valid()
13.97.3.3 set_defaults()
13.97.3.4 name()
13.97.3.5 channel_count()
13.97.3.6 channel_max()
13.97.3.7 instrument()
13.97.3.8 set_instrument()
13.97.3.9 set_name()
13.97.3.10copy_definitions()
13.97.4 Field Documentation
13.97.4.1 m_is_valid
13.97.4.2 m_channel_count
13.97.4.3 m_midi_bus_def
13.98seq64::user_midi_bus_t Struct Reference
13.98.1 Field Documentation
13.98.1.1 alias
13.98.1.2 instrument
13.99seq64::user_settings Class Reference

13.99.1 Detailed Description
13.99.2 Member Typedef Documentation
13.99.2.1 Busses
13.99.2.2 BussIterator
13.99.2.3 BussConstlterator
13.99.2.4 Instruments
13.99.2.5 InstrumentIterator
13.99.2.6 InstrumentConstIterator
13.99.3 Member Enumeration Documentation
13.99.3.1 mainwid_grid_style_t
13.99.4 Constructor & Destructor Documentation
13.99.4.1 user_settings() [1/2]109
13.99.4.2 user_settings() [2/2]109
13.99.5 Member Function Documentation
13.99.5.1 operator=()
13.99.5.2 set_defaults()
13.99.5.3 normalize()
13.99.5.4 add_bus()
13.99.5.5 add_instrument()
13.99.5.6 bus()
13.99.5.7 instrument()
13.99.5.8 bus_count()
13.99.5.9 set_bus_instrument()
13.99.5.10bus_instrument()
13.99.5.11bus_name()
13.99.5.12instrument_count()
13.99.5.13set_instrument_controllers()
13.99.5.14instrument_name() [1/2]
13.99.5.15instrument_name() [2/2]
13.99.5.16instrument_controller_active()

13.99.5.17controller_active()
13.99.5.18nstrument_controller_name()
13.99.5.19controller_name()
13.99.5.20grid_style() [1/2]
13.99.5.21grid_is_normal()
13.99.5.22grid_is_white()
13.99.5.23grid_is_black()
13.99.5.24grid_brackets() [1/2]
13.99.5.25mainwnd_rows() [1/2]1097
13.99.5.26mainwnd_cols() [1/2]
13.99.5.27seqs_in_set()
13.99.5.28gmute_tracks()
13.99.5.29max_sets() [1/2]1097
13.99.5.30max_sequence()
13.99.5.3†text_x() [1/2]1097
13.99.5.32ext_y() [1/2]1097
13.99.5.33seqchars_x() [1/2]1098
13.99.5.34seqchars_y() [1/2]1098
13.99.5.35seqarea_x() [1/2]
13.99.5.36seqarea_y() [1/2]
13.99.5.37seqarea_seq_x() [1/2]1098
13.99.5.38seqarea_seq_y() [1/2]1098
13.99.5.39mainwid_border() [1/2]
13.99.5.40mainwid_spacing() [1/2]1098
13.99.5.41mainwid_x()
13.99.5.42mainwid_y()
13.99.5.43control_height() [1/2]
13.99.5.44zoom() [1/2]1099
13.99.5.45zoom() [2/2]1099
13.99.5.46global_seq_feature() [1/2]

13.99.5.47global_seq_feature() [2/2]1099
13.99.5.48seqedit_scale() [1/2]1100
13.99.5.49seqedit_scale() [2/2]1100
13.99.5.50seqedit_key() [1/2]1100
13.99.5.51seqedit_key() [2/2]1100
13.99.5.52seqedit_bgsequence() [1/2]
13.99.5.53seqedit_bgsequence() [2/2]
13.99.5.54use_new_font() [1/2]1100
13.99.5.55allow_two_perfedits() [1/2]
13.99.5.5\(\text{perf_h_page_increment}\) [1/2]
13.99.5.57perf_v_page_increment() [1/2]
13.99.5.5&progress_bar_colored() [1/2]
13.99.5.59progress_bar_thick() [1/2]
13.99.5.6@nverse_colors() [1/2]
13.99.5.61window_redraw_rate() [1/2]
13.99.5.62use_more_icons() [1/2]
13.99.5.63save_user_config() [1/2]1102
13.99.5.64save_user_config() [2/2]1102
13.99.5.65grid_brackets() [2/2]1102
13.99.5.66grid_style() [2/2]1102
13.99.5.67mainwnd_rows() [2/2]1102
13.99.5.68mainwnd_cols() [2/2]
13.99.5.69max_sets() [2/2]1102
13.99.5.70text_x() [2/2]1103
13.99.5.7 ttext_y() [2/2]1103
13.99.5.72seqchars_x() [2/2]1103
13.99.5.73seqchars_y() [2/2]1103
13.99.5.74seqarea_x() [2/2]1103
13.99.5.75seqarea_y() [2/2]1103
13.99.5.76seqarea_seq_x() [2/2]1103

CONTENTS cxlvii

13.99.5.77seqarea_seq_y() [2/2]1104
13.99.5.78mainwid_border() [2/2]
13.99.5.79mainwid_spacing() [2/2]1104
13.99.5.80control_height() [2/2]
13.99.5.81dump_summary()
13.99.5.82midi_ppqn() [1/2]
13.99.5.83midi_beats_per_bar() [1/2]
13.99.5.84midi_beats_per_minute() [1/2]
13.99.5.85midi_beat_width() [1/2]
13.99.5.86midi_buss_override() [1/2]
13.99.5.87velocity_override() [1/2]1105
13.99.5.8±_precision() [1/2]
13.99.5.89bpm_step_increment() [1/2]
13.99.5.90bpm_page_increment() [1/2]
13.99.5.91min_zoom()
13.99.5.92max_zoom()
13.99.5.93baseline_ppqn()
13.99.5.94use_new_font() [2/2]1106
13.99.5.95allow_two_perfedits() [2/2]
13.99.5.9@perf_h_page_increment() [2/2]
13.99.5.97perf_v_page_increment() [2/2]
13.99.5.9&progress_bar_colored() [2/2]
13.99.5.99progress_bar_thick() [2/2]1107
13.99.5.100verse_colors() [2/2]
13.99.5.10\(\mathbb{W}\) indow_redraw_rate() [2/2]
13.99.5.102se_more_icons() [2/2]
13.99.5.103idi_ppqn() [2/2]1107
13.99.5.104idi_buss_override() [2/2]
13.99.5.10/Eelocity_override() [2/2]1107
13.99.5.10@om_precision() [2/2]

13.99.5.107pm_step_increment() [2/2]
13.99.5.102pm_page_increment() [2/2]
13.99.5.109idi_beats_per_bar() [2/2]1108
13.99.5.11m0idi_beats_per_minute() [2/2]1108
13.99.5.11rhidi_beat_width() [2/2]
13.99.5.11p2rivate_bus()
13.99.5.11parivate_instrument()
13.99.6 Friends And Related Function Documentation
13.99.6.1 userfile
13.99.7 Field Documentation
13.99.7.1 m_midi_buses
13.99.7.2 m_instruments
13.99.7.3 m_grid_style
13.99.7.4 m_grid_brackets
13.99.7.5 m_mainwnd_rows
13.99.7.6 m_mainwnd_cols
13.99.7.7 m_max_sets
13.99.7.8 m_mainwid_border
13.99.7.9 m_mainwid_spacing
13.99.7.10m_control_height
13.99.7.11m_current_zoom
13.99.7.12m_global_seq_feature_save
13.99.7.13m_seqedit_scale
13.99.7.14m_seqedit_key
13.99.7.15m_seqedit_bgsequence
13.99.7.16m_use_new_font
13.99.7.17m_allow_two_perfedits
13.99.7.18m_h_perf_page_increment
13.99.7.19m_v_perf_page_increment
13.99.7.20m_progress_bar_colored

CONTENTS cxlix

13.99.7.21m_progress_bar_thick	1112
13.99.7.22m_inverse_colors	1112
13.99.7.23m_window_redraw_rate_ms	1113
13.99.7.24m_use_more_icons	1113
13.99.7.25m_text_x	1113
13.99.7.26m_text_y	1113
13.99.7.27m_seqchars_x	1113
13.99.7.28m_seqchars_y	1114
13.99.7.29m_midi_ppqn	1114
13.99.7.30m_midi_beats_per_measure	1114
13.99.7.31m_midi_beats_per_minute	1114
13.99.7.32m_midi_beat_width	1114
13.99.7.33m_midi_buss_override	1114
13.99.7.34m_velocity_override	1115
13.99.7.35m_bpm_precision	1115
13.99.7.36m_bpm_step_increment	1115
13.99.7.37m_bpm_page_increment	1115
13.99.7.38m_total_seqs	1115
13.99.7.39m_seqs_in_set	1116
13.99.7.40m_gmute_tracks	1116
13.99.7.41m_max_sequence	1116
13.99.7.42m_seqarea_x	1116
13.99.7.43m_seqarea_y	1116
13.99.7.44m_seqarea_seq_x	1116
13.99.7.45m_seqarea_seq_y	1117
13.99.7.46m_mainwid_x	1117
13.99.7.47m_mainwid_y	1117
13.99.7.48m_save_user_config	1117
13.99.7.49mc_min_zoom	1117
13.99.7.50mc_max_zoom	1117
13.99.7.51mc_baseline_ppqn	1118
13.108eq64::userfile Class Reference	1118
13.100.1Constructor & Destructor Documentation	1119
13.100.1.1userfile()	1119
13.100.1.2~userfile()	1119
13.100.2Member Function Documentation	1119
13.100.2.1parse()	
13.100.2.2write()	
13.100.2.3dump_setting_summary()	1120
Index	1121

Chapter 1

Sequencer64

Author(s) Chris Ahlstrom 2016-10-23

1.1 Introduction

Sequencer64 is a major cleanup, refactoring, and documentation of the Seq24 live-play MIDI sequencer.

The current document, generated by Doxygen, describes the functions, classes, modules, and other entities used in this project.

Also read the ROADMAP, README, and contrib/bugs_to_investigate files to understand the genesis of this project and the things that still need to be done with Sequencer64.

Also, we have pretty deeply documented *Seq24* and *Sequencer64* with PDF files that can be generated by git-cloning the following projects, installing a number of tools related to PDF and LaTeX, and running "make":

- https://github.com/ahlstromcj/seq24-doc.git
- https://github.com/ahlstromcj/sequencer64-doc.git

These project also have prebuilt PDFs should one not want to bother building them.

In the present document, we've left out a some side-code to cut down on the size of the document. Still, the resulting PDF is over 1000 pages long.

Some useful references:

- http://www.midimusicadventures.com/qs/midi-zips/soundtracks/kq6gm.zip

2 Sequencer64

Chapter 2

MIDI File Parsing in Sequencer64

Author(s) Chris Ahlstrom 2016-02-13

2.1 Introduction

This section describes the parsing of a MIDI file (and a few other topics). We wanted to add the reading of SMF 0 files to *Sequencer64*. We started with the main format that is supported, SMF 1. Once we understood that we, we figured out how to split a SMF 0 tracks correctly.

We split the midifile::parse() function into two sections. The first section analyzes the header of the MIDI. Then, based on whether the file is SMF 1 (the normal case) or SMF 0, either the parse_smf_1() function of or the parse—smf_0() function is called. The parse_smf_0() function creates one sequence object per channel present in the SMF 0 file, plus the original track. The last pattern slot (sequence 16) will contain the original track data, and the rest will contain common data and then channel data for each channel. After the parsing is done, all the tracks (including the original track) will be added to the performance. The user then has the option of deleting the original track, which will be the last track.

2.2 SMF 1 Parsing

This section describes the parsing of the header chunk, MThd, and the track chunk, MTrk.

The midifile::parse() function starts by opening the MIDI file, getting its file-size, pre-allocating the data vector to that size, reading all of the characters into that vector, and then closing the file.

2.2.1 MIDI File Header, MThd

The data of the header is read:

```
Header ID: "MThd" read_long() 4 bytes
MThd length: 6 read_long() 4 bytes
Format: 0, 1, 2 read_short() 2 bytes
No. of track: 1 or more read_short() 2 bytes
PPQN: 192 read_short() 2 bytes
```

The header ID and it's length are always the same values. The formats that Sequencer64 supports are 0 or 1. SMF 0 has only one track, while SMF 1 can support an arbitary number of tracks. The last value in the header is the PPQN value, which specifies the "pulses per quarter note", which is the basic time-resolution of events in the MIDI file. Common values are 96 or 192, but higher values are also common. Sequencer64 and its precursor, Seq24, default to 192.

2.2.2 MIDI Track, MTrk

Sequencer64 next reads the tracks specified in the file. Each track is assumed to cover a different MIDI channel, but always the same MIDI buss. (The MIDI buss is not a data item in standard MIDI files, but it is a special data item in Seq24/Sequencer64 MIDI files.) Each track is tagged by a standard chunk marker, "MTrk". Other markers are possible, and are to be ignored, if nothing else. Here are the values read at the beginning of a track:

```
Track ID: "MTrk" read_long() 4 bytes
Track length: varies read_long() 4 bytes
```

The track length is the number of bytes that need to be read in order to get all of the data in the track.

Next, a new sequence object is created, with the PPQN value passed to its constructor. The sequence then is hooked to the master MIDI buss object. The "RunningTime" accumulator is set to 0 for that track.

Next, the parse() function loops through the rest of the track, reading data and logging it to the sequence. Let's go through the loop, which is the meat of the processing.

TODO: An empty event is created before track processing, and re-used for every track and event. This seems dangerous. We moved the event constructor two levels of nesting deeper, and it seems to work fine.

Delta time. The amount time that passes from one event to the next is the *delta time*. For some events, the time doesn't matter, and is set to 0. This values is a *variable length value*, also known as a "VLV" or a "varinum". It provides a way of encoding arbitrarily large values, a byte at a time. For now, just note that a varinum is 1 or more bytes, and MIDI provides a way to tell when the varinum is complete.

```
Delta time: varies read_varinum() 1 or more bytes
```

2.2.2.1 Channel Events

Status. The byte after the delta time is examined by masking it against 0x80 to check the high bit. If not set, it is a "running status", it is replaced with the "last status", which is 0 at first.

```
Status byte: varies read_byte() 1 byte
```

If the high bit is set, it is a status, and is passed to the setter $event::set_status()$.

The "RunningTime" accumulator is incremented by the delta-time. The current time is adjusted as per the PPQN ratio, if needed, and passed to the setter $event::set_timestamp()$.

Now what does the status mean? First, the channel part of the status is masked out using the 0xF0 mask.

If it is a 2-data-byte event (note on, note off, aftertouch, control-change, or pitch-wheel), then the two data bytes are read:

```
Data byte 0: varies read_byte() 1 byte
Data byte 1: varies read_byte() 1 byte
```

If the status is a note-on event, with data[1] = 0, then it is converted to a note-off event, a fix for the output quirks of some MIDI devices, and the status of the event is amended to EVENT_NOTE_OFF.

If it is a 1-data-btye event (program change or channel pressure), then only data byte 0 is read.

Then the one or two data bytes are added to the event by overloads of event::set_data(), the event is added to the current sequence by sequence::add_event(), and the MIDI channel of the sequence is set by sequence::set_midi_channel().

Note that this is the point where parsing could detect a change in channel, and select a new sequence to support that channel, and add the events to that sequence, if the file were SMF 0.

Also note that the channel of the sequence is set every a new channel event/status is read. This should be done once, and then simply warned about if a non-matching channel occurs.

Lastly, note that it might be better to do the sequence function calls at the end of processing the event.

2.2 SMF 1 Parsing 5

2.2.2.2 Meta Events

If the event status masks off to 0xF0 (0xF0 to 0xFF), then it is a meta event. If the status is 0xFF, it is called a "Sequencer-specific", or "SeqSpec" event. For this kind of event, then a type byte and the length of the event are read.

```
Meta type: varies read_byte() 1 byte
Meta length: varies read_varinum() 1 or more bytes
```

If the type of the SeqSpec (0xFF) meta event is 0x7F, parsing checks to see if it is one of the Seq24 "proprietary" events. These events are tagged with various values that mask off to 0x24240000. The parser reads the tag:

```
Prop tag: 0x242400nn read_long() 4 bytes
```

These tags provide a way to save and recover Seq24/Sequencer64 properties from the MIDI file: MIDI buss, MIDI channel, time signature, sequence triggers, and (new), the key, scale, and background sequence to use with the track/sequence. Any leftover data for the tagged event is let go. Unknown tags ate skipped.

If the type of the SeqSpec (0xFF) meta event is 0x2F, then it is the End-of-Track marker. The current time is set using $sequence::set_length()$ and then $sequence::zero_markers()$ is called, and parsing is done for that track.

If the type of the SeqSpec (0xFF) meta event is 0x03, then it is the sequence name. The "length" number of bytes are read, and loaded by $sequence::set_name()$.

If the type of the SeqSpec (0xFF) meta event is 0x00, then it is the sequence number, which is read:

```
Seq number: varies read_short() 2 bytes
```

Note that the sequence number might be modified latter to account for the current screenset in force for a file import operation.

Anything other SeqSpec type is simply skipped by reading the "length" number of bytes.

To summarize the process, here are the relevant event and sequence setter calls typically made while parsing a MIDI track:

```
1. perform::add_sequence()
   (a) sequence::sequence()
   (b) sequence::set_master_midi_bus())
   (c) sequence::add_event()
        i. event::event()
        ii. event::set_status()
        iii. event::set_timestamp()
        iv. event::set_data()
   (d) sequence::set_midi_channel()
        (e) sequence::set_length()
        (f) sequence::set_name()
        (g) sequence::set_midi_bus()
2. xxxxx::yyyy()
```

2.2.3 Meta Events Summary

Here, we summarize the MIDI meta events for your edification.

```
1. FF 00 02 ssss: Sequence Number.
```

- 2. FF 01 len text: Text Event.
- 3. FF 02 len text: Copyright Notice.
- 4. FF 03 len text: Sequence/Track Name.
- 5. FF 04 len text: Instrument Name.
- 6. FF 05 len text: Lyric.
- 7. FF 06 len text: Marker.
- 8. FF 07 len text: Cue Point.
- 9. FF 08 len text: Patch/program Name.
- 10. FF 09 len text: Device Name.
- 11. FF 0A through 0F len text: Other kinds of text events.
- 12. FF 20 01 cc: MIDI channel (obsolete, used by Cakewalk)
- 13. FF 21 01 pp: MIDI port (obsolete, used by Cakewalk)
- 14. FF 2F 00: End of Track.
- 15. FF 51 03 tttttt: Set Tempo, us/qn.
- 16. FF 54 05 hr mn se fr ff: SMPTE Offset.
- 17. FF 58 04 nn dd cc bb: Time Signature.
- 18. FF 59 02 sf mi: Key Signature.
- 19. FF 7F len data: Sequencer-Specific.

The next sections describe the events that Sequencer tries to handle. These are

- Sequence Number (0x00)
- Track Name (0x03)
- End-of-Track (0x2F)
- Set Tempo (0x51) (Sequencer64 only)
- Time Signature (0x58) (Sequencer64 only)
- Sequencer-Specific (0x7F)
- System Exclusive (0xF0) Sort of handled, functionality incomplete..

2.2.3.1 Sequence Number (0x00)

```
FF 00 02 ss ss
```

This optional event must occur at the beginning of a track, before any non-zero delta-times, and before any transmittable MIDI events. It specifies the number of a sequence.

2.2 SMF 1 Parsing 7

2.2.3.2 Track/Sequence Name (0x03)

```
FF 03 len text
```

If in a format 0 track, or the first track in a format 1 file, the name of the sequence. Otherwise, the name of the track.

2.2.3.3 End of Track (0x2F)

```
FF 2F 00
```

This event is not optional. It is included so that an exact ending point may be specified for the track, so that it has an exact length, which is necessary for tracks which are looped or concatenated.

2.2.3.4 Set Tempo Event (0x51)

The MIDI Set Tempo meta event sets the tempo of a MIDI sequence in terms of the microseconds per quarter note. This is a meta message, so this event is never sent over MIDI ports to a MIDI device.

After the delta time, this event consists of six bytes of data:

```
FF 51 03 tt tt tt
```

Example:

```
FF 51 03 07 A1 20
```

- 1. 0xFF is the status byte that indicates this is a Meta event.
- 2. 0x51 the meta event type that signifies this is a Set Tempo event.
- 3. 0x03 is the length of the event, always 3 bytes.
- 4. The remaining three bytes carry the number of microseconds per quarter note. For example, the three bytes above form the hexadecimal value 0x07A120 (500000 decimal), which means that there are 500,000 microseconds per quarter note.

Since there are 60,000,000 microseconds per minute, the event above translates to: set the tempo to 60,000,000 / 500,000 = 120 quarter notes per minute (120 beats per minute). This is a 24-bit binary value, so each byte covers the full range of 0x00 to 0xFF.

This event normally appears in the first track. If not, the default tempo is 120 beats per minute. This event is important if the MIDI time division is specified in "pulses per quarter note", which does not itself define the length of the quarter note. The length of the quarter note is then determined by the Set Tempo meta event.

Representing tempos as time per beat instead of beat per time allows absolutely exact DWORD-term synchronization with a time-based sync protocol such as SMPTE time code or MIDI time code. This amount of accuracy provided by this tempo resolution allows a four-minute piece at 120 beats per minute to be accurate within 500 usec at the end of the piece.

We have now added the Tempo meta event (and the Time Signature meta event) to the track, which allows other sequencers to obtain these values from a Sequencer64 MIDI file. Here are the original headers for a normal MIDI file and its legacy (Seq24) conversion, as shown by the midicvt application:

```
hymne.asc
                                        hymne-ppqn-384.asc
MThd 1 4 96
                                        MThd 1 4 384
                                        MTrk
0 Meta SeqName "Vangelis: Hymne"
                                        0 SeqNr 0
0 TimeSig 4/4 24 8
                                        0 Meta SeqName "Vangelis: Hymne"
                                       0 SeqSpec 24 24 00 08 (no triggers)
0 SeqSpec 24 24 00 01 00 (MIDI buss 0)
0 Tempo 750000
0 Meta TrkEnd
                                        0 SeqSpec 24 24 00 06 04 04 (beats, width)
TrkEnd
                                        0 SeqSpec 24 24 00 02 00 (MIDI ch. 0)
                                        96 Meta TrkEnd
                                        TrkEnd
```

Here is the header data that result from the new conversion, which is used if the "legacy" option is not in force:

```
MThd 1 4 192
MTrk
0 SeqNr 0
0 Meta SeqName "Vangelis: Hymne"
0 TimeSig 4/4 24 8
0 Tempo 750000
0 SeqSpec 24 24 00 08
0 SeqSpec 24 24 00 01 00
0 SeqSpec 24 24 00 06 04 04
0 SeqSpec 24 24 00 02 00
48 Meta TrkEnd
TrkEnd
```

2.2.3.5 Time Signature Event (0x58)

After the delta time, this event consists of seven bytes of data:

```
FF 58 04 nn dd cc bb
```

The time signature is expressed as four numbers. nn and dd represent the numerator and denominator of the time signature as it would be notated. The numerator counts the number of beats in a measure (beats per measure or beats per bar). The denominator is a negative power of two: 2 represents a quarter-note, 3 represents an eighthnote, etc. The denominator specifies the unit of the beat (e.g. 4 or 8). In Seq24/Sequencer64, this value is also called the "beat width".

The CC parameter expresses the number of MIDI clocks (or "ticks", or "pulses") in a metronome click. The standard MIDI clock ticks 24 times per quarter note, so a value of 6 would mean the metronome clicks every 1/8th note. A CC value of 6 would mean that the metronome clicks once every 1/8th of a note (quaver). This MIDI clock is different from the clock (PPQN) that determines the start time and duration of the notes.

The bb parameter expresses the number of notated 32nd-notes in a MIDI quarter note (24 MIDI Clocks). The usual value for this parameter is 8, though some sequencers allow the user to specify that what MIDI thinks of as a quarter note, should be notated as something else. For example, a value of 16 means that the music plays two quarter notes for each quarter note metered out by the MIDI clock, so that the music plays at double speed.

Examples:

```
FF 58 04 04 02 18 08
```

- 1. 0xFF is the status byte that indicates this is a Meta event.
- 2. 0x58 the meta event type that signifies this is a Time Signature event.

2.2 SMF 1 Parsing 9

- 3. 0x04 is the length of the event, always 4 bytes.
- 4. 0x04 is the numerator of the time signature, and ranges from 0x00 to 0xFF.
- 5. 0x02 is the log base 2 of the denominator, and is the power to which 2 must be raised to get the denominator. Here, the denominator is 2 to 0x02, or 4, so the time signature is 4/4.
- 6. 0x18 is the metronome pulse in terms of the number of MIDI clock ticks per click. Assuming 24 MIDI clocks per quarter note, the value here (0x18 = 24) indidicates that the metronome will tick every 24/24 quarter note. If the value of the sixth byte were 0x30 = 48, the metronome clicks every two quarter notes, i.e. every half-note.
- 7. 0x08 defines the number of 32nd notes per beat. This byte is usually 8 as there is usually one quarter note per beat, and one quarter note contains eight 32nd notes.

A time signature of 6/8, with a metronome click every 3rd 1/8 note, would be encoded:

```
FF 58 04 06 03 24 08
```

Remember, a 1/4 note is 24 MIDI Clocks, therefore a bar of 6/8 is 72 MIDI Clocks. Hence 3 1/8 notes is 36 (=0x24) MIDI Clocks.

There should generally be a Time Signature Meta event at the beginning of a track (at time = 0), otherwise a default 4/4 time signature will be assumed. Thereafter they can be used to effect an immediate time signature change at any point within a track.

For a format 1 MIDI file, Time Signature Meta events should only occur within the first MTrk chunk.

If a time signature event is not present in a MIDI sequence, 4/4 signature is assumed.

In Sequencer64, the c_timesig SeqSpec event is given priority. The conventional time signature is used only if the c_timesig SeqSpec is not present in the file. NEEDS TO BE TESTED.

```
2.2.3.6 SysEx Event (0xF0)
```

If the meta event status value is 0xF0, it is called a "System-exclusive", or "SysEx" event.

```
F0 len data F7
```

Sequencer64 has some code in place to store these messages, but the data is currently not actually stored or used. Although there is some infrastructure to support storing the SysEx event within a sequence, the SysEx information is simply skipped. Sequencer64 warns if the terminating 0xF7 SysEx terminator is not found at the expected length. Also, some malformed SysEx events have been encountered, and those are detected and skipped as well.

2.2.3.7 Sequencer Specific (0x7F)

This data, also known as SeqSpec data, provides a way to encode information that a specific sequencer application needs, while marking it so that other sequences can safely ignore the information.

```
FF 7F len data
```

In *Seq24* and *Sequencer64*, the data portion starts with four bytes that indicate the kind of data for a particular SeqSpec event:

In Seq24, these events are placed at the end of the song, but are not marked as SeqSpec data. Most MIDI applications handle this situation fine, but some (e.g. midicvt) do not. Therefore, Sequencer64 makes sure to wrap each data item in the 0xFF 0x7F wrapper.

Also, the last three items above (key, scale, and background sequence) can also be stored (by *Sequencer64*) with a particular sequence/track, as well as at the end of the song. Not sure if this bit of extra flexibility is useful, but it is there.

2.2.3.8 Non-Specific End of Sequence

Any other statuses are deemed unsupportable in Sequencer64, and abort parsing with an error.

If the —bus option is in force, sequence::set_midi_bus() is called to override the buss number (if any) stored with the sequence.

Finally, $perform::add_sequence()$ adds the sequence to the encoded tune.

2.3 SMF 0 Parsing

After parsing SMF 1 track data, we end up with a number of sequences, each on a different MIDI channel. With SMF 0, data for all channels is present in a single track. Sequencer64 will read SMF 0 data, but we really need to be able to have one MIDI channel per track. So we need to take the data from the sequence and use it to make more sequences.

```
sequence::add_event().
sequence::set_midi_channel().
sequence::set_length().
sequence::set_midi_bus().
perform::add_sequence().
```

This code basically works. For now, please look at the source code for more details. Also, the reading of SMF 0 MIDI files is described in the *sequencer64-doc* project on GitHub.

2.4 Running Status

2.4 Running Status

When we apply the midicvt application to a file saved by Sequencer64, we can end up with a successful ASCII conversion that ends with an error message:

```
$ midicvt hymne-seq64.midi -o hymne-seq64.asc
? Error at MIDI file offset 12155 [0x2f7b]
Error: Garbage at end 'readtrack(): unexpected running status'
```

Is this a problem in midicvt or Sequencer4? Let's learn about running status.

Running status is a way to speed up the sending of MIDI bytes to a synthesizer or sequencer by taking advantage of redundancy where possible. For example, if we're sending a consecutive group of Note On and Note Off messages to a particular channel, we can save some time by not sending the channel status byte after the first time. Here's an example with Note On on channel 1:

```
0x90 3C 7F
0x90 40 7F
0x90 43 F3
```

Since no change in status occurs after the first of these three events, we can drop the subsequent status bytes:

```
0x90 3C 7F
40 7F
43 F3
```

The 0x90 byte is saved in a "running status buffer" (RSB), and is filled in by the receiving device.

Here is the sequence of events for operating with running status.

- 1. Clear the RSB buffer (RSB = 0) to start.
- 2. If a **Voice Category Status** (VCS) byte is received, then set RSB = VCS. VCS bytes range from 0x80 to 0xEF. This is binary 1000000 to 11100000.
- 3. If a data byte is received (data bytes range from 0x00 to 0x7F, binary 0000000 to 0111111; that is, bit 7 is always 0 in a data byte):
 - (a) If RSB != 0, first insert the RSB into the incoming data stream, then insert the data byte.
 - (b) If RSB == 0, then just insert the data byte into the incoming data stream.
- 4. Clear the RSB buffer (RSB = 0) when a System Common Message (SCM) status byte is received. SCM bytes range from 0xF0 to 0xF7.
- 5. The message after an SCM must begin with a status byte. That is a byte with bit 7 set.
- 6. Do no special action when a Realtime Category Message (RCM) byte is received. RCM bytes range from 0xF8 to 0xFF.

Note that some events, such as Tempo, assume that its bytes are all data bytes.

Chapter 3

JACK, Live, and Song Modes in Sequencer64

Author(s) Chris Ahlstrom 2017-04-01

3.1 Introduction

This section describes the interactions between JACK settings and the Live/Song Mode settings, with an eye to describing the proper behavior of Sequencer64 with JACK settings, how the Live/Song modes are supposed to work, and what bugs or issues remain in Sequencer64's JACK handling.

Note: There is currently no description of the 0.90.x line's native JACK MIDI support, except what is found in the developer's reference manual and in the updated sequencer64-doc project at GitHub.

I'm not sure why Doxygen is applying the "code" font so often here. Weird, annoying.

3.2 JACK Functions

Please study the following URL and note these important points:

http://jackaudio.org/files/docs/html/transport-design.html

- The timebase master continuously updates position information, beats, timecode, etc. There is at most one
 master active at a time. If no client is registered as timebase master, frame numbers will be the only position
 information available.
- The timebase master registers a callback that updates position information while transport is rolling. Its output affects the following process cycle. This function is called immediately after the process callback in the same thread whenever the transport is rolling, or when any client has set a new position in the previous cycle.
- Clients that don't declare a sync callback are assumed ready immediately, anytime the transport wants to start. If a client doesn't require slow-sync processing, it can set its sync callback to NULL.
- The transport state is always valid; initially it is JackTransportStopped.
- When someone calls <code>jack_transport_start()</code>, the engine resets the poll bits and changes to a new state, <code>JackTransportStarting</code>.
- When all slow-sync clients are ready, the state changes to JackTransportRolling.

Does Sequencer64 need a latency callback?

http://jackaudio.org/files/docs/html/group__ClientCallbacks.html

(We need to see why most of the following is in a monospaced font. Is there a new Doxygen feature?)

Here are summaries of the JACK functions used in the jack_assistant module:

3.2.1 jack_client_open()

```
Open a client session with a JACK server. More complex and powerful than <tt>jack_client_new()</tt>.

Clients choose which of several servers to connect, and how to start the server automatically, if not already running. There is also an option for JACK to generate a unique client name.

const char * client_name, jack_options_t options, jack_status_t * status,
```

client_name of at most jack_client_name_size() characters. The name scope is local to each server. Unless forbidden by the JackUseExactName option, the server will modify this name to create a unique variant, if needed.

options formed by OR-ing together JackOptions bits. Only the JackOpenOptions bits are allowed.

status (if non-NULL) an address for JACK to return information from the open operation. This status word is formed by OR-ing together the relevant JackStatus bits.

Optional parameters: depending on corresponding [options bits] additional parameters may follow status (in this order).

[JackServerName] (char *) server_name selects from among several possible concurrent server instances. Server names are unique to each user. If unspecified, use "default" unless \$JACK_DEFAULT_SERVER is defined in the process environment.

Returns:

Opaque client handle if successful. If this is NULL, the open operation failed, and *status includes JackFailure, and the caller is not a JACK client.

3.2.2 jack_on_shutdown()

Registers a function to call when the JACK server shuts down the client thread. It must be an asynchonrous POSIX signal handler: only async-safe functions, executed from another thread. A typical function might set a flag or write to a pipe so that the rest of the application knows that the JACK client thread has shut down. Clients do not need to call this function. It only helps clients understand what is going on. It should be called before <tt>jack_client_activate()</tt>.

3.2.3 jack_set_sync_callback()

Register/unregister as a slow-sync client; it can't respond immediately to transport position changes. The callback is run at the first opportunity after registration: if the client is active, this is the next process cycle, otherwise it is the first cycle after <tt>jack_activate()</tt>. After that, it runs as per JackSyncCallback rules. Clients that don't set this callback are assumed ready immediately any time the transport wants to start.

3.2 JACK Functions 15

3.2.4 jack_set_process_callback()

Tells the JACK server to call the callback whenever there is work. The function must be suitable for real-time execution, it cannot call functions that might block for a long time: malloc(), free(), printf(), pthread_mutex_lock(), sleep(), wait(), poll(), select(), pthread_join(), pthread_cond_wait(), etc. In the current class, this function is a do-nothing function.

3.2.5 jack set session callback()

Tells the JACK server to call the callback when a session event is delivered. Setting more than one session callback per process is probably a design error. For a multiclient application, it's more sensible to create a JACK client with only one session callback.

3.2.6 jack_activate()

Tells the JACK server that the application is ready to start processing.

3.2.7 jack release timebase()

TODO

3.2.8 jack client close()

TODO

3.2.9 jack_transport_start()

Starts the JACK transport rolling. Any client can make this request at any time. It takes effect no sooner than the next process cycle, perhaps later if there are slow-sync clients. This function is realtime-safe. No return code.

3.2.10 jack_transport_stop()

3.2.11 jack_transport_locate()

Repositions the transport to a new frame number. May be called at any time by any client. The new position takes effect in two process cycles. If there are slow-sync clients and the transport is already rolling, it will enter the JackTransportStarting state and begin invoking their sync_callbacks until ready. This function is realtime-safe.

3.2.12 jack_transport_reposition()

```
Request a new transport position. May be called at any time by any client. The new position takes effect in two process cycles. If there are slow-sync clients and the transport is already rolling, it will enter the JackTransportStarting state and begin invoking their sync_callbacks until ready. This function is realtime-safe. This call, made in the position() function, is currently disabled.
```

3.2.13 jack transport query()

Query the current transport state and position. This function is realtime-safe, and can be called from any thread. If called from the process thread, pos corresponds to the first frame of the current cycle and the state returned is valid for the entire cycle.

The first parameter is the client, which is a pointer to the JACK client structure.

The second parameter is a pointer to structure for returning current transport position; pos->valid will show which fields contain valid data. If pos is NULL, do not return position information.

This function returns the current transport state.

3.3 Modes Operation

3.3.1 No JACK, Live Mode

In ~/.config/sequencer64/sequencer64.rc, set:

- jack_transport = 0
- jack_master = 0
- jack master cond = 0
- song_start_mode = 0

By changing the start mode to 0 (false), Sequencer64 is put into Live Mode. With this setting, control of the muting and unmuting of patterns resides in the main window (the patterns window). One can start the playback in the performance (song) window, but it will not affect which patterns play, at all.

Note that this option is part of the File / Options / JACK/LASH configuration page.

3.3.2 No JACK, Song Mode

In \sim /.config/sequencer64/sequencer64.rc, set:

- jack_transport = 0
- jack master = 0
- jack_master_cond = 0
- song_start_mode = 1

By changing the start mode to 1 (true), Sequencer64 is put into Song Mode.

With this setting, control of the muting and unmuting of patterns resides in the song window (the performance window). The patterns shown in the pattern slots of the main window turn on and off whenever the progress bar is in the pattern as drawn in the perforance window.

Note that this option is part of the File / Options / JACK/LASH configuration page.

3.4 Breakage 17

3.3.3 JACK Transport

In ~/.config/sequencer64/sequencer64.rc, set:

- jack_transport = 1
- jack_master = 0
- jack master cond = 0
- song_start_mode = 0 or 1 (see previous section)

The current behavior is that gjackctl and sequencer64 playback/progress seem to be independent of each other.

The workaround seems to be to set seq24/sequencer64 as JACK Master, or if another application (e.g. Qtractor) is JACK Master.

OLD BEHAVIOR:

```
Start qjackctl, verify that it sets up correctly, then click it's
"play" button to start the transport rolling. Run sequencer64, load a
file. Then note that starting playback (whether in the main window or
in the performance window) is ineffective, but resets the time counter
in qjackctl. Why? With JACK sync enabled by the macro:

[JACK transport slave]
    jack sync(): zero frame rate [single report]!?

[JackTransportRolling]
```

[JackTransportStarting] (every time space bar pressed)

END OF OLD BEHAVIOR.

3.4 Breakage

Old message about seq24 being broken:

[Start playback]

 $\verb|http://lists.linuxaudio.org/pipermail/linux-audio-user/2010-November/073848. \leftarrow \verb|htm||$

```
i dont see the transport synchronisation working with a jackl svn version. you are still using only a sync callback.

and you are relying on the transport to go through the JackTransportStarting state.

this issue should be fixed.
iirc we came to the conclusion, that seq24 is broken, and we will not revert the changes in jack, which break it.

the quick and dirty fix on your side, would be to register an empty process_callback.

but the issue still remains. seq24 is NOT a slow sync client. but it registers a sync_callback.
and it even takes a lock in the sync callback.

the patch for jack-session support didnt get merged either.
```

Another one (no need for a URL):

I use seq24 for the majority of my projects but it isn't ideal (I should point out that I never finish anything). I don't like seq24's pianoroll editor, the way you do CC envelopes isn't ideal, it uses alsa-midi, there's unnecessary complexity in switching from pattern-trigger mode to song mode, and its insistence on being transport master while not even being able to adjust tempo when live is annoying

3.5 JACK References

- $\bullet \ \text{http://libremusicproduction.com/articles/demystifying-jack-\$E2\$80\$93-beginners-guidents.} \\$
- http://jackaudio.org/files/docs/html/transport-design.html
- http://kxstudio.linuxaudio.org/Repositories

Chapter 4

User Testing of Sequencer64 with Yoshimi

Author(s) Chris Ahlstrom 2016-03-04

4.1 Introduction

This section describes user testing of Sequencer64 using Yoshimi. It will expand as we work our way through all the many use-cases that can be achieved with Sequencer64 and Yoshimi.

Please note that the most advanced and recent testing can be found currently in the document contrib/notes/jack-testing txt. We will eventually merge the final tests here... someday.

4.2 Smoke Test

Every so often we run Sequencer64 with a software synthesizer to make sure we haven't broken any functionality via our major refactoring efforts. We call it a "smoke test". We fire up the two application, and see if anything smokes.

This smoke test sets up Yoshimi with a very simple ALSA setup, and no instruments are loaded. Instead, only the "Simple Sound" is used on all channels. We've been doing this test with Yoshimi 1.3.6. The current Debian Sid ("testing") version of Yoshimi is 1.3.6-2, pulled from SourceForge. It seems to have issues, so we've been cloning and pulling the code from:

```
https://github.com/Yoshimi/yoshimi.git
```

After getting the application build and installed, the next step is to run it, using ALSA for MIDI and for audio:

```
$ yoshimi -a -A &
```

Next, fix up the configuration files for Sequencer64, \sim /.config/sequencer64/sequencer64.rc and \sim /.config/sequencer64/sequencer64.usr.

First hide sequencer64.usr somewhere, or delete it, as it will determine what MIDI devices are available, and we don't want that (yet). Second, make sure that sequencer64.rc makes the following setting:

```
[manual-alsa-ports]
# Set to 1 if you want sequencer64 to create its own ALSA ports and
# not connect to other clients
0  # number of manual ALSA ports
```

Next, run the newly-built version of Sequencer64. If desired, use the –bus option described below to force the buss number to the buss you need, as shown in the second version of the command:

```
$ sequencer64/sequencer64 &
$ sequencer64/sequencer64 --bus 5 &
```

In File / Options / MIDI Clock, observe the MIDI inputs made available by your system. Our system shows:

```
[0] 14:0 (Midi Through Port-0)

[1] 128:0 (TiMidity port 0)

[2] 128:0 (TiMidity port 1)

[3] 128:0 (TiMidity port 2)

[4] 128:0 (TiMidity port 3)

[5] 129:0 (input)
```

For some reason (a bug in Yoshimi?), input "[5]" doesn't indicate that it is Yoshimi, but it is. Take note of that input number... that is the MIDI buss number that is needed to drive Yoshimi.

Also make sure that of the clock settings for those busses are "Off".

The next instruction still works, but it is easier to simply pass the option -bus 5 to Sequencer64 when starting it up.

Now open the file sequencer64/contrib/midi/b4uacuse-GM-format.midi in Sequencer64. For all of the patterns (slots) that have lots of data in them, right click on the pattern and select *Midi Bus / [5] 129:0 (input)* and the desired channel number. (Doesn't matter much, just use up the lower channel numbers first).

Back in Yoshimi, select each Part corresponding to the channels you selected. Make sure *Enabled* is checked for each desired channel.

Back in Sequencer64, click on each pattern you want to hear, which highlights them in black. Now click the play button (green triangle). The song should play, with each part using the "Simple Sound". Not too bad for a bunch of sine waves, eh?

Now we can test the application more fully. Note that the instructions here are very light. Detailed instructions on the usage of Sequencer64 can be found in the following project, which contains a PDF file and the LaTeX code used to build it:

```
https://github.com/ahlstromcj/sequencer64-doc.git
```

Although it applies to an earlier version of the project, it still mostly holds true for Sequencer64.

4.3 Tests in the Patterns Window

The Patterns window is the inside portion of the main window, supported by the mainwid class. it contains a grid of boxes or slots, with each slot potentially containing a pattern, sequence, or track. Empty tracks (i.e. tracks that contain no events, like title-only tracks) are highlighted in yellow.

This window supports only a single variant of mouse-handling.

4.3.1 Button Clicks on a Pattern

A left-click on a pattern slot should cause the following to happen:

- 1. The pattern will be highlighted (white on a black background). This won't occur until the button is released.
- 2. During playback, the pattern will emit MIDI events and play its sequence.
- 3. If the pattern is dragged to another slot, whether playing is in progress or not, releasing the button in the destination slot will move the pattern to that slot.

A right-click on a pattern slot should cause the following to happen:

- 1. If the pattern is empty, then a pop-up menu to make a New pattern, paste a pattern, or make other selections will appear.
- 2. If the pattern is active, then a pop-up menu to Edit the pattern or make other selections will appear.
- 3. A second right-click, just off the menu, will dismiss the menu.

4.3.2 Patterns Window Key Shortcuts

First, note the selection of the File / Options / Keyboard / Show keys option. The tests here should work whether or not it is selected. The only difference is if the keys are shown.

We got a segfault during this test, when we weren't being systematic about it.

4.3.3 The Sequencer64 User File

To be discussed.

4.4 Tests Using Valgrind

Valgrind is a very useful tool for unearthing memory issues and other issues in an application, especially when one has the source code and can build the code with debugging information.

One runs the application from the command line, preceding its command line with valgrind and some of its options.

4.4.1 Valgrind Suppressions

One problem with valgrind is that it also uncovers errors in system libraries that one has no control over. These errors clutter the output, so we suppress them using a valgrind "suppressions" file. Here's how to create one:

```
$ valgrind --gen-suppressions=yes --log-file=val.supp ./Sequencer64/sequencer64
$ valgrind --gen-suppressions=all --log-file=val.supp ./Sequencer64/sequencer64
```

As the program runs, one is asked to print a suppression. If the error is due to a system or third-party library, answer "Y return", and then copy-and-paste the suppression to a file, giving it a name. For example, we provide a file contrib/seq64.supp containing suppressions of errors that annoy us. There are way too many "errors" in ALSA, GTK+, gtkmm, glibc, and more.

The second command collects all the suppressions. Passing the val.supp file through sed makes it immediately usable:

```
$ sed -i -e /^==/g val.supp
```

Running valgrind like this then shows mostly the errors we care about:

```
$ valgrind --suppressions=val.supp ./Sequencer64/sequencer64
```

We've added some other suppression files to the contrib directory. Too much! For example:

```
https://github.com/dtrebbien/GNOME.supp
```

However, overall this process is very painful, and we're going to eventually do all the valgrind work on the unit-test project for Sequencer64:

```
https://github.com/ahlstromcj/seq64-tests
```

4.4.2 Full Valgrind Leak-Checking

Here's how to capture errors, while suppressing the system errors and while generating a log file:

```
$ valgrind --suppressions=contrib/seq64.supp --leak-check=full \
    --track-origins=yes --log-file=valgrind.log --show-leak-kinds=all \
    ./Sequencer64/sequencer64
```

The errors can be also be re-routed to a log-file via the "2> valgrind.log" shell redirection.

Another idea is to precede the valgrind command with the following construct:

```
$ G_SLICE=debug-blocks valgrind ...
```

G_SLICE=debug-blocks will turn off gtk's advanced memory management to allow valgrind to show correct results. This results in an amazing plethora or invalid read and invalid write errors in GNOME-related libraries. Sheesh!

And don't forget about Valgrind's "massif" memory-tracking tool! (More to come!)

4.4.2.1 Leak-Checking Basic Operation

For the first pass, just run Sequencer64, then immediately exit. Then scan the log file to see if any "errors" can be pinpointed to the application and library code.

Don't forget to run the same scenario without valgrind, in a console window, to see if any of our own debug/problem output occurs.

In any case, leakage tagged as "still reachable" isn't as bad as leakage tagged as "definitely lost" or "indirectly lost".

But good luck finding a Sequencer64 bug buried in the chaff of 3rd-party valgrind reports, even with some suppressions enabled. Apparently a lot of them have to do with data structures that are intended to last the full life of the application.

One can make the search a little easier by searching for the "seq64" namespace in the valgrind log.

4.5 Specific Fault Debugging

This section goes through specific debugging cases we encountered. They should be part of the regular testing of Sequencer64.

4.6 Snipping of a MIDI file.

In order to have a test file for the *seq64-tests* project, we loaded up the b4uacuse-GM-format.midi file, removed all but four of the tracks, and saved it as b4uacuse-snipped.midi. Loading this file into Sequencer64 caused the following:

```
$ ./Sequencer64/sequencer64
[Reading user configuration /home/ahlstrom/.config/sequencer64/sequencer64.usr]
[Reading rc configuration /home/ahlstrom/.config/sequencer64/sequencer64.rc]
get_sequence(): m_seqs[4] not null
Segmentation fault
```

First step, fire up a debugger and see what happened. We use cgdb, a text-based front-end for gdb with a "vi" feel.

```
$ cgdb ./Sequencer64/sequencer64
```

Just hit "r", do File / Open, navigate to b4uacuse-snipped.midi, select it, and watch what happens.

The "bt" (backtrace) command shows a pretty large stack, 52 items. Page up to the top of the stack, and select frame 1 ("fr 1"). This shows a mutex at a very low address, 0x650! Frame 2 shows we are in the automutex constructor, calling lock() on that same badly-located mutex. Frame 3 is in sequence::event_count(), same bad mutex, and the m_events member is at address 0x0. Obviously, we're dealing with an unallocated sequence.

Frame 4 is in mainwid::draw_sequence_on_pixmap(), just after we've retrieved the next sequence via perform ::get_sequence(4). But that would be the fifth sequence (the sequence numbers start at 0), and we snipped all but 4 from the file before we saved it.

So, one thing we need to do is *check* the value returned by get_sequence() before we try to use it. The other thing to do is figure out how we got to the fifth sequence, and fix that code as well. Using the command "p perf(). \leftarrow sequence_count()", we verify that there are indeed only 4 sequences allocated.

Frame 5 is in mainwid::draw_sequences_on_pixmap(). That function tries to load all sequences on the current screen-set, from 0 to 31, without checking to see how many their actually are. Inefficient and dangerous.

Frame 6 is in mainwid::reset(). We could pass perf().sequence_count() here for checking, or get it in mainwid ::draw_sequences_on_pixmap().

Before we fix this issue, we need to load a file that works, to see why it does not fail for most files. We will put a breakpoint at the top mainwid::draw_sequences_on_pixmap().

We hit the breakpoint before even loading a file, with a sequence_count() of 0. The call to valid_sequence(0) passes the test. We may want to make valid_sequence() take the sequence_count() into account. But the call to perf().is_active(0) prevents anything bad from happening at startup time.

Once we load a good file, the sequence_count() is 14 in mainwid::draw_sequences_on_pixmap(). We turn on the display of "offset" using the command "display offset", and "c" (for "continue") until offset = 14, which means we are beyond that last sequence. That bad access is prevented by perf().is active(14).

So the fundamental problem is that perf().is_active(4) is not protecting the access when we load the "bad file". We need to find and fix that issue before papering over the problem with better access checks.

Start again, putting a breakpoint in the call to "new sequence(m_ppqn)" in midifile. This call sets up some members and clears the list of 256 playing notes. Add another breakpoint at "a_perf.add_sequence()" to see what's happening there.

What we find is that the first two tracks have proper sequence numbers as read from the MIDI file, 0 and 1. But the third one preserves the number from the old file, 4. We have a disjunction between the track number and the sequence number, a conceptual problem. We can leave it as is, and beef up the error-checking, or replace the sequence number with the track number when loading the file. What to do?

- Make sure that the is-active flag for all sequences is "false", that the pointers are always null, and make sure to test both of these items (depending on context) before doing anything with the sequence.
- Convert the sequence number to the track number upon saving the MIDI file, or upon reading the MIDI file, and use that number when adding the sequence to the perform object. This might affect some seq24/sequencer64 functionality, however. It's big move.

We need information on reading and importing.

First, if we look at a file that we created long ago by importing b4uacuse.mid, b4uacuse-GM-format. \leftarrow midi, it has its fourteen sequence numbers identical to their track numbers. No problem.

Second, if we just read b4uacuse.mid, a non-seq24-created MIDI file, we see that each of its tracks have no sequence number – they are all zero. The perform::add_sequence() simple iterates from the beginning of m_seqs[] until it finds an inactive m_seqs[i], and uses that element to hold the sequence pointer.

But now it also segfaults! Let's fix all the non-checked get_sequence() calls right away, it is too big an issue to ignore.

In the end, we have to be aware that a screen-set can have blank (null) slots interspersed amongst the active slots.

Chapter 5

Speed Issue of Sequencer64

Author(s) Chris Ahlstrom 2017-03-27

5.1 Introduction

This section describes some speed issues of Sequencer64. Early on in our reboot of *seq24*, we noticed that some of our larger files took a noticeable time to load. It was only a few seconds, but seemed like a long time for such small files. We fixed the issue using an alternate container implementation.

We recently added a large MIDI file to test, b4uacuse-stress.midi, and all hell broke loose.

5.2 Initial Change of Containers

In the beginning, we noticed that the MIDI container implementation used the std::list container, and also that it called std::list::sort() after each event was added to the container.

Our first thought was to replace the std::list with an std::multimap. Insertions into this container are made in the appropriate location (rather than at the end), and so are automatically sorted. We kept the old code around, but enabled the new multimap code via the SEQ64_USE_EVENT_MAP macro. This decreased the time of loading.

(It also exposed a small number of bugs that users of Sequencer64 discovered and fixed.)

At the back of our minds was the possibility that the longer time needed to increment a multimap iterator versus a list iterator would prove to limit the amount of data that could be played back. Once we finally created a large file, b4uacuse-stress.midi, a 1.5 Mb file, we experienced the limitations of that iterator during playback. On our main development laptop, a near-gaming Intel i7 machine, there were minor artifacts in playback. On our old single-core laptop with 32-bit Debian installed, the sequence would not play, and would continually and visibly refresh the main window display.

5.3 Back to the Original Container

So then we re-enabled the old seq24 list implementation, and found that the time needed to load b4uacuse-stress.midi was over 6 minutes on our near-gamer laptop and 13 minutes on the single-core laptop.

So, we had to find a way to get the fast loading speed of the std::multimap and the faster speed of the std::multimap and the faster speed of the std::list container and stop sorting the container after every insert, when loading the MIDI file.

This worked, but had some side-effects that had to be fixed. We found that the <code>sequence::verify_and_</code> \leftarrow link function required that the container be sorted first, and so we had to find the places where that function was called, and make sure that the sorting had occurred.

Anyway, the current configuration for the usage of std::multimap versus std::list and the sorting of the MIDI container after every event insertion versus after all the events are now permanent and hardwired. The default is to not use the event multimap, and to not presort events. This makes loading fast, and playback able to handle more sequences. One can also try to use the multimap, or use the list with pre-sorting, if bugs appear when building the application with the default setting. However, we really want to get the post-sorted list implementation to work, to get fast loading speed and higher throughput at the same time.

The other options are available as a fallback in case one gets struck by bugs in the default, and can afford slower loads or less throughput.

5.4 What is Next, the Vector?

Or, what's your vector, Victor?

For playback, the std::vector iterator can be even faster than the std::list iterator, because the vector does not use stored pointers to the next element, and, since its storage layout is essentially like a standard C array, processor caching can add even more to the speed of access.

However, this change will require carefully analysis and even more careful work to correctly implement the change. We will log this as a future improvement.

Actually, now we have reverted back to the <code>std::list</code> implementation, with the key difference that, when loading a file, we do not sort until we have read all of the events. So we get a fast load time and higher maximum throughput during playback.

Now to test the hell out of the next version!

Chapter 6

Licenses

Library This application and its libraries, sub-applications, and documents.

Author(s) Chris Ahlstrom 2015-09-10

6.1 License Terms for the This Project.

Wherever the tag \$XPC_SUITE_GPL_LICENSE\$ appears, or wherever reference to the GPL licensing scheme (any version) is mentioned, substitute the appropriate license text, depending on whether the project is a library, application, documentation, or server software. We're not going to include paragraphs of licensing information in every module; you are responsible for coming here to read the licensing information.

These licenses apply to each sub-project and file artifact in the project with which this license description was packaged.

Wherever the term **XPC** is encountered in this project, it refers to my projects, which go beyond the package that contains this document.

6.2 XPC Application License

The **XPC** application license is either the **GNU GPLv2**. or the **GNU GPLv3**. Generally, projects that originate with me use the latter language, while projects I have extended may specify the former license.

Copyright (C) 2015-2015 by Chris Ahlstrom

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the

```
Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor Boston, MA 02110-1301, USA.
```

The text of the GNU GPL version 3 license can also be found here:

```
http://www.gnu.org/licenses/gpl-3.0.txt
```

28 Licenses

6.3 XPC Library License

The XPC library license is the GNU LGPLv3.

Copyright (C) 2015-2015 by Chris Ahlstrom

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Lesser Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to

```
Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor Boston, MA 02110-1301, USA.
```

The text of the GNU LGPL version 3 license can also be found here:

```
http://www.gnu.org/licenses/lgpl-3.0.txt
```

6.4 XPC Documentation License

The XPC documentation license is the GNU FDLv1.3.

Copyright (C) 2015-2015 by Chris Ahlstrom

This documentation is free documentation; you can redistribute it and/or modify it under the terms of the GNU Free Documentation License as published by the Free Software Foundation; either version 1.3 of the License, or (at your option) any later version.

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Free Documentation License for more details.

You should have received a copy of the GNU Free Documentation License along with this documentation; if not, write to the

```
Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor Boston, MA 02110-1301, USA.
```

The text of the GNU FDL version 1.3 license can also be found here:

```
http://www.gnu.org/licenses/fdl.txt
```

6.5 XPC Affero License 29

6.5 XPC Affero License

The XPC "Affero" license is the GNU AGPLv3.

Copyright (C) 2015-2015 by Chris Ahlstrom

This server software is free server software; you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation; either version 1.3 of the License, or (at your option) any later version.

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Free Documentation License for more details.

You should have received a copy of the GNU Affero General Public License along with this server software; if not, write to the

```
Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor Boston, MA 02110-1301, USA.
```

The text of the GNU AGPL version 3 license can also be found here:

```
http://www.gnu.org/licenses/agpl-3.0.txt
```

At the present time, no XPC project uses the "Affero" license.

6.6 XPC License Summary

Include one of these licenses in your Doxygen documentation with one of the following Doxygen tags specified above:

```
\ref gpl_license_subproject
\ref gpl_license_application
\ref gpl_license_library
\ref gpl_license_documentation
\ref gpl_license_affero
```

For more information on navigating GNU licensing, see this page:

```
http://www.gnu.org/licenses/
```

Copies of these licenses (and some logos) are provided in the licenses directory of the main project (or you can search for them at *gnu.org*).

30 Licenses

Chapter 7

Todo List

File calculations.cpp

There are additional user-interface and MIDI scaling variables in the perfroll module that we need to move here.

File perfnames.cpp

When bringing up this dialog, and starting play from it, some extra horizontal lines are drawn for some of the sequences. This happens even in seq24, so this is long standing behavior. Is it useful, and how? Where is it done? In perfroll?

Global seg64::editable events::save events ()

Consider what to do about the sequence::m_is_modified flag.

Global seq64::eventedit::handle_save ()

Could also support writing the events to a new sequence, for added flexibility.

Global seq64::mainwid::timeout ()

We should use this callback to display the current time in the playback.

Global seq64::mainwnd::mainwnd (perform &p, bool allowperf2=true, int ppqn=SEQ64_USE_DEFAULT_← PPQN)

Offload most of the work into an initialization function like options does; make the perform parameter a reference; valgrind flags m_tooltips as lost data, but if we try to manage it ourselves, many more leaks occur.

Global seq64::mainwnd::on_key_release_event (GdkEventKey *ev)

Test this functionality in old and new application.

Global seq64::midi_jack::api_sysex (event *e24)

Flesh out this routine.

Global seq64::perfedit::perfedit (perform &p, bool second_perfedit=false, int ppqn=SEQ64_USE_DEFAU← LT_PPQN)

Offload most of the work into an initialization function like options does.

Global seq64::perform::add_sequence (sequence *seq, int perf)

Shouldn't we wrap around the sequence list if we can't find an empty sequence slot after prefnum?

This function needs some deeper analysis against the original, in my opinion.

Global seq64::perform::launch (int ppqn)

We probably need a bpm parameter for consistency at some point.

Global seq64::perform::m_seqs [c_max_sequence]

First, make the sequence array a vector, and second, put allof these flags into a structure and access those members indirectly.

32 Todo List

Global seq64::perform::pop_trigger_undo()

Look at seq32/src/perform.cpp and the perform :: push_trigger_undo(track) function, which has a track parameter that has a -1 values the supports all tracks. It requires two new vectors (one for undo, one for redo), two new flags (likewise). We've put this code in place, no longer macroed out, now permanent.

Global seq64::perform::set left tick (midipulse tick, bool setstart=true)

The perform::m_one_measure member is currently hardwired to PPQN * 4.

Global seq64::perfroll::set_ppqn (int ppqn)

Resolve the issue of c_perf_scale_x versus m_perf_scale_x in perfroll.

Global seq64::perftime::set_ppqn (int ppqn)

We need make the 4 constant variable per the number of beats (quarter-notes) per bar, and also at least make 16 (4x4) a meaningful manifest constant.

Global seq64::pulses to string (midipulse p)

Still needs to be unit tested.

Global seg64::pulses to timestring (midipulse p, const midi timing &timinginfo)

Still needs to be unit tested.

Global seq64::seqdata::on_scroll_event (GdkEventScroll *ev)

DOCUMENT the segdata scrolling behavior in the documentation projects.

Global seq64::seqedit::get_measures ()

Create a sequence::set units() function or a sequence::get measures() function to forward to.

Global seq64::seqedit::seqedit (perform &perf, sequence &seq, int pos, int ppqn=SEQ64_USE_DEFAULT → PPQN)

Offload most of the work into an initialization function like options does.

Global seq64::seqedit::set_beat_width (int bw)

Check if verification is needed at this point.

Global seq64::seqedit::set_beats_per_bar (int bpm)

Check if verification is needed at this point.

Global seq64::seqedit::set_measures (int lim)

Check if verification is needed at this point.

Global seq64::segmenu::m_modified

We need to make sure that the perform object is in control of the modification flag.

Global seq64::seqmenu::seq_copy ()

Can be offloaded to a perform member function that accepts a sequence clipboard non-const reference parameter.

Global seq64::seqmenu::seq_cut ()

A lot of seq_cut() can be offloaded to a (new) perform member function that takes a sequence clipboard non-const reference parameter.

Global seq64::seqmenu::seq_paste()

All of seq_paste() can be offloaded to a (new) perform member function with a const clipboard reference parameter.

Global seq64::seqtime::update_pixmap()

Sizing needs to be controlled by font parameters. Instead of 19 or 20, estimate the width of 3 letters. Instead of 9 pixels down, use the height of the seqtime and the height of a character.

Global seq64::sequence::add_chord (int chord, midipulse tick, midipulse len, int note)

Add the ability to preserve the incoming velocity.

Global seg64::sequence::get minmax note events (int &lowest, int &highest)

For efficency, we should calculate this only when the event set changes, and save the results and return them if good.

Global seq64::sequence::stream_event (event &ev)

When we feel like debugging, we will replace the global is-playing call with the parent perform's is-running call.

Global seq64::triggers::next (midipulse *tick_on, midipulse *tick_off, bool *selected, midipulse *tick_← offset)

It would be a bit simpler to simply return a trigger object, wouldn't it?

34 Todo List

Chapter 8

Deprecated List

Global seq64::clock_tick_duration_bogus (midibpm bpm, int ppqn)

This is a somewhat bogus calculation used only for "statistical" output in the old perform module. Name changed to reflect this unfortunate fact. Use pulse_length_us() instead.

Global seq64::sequence::get_name () const

36 Deprecated List

Chapter 9

Namespace Index

9.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Gtk		 	 	 	49
seq64					
	Define this macro to use the new seq24 v	 	 	 	49

38 Namespace Index

Chapter 10

Hierarchical Index

10.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

seq64::AbstractPerfInput	33
seg64::FruityPerfInput	63
seg64::Seg24PerfInput	51
seg64::automutex	
seq64::busarray	
seq64::businfo	
seq64::click	
seq64::configfile	
seg64::optionsfile	
seq64::userfile	
·	10
Dialog	۰,
seq64::options	36
DrawingArea seg64::qui palette qtk2	۰,
1 0 1 10	
seq64::gui_drawingarea_gtk2	
seq64::eventslots	
seq64::maintime	
seq64::mainwid	
seq64::perfnames	
seq64::perfroll	
seq64::perftime	
seq64::seqdata	
seq64::seqevent	
seq64::seqkeys	
seq64::seqroll	
·	
seq64::editable_events	78
Entry 2000 Autoubin donton	4 -
seq64::keybindentry	
seq64::event	
seq64::editable_event	
seq64::event_list::event_key	
seq64::event_list	11
exception	

40 Hierarchical Index

seq64::rterror	18
seq64::font	
seq64::FruitySeqEventInput	
seq64::FruitySeqRollInput	
seq64::gui_assistant	
seq64::gui_assistant_gtk2	
seq64::jack_assistant 31 seq64::jack_scratchpad 33	
seq64::jack_status_pair_t	
seq64::keys_perform	
seq64::keys_perform_gtk2	
seq64::keys perform transfer	
seq64::keystroke	
seq64::lash	36
seq64::mastermidibase	35
seq64::mastermidibus	
seq64::mastermidibus	
seq64::midi_container	
seq64::midi_list	
seq64::midi_vector	
seq64::midi_control	
seq64::midi_info	
seq64::midi_alsa_info	
seq64::midi_jack_info	
seq64::midi_jack_data 53 seq64::midi measures 54	
seq64::midi_message	
seq64::midi_port_info	
seq64::midi_queue	64
seq64::midi_splitter	
seq64::midi_timing	
seq64::midibase	
seq64::midi_api	
seq64::midi_alsa	
seq64::midi_m_aisa	
seq64::midi_jack	
seq64::midi in jack	
seq64::midi_out_jack	
seq64::rtmidi	21
seq64::rtmidi_in	
seq64::rtmidi_out	
seq64::midibus	
seq64::midibus	
seq64::mutex	
seq64::condition_var	
seq64::editable event::name value t	
ObjectBase	١,
seq64::seqmenu	29
seq64::mainwid	00
seq64::perfnames	
seq64::perform	
seq64::performcallback	
seq64::mainwnd	15

10.1 Class Hierarchy 41

seq64::midi_port_info::port_info_t
seq64::rc_settings
seq64::rect
seq64::gui_drawingarea_gtk2::rect
seq64::rtmidi_in_data
seq64::rtmidi_info
seq64::Seq24SeqEventInput
seq64::sequence
seq64::trigger
seq64::triggers
seq64::user_instrument
seq64::user_instrument_t
seq64::user_midi_bus
seq64::user_midi_bus_t
seq64::user_settings
Window
seq64::gui_window_gtk2
seq64::eventedit
seq64::Ifownd
seq64::mainwnd
seq64::perfedit
seq64::seqedit

42 Hierarchical Index

Chapter 11

Data Structure Index

11.1 Data Structures

Here are the data structures with brief descriptions:

seq64::AbstractPerfInput	
Provides an abstract base class to provide the minimal interface for the various "perf input"	
classes	133
seq64::automutex	
Provides a mutex that locks automatically when created, and unlocks when destroyed	137
seq64::busarray	
Holds a number of businfo objects	138
seq64::businfo	
A new class to consolidate a number of bus-related arrays into one array	148
seq64::click	
Encapsulates any possible mouse click	154
seq64::condition_var	4.50
A mutex works best in conjunction with a condition variable	159
seq64::configfile	101
This class is the abstract base class for optionsfile and userfile	161
seq64::editable_event Provides for the management of MIDI editable events	166
seq64::editable events	100
Provides for the management of an ordered collection MIDI editable events	179
seq64::event	170
Provides events for management of MIDI events	188
seq64::event_list::event_key	
Provides a key value for an event map	209
seq64::event_list	
Receptable for MIDI events	211
seq64::eventedit	
This class supports an Event Editor that is used to tweak the details of events and get a better	
idea of the mix of events in a sequence	223
seq64::eventslots	
This class implements the left-side list of events in the pattern event-edit window	239
seq64::font	
This class provides a wrapper for rendering fonts that are encoded as a 16 x 16 pixmap file in	
XPM format	258
seq64::FruityPerfInput	
Implements the performance input of that certain fruity sequencer that people seem to like	263

44 Data Structure Index

seq64::FruitySeqEventInput	
This structure implements the interaction methods for the "fruity" mode of operation	269
	272
seq64::gui_assistant	
	274
seq64::gui_assistant_gtk2 This class provides an interface for some of the Gtk/Gdk/Glib support needed in Sequencer64	277
seq64::gui_drawingarea_gtk2	
	280
seq64::gui_palette_gtk2	200
Implements a stock palette of Gdk::Color elements	298
	310
This class supports a basic interface for Gtk::Window-derived objects seq64::jack_assistant	310
	314
seq64::jack_scratchpad	314
Provide a temporary structure for passing data and results between a perform and jack_assistant	
object	338
seq64::jack status pair t	330
Provides an internal type to make it easier to display a specific and accurate human-readable	
message when a JACK operation fails	240
-	340
seq64::keybindentry	0.44
3 11 , 3	341
seq64::keys_perform	244
This class supports the performance mode	344
seq64::keys_perform_gtk2	074
	374
seq64::keys_perform_transfer	
Provides a data-transfer structure to make it easier to fill in a keys_perform object's members	077
using sscanf()	377
seq64::keystroke	001
	381
seq64::lash This class supports I ASII energiage if compiled with I ASII support (i.e.	200
	386
seq64::lfownd	000
	390
seq64::maintime	
This class provides the drawing of the progress bar at the top of the main window, along with two	205
"pills" that move in time with the beat and measure	395
seq64::mainwid	400
This class implements the piano roll area of the application	400
seq64::mainwnd	
This class implements the functionality of the main window of the application, except for the	445
<i>7</i> , 1	415
seq64::mastermidibase	405
,	435
seq64::mastermidibus	450
· · ·	456
seq64::midi_alsa	
	465
seq64::midi_alsa_info	
	475
seq64::midi_api	
Subclasses of midi_in_api and midi_out_api contain all API- and OS-specific code necessary to	
fully implement the rtmidi API	481
seq64::midi_container	
This class is the abstract base class for a container of MIDI track information	491

11.1 Data Structures 45

seq64::midi_control	
This class (formerly a struct) contains the control information for sequences that make up a live set	500
seq64::midi_in_alsa	
This class implements the ALSA version of a MIDI input object	505
seq64::midi_in_jack The class for handling JACK MIDI input	507
·	507
seq64::midi_info The class for holding basic information on the MIDI input and output ports currently present in	
the system	511
seq64::midi_jack This class implements with JACK version of the midi_alsa object	522
seq64::midi_jack_data	
Contains the JACK MIDI API data as a kind of scratchpad for this object	535
seq64::midi_jack_info	
The class for handling JACK MIDI port enumeration	538
seq64::midi_list	
This class is the std::list implementation of the midi_container seq64::midi_measures	545
Provides a data structure to hold the numeric equivalent of the measures string "measures.	
:beats:divisions" ("m:b:d")	549
seq64::midi message	
Provides a handy capsule for a MIDI message, based on the std::vector <unsigned char=""> data</unsigned>	
type from the RtMidi project	552
seq64::midi_out_alsa	
This class implements the ALSA version of a MIDI output object	555
seq64::midi_out_jack	
The JACK MIDI output API class	557
seq64::midi_port_info	
A class for holding port information	560
seq64::midi_queue	
Provides a queue of midi_message structures	564
seq64::midi_splitter	
This class handles the parsing and writing of MIDI files	567
seq64::midi_timing	
We anticipate the need to have a small structure holding the parameters needed to calculate	
MIDI times within an arbitrary song	573
seq64::midi_vector	
This class is the std::vector implementation of the midi_container	5//
seq64::midibase	E04
This class implements with ALSA version of the midibase object	581
seq64::midibus This class implements with ALSA version of the midibus object	603
seq64::midifile	003
This class handles the parsing and writing of MIDI files	615
seq64::mutex	013
Simple wrapper for the pthread_mutex_t type used as a recursive mutex	636
seq64::editable event::name value t	000
Provides a type that contains the pair of values needed for the various lookup maps that are	
needed to manage editable events	637
seq64::options	
This class supports a full tabbed options dialog	638
seq64::optionsfile	0.45
Provides a file for reading and writing the application' main configuration file	645
seq64::perfedit	
This class supports a Performance Editor that is used to arrange the patterns/sequences defined in the patterns panel	649
iii uie patteriio pariei	043

46 Data Structure Index

seq64::perfnames	
This class implements the left-side keyboard in the patterns window	667
seq64::perform This class supports the performance mode	675
seq64::performcallback Provides for notification of events	761
seq64::perfroll This class implements the performance roll user interface	763
seq64::perftime	
This class implements drawing the piano time at the top of the "performance window" (the "song editor")	783
seq64::midi_port_info::port_info_t	
Hold the information for a single port	792
seq64::rc_settings This class contains the options formerly named "global_xxxxxx"	794
seq64::rect	, , ,
A small helper class representing a rectangle	816
A small helper structure representing a rectangle	817
seq64::rterror	
Exception handling class for rtexmidi	818
The main class of the rtmidi API	821
seq64::rtmidi_in	
A realtime MIDI input class	829
seq64::rtmidi_in_data The rtmidi_in_data structure is used to pass private class data to the MIDI input handling function	
or thread	833
seq64::rtmidi_info	
A class for enumerating MIDI clients and ports	838
seq64::rtmidi_out A realtime MIDI output class	849
seq64::Seq24PerfInput	
Implements the default (Seq24) performance input characteristics of this application seq64::Seq24SeqEventInput	851
This structure implement the normal interaction methods for Seq24	855
seq64::seqdata This class supports drawing piano-roll eventis on a window	857
seq64::seqedit	
Implements the Pattern Editor, which has references to:	870
Implements the piano event drawing area	903
seq64::seqkeys	
This class implements the left side piano of the pattern/sequence editor	917
seq64::seqmenu This class handles the right-click menu of the sequence slots in the pattern window	929
seq64::seqroll	020
Implements the piano roll section of the pattern editor	942
seq64::seqtime	075
This class implements the piano time, whatever that is	975
Firstly a receptable for a single track of MIDI data read from a MIDI file or edited into a pattern	981
seq64::trigger This class hold a single trigger for a sequence object	1052
seq64::triggers	
Receptable the triggers that can be used with a sequence object	1056
seq64::user_instrument Provides data about the MIDI instruments, readable from the "user" configuration file	1073

11.1 Data Structures 47

seq64::user_instrument_t
This structure corresponds to [user-instrument-N] definitions in the \sim /.seq24usr
<pre>or ~/.config/sequencer64/sequencer64.usr file</pre>
seq64::user_midi_bus
Provides data about the MIDI busses, readable from the "user" configuration file 1078
seq64::user_midi_bus_t
This structure corresponds to [user-midi-bus-0] definitions in the \sim /.seq24usr
("user") file (\sim /.config/sequencer64/sequencer64.usr in the latest version of the
application)
seq64::user_settings
Holds the current values of sequence settings and settings that can modify the number of se-
quences and the configuration of the user-interface
seq64::userfile
Supports the user's \sim /.config/sequencer64/sequencer64.usr and \sim /.seq24usr
configuration file

48 Data Structure Index

Chapter 12

Namespace Documentation

12.1 Gtk Namespace Reference

12.2 seq64 Namespace Reference

Define this macro to use the new seq24 v.

Data Structures

class AbstractPerfInput

Provides an abstract base class to provide the minimal interface for the various "perf input" classes.

· class automutex

Provides a mutex that locks automatically when created, and unlocks when destroyed.

class busarray

Holds a number of businfo objects.

· class businfo

A new class to consolidate a number of bus-related arrays into one array.

· class click

Encapsulates any possible mouse click.

class condition_var

A mutex works best in conjunction with a condition variable.

· class configfile

This class is the abstract base class for optionsfile and userfile.

· class editable_event

Provides for the management of MIDI editable events.

class editable_events

Provides for the management of an ordered collection MIDI editable events.

· class event

Provides events for management of MIDI events.

class event_list

The event_list class is a receptable for MIDI events.

· class eventedit

This class supports an Event Editor that is used to tweak the details of events and get a better idea of the mix of events in a sequence.

· class eventslots

This class implements the left-side list of events in the pattern event-edit window.

class font

This class provides a wrapper for rendering fonts that are encoded as a 16 x 16 pixmap file in XPM format.

class FruityPerfInput

Implements the performance input of that certain fruity sequencer that people seem to like.

struct FruitySeqEventInput

This structure implements the interaction methods for the "fruity" mode of operation.

class FruitySeqRollInput

Implements the fruity mouse interaction paradigm for the segroll.

· class gui assistant

This class provides an interface for some of the GUI support needed in Sequencer64.

class gui_assistant_gtk2

This class provides an interface for some of the Gtk/Gdk/Glib support needed in Sequencer64.

• class gui_drawingarea_gtk2

Implements the basic drawing areas of the application.

class gui_palette_gtk2

Implements a stock palette of Gdk::Color elements.

class gui_window_gtk2

This class supports a basic interface for Gtk::Window-derived objects.

· class jack assistant

This class provides the performance mode JACK support.

class jack_scratchpad

Provide a temporary structure for passing data and results between a perform and jack_assistant object.

struct jack_status_pair_t

Provides an internal type to make it easier to display a specific and accurate human-readable message when a JACK operation fails.

· class keybindentry

Class for management of application key-bindings.

class keys_perform

This class supports the performance mode.

class keys_perform_gtk2

This class supports the performance mode.

struct keys_perform_transfer

Provides a data-transfer structure to make it easier to fill in a keys_perform object's members using sscanf().

· class keystroke

Encapsulates any practical keystroke.

· class lash

This class supports LASH operations, if compiled with LASH support (i.e.

· class Ifownd

One LFO window class.

class maintime

This class provides the drawing of the progress bar at the top of the main window, along with two "pills" that move in time with the beat and measure.

· class mainwid

This class implements the piano roll area of the application.

class mainwnd

This class implements the functionality of the main window of the application, except for the Patterns Panel functionality, which is implemented in the mainwid class.

· class mastermidibase

The class that "supervises" all of the midibus objects?

class mastermidibus

The class that "supervises" all of the midibus objects.

· class midi alsa

This class implements the ALSA version of the midi_api.

class midi_alsa_info

The class for handling ALSA MIDI input.

· class midi api

Subclasses of midi_in_api and midi_out_api contain all API- and OS-specific code necessary to fully implement the rtmidi API.

· class midi container

This class is the abstract base class for a container of MIDI track information.

· class midi control

This class (formerly a struct) contains the control information for sequences that make up a live set.

· class midi_in_alsa

This class implements the ALSA version of a MIDI input object.

· class midi in jack

The class for handling JACK MIDI input.

· class midi info

The class for holding basic information on the MIDI input and output ports currently present in the system.

· class midi jack

This class implements with JACK version of the midi_alsa object.

· struct midi jack data

Contains the JACK MIDI API data as a kind of scratchpad for this object.

class midi_jack_info

The class for handling JACK MIDI port enumeration.

class midi_list

This class is the std::list implementation of the midi container.

· class midi measures

Provides a data structure to hold the numeric equivalent of the measures string "measures:beats:divisions" ("m:b:d").

· class midi message

Provides a handy capsule for a MIDI message, based on the std::vector<unsigned char> data type from the RtMidi project.

· class midi_out_alsa

This class implements the ALSA version of a MIDI output object.

class midi_out_jack

The JACK MIDI output API class.

· class midi_port_info

A class for holding port information.

· class midi_queue

Provides a queue of midi_message structures.

· class midi_splitter

This class handles the parsing and writing of MIDI files.

class midi_timing

We anticipate the need to have a small structure holding the parameters needed to calculate MIDI times within an arbitrary song.

· class midi vector

This class is the std::vector implementation of the midi_container.

class midibase

This class implements with ALSA version of the midibase object.

· class midibus

This class implements with ALSA version of the midibus object.

· class midifile

This class handles the parsing and writing of MIDI files.

· class mutex

The mutex class provides a simple wrapper for the pthread_mutex_t type used as a recursive mutex.

· class options

This class supports a full tabbed options dialog.

· class optionsfile

Provides a file for reading and writing the application' main configuration file.

· class perfedit

This class supports a Performance Editor that is used to arrange the patterns/sequences defined in the patterns panel.

class perfnames

This class implements the left-side keyboard in the patterns window.

class perform

This class supports the performance mode.

struct performcallback

Provides for notification of events.

· class perfroll

This class implements the performance roll user interface.

· class perftime

This class implements drawing the piano time at the top of the "performance window" (the "song editor").

· class rc settings

This class contains the options formerly named "global_xxxxxx".

· class rect

A small helper class representing a rectangle.

· class rterror

Exception handling class for rtexmidi.

· class rtmidi

The main class of the rtmidi API.

• class rtmidi_in

A realtime MIDI input class.

• class rtmidi_in_data

The rtmidi_in_data structure is used to pass private class data to the MIDI input handling function or thread.

· class rtmidi info

A class for enumerating MIDI clients and ports.

class rtmidi_out

A realtime MIDI output class.

class Seq24PerfInput

Implements the default (Seq24) performance input characteristics of this application.

struct Seq24SeqEventInput

This structure implement the normal interaction methods for Seq24.

· class segdata

This class supports drawing piano-roll eventis on a window.

· class seqedit

Implements the Pattern Editor, which has references to:

· class segevent

Implements the piano event drawing area.

· class segkeys

This class implements the left side piano of the pattern/sequence editor.

class segmenu

This class handles the right-click menu of the sequence slots in the pattern window.

class segroll

Implements the piano roll section of the pattern editor.

· class segtime

This class implements the piano time, whatever that is.

· class sequence

The sequence class is firstly a receptable for a single track of MIDI data read from a MIDI file or edited into a pattern.

· class trigger

This class hold a single trigger for a sequence object.

· class triggers

The triggers class is a receptable the triggers that can be used with a sequence object.

class user_instrument

Provides data about the MIDI instruments, readable from the "user" configuration file.

struct user_instrument_t

This structure corresponds to [user-instrument-N] definitions in the \sim /.seq24usr or \sim /.config/sequencer64/sequencer file.

· class user midi bus

Provides data about the MIDI busses, readable from the "user" configuration file.

struct user_midi_bus_t

This structure corresponds to [user-midi-bus-0] definitions in the \sim /.seq24usr ("user") file (\sim /.config/sequencer64/sequencer64.usr in the latest version of the application).

· class user settings

Holds the current values of sequence settings and settings that can modify the number of sequences and the configuration of the user-interface.

· class userfile

Supports the user's \sim /.config/sequencer64/sequencer64.usr and \sim /.seq24usr configuration file.

Typedefs

· typedef unsigned char midibyte

Provides a fairly common type definition for a byte value.

• typedef unsigned char bussbyte

Distinguishes a buss/bus number from other MIDI bytes.

· typedef unsigned short midishort

Distinguishes a short value from the unsigned short values implicit in short-valued MIDI numbers.

· typedef unsigned long midilong

Distinguishes a long value from the unsigned long values implicit in long-valued MIDI numbers.

· typedef long midipulse

Distinguishes a long value from the unsigned long values implicit in MIDI time measurements.

typedef double midibpm

Provides the data type for BPM (beats per minute) values.

typedef void(* rterror_callback) (rterror::Type type, const std::string &errormsg, void *userdata)

rtmidi error callback function prototype.

typedef void(* rtmidi callback t) (midi message &message, void *userdata)

MIDI caller callback function type definition.

Enumerations

```
enum wave type t {
 WAVE NONE.
 WAVE_SINE,
 WAVE_SAWTOOTH,
 WAVE REVERSE SAWTOOTH,
 WAVE TRIANGLE }
    Provides a clear enumation of wave types supported by the wave function.
enum seq_modifier_t {
 SEQ64 NO MASK,
 SEQ64_SHIFT_MASK,
 SEQ64_LOCK_MASK,
 SEQ64_CONTROL_MASK,
 SEQ64 MOD1 MASK,
 SEQ64_MOD2_MASK,
 SEQ64_MOD3_MASK,
 SEQ64 MOD4 MASK,
 SEQ64 MOD5 MASK,
 SEQ64 BUTTON1 MASK,
 SEQ64 BUTTON2 MASK,
 SEQ64_BUTTON3_MASK,
 SEQ64 BUTTON4 MASK,
 SEQ64_BUTTON5_MASK,
 SEQ64_SUPER_MASK,
 SEQ64_HYPER_MASK,
 SEQ64_META_MASK,
 SEQ64_RELEASE_MASK,
 SEQ64_MASK_MAX }
    Types of modifiers, essentially copied from gtk-2.0/gdk/gdktypes.h.
enum seq_event_type_t {
 SEQ64_NOTHING,
 SEQ64_DELETE,
 SEQ64 DESTROY,
 SEQ64_EXPOSE,
 SEQ64_MOTION_NOTIFY,
 SEQ64_BUTTON_PRESS,
 SEQ64 2BUTTON PRESS,
 SEQ64_3BUTTON_PRESS,
 SEQ64_BUTTON_RELEASE,
 SEQ64_KEY_PRESS,
 SEQ64 KEY RELEASE,
 SEQ64_SCROLL,
 SEQ64 EVENT LAST }
    Event types copped from gtk-2.0/gdk/gdkevents.h for use with this application.
enum seq_scroll_direction_t {
 SEQ64_SCROLL_UP,
 SEQ64_SCROLL_DOWN,
 SEQ64_SCROLL_LEFT,
 SEQ64 SCROLL RIGHT }
     Types of scroll events, essentially copied from gtk-2.0/gdk/gdkevents.h.
enum clock_e {
 e_clock_off,
 e_clock_pos,
 e_clock_mod }
    A clock enumeration, as used in the File / Options / MIDI Clock dialog.
```

```
enum interaction_method_t {
 e_seq24_interaction,
 e_fruity_interaction,
 e_number_of_interactions }
     Provides codes for the mouse-handling used by the application.
• enum c_music_scales {
 c_scale_off,
 c_scale_major,
 c_scale_minor,
 c_scale_harmonic_minor,
 c_scale_melodic_minor,
 c_scale_c_whole_tone,
 c_scale_blues,
 c_scale_major_pentatonic,
 c_scale_minor_pentatonic,
 c scale size }
     Corresponds to the small number of musical scales that the application can handle.
enum draw_type_t {
 DRAW_FIN,
 DRAW NORMAL LINKED,
 DRAW NOTE ON,
 DRAW NOTE OFF }
     Provides a set of methods for drawing certain items.
enum mouse_action_e {
 e_action_select,
 e_action_draw,
 e_action_grow }
     Mouse actions, for the Pattern Editor.
enum edit_action_t {
 c_select_all_notes,
 c_select_all_events,
 c_select_inverse_notes,
 c_select_inverse_events,
 c_quantize_notes,
 c_quantize_events,
 c_tighten_events,
 c tighten notes,
 c transpose notes,
 c_reserved,
 c_transpose_h,
 c expand pattern,
 c_compress_pattern,
 c_select_even_notes,
 c_select_odd_notes,
 c_swing_notes }
     Actions.
enum rtmidi_api {
 RTMIDI API UNSPECIFIED,
 RTMIDI_API_LINUX_ALSA,
 RTMIDI_API_UNIX_JACK,
 RTMIDI_API_MAXIMUM }
     MIDI API specifier arguments.
```

Functions

• void swap (busarray &buses0, busarray &buses1)

This free function swaps the contents of two busarray objects.

std::string wave_type_name (wave_type_t wavetype)

Converts a wave type value to a string.

bool extract_timing_numbers (const std::string &s, std::string &part_1, std::string &part_2, std::string &part ← __3, std::string &fraction)

Extracts up to 4 numbers from a colon-delimited string.

std::string pulses to string (midipulse p)

Converts MIDI pulses (also known as ticks, clocks, or divisions) into a string.

std::string pulses_to_measurestring (midipulse p, const midi_timing &seqparms)

Converts a MIDI pulse/ticks/clock value into a string that represents "measures:beats:ticks" ("measures:beats⇔:division").

• bool pulses to midi measures (midipulse p, const midi timing &segparms, midi measures &measures)

Converts a MIDI pulse/ticks/clock value into a string that represents "measures:beats:ticks" ("measures:beats⇔:division").

std::string pulses_to_timestring (midipulse p, midibpm bpm, int ppqn)

Converts a MIDI pulse/ticks/clock value into a string that represents "hours:minutes:seconds.fraction".

std::string pulses_to_timestring (midipulse p, const midi_timing &timinginfo)

Converts a MIDI pulse/ticks/clock value into a string that represents "hours:minutes:seconds.fraction".

midipulse measurestring_to_pulses (const std::string &measures, const midi_timing &seqparms)

Converts a string that represents "measures:beats:division" to a MIDI pulse/ticks/clock value.

midipulse midi_measures_to_pulses (const midi_measures &measures, const midi_timing &seqparms)

Converts a string that represents "measures:beats:division" to a MIDI pulse/ticks/clock value.

- midipulse timestring to pulses (const std::string ×tring, int bpm, int ppqn)
- midipulse string_to_pulses (const std::string &s, const midi_timing &mt)

Converts a time string to pulses.

midibyte string_to_midibyte (const std::string &s)

Converts a string to a MIDI byte.

std::string shorten_file_spec (const std::string &fpath, int leng)

Shortens a file-specification to make sure it is no longer than the provided length value.

bool string_not_void (const std::string &s)

Tests that a string is not empty and has non-space characters.

bool string_is_void (const std::string &s)

Tests that a string is empty or has only white-space characters.

bool strings_match (const std::string &target, const std::string &x)

Compares two strings for a form of semantic equality, for the purposes of editable event(), for example.

• int log2_time_sig_value (int tsd)

Calculates the log-base-2 value of a number that is already a power of 2.

void tempo_us_to_bytes (midibyte t[3], int tempo_us)

Provide a way to convert a tempo value (microseconds per quarter note) into the three bytes needed as value in a Tempo meta event.

int zoom_power_of_2 (int ppqn)

Calculates a suitable starting zoom value for the given PPQN value.

double bpm_from_tempo_us (double tempous)

This function calculates the effective beats-per-minute based on the value of a Tempo meta-event.

double tempo_us_from_bpm (midibpm bpm)

Converts tempo (e.g.

double pulse_length_us (midibpm bpm, int ppqn)

Calculates pulse-length from the BPM (beats-per-minute) and PPQN (pulses-per-quarter-note) values.

• double delta_time_us_to_ticks (unsigned long us, midibpm bpm, int ppqn)

Converts delta time in microseconds to ticks.

• double ticks_to_delta_time_us (midipulse delta_ticks, midibpm bpm, int ppqn)

Converts the time in ticks ("clocks") to delta time in microseconds.

double clock_tick_duration_bogus (midibpm bpm, int ppqn)

Calculates the duration of a clock tick based on PPQN and BPM settings.

int clock_ticks_from_ppqn (int ppqn)

A simple calculation to convert PPQN to MIDI clock ticks.

double double_ticks_from_ppqn (int ppqn)

A simple calculation to convert PPQN to MIDI clock ticks.

midipulse pulses_per_measure (int ppqn=SEQ64_DEFAULT_PPQN)

Calculates the pulses per measure.

• midipulse measures_to_ticks (midibpm bpm, int ppqn, int bw, int measures=1)

Calculates the length of an integral number of measures, in ticks.

double wave_func (double angle, wave_type_t wavetype)

Calculates a wave function for use as an LFO (low-frequency oscillator) for modifying data values in a sequence.

• bool extract_port_names (const std::string &fullname, std::string &clientname, std::string &portname)

Extracts the two names from the ALSA/JACK client/port name format, "[0] 128:0 clientname:portname".

std::string extract_bus_name (const std::string &fullname)

Extracts the buss name from "bus:port".

std::string extract port name (const std::string &fullname)

Extracts the port name from "bus:port".

bool help_check (int argc, char *argv [])

Checks to see if the first option is a help or version argument, just so we can skip the "Reading configuration ..." messages.

bool parse_options_files (perform &p, std::string &errmessage, int argc, char *argv [])

Provides the command-line option support, as well as some setup support, extracted from the main routine of Sequencer64.

• int parse_command_line_options (perform &p, int argc, char *argv [])

Parses the command-line options on behalf of the application.

bool write options files (const perform &p)

Saves all options to the "rc" and "user" configuration files.

std::string build_details ()

Generates a string describing the features of the build.

• std::string message_concatenate (const char *m1, const char *m2)

This function concatenates two C string pointers and returns them as a string message.

bool info_message (const std::string &msg)

Common-code for console messages.

bool error_message (const std::string &msg)

Common-code for error messages.

std::string to_string (const event &ev)

A free function to convert an event into an informative string, just enough to save some debugging time.

- bool file_access (const std::string &targetfile, int mode)
- bool file_exists (const std::string &filename)

Checks a file for existence.

bool file_readable (const std::string &filename)

Checks a file for readability.

bool file writable (const std::string &filename)

Checks a file for writability.

bool file_accessible (const std::string &filename)

Checks a file for readability and writability.

bool file_executable (const std::string &filename)

Checks a file for the ability to be executed.

bool file_is_directory (const std::string &filename)

Checks a file to see if it is a directory.

bool make_directory (const std::string &pathname)

A function to ensure that the \sim /.config/sequencer64 directory exists.

bool ppqn_is_valid (int ppqn)

Common code for handling PPQN settings.

• int jack_sync_callback (jack_transport_state_t state, jack_position_t *pos, void *arg)

Global functions for JACK support and JACK sessions.

void jack shutdown callback (void *arg)

This callback is to shut down JACK by clearing the jack_assistant :: m_jack_running flag.

void jack_timebase_callback (jack_transport_state_t state, jack_nframes_t nframes, jack_position_t *pos, int new pos, void *arg)

The JACK timebase function defined here sets the JACK position structure.

- int jack_transport_callback (jack_nframes_t nframes, void *arg)
- jack client t * create jack client (const std::string &clientname, const std::string &uuid)

A more full-featured initialization for a JACK client, which is meant to be called by the init() function.

void show jack statuses (unsigned bits)

Loops through the full set of JACK bits, showing the information for any bits that are set in the given parameter.

- long get_current_jack_position (void *arg)
- void jack_session_callback (jack_session_event_t *ev, void *arg)

Set the m_jsession_ev (event) value of the perform object.

- bool invalid key (unsigned int key)
- std::string keyval_name (unsigned int key)

Obtains the name of the key.

void keyval_normalize (keys_perform_transfer &k)

For the case in which the "rc" file is missing or corrupt, this function makes sure that each control key has a reasonable value.

bool create_lash_driver (perform &p, int argc, char **argv)

Creates and starts a lash object.

• lash * lash_driver ()

Provides access to the lash object.

void delete_lash_driver ()

Deletes the last object.

- void millisleep (unsigned long ms)
- bool is_null_midipulse (midipulse p)

Compares a midipulse value to SEQ64_NULL_MIDIPULSE.

void * output_thread_func (void *p)

Global functions defined in perform.cpp.

void * input_thread_func (void *myperf)

Set up the performance, and set the process to realtime privileges.

• rc settings & rc ()

Returns a reference to the global rc_settings object.

user_settings & usr ()

Returns a reference to the global user_settings object, for better encapsulation.

int choose_ppqn (int ppqn)

Common code for handling PPQN settings.

• midipulse timestring_to_pulses (const std::string ×tring, midibpm bpm, int ppqn)

Converts a string that represents "hours:minutes:seconds.fraction" into a MIDI pulse/ticks/clock value.

int jack_dummy_callback (jack_nframes_t, void *arg)

Provides a dummy callback.

const std::string & seq_app_name ()

Returns the name of the application.

const std::string & seq_client_name ()

Returns the name of the client for the application.

const std::string & seq_version ()

Returns the version of the application.

• static std::string make_section_name (const std::string &label, int value)

Provides a purely internal, ad hoc helper function to create numbered section names for the userfile class.

• font & font render ()

The p_font_renderer pointer was once created in the main module, sequencer64.cpp.

Gtk::Adjustment & adjustment dummy ()

Provides a way to provide a dummy Gtk::Adjustment object, but not create one until it is actually needed, so that the Glib/Gtk infrastructure is ready for it.

bool is ctrl key (GdkEventKey *ev)

Encapsulates the safe test for the control key, as described here: $https://developer.gnome. \leftarrow org/gtk3/stable/checklist-modifiers.html.$

bool is_shift_key (GdkEventKey *ev)

Encapsulates the safe test for the shift key.

bool is_no_modifier (GdkEventScroll *ev)

Encapsulates the safe test for no modifier keys, for a scroll event.

bool is_ctrl_key (GdkEventScroll *ev)

Encapsulates the safe test for the control key for scrolling.

bool is_shift_key (GdkEventScroll *ev)

Encapsulates the safe test for the shift key.

bool is ctrl key (GdkEventButton *ev)

Encapsulates the safe test for the control key for buttons.

bool is_shift_key (GdkEventButton *ev)

Encapsulates the safe test for the shift key.

bool is_ctrl_shift_key (GdkEventButton *ev)

Encapsulates the safe test for the ctrl-shift key combination.

bool is_super_key (GdkEventButton *ev)

Encapsulates the test for the super (mod4, windows) key for buttons.

- void test widget click (GtkWidget *w)
- void update_mainwid_sequences ()

This global function in the seq64 namespace calls mainwid :: update_sequences_on_window(), if the global mainwid object exists.

· void update perfedit sequences ()

This global function in the seq64 namespace calls perfedit :: draw_sequences(), if the global perfedit objects exist.

int FF_RW_timeout (void *arg)

This global function in the seq64 namespace is passed to the gtk_timeout_add callback.

static long clamp (long val, long low, long hi)

An internal function used by the FruitySeqRollInput class.

static long clamp (long val, long low, long hi)

An internal function used by the FruitySeqRollInput class.

void silence_jack_errors (bool silent)

This function silences JACK error output to the console.

void silence_jack_info (bool silent)

This function silences JACK info output to the console.

• std::string midi_api_name (int i)

Function to get RtMidi API names in a reusable manner.

• int midi_probe ()

Formerly the main program of the RtMidi test program midiprobe.

bool midi_input_test (rtmidi_info &info, int portindex)

Provides testing the MIDI input process for 10 seconds.

• long min (long a, long b)

min() for long values.

int jack process rtmidi input (jack nframes t nframes, void *arg)

Defines the JACK input process callback.

int jack process rtmidi output (jack nframes t nframes, void *arg)

Defines the JACK process input callback.

int jack process io (jack nframes t nframes, void *arg)

Provides a JACK callback function that uses the callbacks defined in the midi jack module.

static void jack_message_bit_bucket (const char *)

This function merely eats the string passed as a parameter.

static void midi_input_callback (midi_message &message, void *)

Provides the callback for midi_input_test().

Variables

std::string c controller names [SEQ64 MIDI COUNT MAX]

Provides the default names of MIDI controllers.

· const midibyte EVENT STATUS BIT

This highest bit of the status byte is always 1.

· const midibyte EVENT_ANY

Channel Voice Messages.

- · const midibyte EVENT NOTE OFF
- const midibyte EVENT NOTE ON
- const midibyte EVENT_AFTERTOUCH
- · const midibyte EVENT CONTROL CHANGE
- const midibyte EVENT_PROGRAM_CHANGE
- · const midibyte EVENT CHANNEL PRESSURE
- const midibyte EVENT_PITCH_WHEEL
- · const midibyte EVENT MIDI SYSEX

System Messages.

- const midibyte EVENT MIDI QUARTER FRAME
- · const midibyte EVENT MIDI SONG POS
- · const midibyte EVENT MIDI SONG SELECT
- const midibyte EVENT MIDI SONG F4
- const midibyte EVENT MIDI SONG F5
- const midibyte EVENT_MIDI_TUNE_SELECT
- const midibyte EVENT_MIDI_SYSEX_END
- const midibyte EVENT_MIDI_SYSEX_CONTINUE
- const midibyte EVENT_MIDI_CLOCK
- · const midibyte EVENT MIDI SONG F9
- · const midibyte EVENT MIDI START
- const midibyte EVENT MIDI CONTINUE
- const midibyte EVENT_MIDI_STOP
- · const midibyte EVENT MIDI SONG FD
- const midibyte EVENT_MIDI_ACTIVE_SENS
- const midibyte EVENT_MIDI_RESET
- const midibyte EVENT_MIDI_META

0xFF is a MIDI "escape code" used in MIDI files to introduce a MIDI meta event.

const midibyte EVENT NULL CHANNEL

This value of 0xFF is Sequencer64's channel value that indicates that the event's m_channel value is bogus.

• const midibyte EVENT_GET_CHAN_MASK

These file masks are used to obtain or to mask off the channel data from a status byte.

- const midibyte EVENT_CLEAR_CHAN_MASK
- · const int EVENTS ALL

Variable from the "stazed" extras.

- const int EVENTS UNSELECTED
- const int c_midibus_output_size

Manifest global constants.

· const int c midibus input size

The c_midibus_input_size value is passed, in mastermidibus, to snd_seq_set_input_buffer_size().

• const int c_midibus_sysex_chunk

Controls the amount a SysEx data sent at one time, in the midibus module.

· const midilong c midibus

Provides tags used by the midifile class to control the reading and writing of the extra "proprietary" information stored in a Seq24 MIDI file.

· const midilong c_midich

Track channel number.

· const midilong c_midiclocks

Track clocking.

const midilong c_triggers

See c_triggers_new.

const midilong c_notes

Song data.

• const midilong c_timesig

Track time signature.

• const midilong c_bpmtag

Song beats/minute.

const midilong c_triggers_new

Track trigger data.

• const midilong c_mutegroups

Song mute group data.

const midilong c_midictrl

Song MIDI control.

· const midilong c_musickey

The track's key.

• const midilong c_musicscale

The track's scale.

• const midilong c_backsequence

Track background sequence.

const midilong c_transpose

Track transpose value.

const midilong c_perf_bp_mes

Perfedit beats/measure.

const midilong c_perf_bw

Perfedit beat-width.

· const int c midi track ctrl

Pseudo control values for associating MIDI events (I think) with automation of some of the controls in seq24.

- const int c_midi_control_bpm_up
- const int c_midi_control_bpm_dn
- const int c_midi_control_ss_up
- const int c_midi_control_ss_dn
- const int c_midi_control_mod_replace

- · const int c_midi_control_mod_snapshot
- · const int c_midi_control_mod_queue
- · const int c midi control mod gmute
- · const int c midi control mod glearn
- const int c_midi_control_play_ss
- const int c_midi_controls
- · const int c_midi_control_playback
- · const int c midi control record
- · const int c midi control solo
- · const int c midi control thru
- · const int c midi control bpm page up
- const int c_midi_control_bpm_page_dn
- const int c_midi_control_16
- const int c_midi_control_17
- · const int c midi control 18
- const int c_midi_control_19
- const int c_midi_controls_extended
- · int g midi control limit
- const bool c_scales_policy [c_scale_size][SEQ64_OCTAVE_SIZE]

Each value in the kind of scale is denoted by a true value in these arrays.

const int c_scales_transpose_up [c_scale_size][SEQ64_OCTAVE_SIZE]

Increment values needed to transpose each scale up so that it remains in the same key.

const int c_scales_transpose_dn [c_scale_size][SEQ64_OCTAVE_SIZE]

Making these positive makes it easier to read, but the actual array contains negative values.

const char c_scales_text [c_scale_size][20]

The names of the currently-supported scales.

const char c_key_text [SEQ64_OCTAVE_SIZE][4]

Provides the entries for the Key dropdown menu in the Pattern Editor window.

const char c_interval_text [16][4]

Provides the entries for the Interval dropdown menu in the Pattern Editor window.

const char c_chord_text [8][6]

Provides the entries for the Chord dropdown menu in the Pattern Editor window.

const int c_chord_number

Additional support data for the chord-generation feature from Stazed's seg32 project.

• const char c_chord_table_text [c_chord_number][12]

Additional support data for the chord-generation feature from Stazed's seq32 project.

• const int c_chord_size

Provides the number of chord values in each chord's specification array.

• const int c_chord_table [c_chord_number][c_chord_size]

Additional support data for the chord-generation feature from Stazed's seq32 project.

• const int c_max_instruments

Provides the maximum number of instruments that can be defined in the \sim /.seq24usr or \sim /.config/sequencer64/sequencer6.

• const int c_max_busses

Provides the maximum number of MIDI buss definitions supported in the "user" file.

static const std::string versiontext

Sets up the "hardwired" version text for Sequencer64.

static struct option long_options []

A structure for command parsing that provides the long forms of command-line arguments, and associates them with their respective short form.

static const std::string s arg list

Provides a complete list of the short options, and is passed to getopt_long().

static const char *const s_help_1a

Provides help text.

• static const char *const s help 1b

More help text.

static const char *const s_help_2

Still more help text.

static const char *const s help 3

Still more help text.

static const char *const s_help_4

Still more help text.

· static const std::string s build alsamidi support

This section of variables provide static information about the options enabled or disabled during the build.

- static const std::string s_build_portmidi_support
- static const std::string s build rtmidi support
- static const std::string s build highlight empty
- static const std::string s_build_lash_support
- static const std::string s_build_jack_support
- static const std::string s_build_jack_session
- static const std::string s_event_editor
- static const std::string s build pause support
- static const std::string s_build_use_event_map
- · static const std::string s build chord generator
- · static const std::string s_build_edit_highlight
- static const std::string s_build_timesig_tempo
- static const std::string s build midi vector
- static const std::string s_build_solid_grid
- static const std::string s_build_follow_progress
- static const std::string s_statistics_support
- static const std::string s_strip_empty_mutes
- static const std::string s_seq32_jack_support
- static const std::string s_seq32_transport
- static const std::string s_seq32_transpose
- static const std::string s_seq32_menu_buttons
- static const std::string s_seq32_lfo_support
- static const std::string s_debug_mode
- jack_status_pair_t s_status_pairs []

Provides a list of JACK status bits, and a brief string to explain the status bit.

struct charpair t s character mapping []

The array of mappings of the non-alphabetic characters.

static lash * s_global_lash_driver

The global pointer to the LASH driver instance.

• static const int c_status_replace

Purely internal constants used with the functions that implement MIDI control for the application.

static const int c_status_snapshot

This value signals the "snapshot" functionality.

· static const int c status queue

This value signals the "queue" functionality.

static rc_settings g_rc_settings

Provides the replacement for all of the other "global_xxx" variables.

· static user settings g user settings

Provides the replacement for all of the other settings in the "user" configuration file, plus some of the "constants" in the globals module.

• static const long s_handlesize

An internal variable for handle size.

· static const int s jitter amount

An internal variable for user-jitter control.

static mainwid * gs_mainwid_pointer

Holds a pointer to the single instance of mainwnd for the entire application, once it is created.

· const int c mainwid x

The width of the main pattern/sequence grid, in pixels.

- const int c mainwid y
- static perfedit * gs_perfedit_pointer_0

Holds a pointer to the first instance of perfedit for the entire application, once it is created.

static perfedit * gs_perfedit_pointer_1

Holds a pointer to the second instance of perfedit for the entire application, once it is created.

• static const long s_handlesize

An internal variable for handle size.

12.2.1 Detailed Description

Do not document a namespace; it breaks Doxygen.

Obsolete Now a permanent option.

0.9.3 delta-tick calculation code. This code doesn't quite work for generating the proper rate of MIDI clocks, and so have disabled that code until we can figure out what it is we're doing wrong. Do not enable it unless you are willing to test it.

Provides a new option to save the Time Signature and Tempo data that may be present in a MIDI file (in the first track) in the sequence object, and write them back to the MIDI file when saved again, in Sequencer64 format. The SeqSpec events that Seq24 and Sequencer64 save for these "events" are not readable by other MIDI applications, such as QTractor. By enabling this macro, other sequencers can read the correct time-signature and tempo values.

```
#define SEQ64_HANDLE_TIMESIG_AND_TEMPO
```

12.2.2 Typedef Documentation

12.2.2.1 midibyte

typedef unsigned char seq64::midibyte

This can be used for a MIDI buss/port number or for a MIDI channel number. See the SEQ64_INVALID_MIDIBYTE macro.

12.2.2.2 bussbyte

typedef unsigned char seq64::bussbyte

12.2.2.3 midishort

typedef unsigned short seq64::midishort

12.2.2.4 midilong

typedef unsigned long seq64::midilong

12.2.2.5 midipulse

typedef long seq64::midipulse

HOWEVER, CURRENTLY, if you make this value unsigned, then perfroll won't show any notes in the sequence bars!!! Also, a number of manipulations of this type currently depend upon it being a signed value.

12.2.2.6 midibpm

typedef double seq64::midibpm

This value used to be an integer, but we need to provide more precision in order to support better tempo matching.

12.2.2.7 rterror_callback

typedef void(* seq64::rterror_callback) (rterror::Type type, const std::string &errormsg, void
*userdata)

Note that class behaviour is undefined after a critical error (not a warning) is reported.

Parameters

type	Type of error.
errorText	Error description.

12.2.2.8 rtmidi_callback_t

```
typedef void(* seq64::rtmidi_callback_t) (midi_message &message, void *userdata)
```

Used to be nested in the rtmidi_in class. The timestamp parameter has been folded into the midi_message class (a wrapper for std::vector<unsigned char>), and the pointer has been replaced by a reference.

12.2.3 Enumeration Type Documentation

12.2.3.1 wave_type_t

```
enum seq64::wave_type_t
```

We still have to clarify these type values, though.

Enumerator

WAVE_NONE	No waveform, never used.
WAVE_SINE	Sine wave modulation.
WAVE_SAWTOOTH	Saw-tooth (ramp) modulation.
WAVE_REVERSE_SAWTOOTH	Reverse saw-tooth (decay).
WAVE_TRIANGLE	No waveform, never used.

12.2.3.2 seq_modifier_t

```
enum seq64::seq_modifier_t
```

We have to tweak the names to avoid redeclaration errors and to "personalize" the values. We change "GDK" to "SEQ64".

Since we're getting events from, say Gtk-2.4, but using our (matching) values for comparison, use the CAST_EQ UIVALENT() macro to compare them. Note that we might still end up having to a remapping (e.g. if trying to get the code to work with the Qt framework).

Enumerator

SEQ64_NO_MASK	
SEQ64_SHIFT_MASK	
SEQ64_LOCK_MASK	
SEQ64_CONTROL_MASK	
SEQ64_MOD1_MASK	
SEQ64_MOD2_MASK	
SEQ64_MOD3_MASK	
SEQ64_MOD4_MASK	
SEQ64_MOD5_MASK	
SEQ64_BUTTON1_MASK	
SEQ64_BUTTON2_MASK	
SEQ64_BUTTON3_MASK	
SEQ64_BUTTON4_MASK	
SEQ64_BUTTON5_MASK	
SEQ64_SUPER_MASK	Bits 13 and 14 are used by XKB, bits 15 to 25 are unused. Bit 29 is used internally.
SEQ64_HYPER_MASK	
SEQ64_META_MASK	
SEQ64_RELEASE_MASK	
SEQ64_MASK_MAX	Generated by Doxygen

12.2.3.3 seq_event_type_t

enum seq64::seq_event_type_t

Only the values we need have been grabbed. We have to tweak the names to avoid redeclaration errors and to "personalize" the values. We change "GDK" to "SEQ64", but, for convenience (to hide errors? :-D), we keep the number the same.

Since we're getting events from, say Gtk-2.4, but using our (matching) values for comparison, use the CAST_EQ UIVALENT() macro to compare them. Note that we might still end up having to a remapping (e.g. if trying to get the code to work with the Qt framework).

Enumerator

SEQ64_NOTHING	
SEQ64_DELETE	
SEQ64_DESTROY	
SEQ64_EXPOSE	
SEQ64_MOTION_NOTIFY	
SEQ64_BUTTON_PRESS	
SEQ64_2BUTTON_PRESS	
SEQ64_3BUTTON_PRESS	
SEQ64_BUTTON_RELEASE	
SEQ64_KEY_PRESS	
SEQ64_KEY_RELEASE	
SEQ64_SCROLL	
SEQ64_EVENT_LAST	

12.2.3.4 seq_scroll_direction_t

enum seq64::seq_scroll_direction_t

We have to tweak the names to avoid redeclaration errors and to "personalize" the values. We change "SEQ64" to "SEQ64".

Since we're getting events from, say Gtk-2.4, but using our (matching) values for comparison, use the CAST_EQ UIVALENT() macro to compare them. Note that we might still end up having to a remapping (e.g. if trying to get the code to work with the Qt framework).

Enumerator

SEQ64_SCROLL_UP	
SEQ64_SCROLL_DOWN	
SEQ64_SCROLL_LEFT	
SEQ64_SCROLL_RIGHT	

12.2.3.5 clock_e

enum seq64::clock_e

This enumeration was also defined in midibus_portmidi.h, but we put it into this common module to avoid duplication.

Enumerator

e_clock_off	Corresponds to the "Off" selection in the MIDI Clock tab. With this setting, the MIDI Clock is disabled for the buss using this setting. Notes will still be sent that buss, of course. Some software synthesizer might require this setting in order to make a sound.
e_clock_pos	Corresponds to the "Pos" selection in the MIDI Clock tab. With this setting, MIDI Clock will be sent to this buss, and, if playback is starting beyond tick 0, then MIDI Song Position and MIDI Continue will also be sent on this buss.
e_clock_mod	Corresponds to the "Mod" selection in the MIDI Clock tab. With this setting, MIDI Clock and MIDI Start will be sent. But clocking won't begin until the Song Position has reached the start modulo (in 1/16th notes) that is specified.

12.2.3.6 interaction_method_t

enum seq64::interaction_method_t

Moved here from the globals.h module.

Enumerator

e_seq24_interaction	Use the normal mouse interactions.
e_fruity_interaction	The "fruity" mouse interactions.
e_number_of_interactions	Keep this last a size value.

12.2.3.7 c_music_scales

enum seq64::c_music_scales

Scales can be shown in the piano roll as gray bars for reference purposes.

We've added three more scales; there are still a number of them that could be fruitfully added to the list of scales.

It would be good to offload this stuff into a new "scale" class.

Enumerator

c_scale_off

Enumerator

c_scale_major c_scale_minor c_scale_harmonic_minor c_scale_melodic_minor c_scale_c_whole_tone c_scale_blues c_scale_major_pentatonic c_scale_minor_pentatonic c_scale_size		
c_scale_harmonic_minor c_scale_melodic_minor c_scale_c_whole_tone c_scale_blues c_scale_major_pentatonic c_scale_minor_pentatonic	c_scale_major	
c_scale_melodic_minor c_scale_c_whole_tone c_scale_blues c_scale_major_pentatonic c_scale_minor_pentatonic	c_scale_minor	
c_scale_c_whole_tone	c_scale_harmonic_minor	
c_scale_blues c_scale_major_pentatonic c_scale_minor_pentatonic	c_scale_melodic_minor	
c_scale_major_pentatonic c_scale_minor_pentatonic	c_scale_c_whole_tone	
c_scale_minor_pentatonic	c_scale_blues	
	c_scale_major_pentatonic	
c_scale_size	c_scale_minor_pentatonic	
	c_scale_size	

12.2.3.8 draw_type_t

enum seq64::draw_type_t

These values are used in the sequence, seqroll, perfroll, and mainwid classes.

Enumerator

DRAW_FIN	Indicates that drawing is finished.
DRAW_NORMAL_LINKED	Used for drawing linked notes.
DRAW_NOTE_ON	For starting the drawing of a note.
DRAW_NOTE_OFF	For finishing the drawing of a note.

12.2.3.9 mouse_action_e

enum seq64::mouse_action_e

Enumerator

e_action_select	
e_action_draw	
e_action_grow	

12.2.3.10 edit_action_t

enum seq64::edit_action_t

These variables represent actions that can be applied to a selection of notes. One idea would be to add a swing-quantize action. We will reserve the value here, for notes only; not yet used or part of the action menu.

Enumerator

c_select_all_notes	
c_select_all_events	
c_select_inverse_notes	
c_select_inverse_events	
c_quantize_notes	
c_quantize_events	
c_tighten_events	
c_tighten_notes	
c_transpose_notes	
c_reserved	
c_transpose_h	
c_expand_pattern	
c_compress_pattern	
c_select_even_notes	
c_select_odd_notes	
c_swing_notes	

12.2.3.11 rtmidi_api

```
enum seq64::rtmidi_api
```

These items used to be nested in the rtmidi class, but that only worked when RtMidi.cpp/h were large monolithic modules.

Enumerator

RTMIDI_API_UNSPECIFIED	Search for a working compiled API.
RTMIDI_API_LINUX_ALSA	Advanced Linux Sound Architecture API.
RTMIDI_API_UNIX_JACK	JACK Low-Latency MIDI Server API.
RTMIDI_API_MAXIMUM	A count of APIs; an erroneous value.

12.2.4 Function Documentation

12.2.4.1 swap()

```
void seq64::swap (
                busarray & buses0,
                busarray & buses1)
```

Parameters

buses0	Provides the first buss in the swap.
buses1	Provides the second buss in the swap.

12.2.4.2 wave_type_name()

These names are short because I cannot figure out how to get the window pad out to show the longer names.

Parameters

yed.

Returns

Returns a short description of the wave type.

12.2.4.3 extract_timing_numbers()

- · measures : beats : divisions
 - "213:4:920"
 - "0:1:0"
- · hours : minutes : seconds . fraction
 - "2:04:12.14"
 - "0:1:2"

Warning

This is not the most efficient implementation you'll ever see. At some point we will tighten it up. This function is tested in the seq64-tests project, in the "calculations_unit_test" module.

Parameters

	s	Provides the input time string, in measures or time format, to be processed.
out	part⊷	The destination reference for the first part of the time.
	_1	
out	part⇔	The destination reference for the second part of the time.
	_2	
out	part←	The destination reference for the third part of the time.
	_3	
out	fraction	The destination reference for the fractional part of the time.

Returns

Returns true if a reasonable portion (3 numbers) was good for extraction. The fraction part will start with a period for easier conversion to fractional seconds.

12.2.4.4 pulses_to_string()

```
\begin{tabular}{ll} {\tt std::string seq64::pulses\_to\_string (} \\ & {\tt midipulse } p \end{tabular} \end{tabular}
```

Todo Still needs to be unit tested.

Parameters

p The MIDI pulse/tick value to be converted.

Returns

Returns the string as an unsigned ASCII integer number.

12.2.4.5 pulses_to_measurestring()

Parameters

p	The number of MIDI pulses (clocks, divisions, ticks, you name it) to be converted. If the value is SEQ64_NULL_MIDIPULSE, it is converted to 0, because callers don't generally worry about such niceties, and the least we can do is convert illegal measure-strings (like "000:0:000") to a legal value.
seqparms	This small structure provides the beats/measure, beat-width, and PPQN that hold for the sequence involved in this calculation. These values are needed in the calculations.

Returns the string, in measures notation, for the absolute pulses that mark this duration.

12.2.4.6 pulses_to_midi_measures()

Parameters

	р	Provides the MIDI pulses (as in "pulses per quarter note") that are to be converted to MIDI measures format.
	seqparms	This small structure provides the beats/measure (B), beat-width (W), and PPQN (P) that hold for the sequence involved in this calculation. The beats/minute (T for tempo) value is not needed.
out	measures	Provides the current MIDI song time structure to hold the results, which are the measures, beats, and divisions values for the time of interest. Note that the measures and beats are corrected to be re 1, not 0.

Returns

Returns true if the calculations were able to be made. The P, B, and W values all need to be greater than 0.

12.2.4.7 pulses_to_timestring() [1/2]

If the fraction part is 0, then it is not shown. Examples:

```
- "0:0:0"

- "0:0:0.102333"

- "12:3:1"

- "12:3:1.000001"
```

Parameters

p	Provides the number of ticks, pulses, or divisions in the MIDI event time.
bpm	Provides the tempo of the song, in beats/minute.
ppqn	Provides the pulses-per-quarter-note of the song.

Returns the time-string representation of the pulse (ticks) value.

See the other pulses_to_timestring() overload.

Todo Still needs to be unit tested.

Parameters

р	Provides the number of ticks, pulses, or divisions in the MIDI event time.
timinginfo	Provides the tempo of the song, in beats/minute, and the pulse-per-quarter-note of the song.

Returns

Returns the return-value of the other pulses_to_timestring() function.

12.2.4.9 measurestring_to_pulses()

Parameters

measures	Provides the current MIDI song time in "measures:beats:divisions" format, where divisions are the MIDI pulses in "pulses-per-quarter-note".
seqparms	This small structure provides the beats/measure, beat-width, and PPQN that hold for the sequence involved in this calculation.

Returns

Returns the absolute pulses that mark this duration. If the input string is empty, then 0 is returned.

12.2.4.10 midi_measures_to_pulses()

p = 4 * P * m * B / W p == pulse count (ticks or pulses) m == number of measures B == beats per measure (constant) P == pulses per quarter-note (constant) W == beat width in beats per measure (constant)

Note that the 0-pulse MIDI measure is "1:1:0", which means "at the beginning of the first beat of the first measure, no pulses'. It is not "0:0:0" as one might expect.

Parameters

measures	Provides the current MIDI song time structure holding the measures, beats, and divisions values
	for the time of interest.
seqparms	This small structure provides the beats/measure, beat-width, and PPQN that hold for the
	sequence involved in this calculation.

Returns

Returns the absolute pulses that mark this duration. If the pulse-value cannot be calculated, then SEQ64_← NULL MIDIPULSE is returned.

12.2.4.11 timestring_to_pulses() [1/2]

12.2.4.12 string_to_pulses()

First, the type of string is deduced by the characters in the string. If the string contains two colons and a decimal point, it is assumed to be a time-string ("hh:mm:ss.frac"); in addition ss will have to be less than 60.

If the string just contains two colons, then it is assumed to be a measure-string ("measures:beats:divisions").

If it has none of the above, it is assumed to be pulses. Testing is not rigorous.

Parameters

s	Provides the string to convert to pulses.	
mt	Provides the structure needed to provide BPM and other values needed for some of the conversions done	
Genera	by this function. Generated by Doxygen	

Returns the string as converted to MIDI pulses (or divisions, clocks, ticks, whatever you call it).

12.2.4.13 string_to_midibyte()

```
midibyte seq64::string_to_midibyte ( {\tt const\ std::string\ \&\ s\ )}
```

This function bypasses characters until it finds a digit (whether part of the number or a "0x" construct), and then converts it.

Parameters

s Provides the string to convert to a MIDI byte.

Returns

Returns the MIDI byte value represented by the string.

12.2.4.14 shorten_file_spec()

This is done by removing character in the middle, if necessary, and replacing them with an ellipse.

This function operates by first trying to find the <code>/home directory</code>. If found, it strips off <code>/home/username and</code> replace it with the Linux \sim replacement for the \$HOME environment variable. This function assumes that the "username" portion *must* exist, and that there's no goofy stuff like double-slashes in the path.

Parameters

fpath	The file specification, including the full path to the file, and the name of the file.
leng	Provides the length to which to limit the string.

Returns

Returns the fpath parameter, possibly shortened to fit within the desired length.

12.2.4.15 string_not_void()

Provides essentially the opposite test that string_is_void() provides. The definition of white-space is provided by the std::isspace() function/macro.

Parameters

s The string pointer to check for emptiness.

Returns

Returns true if the pointer is valid, the string has a non-zero length, and is not just white-space.

12.2.4.16 string_is_void()

```
bool seq64::string_is_void ( {\tt const\ std::string\ \&\ s\ )}
```

Meant to have essentially the opposite result of string_not_void(). The meaning of empty is special here, as it refers to a string being useless as a token:

```
    The string is of zero length.
    The string has only white-space characters in it, where the isspace() macro provides the definition of white-space.
```

Parameters

s The string pointer to check for emptiness.

Returns

Returns true if the string has a zero length, or is only white-space.

12.2.4.17 strings_match()

The strings_match() function returns true if the comparison items are identical, without case-sensitivity in character content up to the length of the secondary string. This allows abbreviations to match. (And, in scanning routines, the first match is immediately accepted.)

Parameters

target	The primary string in the comparison. This is the target string, the one we hope to match. It is
	assumed to be non-empty, and the result is false if it is empty
X	The secondary string in the comparison. It must be no longer than the target string, or the match is
	false.

Returns true if both strings are are identical in characters, up to the length of the secondary string, with the case of the characters being insignificant. Otherwise, false is returned.

12.2.4.18 log2_time_sig_value()

```
int seq64::log2_time_sig_value ( int \ tsd \ )
```

Useful in converting a time signature's denominator to a Time Signature meta event's "dd" value.

Parameters

tsd The time signature denominator, which must be a power of 2: 2, 4, 8, 16, or 32.

Returns

Returns the power of 2 that achieves the tsd parameter value.

12.2.4.19 tempo_us_to_bytes()

Recall the format of a Tempo event:

0 FF 51 03 t2 t1 t0 (tempo as number of microseconds per quarter note)

This code is the inverse of the lines of code around line 768 in midifile.cpp, which is basically ((t2 * 256) + t1) * 256 + t0.

As a test case, note that the default tempo is 120 beats/minute, which is equivalent to ttttt=500000 (0x07A120).

Parameters

t	Provides a small array of 3 elements to hold each tempo byte.
tempo_us	Provides the temp value in microseconds per quarter note.

12.2.4.20 zoom_power_of_2()

The default starting zoom is 2, but this value is suitable only for PPQN of 192 and below. Also, zoom currently works consistently only if it is a power of 2. For starters, we scale the zoom to the selected ppqn, and then shift it each way to get a suitable power of two.

Parameters

p	pqn	The ppqn of interest.
---	-----	-----------------------

Returns

Returns the power of 2 appropriate for the given PPQN value.

12.2.4.21 bpm_from_tempo_us()

The tempo event's numeric value is given in 3 bytes, and is in units of microseconds-per-quarter-note (us/qn).

Parameters

tempous	The value of the Tempo meta-event, in units of us/qn. If this value is 0, we'll get an arithmetic	
	exception.	

Returns

Returns the beats per minute. If the tempo value is 0, then 0 is returned.

12.2.4.22 tempo_us_from_bpm()

120 beats/minute) to microseconds. This function is the inverse of bpm_from_tempo_us().

Parameters

bpm The value of beats-per-minute. If this value is 0, we'll get an arithmetic exception.

Returns

Returns the tempo in qn/us. If the bpm value is 0, then 0 is returned.

12.2.4.23 pulse_length_us()

The formula for the pulse-length in seconds is:

Parameters

bpm	Provides the beats-per-minute value. No sanity check is made. If this value is 0, we'll get an arithmetic exception.
ppqn	Provides the pulses-per-quarter-note value. No sanity check is made. If this value is 0, we'll get an arithmetic exception.

Returns

Returns the pulse length in microseconds. If either parameter is invalid, then this function will crash. :-D

12.2.4.24 delta_time_us_to_ticks()

This function is the inverse of ticks_to_delta_time_us().

Please note that terms "ticks" and "pulses" are equivalent, and refer to the "pulses" in "pulses per quarter note".

```
beats pulses 1 minute 1 sec
P = 120 ----- * 192 ----- * T us * ------ * -----
minute beats 60 sec 1,000,000 us
```

Note that this formula assumes that a beat is a quarter note. If a beat is an eighth note, then the P value would be halved, because there would be only 96 pulses per beat. We will implement an additional function to account for the beat; the current function merely blesses some calculations made in the application.

Parameters

us The number of microseconds in the delta time.	
bpm	Provides the beats-per-minute value, otherwise known as the "tempo".
ppqn	Provides the pulses-per-quarter-note value, otherwise known as the "division".

Returns the tick value.

12.2.4.25 ticks_to_delta_time_us()

The inverse of delta_time_us_to_ticks().

Please note that terms "ticks" and "pulses" are equivalent, and refer to the "pulses" in "pulses per quarter note".

Old: 60000000.0 * double(delta_ticks) / (double(bpm) * double(ppqn));

Parameters

delta_ticks	The number of ticks or "clocks".
bpm	Provides the beats-per-minute value, otherwise known as the "tempo".
ppqn	Provides the pulses-per-quarter-note value, otherwise known as the "division".

Returns

Returns the time value in microseconds.

12.2.4.26 clock_tick_duration_bogus()

Deprecated This is a somewhat bogus calculation used only for "statistical" output in the old perform module. Name changed to reflect this unfortunate fact. Use pulse_length_us() instead.

Parameters

bpm	Provides the beats-per-minute value. No sanity check is made. If this value is 0, we'll get an arithmetic exception.
ppqn	Provides the pulses-per-quarter-note value. No sanity check is made. If this value is 0, we'll get an arithmetic exception.

Returns

Returns the clock tick duration in microseconds. If either parameter is invalid, this will crash. Who wants to waste time on value checks here? :-D

12.2.4.27 clock_ticks_from_ppqn()

Parameters

ppqn The number of pulses per quarter note. For example, the default value for Seq24 is 192.

Returns

The integer value of ppqn / 24 [MIDI_CLOCK_IN_PPQN] is returned.

12.2.4.28 double_ticks_from_ppqn()

The same as clock_ticks_from_ppqn(), but returned as a double float.

Parameters

ppqn The number of pulses per quarter note.

Returns

The double value of ppqn / 24 [SEQ64_MIDI_CLOCK_IN_PPQN]_is returned.

12.2.4.29 pulses_per_measure()

This calculation is extremely simple, and it provides an important constraint to pulse (ticks) calculations: the number of pulses in a measure is always 4 times the PPQN value, regardless of the time signature. The number pulses in a 7/8 measure is the the same as in a 4/4/ measure.

12.2.4.30 measures_to_ticks()

This function is called in seqedit::apply_length(), when the user selects a sequence length in measures. That function calculates the length in ticks. The number of pulses is given by the number of quarter notes times the pulses per quarter note. The number of quarter notes is given by the measures times the quarter notes per measure. The quarter notes per measure is given by the beats per measure times 4 divided by beat_width beats. So:

Parameters

bpm	The B value in the equation, beats/measure.
ppqn	The P value in the equation, pulses/qn.
bw	The W value in the equation, the denominator of the time signature. If this value is 0, we'll get an arithmetic exception (crash), so we just return 0 in this case.
measures	The M value in the equation. It defaults to 1, in case one desires a simple "ticks per measure" number.

Returns

Returns the L value (ticks or pulses) as calculated via the given equation. If bw is 0, then 0 is returned.

12.2.4.31 wave_func()

We extracted this function from mattias's Ifownd module, as it is more generally useful. The angle parameter is provided by the Ifownd object. It is calculated by

The speed ranges from 0 to 16; the ratio of tick/seqlength ranges from 0 to 1; BW (beat width) is generally 4; the phase ranges from 0 to 1.

Parameters

angle	Provides the radial angle to be applied. Units of radians, apparently.	
wavetype	Provides the wave_type_t value to select the type of wave data-point to be generated.	1

12.2.4.32 extract_port_names()

It's a bit krufty to have to rely on that strict format; changes in the bus/port code could break this function.

And when a2jmidid is running, indeed this function breaks. The name of a port changes to

```
a2j:Midi Through [14] (playback): Midi Through Port-0
```

with "a2j" as the client name and the rest, including the second colon, as the port name.

TODO: FIX ME!!!!!!!

Parameters

		fullname	The full port specification to be split.
	out	clientname	The destination for the client name portion, "clientname".
Ī	out	portname	The destination for the port name portion, "portname".

Returns

Returns true if all items are non-empty after the process.

12.2.4.33 extract_bus_name()

Sometimes we don't need both parts at once.

However, when a2jmidid is active. the port name will have a colon in it.

Parameters

fullname	The "bus:port" name.
----------	----------------------

Returns

Returns the "bus" portion of the string. If there is no colon, then it is assumed there is no buss name, so an empty string is returned.

12.2.4.34 extract_port_name()

Sometimes we don't need both parts at once.

However, when a2jmidid is active. the port name will have a colon in it.

Parameters

```
fullname The "bus:port" name.
```

Returns

Returns the "port" portion of the string. If there is no colon, then it is assumed that the name is a port name, and so *fullname* is returned.

12.2.4.35 help_check()

Also check for the –legacy option. Finally, it also checks for the "?" option that people sometimes use as a guess to get help.

Parameters

argc	The number of command-line arguments.
argv	The array of command-line argument pointers.

Returns

Returns true only if -v, -V, -version, -h, -help, or "?" were encountered. If the legacy options occurred, then rc().legacy format(true) is called, as a side effect, because it will be needed before we parse the options.

12.2.4.36 parse_options_files()

It probably requires this call preceding: Gtk::Main kit(argc, argv), to strip any GTK+-specific parameters the knowledgeable user may have added. Usage:

```
Gtk::Main kit(argc, argv);
seq64::gui_assistant_gtk2 gui;
seq64::perform p(gui);
```

It also requires the caller to call rc().set_defaults() and usr().set_defaults(). The caller can then use the command-line to make any modifications to the setting that will be used here. The biggest example is the -r/-reveal-alsa-ports option, which determines if the MIDI buss definition strings are read from the 'user' configuration file.

Instead of the legacy Seq24 names, we use the new configuration file-names, located in the \sim /.config/sequencer64 directory. However, if they are not found, we no longer fall back to the legacy configuration file-names. If the – legacy option is in force, use only the legacy configuration file-name. The code also ensures the directory exists. CURRENTLY LINUX-SPECIFIC. See the rc_settings class for how this works.

```
std::string cfg_dir = seq64::rc().home_config_directory();
if (cfg_dir.empty())
    return EXIT_FAILURE;
```

Change Note ca 2016-04-03 We were parsing the user-file first, but we now need to parse the rc-file first, to get the manual-alsa-ports option, so that we can avoid overriding the port names that the ALSA system provides, if the manual-alsa-option is false.

Parameters

	p	Provides the perform object that will be affected by the new parameters.
out	errmessage	Provides a string into which to dump any error-message that might occur. Still not
		thoroughly supported in the "rc" and "user" configuration files.
	argc	The number of command-line arguments.
	argv	The array of command-line argument pointers.

Returns true if the reading of both configuration files succeeded.

12.2.4.37 parse_command_line_options()

Note that, since we call this function twice (once before the configuration files are parsed, and once after), we have to make sure that the global value optind is reset to 0 before calling this function. Note that the traditional reset value for optind is 1, but 0 is used in GNU code to trigger the internal initialization routine of get_opt().

Parameters

р	The performance object that implements some of the command-line options.
argc	The number of command-line arguments.
argv	The array of command-line argument pointers.

Returns

Returns the value of optind if no help-related options were provided.

12.2.4.38 write_options_files()

This function gets any legacy global variables, on the theory that they might have been changed.

Parameters

p Provides the perform object that may provide new values for the parameters.

Returns

Returns true if both files were saved successfully. Otherwise returns false. But even if one write failed, the other might have succeeded.

12.2.4.39 build_details()

```
std::string seq64::build_details ( )
```

Returns an ordered, human-readable string enumerating the built-in features of this application.

12.2.4.40 message_concatenate()

Note that we don't bother with error-checking the pointers. You're on your own, Hoss.

Parameters

m1	The first message, often a func macro.
m2	The second message.

Returns

Returns "m1: m2" as a standard C++ string.

12.2.4.41 info_message()

Adds markers and a newline.

Parameters

```
msg The message to print, sans the newline.
```

Returns

Returns true.

12.2.4.42 error_message()

Adds markers, and returns false.

Parameters

msg The message to print, sans the newline.

Returns

Returns false for convenience/brevity in setting function return values.

12.2.4.43 to_string()

Nothing fancy. If you want that, use the midicvt project.

Parameters

ev The event to put on show.

Returns

Returns the string representation of the event parameter.

12.2.4.44 file_access()

12.2.4.45 file_exists()

Parameters

filename Provides the name of the file to be checked.

Returns

Returns 'true' if the file exists.

12.2.4.46 file_readable()

Parameters

filename Provides the name of the file to be checked.

Returns

Returns 'true' if the file is readable.

12.2.4.47 file_writable()

Parameters

Returns

Returns 'true' if the file is writable.

12.2.4.48 file_accessible()

An even stronger test than file_exists. At present, we see no need to distinguish read and write permissions. We assume the file is accessible only if the file has both permissions.

Parameters

Returns

Returns 'true' if the file is readable and writable.

12.2.4.49 file_executable()

Parameters

filename	Provides the name of the file to be checked.
----------	--

Returns

Returns 'true' if the file exists.

12.2.4.50 file_is_directory()

This function is also used in the function of the same name in fileutilities.cpp.

Parameters

filename	Provides the name of the directory to be checked.
----------	---

Returns

Returns 'true' if the file is a directory.

12.2.4.51 make_directory()

This function is actually a little more general than that, but it is not sufficiently general, in general.

Parameters

pathname | Provides the name of the path to create. The parent directory of the final directory must already exist.

Returns true if the path-name exists.

12.2.4.52 ppqn_is_valid()

Validates a PPQN value.

Parameters

ppqn	Provides the PPQN value to be used.
------	-------------------------------------

Returns

Returns true if the ppqn parameter is between MINIMUM_PPQN and MAXIMUM_PPQN, or is set to SE ← Q64 USE DEFAULT PPQN (-1).

12.2.4.53 jack_sync_callback()

```
int seq64::jack_sync_callback (
          jack_transport_state_t state,
          jack_position_t * pos,
          void * arg )
```

This JACK synchronization callback informs the specified perform object of the current state and parameters of JACK.

The transport state will be:

- JackTransportStopped when a new position is requested.
- JackTransportStarting when the transport is waiting to start.
- JackTransportRolling when the timeout has expired, and the position is now a moving target.

This is the slow-sync callback, which the stazed code replaces with jack_transport_callback().

Parameters

Ī	state	The JACK Transport state.	
	pos	The JACK position value.	
Ī	arg	The pointer to the jack_assistant object. Currently not checked for nullity, nor dynamic-casted.	

Returns 1 if the function works, and 0 if something was wrong.

12.2.4.54 jack_shutdown_callback()

Parameters

arq

Points to the jack_assistant in charge of JACK support for the perform object.

12.2.4.55 jack_timebase_callback()

```
void seq64::jack_timebase_callback (
          jack_transport_state_t state,
          jack_nframes_t nframes,
          jack_position_t * pos,
          int new_pos,
          void * arg )
```

The original version of the function worked properly with Hydrogen, but not with Klick. The new code seems to work with both. More testing and clarification is needed. This new code was "discovered" in the source-code for the "SooperLooper" project:

```
http://essej.net/sooperlooper/
```

The first difference with the new code is that it handles the case where the JACK position is moved (new_pos == true). If this is true, and the JackPositionBBT bit is off in pos->valid, then the new BBT value is set.

The seconds set of differences are in the "else" clause. In the new code, it is very simple: calculate the new tick value, back it off by the number of ticks in a beat, and perhaps go to the first beat of the next bar.

In the old code (complex!), the simple BBT adjustment is always made. This changes (perhaps) the beats_per_bar, beat_type, etc. We need to make these settings use the actual global values for beats set for Sequencer64. Then, if transitioning from JackTransportStarting to JackTransportRolling (instead of checking new_pos!), the BBT values (bar, beat, and tick) are finally adjusted. Here are the steps, with old and new steps noted:

```
-# Calculate the "delta" ticks based on the current frame, the
    ticks_per_beat, the beats_per_minute, and the frame_rate. The old
    code saves this in a local, the new code assigns it to pos->tick.
-# Old code: save this delta as a positive value.
-# Figure out the settings and modify bar, beat, tick, and
    bar_start_tick. The old and new code seem to have the same intent,
    but it seems like the new code is faster and also correct.
- Old code: Calculations are made by division and mod
        operations.
- New code: Calculations are made by increments and decrements
    in a while loop.
```

Stazed:

The call to jack_timebase_callback() to supply JACK with BBT, etc. would occasionally fail when the pos information had zero or some garbage in the pos.frame_rate variable. This would occur when there was a rapid change of frame position by another client... i.e. qjackctl. From the JACK API:

pos address of the position structure for the next cycle; pos->frame will be its frame number. If new_pos is FALSE, this structure contains extended position information from the current cycle. If TRUE, it contains whatever was set by the requester. The timebase_callback's task is to update the extended information here."

The "If TRUE" line seems to be the issue. It seems that qjackctl does not always set pos.frame_rate so we get garbage and some strange BBT calculations that display in qjackctl. So we need to set it here and just use m_jack_frame_rate for calculations instead of pos.frame_rate.

Parameters

state	Indicates the current state of JACK transport.
nframes	The number of JACK frames in the current time period.
pos	Provides the position structure to be filled in, the address of the position structure for the next cycle; pos->frame will be its frame number. If new_pos is FALSE, this structure contains extended position information from the current cycle. If TRUE, it contains whatever was set by the requester. The timebase_callback's task is to update the extended information here.
new_pos	TRUE (non-zero) for a newly requested pos, or for the first cycle after the timebase_callback is defined. This is usually 0 in Sequencer64 at present, and 1 if one, say, presses "rewind" in qjackctl.
arg	Provides the jack_assistant pointer, currently unchecked for nullity.

12.2.4.56 jack_transport_callback()

12.2.4.57 create_jack_client()

Do not call this function if the JACK client handle is already open.

Status bits for jack_status_t return pointer:

JackNameNotUnique means that the client name was not unique. With JackUseExactName, this is fatal. Otherwise, the name was modified by appending a dash and a two-digit number in the range "-01" to "-99". The jack_get_client_name() function returns the exact string used. If the specified client_name plus these extra characters would be too long, the open fails instead.

JackServerStarted means that the JACK server was started as a result of this operation. Otherwise, it was running already. In either case the caller is now connected to jackd, so there is no race condition. When the server shuts down, the client will find out.

JackOpenOptions:

```
JackSessionID | JackServerName | JackNoStartServer | JackUseExactName
```

helgrind:

Valgrind's helgrind tool shows

```
Possible data race during read of size 4 at 0xF854E58 by thread #1
  by 0x267602: seq64::create_jack_client(...)
This conflicts with a previous write of size 4 by thread #2
  by 0x267602: seq64::create_jack_client(...)
```

So we add a static mutex to use with our automutex. Does not prevent that message.... WHY?

We've never disabled the SEQ64_JACK_SESSION macro, and we like the error-reporting we get by that method. So we've commented out the following code in favor of using the session-uuid code:

ifdef SEQ64_JACK_SESSION

else

jack_status_t * pstatus = NULL; result = jack_client_open(name, JackNullOption, pstatus);

endif

Parameters

clientname	Provides the name of the client, used in the call to jack_client_open(). By default, this name is the macro SEQ64_PACKAGE (i.e. "sequencer64"). The name scope is local to each server. Unless forbidden by the JackUseExactName option, the server will modify this name to create a unique variant, if needed.
uuid	The optional UUID to assign to the new client. If empty, there is no UUID.

Returns

Returns a pointer to the JACK client if JACK has opened the client connection successfully. Otherwise, a null pointer is returned.

12.2.4.58 show jack statuses()

```
void seq64::show_jack_statuses (
          unsigned bits)
```

For reference, here are the enumeration values from /usr/include/jack/types.h:

```
JackFailure
                  = 0x01
JackInvalidOption = 0x02
JackNameNotUnique = 0x04
JackServerStarted = 0x08
JackServerFailed
                 = 0x10
JackServerError
                 = 0x20
JackNoSuchClient
                 = 0x40
                 = 0x80
JackLoadFailure
JackInitFailure = 0x100
JackPach : 0x200
JackBackendError = 0x800
JackClientZombie = 0x1000
```

Parameters

bits The mask of the bits to be shown in the output.

12.2.4.59 get_current_jack_position()

Warning

Currently valgrind flags j->client() as uninitialized.

12.2.4.60 jack_session_callback()

```
void seq64::jack_session_callback (
          jack_session_event_t * ev,
          void * arg )
```

Glib is then used to connect in perform::jack_session_event(). However, the perform object's GUI-support interface is used instead of the following, so that the libseq64 library can be independent of a specific GUI framework:

```
Glib::signal_idle().
    connect(sigc::mem_fun(*jack, &jack_assistant::session_event));
```

Parameters

ev	The JACK event to be set.	
arg	The pointer to the jack_assistant object. Currently not checked for nullity.	

12.2.4.61 invalid_key()

12.2.4.62 keyval_name()

```
std::string seq64::keyval_name (
          unsigned int key )
```

In gtkmm, this is done via the gdk_keyval_name() function. Here, in the base class, we just provide an easy-to-create string. Note that this is a free function, not a class member.

Parameters

key	Provides the key-number to be converted to a key name.
-----	--

Returns

Returns the key name as looked up by the GDK infrastructure. If the key is not found, then an empty string is returned.

12.2.4.63 keyval_normalize()

Otherwise, random values, unchecked, can cause the application to crash.

Any field that is 0 or greater than 65536 is fixed. Not perfect, but better than allowing random values to be used.

Parameters

k The structure to be validated and normalized.

12.2.4.64 create_lash_driver()

Initializes the lash driver (strips lash-specific command line arguments), then connects to the LASH daemon and polls events.

This function will always be called from the main routine, and called only once. Note that we don't need that darn SEQ64_LASH_SUPPORT macro in client code anymore.

Parameters

р	The perform object that needs to implement LASH support.
argc	The number of command-line arguments.
argv	The command-line arguments.

Returns

This function returns true if a lash object was created. This function will not create one if not configured to, if the command-line options did not specify the creation of the LASH driver, or if the LASH driver was already created.

12.2.4.65 lash_driver()

```
lash * seq64::lash_driver ( )
```

Returns

Returns the pointer to the LASH driver if it exists. Otherwise a null pointer is returned. The caller *must always check* the return value.

12.2.4.66 delete_lash_driver()

```
void seq64::delete_lash_driver ( )
```

This function will always be called from the main routine, once. The other lash-pointer functions will know if the pointer has been deleted.

12.2.4.67 millisleep()

12.2.4.68 is_null_midipulse()

By "null" in this case, we mean "unusable", not 0. Sigh, it's always something.

12.2.4.69 output_thread_func()

Set up the performance, set the process to realtime privileges, and then start the output function.

Parameters

myperf Provides the perform object instance that is to be used. Its output_func() is called. Currently, this parameter is not validated, for speed.

Returns

Always returns nullptr.

12.2.4.70 input_thread_func()

Parameters

myperf Provides the perform object instance that is to be used. Its output_func() is called. Currently, this parameter is not validated, for speed.

Returns

Always returns nullptr.

12.2.4.71 rc()

```
rc_settings & seq64::rc ( )
```

Why a function instead of direct variable access? Encapsulation. We are then free to change the way "global" settings are accessed, without changing client code.

Returns

Returns the global object g_rc_settings.

12.2.4.72 usr()

```
user_settings & seq64::usr ( )
```

Returns

Returns the global object g user settings.

12.2.4.73 choose_ppqn()

Putting it here means we can reduce the reliance on the global ppgn.

However, this function works completely only if the "user" configuration file has already been read. In some cases we may need to retrofit the desired PPQN value!

Parameters

Returns

Returns the ppqn parameter, unless that parameter is SEQ64_USE_DEFAULT_PPQN (-1), then usr().midi ← __ppqn is returned. If that value is also -1, then we return SEQ64_DEFAULT_PPQN (192).

12.2.4.74 timestring_to_pulses() [2/2]

Parameters

	timestring	The time value to be converted, which must be of the form "hh:mm:ss" or "hh:mm:ss.fraction".
	bpm	The beats-per-minute tempo (e.g. 120) of the current MIDI song.
ĺ	ppqn	The parts-per-quarter note precision (e.g. 192) of the current MIDI song.

Returns

Returns 0 if an error occurred or if the number actually translated to 0.

This conversion assumes that the fractional parts of the seconds is padded with zeroes on the left or right to 6 digits.

12.2.4.75 jack_dummy_callback()

Returns

Does nothing, always returns 0.

12.2.4.76 seq_app_name()

```
const std::string& seq64::seq_app_name ( )
```

We could continue to use the macro SEQ64_APP_NAME, but we might eventually want to make this name configurable. Not too likely, but possible.

Returns

Returns SEQ64_APP_NAME.

12.2.4.77 seq_client_name()

```
const std::string& seq64::seq_client_name ( )
```

We could continue to use the macro SEQ64_CLIENT_NAME, but we might eventually want to make this name configurable. More likely to be a configuration option in the future.

Returns

Returns SEQ64_CLIENT_NAME.

12.2.4.78 seq_version()

```
const std::string& seq64::seq_version ( )
```

We could continue to use the macro SEQ64_VERSION, but ... let's be consistent. :-D

Returns

Returns SEQ64_VERSION.

12.2.4.79 make_section_name()

ſ	label	The base-name of the section.
	value	The numeric value to append to the section name.

Returns

Returns a string of the form "[basename-1]".

12.2.4.80 font_render()

```
font& seq64::font_render ( ) [inline]
```

We've going to render this pointer obsolete, though, and use a smart factory function to ensure the existence of this pointer, and return a reference to the font object.

We wanted to make the font a const object, but mainwid::on_realize() calls the font::init() function with its window object, and using const is impractical. We don't want to force every caller to deal with the overhead of passing even a null window pointer, either.

However, at some point we need some quarantee that the init() function is called before rendering a string. Right now, we guarantee it only by build order.

Returns

Returns a reference to the object pointed to by sp_font_renderer.

12.2.4.81 adjustment_dummy()

```
Gtk::Adjustment & seq64::adjustment_dummy ( )
```

This static object is used so we have an Adjustment to assign to the Adjustment members for classes that don't use them. Clumsy? We shall see.

Anyway, the parameters for this constructor are value, lower, upper, step-increment, and two more values.

It's a shame that GdkEventAny doesn't also encapsulate the keyboard state, since that is also available for other events, such as scroll events.

ev The keystroke event to be tested.

Parameters

ev The keystroke event to be tested.

12.2.4.84 is_no_modifier()

Parameters

ev The scroll event to be tested.

```
12.2.4.85 is_ctrl_key() [2/3]
```

Parameters

ev The keystroke event to be tested.

12.2.4.86 is_shift_key() [2/3]

ev The keystroke event to be tested.

```
12.2.4.87 is_ctrl_key() [3/3]
```

Parameters

ev The keystroke event to be tested.

```
12.2.4.88 is_shift_key() [3/3]
```

Parameters

ev The keystroke event to be tested.

12.2.4.89 is_ctrl_shift_key()

```
bool seq64::is_ctrl_shift_key ( {\tt GdkEventButton} \ *\ ev\ )
```

Parameters

ev The keystroke event to be tested.

12.2.4.90 is_super_key()

Basically just masks off the MOD4 bit; the "safe" method does not work for this key.

ev The keystroke event to be tested.

12.2.4.91 test_widget_click()

12.2.4.92 update_mainwid_sequences()

```
void seq64::update_mainwid_sequences ( )
```

It is used by other objects that can modify the currently-edited sequence shown in the mainwid (main window).

12.2.4.93 update_perfedit_sequences()

```
void seq64::update_perfedit_sequences ( )
```

It is used by other objects (sequence shown in the perfedit (song window).

12.2.4.94 FF_RW_timeout()

```
int seq64::FF_RW_timeout ( void * arg ) \quad [inline]
```

Parameters

arg Provides a putative pointer to the perform object that actually implements the timeout functionality.

Returns

Returns the value of the perform::FF_RW_timeout() call if seq32 transport support is enabled and the arg parameter is good, otherwise false is returned.

12.2.4.95 clamp() [1/2]

```
static long seq64::clamp (

long val,
```

```
long low,
long hi ) [inline], [static]
```

12.2.4.96 clamp() [2/2]

12.2.4.97 silence_jack_errors()

```
void seq64::silence_jack_errors (
          bool silent )
```

Probably not good to silence this output, but let's provide the option, for the sake of symmetry, consistency, what have you.

12.2.4.98 silence_jack_info()

We were getting way too many informational message, to the point of obscuring the debug and error output.

12.2.4.99 midi_api_name()

Parameters

i The integer value code for the desired API. Must range from int(RTMIDI_API_UNSPECIFIED) to int(RTMIDI_API_DUMMY).

Returns

Returns a human-readable name for the API.

12.2.4.100 midi_probe()

```
int seq64::midi_probe ( )
```

We will upgrade this function for some better testing eventually. It uses the functionality of the midi_info/rtmidi_info objects, plus its own version of some of that functionality.

Returns

Currently always returns 0.

12.2.4.101 midi_input_test()

12.2.4.102 min()

```
long seq64::min (
          long a,
          long b) [inline]
```

Parameters

а	First operand.
b	Second operand.

Returns

Returns the minimum value of a and b.

12.2.4.103 jack_process_rtmidi_input()

It is the JACK process callback for a MIDI output port (e.g. "system:midi_capture_1", which gives us the output of the Korg nanoKEY2 MIDI controller), also known as a "Readable Client" by qjackctl. It does the following:

This callback receives data from JACK and gives it to our application's input port.

This function does the following:

```
-# Get the JACK port buffer and the MIDI event-count into this
   buffer.
-# For each MIDI event, get the event from JACK and push it into a
   local midi_message object.
-# Get the event time, converting it to a delta time if possible.
-# If it is not a SysEx continuation, then:
   -# If we're using a callback, pass the data to that callback. Do
    we need this callback to interface with the midibus-based
   code?
-# Otherwise, add the midi_message container to the rtmidi input
   queue. One can then grab this data in a midibase ::
   poll_for_midi() call. We still ought to check the add
   success.
```

The ALSA code polls for events, and that model is also available here. We're still working exactly how it will work hest

This function used to be static, but now we make if available to midi_jack_info. Also note the s_null_detected flag. It is used only to have the apiprint() debug messages appear only once, for better trouble-shooting.

Parameters

nframes	The frame number to be processed.
arg	A pointer to the midi_jack_data structure to be processed.

Returns

Returns 0.

12.2.4.104 jack_process_rtmidi_output()

It is the JACK process callback for a MIDI input port (a midi_in_jack object associated with, for example, "system :midi_playback_1", representing, for example, a Korg nanoKEY2 to which we can send information), also known as a "Writable Client" by gjacketl. Here's how it works:

```
-# Get the JACK port buffer, for our local jack port. Clear it.
-# Loop while the number of bytes available for reading [via jack_ringbuffer_read_space()] is non-zero. Note that the second parameter is where the data is copied.
-# Get the size of each event, and allocate space for an event to be written to an event port buffer (the JACK "reserve" function).
-# Read the data from the ringbuffer into this port buffer. JACK should then send it to the remote port.
```

Since this is an output port, "buff" is the area to which we can write data, to send it to the "remote" (i.e. outside our application) port. The data is written to the ringbuffer in api_init_out(), and here we read the ring buffer and pass it to the output buffer.

We were wondering if, like the JACK midiseq example program, we need to wrap out process in a for-loop over the number of frames. In our tests, we are getting 1024 frames, and the code seems to work without that loop.

nframes	The frame number to be processed.
arg	A pointer to the JackMIDIData structure to be processed.

Returns

Returns 0.

12.2.4.105 jack_process_io()

```
int seq64::jack_process_io (
          jack_nframes_t nframes,
          void * arg )
```

12.2.4.106 jack_message_bit_bucket()

12.2.4.107 midi_input_callback()

12.2.5 Variable Documentation

12.2.5.1 c_controller_names

```
std::string seq64::c_controller_names
```

This array is used only by the seqedit class.

12.2.5.2 EVENT_STATUS_BIT

```
const midibyte seq64::EVENT_STATUS_BIT
```

12.2.5.3 **EVENT_ANY**

```
const midibyte seq64::EVENT_ANY
```

The following MIDI events are channel messages. The comments represent the one or two data-bytes of the message.

Note that Channel Mode Messages use the same code as the Control Change, but uses reserved controller numbers ranging from 122 to 127.

The EVENT_ANY (0x00) value may prove to be useful in allowing any event to be dealt with. Not sure yet, but the cost is minimal.

12.2.5.4 EVENT_NOTE_OFF

```
const midibyte seq64::EVENT_NOTE_OFF
```

12.2.5.5 EVENT_NOTE_ON

```
const midibyte seq64::EVENT_NOTE_ON
```

12.2.5.6 EVENT_AFTERTOUCH

```
const midibyte seq64::EVENT_AFTERTOUCH
```

12.2.5.7 EVENT_CONTROL_CHANGE

```
const midibyte seq64::EVENT_CONTROL_CHANGE
```

12.2.5.8 EVENT_PROGRAM_CHANGE

```
const midibyte seq64::EVENT_PROGRAM_CHANGE
```

12.2.5.9 EVENT_CHANNEL_PRESSURE

```
const midibyte seq64::EVENT_CHANNEL_PRESSURE
```

12.2.5.10 EVENT_PITCH_WHEEL

```
const midibyte seq64::EVENT_PITCH_WHEEL
```

12.2.5.11 EVENT_MIDI_SYSEX

```
const midibyte seq64::EVENT_MIDI_SYSEX
```

The following MIDI events have no channel. We have included redundant constant variables for the SysEx Start and End bytes just to make it clear that they are part of this sequence of values, though usually treated separately.

Only the following constants are followed by some data bytes:

```
- EVENT_MIDI_SYSEX = 0xF0

- EVENT_MIDI_QUARTER_FRAME = 0xF1 // undefined?

- EVENT_MIDI_SONG_POS = 0xF2

- EVENT_MIDI_SONG_SELECT = 0xF3
```

A MIDI System Exclusive (SYSEX) message starts with F0, followed by the manufacturer ID (how many? bytes), a number of data bytes, and ended by an F7.

12.2.5.12 EVENT_MIDI_QUARTER_FRAME

```
const midibyte seq64::EVENT_MIDI_QUARTER_FRAME
```

12.2.5.13 EVENT_MIDI_SONG_POS

```
const midibyte seq64::EVENT_MIDI_SONG_POS
```

12.2.5.14 EVENT_MIDI_SONG_SELECT

```
const midibyte seq64::EVENT_MIDI_SONG_SELECT
```

12.2.5.15 EVENT_MIDI_SONG_F4

```
const midibyte seq64::EVENT_MIDI_SONG_F4
```

12.2.5.16 EVENT_MIDI_SONG_F5

```
const midibyte seq64::EVENT_MIDI_SONG_F5
```

12.2.5.17 EVENT_MIDI_TUNE_SELECT

```
const midibyte seq64::EVENT_MIDI_TUNE_SELECT
```

12.2.5.18 EVENT_MIDI_SYSEX_END

```
const midibyte seq64::EVENT_MIDI_SYSEX_END
```

12.2.5.19 EVENT_MIDI_SYSEX_CONTINUE

```
const midibyte seq64::EVENT_MIDI_SYSEX_CONTINUE
```

12.2.5.20 EVENT_MIDI_CLOCK

```
const midibyte seq64::EVENT_MIDI_CLOCK
```

12.2.5.21 EVENT_MIDI_SONG_F9

```
const midibyte seq64::EVENT_MIDI_SONG_F9
```

12.2.5.22 EVENT_MIDI_START

const midibyte seq64::EVENT_MIDI_START

12.2.5.23 EVENT_MIDI_CONTINUE

const midibyte seq64::EVENT_MIDI_CONTINUE

12.2.5.24 EVENT_MIDI_STOP

const midibyte seq64::EVENT_MIDI_STOP

12.2.5.25 EVENT_MIDI_SONG_FD

const midibyte seq64::EVENT_MIDI_SONG_FD

12.2.5.26 EVENT_MIDI_ACTIVE_SENS

const midibyte seq64::EVENT_MIDI_ACTIVE_SENS

12.2.5.27 EVENT_MIDI_RESET

const midibyte seq64::EVENT_MIDI_RESET

12.2.5.28 EVENT_MIDI_META

const midibyte seq64::EVENT_MIDI_META

12.2.5.29 EVENT_NULL_CHANNEL

const midibyte seq64::EVENT_NULL_CHANNEL

However, it also means that the channel is encoded in the m_status byte itself. This is our work around to be able to hold a multi-channel SMF 0 track in a sequence. In a Sequencer64 SMF 0 track, every event has a channel. In a Sequencer64 SMF 1 track, the events do not have a channel. Instead, the channel is a global value of the sequence, and is stuffed into each event when the event is played or is written to a MIDI file.

12.2.5.30 EVENT_GET_CHAN_MASK

const midibyte seq64::EVENT_GET_CHAN_MASK

12.2.5.31 EVENT_CLEAR_CHAN_MASK

```
const midibyte seq64::EVENT_CLEAR_CHAN_MASK
```

12.2.5.32 EVENTS_ALL

```
const int seq64::EVENTS_ALL
```

We reversed the parts of each token for consistency with the macros defined above.

12.2.5.33 EVENTS_UNSELECTED

```
const int seq64::EVENTS_UNSELECTED
```

12.2.5.34 c_midibus_output_size

```
const int seq64::c_midibus_output_size
```

These constants were also defined in midibus_portmidi.h, but we made them common to both implementations here.

The c_midibus_output_size value is passed, in mastermidibus, to snd_seq_set_output_buffer_size(). Not sure if the value needs to be so large.

12.2.5.35 c_midibus_input_size

```
const int seq64::c_midibus_input_size
```

Not sure if the value needs to be so large.

12.2.5.36 c_midibus_sysex_chunk

```
const int seq64::c_midibus_sysex_chunk
```

12.2.5.37 c_midibus

```
const midilong seq64::c_midibus
```

Some of the information is stored with each track (and in the midi_container-derived classes), and some is stored in the proprietary header.

Track (sequencer-specific) data:

```
c_midibus
c_midich
c_timesig
c_triggers (deprecated)
c_triggers_new
c_musickey (can be in footer, as well)
c_musicscale (ditto)
c_backsequence (ditto)
c transpose
```

Footer ("proprietary") data:

```
c_midictrl
c_midiclocks
c_notes
c_bpmtag (beats per minute)
c_mutegroups
c_perf_bp_mes (perfedit's beats-per-measure setting)
c_perf_bw (perfedit's beat-width setting)
```

Also see the PDF file in the following project for more information about the "proprietary" data:

https://github.com/ahlstromcj/sequencer64-doc.git

Note that the track data is read from the MIDI file, but not written directly to the MIDI file. Instead, it is stored in the MIDI container as sequences are edited to used these "sequencer-specific" features. Also note that c_triggers has been replaced by c_triggers_new as the code that marks the triggers stored with a sequence.

As an extension, we can also grab the key, scale, and background sequence value selected in a sequence and write these values as track data, where they can be read in and applied to a specific sequence, when the sequence object is created. These values would not be stored in the legacy format.

Something like this could be done in the "user" configuration file, but then the key and scale would apply to all songs. We don't want that.

We could also add snap and note-length to the per-song defaults, but the "user" configuration file seems like a better place to store these preferences.

Note

The new value c_transpose value is from Stazed's seq32 project. The code to support this option is turned on via the build-configurable SEQ64_STAZED_TRANSPOSE macro, but here we reserved the value even if that option is not enabled by the user. There are additional values from Stazed/seq32, not yet used.Track buss number.

```
12.2.5.38 c_midich
const midilong seq64::c_midich
12.2.5.39 c_midiclocks
const midilong seq64::c_midiclocks
12.2.5.40 c_triggers
const midilong seq64::c_triggers
12.2.5.41 c_notes
const midilong seq64::c_notes
12.2.5.42 c_timesig
const midilong seq64::c_timesig
12.2.5.43 c_bpmtag
const midilong seq64::c_bpmtag
12.2.5.44 c_triggers_new
const midilong seq64::c_triggers_new
12.2.5.45 c_mutegroups
```

const midilong seq64::c_mutegroups

12.2.5.46 c_midictrl

```
const midilong seq64::c_midictrl
```

12.2.5.47 c_musickey

```
const midilong seq64::c_musickey
```

12.2.5.48 c_musicscale

```
const midilong seq64::c_musicscale
```

12.2.5.49 c_backsequence

```
const midilong seq64::c_backsequence
```

12.2.5.50 c_transpose

```
const midilong seq64::c_transpose
```



```
const midilong seq64::c_perf_bp_mes
```

12.2.5.52 c_perf_bw

```
const midilong seq64::c_perf_bw
```

```
12.2.5.53 c_midi_track_ctrl
```

```
const int seq64::c_midi_track_ctrl
```

The lowest value is $c_{seqs_in_set} * 2 = 64$.

I think the reason for that value is to perhaps handle two sets or something like that. Will figure it out later.

The controls are read in from the "rc" configuration files, and are written to the c_midictrl section of the "proprietary" final track in a Seq24/Sequencer64 MIDI file.

Note that we are adding some more MIDI control entries to support the following additional functions:

- Start
- Pause
- Stop
- (any more???)

To help with backward compatibility, the old c_midi_controls limit is supplemented with a new, higher limit, c_midi ← _controls_extended. We also add a number of placeholders so we don't have to adjust the new limit again later. To aid the transition, g_midi_control_limit replaces c_midi_controls, though, for now, it has the same value.

```
12.2.5.54 c_midi_control_bpm_up
```

```
const int seq64::c_midi_control_bpm_up
```

12.2.5.55 c_midi_control_bpm_dn

```
\verb|const| int seq64::c_midi_control_bpm_dn|\\
```

12.2.5.56 c_midi_control_ss_up

```
const int seq64::c_midi_control_ss_up
```

12.2.5.57 c_midi_control_ss_dn

```
\verb|const| int seq64::c_midi_control_ss_dn|\\
```

12.2.5.58 c_midi_control_mod_replace

```
const int seq64::c_midi_control_mod_replace
```

12.2.5.59 c_midi_control_mod_snapshot

const int seq64::c_midi_control_mod_snapshot

12.2.5.60 c_midi_control_mod_queue

 $\verb|const| int seq64::c_midi_control_mod_queue|\\$

12.2.5.61 c_midi_control_mod_gmute

const int seq64::c_midi_control_mod_gmute

12.2.5.62 c_midi_control_mod_glearn

const int seq64::c_midi_control_mod_glearn

12.2.5.63 c_midi_control_play_ss

const int seq64::c_midi_control_play_ss

12.2.5.64 c_midi_controls

const int seq64::c_midi_controls

12.2.5.65 c_midi_control_playback

const int seq64::c_midi_control_playback

12.2.5.66 c_midi_control_record

const int seq64::c_midi_control_record

12.2.5.67 c_midi_control_solo

const int seq64::c_midi_control_solo

12.2.5.68 c_midi_control_thru

const int seq64::c_midi_control_thru

12.2.5.69 c_midi_control_bpm_page_up

const int seq64::c_midi_control_bpm_page_up

12.2.5.70 c_midi_control_bpm_page_dn

const int seq64::c_midi_control_bpm_page_dn

const int seq64::c_midi_control_16

const int seq64::c_midi_control_17

const int seq64::c_midi_control_18

const int seq64::c_midi_control_19

12.2.5.75 c_midi_controls_extended

```
const int seq64::c_midi_controls_extended
```

12.2.5.76 g_midi_control_limit

```
int seq64::g_midi_control_limit
```

12.2.5.77 c_scales_policy

```
const bool seq64::c_scales_policy[c_scale_size][SEQ64_OCTAVE_SIZE]
```

See the following sites for more information:

- http://method-behind-the-music.com/theory/scalesandkeys/
- https://en.wikipedia.org/wiki/Heptatonic_scale
- https://en.wikibooks.org/wiki/Music_Theory/Scales_and_Intervals

Note that melodic minor descends in the same way as the natural minor scale, so it descends differently than it ascends. We don't deal with that trick, at all. In the following table, the scales all start with C, but seq24/sequencer64 allow other starting notes (e.g. "keys").

```
Chromatic
                                 C C# D D# E F F# G G# A A# B Notes, chord
Major
                                       C . D . E F . G . A . B
                                                         Eb .
Minor
                                       С
                                                   D
                                                                     F
                                                                                G
                                                                                       Ab .
                                                                                                  Bb .
Minor

Harmonic Minor

C . D Eb . F . G Ab . . B

Melodic Minor

C . D Eb . F . G . A . B Descending diff.

C Whole Tone

C . D . E . F# . G# . A# . C+7 chord

Blues

C . . Eb . F Gb G . . Bb .

      Major Pentatonic
      C . D . E . . . G . A . .

      Minor Pentatonic
      C . . Eb . F . G . . Bb .

      Octatonic 1
      C . D Eb . F Gb . Ab A . B

      Octatonic 2
      C Db . Eb E F F# G . A Bb .

                                                                                                                 Unimplemented
                                                                                                                 Unimplemented
```

12.2.5.78 c_scales_transpose_up

```
const int seq64::c_scales_transpose_up[c_scale_size][SEQ64_OCTAVE_SIZE]
```

For example, if we simply add 1 semitone to each note, it remains a minor key, but it is in a different minor key. Using the transpositions in these arrays, the minor key remains the same minor key.

Minor Transpose up Result up	C 2 D	0	D 1 D#	D# 2 F	0	F 2 G	0	G 1 G#	G# 2 A#	0	A# 2 C	0
Harmonic minor Transpose up Result up	C 2 D		D 1 Eb	Eb 2 F		F 2 G		G 1 Ab	Ab 3 B			B 1 C
Melodic minor Transpose up Result up	C 2 D		D 1 Eb	Eb 2 F		F 2 G		G 2 A		A 2 B		B 1 C
C Whole Tone Transpose up Result up	C 2 D		D 2 E	· ·			2	· ·	2			· ·
Blues Transpose up Result up	C 3 Eb			Eb 2 F		F 1 Gb	1	G 3 Bb			Bb 2 C	
Major Pentatonic Transpose up Result up	C 2 D		D 2 E		E 3 G			G 2 A		A 3 C		
Minor Pentatonic Transpose up Result up	C 3 Eb			Eb 2 F		F 2 G		G 3 Bb			Bb 2 C	

12.2.5.79 c_scales_transpose_dn

const int seq64::c_scales_transpose_dn[c_scale_size][SEQ64_OCTAVE_SIZE]

```
\mbox{\bf C} . \mbox{\bf D} . \mbox{\bf E} F . \mbox{\bf G} . \mbox{\bf A}
                     . 2
                                   . 2
Transpose down
                   1
                              2 1
                                               . 2
                           .
                   \tt B . C . D \tt E . F . \tt G
Result down
Minor
                   C . D D# . F
                                   . G G# . A# .
                        2
Transpose down
                   2
                           1
                                 2
                                       2
                                         1
Result down
                  A# . C D . D# . F G . G# .
                  C . D Eb . F . G Ab . . B
Harmonic minor
                  1 .
B .
                        2 1 . 2 .
C D . Eb .
                                      2 1 . .
F G . .
Transpose down
Result down
                  В
Melodic minor
                C . D Eb . F . G . A . B
                  1 . 2 1 . 2 . 2 . 2
B . C D . Eb . F . G
Transpose down
Result down
                  C . D
                                 . F# . G# . A# .
C whole tone
                           . E
Transpose down
                   2 . 2 . 2
                                 . 2 . 2 . 2
Result down
                   A# . C . D
                                 . E . F# . G# .
                     . . Eb . F Gb G
Blues
                        . 3 .
                   2
                                 2 1 1 .
Transpose down
                     .
                  Bb. . C . Eb F Gb. . G
Result down
Major Pentatonic C
                  C . D . E 3 . 2
                                 . . G
                                            Α
Transpose down
                                       3
                                            2
Result down
                     . C . D .
                                   . E . G
Minor Pentatonic \, C \, . \, Eb \, F \, G \, . \, Bb \, .
                  2 . . 3 . 2 . 2 .
Bb . . C . Eb . F .
Transpose down
Result down
```

```
const char seq64::c_scales_text[c_scale_size][20]
```

12.2.5.81 c_key_text

```
const char seq64::c_key_text[SEQ64_OCTAVE_SIZE][4]
```

12.2.5.82 c_interval_text

```
const char seq64::c_interval_text[16][4]
```

12.2.5.83 c_chord_text

```
const char seq64::c_chord_text[8][6]
```

However, we have not seen this menu in the GUI! Ah, it only appears if the user has selected a musical scale like Major or Minor.

12.2.5.84 c_chord_number

```
const int seq64::c_chord_number
```

The chord-number is a count of the number of entries in c_chord_table_text. Will never change, luckily.

12.2.5.85 c_chord_table_text

```
const char seq64::c_chord_table_text[c_chord_number][12]
```

These chords appear in the sequence-editor chord-button dropdown menu. The longest string is 11 characters, and we add one for the null terminator. A good case for using std::string here. :-)

12.2.5.86 c_chord_size

```
const int seq64::c_chord_size
```

```
const int seq64::c_chord_table[c_chord_number][c_chord_size]
```

These values indicate the note offsets needed for a particular kind of chord. 0 means no offset, and a -1 ends the list of note offsets for the chord.

```
12.2.5.88 c_max_instruments
```

```
const int seq64::c_max_instruments
```

With a value of 64, this is more of a sanity-check than a realistic number of instruments defined by a user.

```
12.2.5.89 c_max_busses
```

```
const int seq64::c_max_busses
```

12.2.5.90 versiontext

```
const std::string seq64::versiontext [static]
```

This value ultimately comes from the configure.ac script.

This was too redundant:

```
SEQ64_PACKAGE " " SEQ64_VERSION " (" SEQ64_GIT_VERSION ") " DATE "\n"
```

This is out-of-date:

```
SEQ64_PACKAGE " " SEQ64_GIT_VERSION " " DATE "\n";
```

```
12.2.5.91 long_options
```

```
struct option seq64::long_options[] [static]
```

Note the terminating null structure..

```
12.2.5.92 s_arg_list
```

```
const std::string seq64::s_arg_list [static]
```

The following string keeps track of the characters used so far. An 'x' means the character is used; an 'o' means it is used for the legacy spelling of the option, which uses underscores instead of hyphens. An 'a' indicates we could repurpose the key with minimal impact.

Previous arg-list, items missing! "ChVH:lRrb:q:Lni:jJmaAM:pPusSU:x:"

```
12.2.5.93 s_help_1a
const char* const seq64::s_help_la [static]
12.2.5.94 s_help_1b
const char* const seq64::s_help_1b [static]
12.2.5.95 s_help_2
const char* const seq64::s_help_2 [static]
12.2.5.96 s_help_3
const char* const seq64::s_help_3 [static]
12.2.5.97 s_help_4
const char* const seq64::s_help_4 [static]
12.2.5.98 s build alsamidi support
const std::string seq64::s_build_alsamidi_support [static]
12.2.5.99 s_build_portmidi_support
const std::string seq64::s_build_portmidi_support [static]
12.2.5.100 s_build_rtmidi_support
const std::string seq64::s_build_rtmidi_support [static]
```

12.2.5.101 s_build_highlight_empty

const std::string seq64::s_build_highlight_empty [static]

12.2.5.102 s_build_lash_support

const std::string seq64::s_build_lash_support [static]

12.2.5.103 s_build_jack_support

const std::string seq64::s_build_jack_support [static]

12.2.5.104 s_build_jack_session

const std::string seq64::s_build_jack_session [static]

12.2.5.105 s_event_editor

const std::string seq64::s_event_editor [static]

12.2.5.106 s_build_pause_support

const std::string seq64::s_build_pause_support [static]

12.2.5.107 s_build_use_event_map

const std::string seq64::s_build_use_event_map [static]

12.2.5.108 s_build_chord_generator

const std::string seq64::s_build_chord_generator [static]

12.2.5.109 s_build_edit_highlight

const std::string seq64::s_build_edit_highlight [static]

12.2.5.110 s_build_timesig_tempo

const std::string seq64::s_build_timesig_tempo [static]

12.2.5.111 s_build_midi_vector

const std::string seq64::s_build_midi_vector [static]

12.2.5.112 s_build_solid_grid

const std::string seq64::s_build_solid_grid [static]

12.2.5.113 s_build_follow_progress

const std::string seq64::s_build_follow_progress [static]

12.2.5.114 s_statistics_support

const std::string seq64::s_statistics_support [static]

12.2.5.115 s_strip_empty_mutes

const std::string seq64::s_strip_empty_mutes [static]

const std::string seq64::s_seq32_jack_support [static]


```
const std::string seq64::s_seq32_transport [static]
```

12.2.5.118 s_seq32_transpose

```
const std::string seq64::s_seq32_transpose [static]
```

12.2.5.119 s_seq32_menu_buttons

```
const std::string seq64::s_seq32_menu_buttons [static]
```

12.2.5.120 s_seq32_lfo_support

```
const std::string seq64::s_seq32_lfo_support [static]
```

12.2.5.121 s_debug_mode

```
const std::string seq64::s_debug_mode [static]
```

12.2.5.122 s_status_pairs

```
jack_status_pair_t seq64::s_status_pairs[]
```

Terminated by a 0 value and an empty string.

12.2.5.123 s_character_mapping

```
struct charpair_t seq64::s_character_mapping[]
```

12.2.5.124 s_global_lash_driver

```
lash* seq64::s_global_lash_driver [static]
```

It is actually hidden in this module now, so that a function can be used in its place.

Like the font renderer, This item was once created in the main module, sequencer64.cpp. Now we make it a safer, more fool-proof, function. However, unlike the font-render, which always exists, the LASH driver is conditional, and might not be wanted. Therefore, we cannot return a reference, because there's no such thing as a null reference in C++. We have to return a pointer.

12.2.5.125 c_status_replace

```
const int seq64::c_status_replace [static]
```

Note how they specify different bit values, as it they could be masked together to signal multiple functions.

This value signals the "replace" functionality. If this bit is set, then perform::sequence_playing_toggle() unsets this status and calls perform::off sequences(), which calls sequence::set playing(false) for all active sequences.

It works like this:

```
The user presses the Replace key, or the MIDI control message for
    c_midi_control_mod_replace is received.
   This bit is OR'd into perform::m_control_status. This status bit
    is used in perform::sequence_playing_toggle().
       Called in perform::sequence_key() so that keystrokes in
        the main window toggle patterns in the main window.
        Called in peform::toggle_other_segs() to implement
       Shift-click to toggle all other patterns but the one
        clicked.
       Called in seqmenu::toggle_current_sequence(), called in
       mainwid to implement clicking on a pattern.
       Also used in MIDI control to toggle patterns 0 to 31,
       offset by the screen-set.
       perform::sequence_playing_off(), similarly used in MIDI control.
       perform::sequence_playing_on(), similarly used in MIDI control.
-# When the key is released, this bit is AND'd out of
    perform::m_control_status.
Both the MIDI control and the keystoke set the sequence to be
```

12.2.5.126 c status snapshot

```
const int seq64::c_status_snapshot [static]
```

By default, perform::sequence_playing_toggle() calls sequence::toggle_playing() on the given sequence number, plus what is noted for c status snapshot.

It works like this:

"replaced".

```
-# The user presses the Snapshot key.
-# This bit is OR'd into perform::m_control_status.
-# The playing state of the patterns is saved by perform::save_playing_state().
-# When the key is released, this bit is AND'd out of perform::m_control_status.
-# The playing state of the patterns is restored by perform::restore_playing_state().
```

12.2.5.127 c_status_queue

```
const int seq64::c_status_queue [static]
```

If this bit is set, then perform::sequence_playing_toggle() calls sequence::toggle_queued() on the given sequence number.

12.2.5.128 g_rc_settings

```
rc_settings seq64::g_rc_settings [static]
```

12.2.5.129 g_user_settings

```
user_settings seq64::g_user_settings [static]
```

12.2.5.130 s_handlesize [1/2]

```
const long seq64::s_handlesize [static]
```

12.2.5.131 s_jitter_amount

```
const int seq64::s_jitter_amount [static]
```

12.2.5.132 gs_mainwid_pointer

```
mainwid* seq64::gs_mainwid_pointer [static]
```

We have decided that passing along a mainwnd reference among a number of constructors is too much and actually harder to understand and more error prone. This value is set at the end of the mainwnd constructor, but only the first time that constructor is called.

12.2.5.133 c_mainwid_x

```
const int seq64::c_mainwid_x
```

Affected by the c_mainwid_border and c_mainwid_spacing values.

```
12.2.5.134   c_mainwid_y

const int seq64::c_mainwid_y

12.2.5.135   gs_perfedit_pointer_0

perfedit* seq64::gs_perfedit_pointer_0   [static]

12.2.5.136   gs_perfedit_pointer_1

perfedit* seq64::gs_perfedit_pointer_1   [static]

12.2.5.137   s_handlesize [2/2]

const long seq64::s_handlesize [static]
```

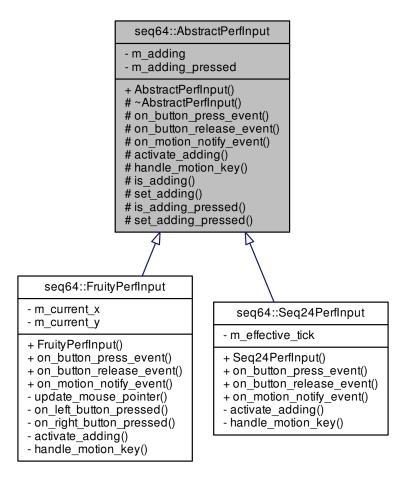
Chapter 13

Data Structure Documentation

13.1 seq64::AbstractPerfInput Class Reference

Provides an abstract base class to provide the minimal interface for the various "perf input" classes.

Inheritance diagram for seq64::AbstractPerfInput:



Public Member Functions

AbstractPerfInput ()

Default constructor.

Protected Member Functions

- virtual ∼AbstractPerfInput ()
 - Destructor, does nothing.
- virtual bool on_button_press_event (GdkEventButton *a_ev, perfroll &roll)=0
- virtual bool on_button_release_event (GdkEventButton *a_ev, perfroll &roll)=0
- virtual bool on_motion_notify_event (GdkEventMotion *a_ev, perfroll &roll)=0
- virtual void activate_adding (bool adding, perfroll &roll)=0
- virtual bool handle_motion_key (bool is_left, perfroll &roll)=0
- bool is_adding () const

'Getter' function for member m_adding

void set_adding (bool flag)

'Setter' function for member m_adding

• bool is_adding_pressed () const

'Getter' function for member m_adding_pressed

void set_adding_pressed (bool flag)

'Setter' function for member m_adding_pressed

Private Attributes

· bool m_adding

Indicates we are in the middle of adding a sequence segment to the performance.

bool m_adding_pressed

Indicates if the left mouse button is pressed while in adding mode.

Friends

· class perfroll

13.1.1 Constructor & Destructor Documentation

```
13.1.1.1 AbstractPerfInput()
```

```
seq64::AbstractPerfInput::AbstractPerfInput ( ) [inline]
```

13.1.1.2 ~AbstractPerfInput()

virtual seq64::AbstractPerfInput::~AbstractPerfInput () [inline], [protected], [virtual]

13.1.2 Member Function Documentation

13.1.2.1 on_button_press_event()

Implemented in seq64::Seq24PerfInput, and seq64::FruityPerfInput.

13.1.2.2 on_button_release_event()

Implemented in seq64::Seq24PerfInput, and seq64::FruityPerfInput.

13.1.2.3 on_motion_notify_event()

Implemented in seq64::Seq24PerfInput, and seq64::FruityPerfInput.

13.1.2.4 activate_adding()

Implemented in seq64::Seq24PerfInput, and seq64::FruityPerfInput.

13.1.2.5 handle_motion_key()

Implemented in seq64::Seq24PerfInput, and seq64::FruityPerfInput.

13.1.3 Friends And Related Function Documentation

13.1.3.1 perfroll

friend class perfroll [friend]

13.1.4 Field Documentation

13.1.4.1 m_adding

bool seq64::AbstractPerfInput::m_adding [private]

13.1.4.2 m_adding_pressed

```
bool seq64::AbstractPerfInput::m_adding_pressed [private]
```

13.2 seq64::automutex Class Reference

Provides a mutex that locks automatically when created, and unlocks when destroyed.

Public Member Functions

automutex (mutex &my_mutex)

Principal constructor gets a reference to a mutex parameter, and then locks the mutex.

• \sim automutex ()

The destructor unlocks the mutex.

Private Member Functions

- automutex ()
- automutex (const automutex &)
- automutex & operator= (const automutex &)

Private Attributes

• mutex & m_safety_mutex

Provides the mutex reference to be used for locking.

13.2.1 Detailed Description

This has a couple of benefits. First, it is threadsafe in the face of exception handling. Secondly, it can be done with just one line of code.

13.2.2 Constructor & Destructor Documentation

Parameters

my_mutex	The caller's mutex to be used for locking.	1
----------	--	---

13.2.2.4 ∼automutex()

```
seq64::automutex::\sim automutex ( ) [inline]
```

13.2.3 Member Function Documentation

13.2.3.1 operator=()

13.2.4 Field Documentation

13.2.4.1 m_safety_mutex

```
mutex& seq64::automutex::m_safety_mutex [private]
```

13.3 seq64::busarray Class Reference

Holds a number of businfo objects.

Public Member Functions

• busarray ()

A new class to hold a number of MIDI busses and flags for more controlled access than using arrays of booleans and pointers.

∼busarray ()

Removes components from the container.

bool add (midibus *bus, clock_e clock)

Creates and adds a new midibus object to the list.

bool add (midibus *bus, bool inputing)

Creates and adds a new midibus object to the list.

bool initialize ()

Initializes all busses.

- · int count () const
- midibus * bus (bussbyte b)
- void start ()

Starts all of the busses; used for output busses only, but no check is made at present.

void stop ()

Stops all of the busses; used for output busses only, but no check is made at present.

void continue_from (midipulse tick)

Continues from the given tick for all of the busses; used for output busses only.

void init clock (midipulse tick)

Initializes the clocking at the given tick for all of the busses; used for output busses only.

void clock (midipulse tick)

Clocks at the given tick for all of the busses; used for output busses only.

void sysex (event *ev)

Handles SysEx events; used for output busses.

void play (bussbyte bus, event *e24, midibyte channel)

Plays an event, if the bus is proper.

• bool set_clock (bussbyte bus, clock_e clocktype)

Sets the clock type for the given bus, usually the output buss.

· void set all clocks ()

Sets the clock type for all busses, usually the output buss.

clock e get clock (bussbyte bus)

Gets the clock type for the given bus, usually the output buss.

• std::string get_midi_bus_name (int bus)

Get the MIDI output buss name (i.e.

void print ()

Print some information about the available MIDI output busses.

void port_exit (int client, int port)

Turn off the given port for the given client.

bool set_input (bussbyte bus, bool inputing)

Set the status of the given input buss, if a legal buss number.

void set_all_inputs ()

Set the status of all input busses.

bool get_input (bussbyte bus)

Get the input for the given (legal) buss number.

bool is_system_port (bussbyte bus)

Get the system-port status for the given (legal) buss number.

bool poll_for_midi ()

Initiate a poll() on the existing poll descriptors.

bool get_midi_event (event *inev)

Gets the first MIDI event in finds on an input bus.

• int replacement_port (int bus, int port)

Provides a function to use in api_port_start(), to determine if the port is to be a "replacement" port.

Private Attributes

std::vector < businfo > m_container

The full set of businfo objects, only some of which will actually be used.

13.3.1 Constructor & Destructor Documentation

13.3.1.1 busarray()

```
seq64::busarray::busarray ( )
```

13.3.1.2 \sim busarray()

```
seq64::busarray::\sim busarray ( )
```

However, now that we swap containers, we cannot call this functionality, because it deletes the bus's midibus pointer and nullifies it.

DISABLED, BUT WE NEED A WAY TO CLEAN UP AT EXIT TIME!!!

13.3.2 Member Function Documentation

```
13.3.2.1 add() [1/2]
```

Then the clock value is set. This function is meant for output ports.

We need to belay the initialization until later, when we know the configured clock settings for the output ports. So initialization has been removed from the constructor and moved to the initialize() function.

Parameters

bus	The midibus to be hooked into the array of busses.
clock	The clocking value for the bus.

Returns

Returns true if the bus was added successfully, though, really, it cannot fail.

Then the inputing value is set. This function is meant for input ports.

We need to belay the initialization until later, when we know the configured inputing settings for the input ports. So initialization has been removed from the constructor and moved to the initialize() function. However, now we know the configured status and can apply it right away.

Parameters

bus	The midibus to be hooked into the array of busses.
inputing	The input flag value for the bus. If true, this value indicates that the user has selected this bus to be the input MIDI bus.

Returns

Returns true if the bus was added successfully, though, really, it cannot fail.

13.3.2.3 initialize()

```
bool seq64::busarray::initialize ( )
```

Not sure we need this function.

Returns

Returns true if all busses initialized successfully.

13.3.2.4 count()

```
int seq64::busarray::count ( ) const [inline]
```

```
13.3.2.5 bus()
midibus* seq64::busarray::bus (
             bussbyte b ) [inline]
13.3.2.6 start()
void seq64::busarray::start ( )
13.3.2.7 stop()
void seq64::busarray::stop ( )
13.3.2.8 continue_from()
void seq64::busarray::continue_from (
              midipulse tick )
Parameters
 tick Provides the tick value for all busses to continue from.
13.3.2.9 init_clock()
void seq64::busarray::init_clock (
              midipulse tick )
Parameters
 tick | Provides the tick value for all busses use as the clock tick.
13.3.2.10 clock()
void seq64::busarray::clock (
```

midipulse tick)

Parameters

tick Provides the tick value for all busses use as the clock tick.

13.3.2.11 sysex()

Parameters

ev Provides the SysEx event to handle.

13.3.2.12 play()

```
void seq64::busarray::play (
          bussbyte bus,
          event * e24,
          midibyte channel )
```

Parameters

bus	The MIDI buss on which to play the event.
e24	A pointer to the event to be played.
channel	The MIDI channel on which to play the event. Sequencer64 controls the actual channel of playback, no matter what the channel specified in the event.

13.3.2.13 set_clock()

```
bool seq64::busarray::set_clock (
          bussbyte bus,
          clock_e clocktype )
```

This code is a bit more restrictive than the original code in mastermidibus::set_clock().

Parameters

bus	The MIDI bus for which the clock is to be set.
clocktype	Provides the type of clocking for the buss.

13.3.2.14 set_all_clocks()

```
void seq64::busarray::set_all_clocks ( )
```

Note that the settings to apply are added when the add() call is made.

13.3.2.15 get_clock()

Parameters

bus The MIDI bus for which the clock is to be set.

Returns

Returns the clock value set for the desired buss. If the buss is invalid, then e_clock_off is returned.

13.3.2.16 get_midi_bus_name()

the full display name) for the given (legal) buss number.

This function adds the retrieval of client and port numbers that are not needed in the portmidi implementation, but seem generally useful to support in all implementations. It's main use is to display the full portname in one of two forms:

```
- "[0] 0:0 clientname:portname"
- "[0] 0:0 portname"
```

The second version is chosen if "clientname" is already included in the port name, as many MIDI clients do that. However, the name gets modified to reflect the remote system port to which it will connect.

Parameters

bus Provides the output buss number. Checked before usage. Actually should now be an index number

Returns

Returns the buss name as a standard C++ string, truncated to 80-1 characters. Also contains an indication that the buss is disconnected or unconnected. If the buss number is illegal, this string is empty.

Both the busses for the given client are stopped: that is, set to inactive.

This function is called by api_get_midi_event() when the ALSA event SND_SEQ_EVENT_PORT_EXIT is received. Since port_exit() has no direct API-specific code in it, we do not need to create a virtual api_port_exit() function to implement the port-exit event.

Parameters

client	The client to be matched and acted on. This value is actually an ALSA concept.
port	The port to be acted on. Both parameter must be matched before the buss is made inactive. This value
	is actually an ALSA concept.

13.3.2.19 set_input()

```
bool seq64::busarray::set_input (
          bussbyte bus,
          bool inputing)
```

There's currently no implementation-specific API function called directly here. What happens is that midibase::set— _input() uses the *inputing* parameter to decide whether to call init_in() or deinit_in(), and these functions ultimately lead to an API specific called.

Note that the call to midibase::set_input() will set its m_inputing flag, and then call init_in() or deinit_in() if that flag changed. This change is important, so we have to call midibase::set_input() first. Then the call to businfo::init_input() will set that flag again (plus another flag). A bit confusing in sequence and in function naming.

This function should be used only for the input busarray, obviously.

Threadsafe

Parameters

bus	Provides the buss number.
inputing	True if the input bus will be inputting MIDI data.

Returns

Returns true if the buss number is valid and was active, and so could be set.

```
13.3.2.20 set_all_inputs()
void seq64::busarray::set_all_inputs ( )
```

There's no implementation-specific API function here. This function should be used only for the input busarray, obviously. Note that the input settings used here were stored when the add() function was called. They can be changed by the user via the Options / MIDI Input tab.

```
13.3.2.21 get_input()
```

```
bool seq64::busarray::get_input (
          bussbyte bus )
```

There's currently no implementation-specific API function here.

Parameters

bus Provides the buss number.

Returns

If the buss is a system buss, always returns true. Otherwise, if the buss is inactive, returns false. Otherwise, the buss's get_input() status is returned.

13.3.2.22 is_system_port()

Parameters

bus Provides the buss number.

Returns

Returns the selected buss's is-system-port status. If the buss number is out of range, then false is returned.

```
13.3.2.23 poll_for_midi()
```

```
bool seq64::busarray::poll_for_midi ( )
```

This is a primitive poll, which exits when some data is obtained. It also applies only to the input busses.

Returns

Returns true if a MIDI event was detected on one of the busses. Note that this is a boolean value, while the midibase::poll_for_midi() function returns an integer.

13.3.2.24 get_midi_event()

Note that this function risks starving the second input device if more than one is enabled in Sequencer64. We will figure that one out later.

Parameters

inev	A pointer to the event to be modified by incoming data, if any.
------	---

Returns

Returns true if an event's data was copied into the event pointer.

13.3.2.25 replacement_port()

This function is meant only for the output buss (so far).

Still need to determine exactly what this function needs to do.

Parameters

bus	The buss to be affected.
port	The prot to be affected.

Returns

Returns -1 if no matching port is found, otherwise it returns the replacement-port number.

13.3.3 Field Documentation

13.3.3.1 m_container

```
std::vector<businfo> seq64::busarray::m_container [private]
```

13.4 seq64::businfo Class Reference

A new class to consolidate a number of bus-related arrays into one array.

Public Member Functions

• businfo ()

A new class to consolidate a number of bus-related arrays into one array.

• businfo (midibus *bus)

Principal constructor.

· businfo (const businfo &rhs)

Copy constructor.

• ∼businfo ()

We can't destroy the bus pointer.

- void remove ()
- const midibus * bus () const
- midibus * bus ()
- bool active () const
- bool initialize ()

This function is called when the businfo object is added to the busarray.

- · bool initialized () const
- clock_e init_clock () const
- bool init_input () const
- void bus (midibus *b)
- · void activate ()
- · void deactivate ()
- void init_clock (clock_e clocktype)
- void init input (bool flag)

Private Member Functions

- void start ()
- void stop ()
- void continue_from (midipulse tick)
- void init_clock (midipulse tick)
- void clock (midipulse tick)
- void sysex (event *ev)

Private Attributes

```
• midibus * m_bus
```

Points to an existing midibus object.

• bool m_active

Indicates if the existing bus is active.

bool m_initialized

Indicates if the existing bus is initialized.

· clock_e m_init_clock

Clock initialization.

• bool m_init_input

Input initialization?

Friends

· class busarray

13.4.1 Detailed Description

There will be in input instance and an output instance of this object contained by mastermidibus.

13.4.2 Constructor & Destructor Documentation

```
13.4.2.1 businfo() [1/3] seq64::businfo::businfo ( )
```

There will be in input instance and an output instance of this object contained by mastermidibus.

Parameters

bus | Provides a pointer to the MIDI buss object to be represented by this object.

Currently it does not replicate the pointed-to object.

Parameters

```
rhs The source object to be copied.
```

```
13.4.2.4 ∼businfo()
```

```
seq64::businfo::~businfo ( ) [inline]
```

13.4.3 Member Function Documentation

```
13.4.3.1 remove()
```

```
void seq64::businfo::remove ( ) [inline]
```

```
13.4.3.2 bus() [1/3]
```

```
const midibus* seq64::businfo::bus ( ) const [inline]
```

13.4.3.3 bus() [2/3]

```
midibus* seq64::businfo::bus ( ) [inline]
```

13.4.3.4 active()

```
bool seq64::businfo::active ( ) const [inline]
```

13.4.3.5 initialize()

```
bool seq64::businfo::initialize ( )
```

It relies on the perform::launch() function to actually activate() all of the ports that have been flagged as "activated" here.

is_input_port():

```
Indicates if the midibus represents an input port (true) versus an output port (false). The way the mastermidibus currently works, it creates the API MIDI input objects there, so it does not need to be done here. This falls under the heading of "tricky code".
```

is_virtual_port():

```
Indicates if the midibus represents a manual/virtual port (true) versus a normal port (false).
```

The rules for port initialization follow those of seq24 for MIDI busses:

- Manual (virtual) input and output ports always get their init functions called. They are unconditionally marked as "active" and "initialized".
- Normal output ports are marked as "active" and "initialized" if init_out() succeeds.
- Normal input ports don't have init_in() called, but are marked as "active" and "initialized" anyway. The settings from the "rc" file determine which inputs will operate.

Returns

Returns true if the buss is value, and it could be initialized (as an output port or a virtual output port.

13.4.3.6 initialized()

```
bool seq64::businfo::initialized ( ) const [inline]
```

13.4.3.7 init_clock() [1/3]

```
clock_e seq64::businfo::init_clock ( ) const [inline]
```

```
13.4.3.8 init_input() [1/2]
bool seq64::businfo::init_input ( ) const [inline]
13.4.3.9 bus() [3/3]
void seq64::businfo::bus (
             midibus * b ) [inline]
13.4.3.10 activate()
void seq64::businfo::activate ( ) [inline]
13.4.3.11 deactivate()
void seq64::businfo::deactivate ( ) [inline]
13.4.3.12 init_clock() [2/3]
void seq64::businfo::init_clock (
             clock_e clocktype ) [inline]
13.4.3.13 init_input() [2/2]
void seq64::businfo::init_input (
             bool flag ) [inline]
13.4.3.14 start()
void seq64::businfo::start ( ) [inline], [private]
```

```
13.4.3.15 stop()
void seq64::businfo::stop ( ) [inline], [private]
13.4.3.16 continue_from()
void seq64::businfo::continue_from (
             midipulse tick ) [inline], [private]
13.4.3.17 init_clock() [3/3]
void seq64::businfo::init_clock (
             midipulse tick ) [inline], [private]
13.4.3.18 clock()
void seq64::businfo::clock (
            midipulse tick ) [inline], [private]
13.4.3.19 sysex()
void seq64::businfo::sysex (
            event * ev ) [inline], [private]
13.4.4 Friends And Related Function Documentation
```

```
13.4.4.1 busarray
friend class busarray [friend]
```

13.4.5 Field Documentation

```
13.4.5.1 m_bus
midibus* seq64::businfo::m_bus [private]
13.4.5.2 m_active
bool seq64::businfo::m_active [private]
13.4.5.3 m_initialized
bool seq64::businfo::m_initialized [private]
13.4.5.4 m_init_clock
clock_e seq64::businfo::m_init_clock [private]
13.4.5.5 m_init_input
```

13.5 seq64::click Class Reference

bool seq64::businfo::m_init_input [private]

Encapsulates any possible mouse click.

Public Member Functions

· click ()

The constructor for class click.

click (int x, int y, int button=SEQ64_CLICK_BUTTON_LEFT, bool press=true, seq_modifier_t modkey=SE
 —
 Q64_NO_MASK)

Principal constructor for class click.

click (const click &rhs)

Provides a stock copy constructor.

click & operator= (const click &rhs)

Provides a stock principal assignment operator.

• bool is_press () const

'Getter' function for member m_is_press

• bool is left () const

'Getter' function for member m_button to test for the left button.

bool is_middle () const

'Getter' function for member m_button to test for the middle button.

bool is_right () const

'Getter' function for member m_button to test for the right button.

• int x () const

'Getter' function for member m_x

• int y () const

'Getter' function for member m_y

• int button () const

'Getter' function for member m_button

seq_modifier_t modifier () const

'Getter' function for member m_modifier

• bool mod_control () const

'Getter' function for member m_modifier tested for Ctrl key.

• bool mod_control_shift () const

'Getter' function for member m_modifier tested for Ctrl and Shift key.

• bool mod_super () const

'Getter' function for member m_modifier tested for Mod4/Super/Windows key.

Private Attributes

• bool m_is_press

Determines if the click was a press or a release event.

• int m_x

The x-coordinate of the click.

• int m y

The y-coordinate of the click.

• int m_button

The button that was pressed or released.

· seq_modifier_t m_modifier

The optional modifier value.

13.5.1 Detailed Description

Useful in passing more generic events to non-GUI classes.

13.5.2 Constructor & Destructor Documentation

```
13.5.2.1 click() [1/3] seq64::click::click ( )
```

Sets all members to false, zero, or the lowest good value.

```
13.5.2.2 click() [2/3]
seq64::click::click (
    int x,
    int y,
    int button = SEQ64_CLICK_BUTTON_LEFT,
    bool press = true,
    seq_modifier_t modkey = SEQ64_NO_MASK )
```

This function is the only way to set value for the click members (other than the copy constructor and principal assignment operator.

Parameters

X	The putative x value of the button click.	
У	The putative y value of the button click.	
button	The value of the button that was clicked, set to 1, 2, or 3.	
press	ress Set to true if the event was a button press, false if it was a button release.	
modkey	modkey Indicates which modifier key (such as Ctrl or Alt), if any, was pressed at the same time as the click	
	action.	

It is nice to be explicit about these kinds of functions, even if it gets tedious.

Parameters

rhs Provies the source object to be copied.

13.5.3 Member Function Documentation

13.5.3.1 operator=()

It is nice to be explicit about these kinds of functions, even if it gets tedious.

Parameters

rhs

Provies the source object to be assigned from. The assignment is not made if "this" has the same address as this parameter.

Returns

Returns a reference to self for usage in a string of assignments.

13.5.3.2 is_press()

```
bool seq64::click::is_press ( ) const [inline]
```

13.5.3.3 is_left()

```
bool seq64::click::is_left ( ) const [inline]
```

13.5.3.4 is_middle()

```
bool seq64::click::is_middle ( ) const [inline]
```

13.5.3.5 is_right()

```
bool seq64::click::is_right ( ) const [inline]
```

13.5.3.6 x()

```
int seq64::click::x ( ) const [inline]
```

```
13.5.3.7 y()
int seq64::click::y ( ) const [inline]
13.5.3.8 button()
int seq64::click::button ( ) const [inline]
13.5.3.9 modifier()
seq_modifier_t seq64::click::modifier ( ) const [inline]
13.5.3.10 mod_control()
bool seq64::click::mod_control ( ) const [inline]
13.5.3.11 mod_control_shift()
bool seq64::click::mod_control_shift ( ) const [inline]
13.5.3.12 mod_super()
bool seq64::click::mod_super ( ) const [inline]
13.5.4 Field Documentation
13.5.4.1 m_is_press
bool seq64::click::m_is_press [private]
```

```
13.5.4.2 m_x
```

```
int seq64::click::m_x [private]
```

0 is the left-most coordinate.

13.5.4.3 m_y

```
int seq64::click::m_y [private]
```

0 is the top-most coordinate.

13.5.4.4 m_button

```
int seq64::click::m_button [private]
```

Left is 1, mmiddle is 2, and right is 3. These numbers are defined via macros, and are Linux-specific and Gtk-specific.

13.5.4.5 m_modifier

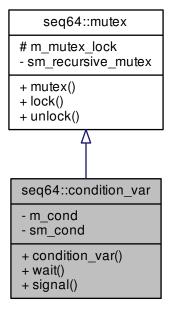
```
seq_modifier_t seq64::click::m_modifier [private]
```

Note that SEQ64_NO_MASK is our word for 0, meaning "no modifier".

13.6 seq64::condition_var Class Reference

A mutex works best in conjunction with a condition variable.

Inheritance diagram for seq64::condition_var:



Public Member Functions

• condition_var ()

Initialize the condition variable with the global variable.

• void wait ()

Waits for the condition variable.

• void signal ()

Signals the condition variable.

Private Attributes

pthread_cond_t m_cond

Provides a class-specific condition variable.

Static Private Attributes

static const pthread_cond_t sm_cond
 Provides a "global" condition variable.

Additional Inherited Members

13.6.1 Detailed Description

Therefore this class derives from the mutex class. A "has-a" relationship might be more logical than this "is-a" relationship.

13.6.2 Constructor & Destructor Documentation

```
13.6.2.1 condition_var()
```

```
seq64::condition_var::condition_var ( )
```

13.6.3 Member Function Documentation

13.6.3.1 wait()

```
void seq64::condition_var::wait ( )
```

```
13.6.3.2 signal()
```

```
void seq64::condition_var::signal ( )
```

13.6.4 Field Documentation

13.6.4.1 sm_cond

```
const pthread_cond_t seq64::condition_var::sm_cond [static], [private]
```

Define the static condition variable used by all mutex locks.

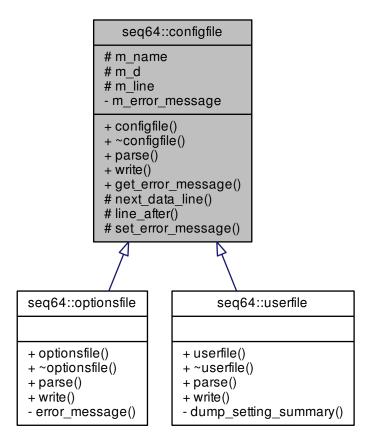
13.6.4.2 m_cond

```
pthread_cond_t seq64::condition_var::m_cond [private]
```

13.7 seq64::configfile Class Reference

This class is the abstract base class for optionsfile and userfile.

Inheritance diagram for seq64::configfile:



Public Member Functions

- configfile (const std::string &name)
 - Provides the string constructor for a configuration file.
- virtual ∼configfile ()

A rote destructor needed for a base class.

- virtual bool parse (perform &perf)=0
- virtual bool write (const perform &perf)=0
- · const std::string & get_error_message () const

Protected Member Functions

- bool next_data_line (std::ifstream &file)
 - Gets the next line of data from an input stream.
- · bool line_after (std::ifstream &file, const std::string &tag)

This function gets a specific line of text, specified as a tag.

void set_error_message (const std::string &msg)

Protected Attributes

• std::string m_name

Provides the name of the configuration file.

char * m_d

Points to an allocated buffer that holds the data for the configuration file.

• char m_line [SEQ64_LINE_MAX]

The current line of text being processed.

Private Attributes

• std::string m_error_message

Holds the last error message, if any.

13.7.1 Constructor & Destructor Documentation

13.7.1.1 configfile()

Parameters

name The name of the configuration file.

13.7.1.2 \sim configfile()

```
virtual seq64::configfile::~configfile ( ) [inline], [virtual]
```

13.7.2 Member Function Documentation

13.7.2.1 next_data_line()

If the line starts with a number-sign, a space (!), or a null, it is skipped, to try the next line. This occurs until an EOF is encountered.

Member m_line is a "global" return value.

Parameters

file

Points to an input stream. We converted this item to a reference; pointers can be subject to problems. For example, what if someone passes a null pointer?

Returns

Returns true if a presumed data line was found. False is returned if not found before an EOF or a section marker ("[") is found. This is a a new (ca 2016-02-14) feature of this function, to assist in adding new data to the file.

13.7.2.2 line_after()

Then it gets the next non-blank line (i.e. data line) after that.

This function always starts from the beginning of the file. Therefore, it can handle reading Sequencer64 configuration files that have had their tagged sections arranged in a different order. This feature makes the configuration file a little more robust against errors.

Parameters

file	Points to the input file stream.
tag	Provides a tag to be found. Lines are read until a match occurs with this tag. Normally, the tag is a
	section marker, such as "[user-interface]". Best to assume an exact match is needed.

Returns

Returns true if the tag was found. Otherwise, false is returned.

13.7.2.3 parse()

Implemented in seq64::userfile, and seq64::optionsfile.

13.7.2.4 write()

Implemented in seq64::userfile, and seq64::optionsfile.

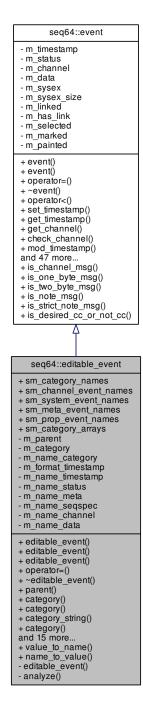
```
13.7.2.5 get_error_message()
const std::string& seq64::configfile::get_error_message ( ) const [inline]
13.7.2.6 set_error_message()
void seq64::configfile::set_error_message (
            const std::string & msg ) [inline], [protected]
13.7.3 Field Documentation
13.7.3.1 m_error_message
std::string seq64::configfile::m_error_message [private]
Not a 100% foolproof yet.
13.7.3.2 m_name
std::string seq64::configfile::m_name [protected]
13.7.3.3 m_d
char* seq64::configfile::m_d [protected]
13.7.3.4 m_line
char seq64::configfile::m_line[SEQ64_LINE_MAX] [protected]
```

This member receives an input line, and so needs to be a character buffer.

13.8 seq64::editable_event Class Reference

Provides for the management of MIDI editable events.

Inheritance diagram for seq64::editable_event:



Data Structures

struct name_value_t

Provides a type that contains the pair of values needed for the various lookup maps that are needed to manage editable events.

Public Types

```
    enum category_t {
        category_name,
        category_channel_message,
        category_system_message,
        category_meta_event,
        category_prop_event }
```

These values determine the major kind of event, which determines what types of events are possible for this editable event object.

```
    enum timestamp_format_t {
        timestamp_measures,
        timestamp_time,
        timestamp_pulses }
```

Provides a code to indicate the desired timestamp format.

Public Member Functions

• editable event (const editable events &parent)

This constructor simply initializes all of the class members.

editable_event (const editable_events &parent, const event &ev)

Event constructor.

editable_event (const editable_event &rhs)

This copy constructor initializes most of the class members.

- editable_event & operator= (const editable_event &rhs)
- virtual ∼editable_event ()

This destructor current is a rote virtual function override.

const editable_events & parent () const

'Getter' function for member m parent

· category_t category () const

'Getter' function for member m_category

void category (category_t c)

'Setter' function for member m category by value Also keeps the m name category member in synchrony.

· const std::string & category_string () const

'Getter' function for member m_category

void category (const std::string &cs)

'Setter' function for member m_category by name Also keeps the m_name_category member in synchrony, but looks up the name, rather than using the name parameter, to avoid storing abbreviations.

const std::string & timestamp_string () const

'Getter' function for member m_name_timestamp

• midipulse timestamp () const

'Getter' function for member event::get_timestamp() Implemented to allow a uniform naming convention that is not slavish to the get/set crowd [this ain't Java].

· void timestamp (midipulse ts)

'Setter' function for member event::set_timestamp() Implemented to allow a uniform naming convention that is not slavish to the get/set crowd [this ain't Java].

void timestamp (const std::string &ts string)

'Setter' function for member event::set_timestamp() [string version]

std::string time_as_pulses ()

Converts the current time-stamp to a string representation in units of pulses.

std::string time_as_measures ()

Converts the current time-stamp to a string representation in units of measures, beats, and divisions.

std::string time_as_minutes ()

Converts the current time-stamp to a string representation in units of hours, minutes, seconds, and fraction.

void set_status_from_string (const std::string &ts, const std::string &ts, con

Converts a string into an event status, along with timestamp and data bytes.

• std::string format_timestamp ()

Formats the current timestamp member as a string.

std::string stock_event_string ()

Converts the event into a string desribing the full event.

• std::string status_string () const

'Getter' function for member m_name_status

• std::string meta_string () const

'Getter' function for member m_name_meta

std::string seqspec_string () const

'Getter' function for member m_name_seqspec

std::string channel_string () const

'Getter' function for member m name channel

std::string data_string () const

'Getter' function for member m_name_data

Static Public Member Functions

• static std::string value to name (midibyte value, category t cat)

Provides a static lookup function that returns the name, if any, associated with a midibyte value.

static unsigned short name_to_value (const std::string &name, category_t cat)

Provides a static lookup function that returns the value, if any, associated with a name string.

Static Public Attributes

static const name_value_t sm_category_names []

An array of event categories and their names.

static const name_value_t sm_channel_event_names []

An array of MIDI channel events and their names.

• static const name_value_t sm_system_event_names []

An array of MIDI system events and their names.

static const name_value_t sm_meta_event_names []

An array of Meta events and their names.

static const name_value_t sm_prop_event_names[]

An array of Sequencer64-specific events and their names.

static const name_value_t *const sm_category_arrays[]

Provides for fast access (no ifs) to the correct name array for the given category.

Private Member Functions

- editable_event ()
- void analyze ()

Analyzes an editable-event to make all the settings it needs.

Private Attributes

· const editable_events & m_parent

Provides a reference to the container that holds this event.

category_t m_category

Indicates the overall category of this event, which will be category_channel_message, category_system_message, category_meta_event, and category_prop_event.

std::string m_name_category

Holds the name of the event category for this event.

· timestamp format t m format timestamp

Indicates the format to display the time-stamp.

std::string m_name_timestamp

Holds the string version of the MIDI pulses time-stamp.

std::string m name status

Holds the name of the status value for this event.

• std::string m_name_meta

Holds the name of the meta message, if applicable.

· std::string m_name_seqspec

If we eventually implement the editing of the Seq24/Sequencer64 "proprietary" meta sequencer-specific events, the name of the SeqSpec will be stored here.

std::string m_name_channel

Holds the channel description, if applicable.

std::string m_name_data

Holds the data description, if applicable.

13.8.1 Detailed Description

It makes the following members of an event modifiable using human-readable strings:

- m_timestamp
- m_status
- m_channel
- m_data[]

Eventually, it would be nice to be able to edit, or at least view, the SysEx events and the Meta events. Those two will require extensions to make events out of them (SysEx is partly supported).

To the concepts of event, the editable_event class adds a category field and strings to represent all of these members.

13.8.2 Member Enumeration Documentation

13.8.2.1 category_t

```
enum seq64::editable_event::category_t
```

These tags are accompanied by category names in sm_category_names[]. The enum values are cast to midibyte values for the purposes of using the lookup infrastructure.

Enumerator

category_name	Indicates that the lookup needs to be done on the category names, as listed in sm_category_names[].
category_channel_message	Indicates a channel event, with a value ranging from 0x80 through 0xEF. Some examples are note on/off, control change, and program change. Values are looked up in sm_channel_event_names[].
category_system_message	Indicates a system event, with a value ranging from 0xF0 through 0xFF. Some examples are SysEx start/end, song position, and stop/start/continue/reset. Values are looked up in sm_system_event_names[].
category_meta_event	Indicates a meta event, and there is a second value that is used to look up the name of the meta event, in sm_meta_event_names[].
category_prop_event	Indicates a "proprietary", Sequencer64 event. Indicates to look up the name of the event in sm_prop_event_names[]. Not sure if these kinds of events will be stored separately.

13.8.2.2 timestamp_format_t

enum seq64::editable_event::timestamp_format_t

Three are supported. All editable events will share the same timestamp format, but it seems good to make this a event class member, rather than something imposed from an outside static value. We shall see.

Enumerator

timestamp_measures	This format displays the time in "measures:beats:divisions" format, where measures and beats start at 1. Thus, "1:1:0" is equivalent to 0 pulses or to "0:0:0:0.0" in normal time values.
timestamp_time	This format displays the time in "hh:mm:second.fraction" format. The value displayed should not depend upon the internal timing parameters of the event.
timestamp_pulses	This format specifies a bare pulse format for the timestamp – a long integer ranging from 0 on up. Obviously, this representation depends on the PPQN value for the sequence holding this event.

13.8.3 Constructor & Destructor Documentation

13.8.3.1 editable_event() [1/4]

seq64::editable_event::editable_event () [private]

editable_event::editable_event (): event (), m_category (category_name), m_name_category (), m_format_
timestamp (timestamp_measures), m_name_timestamp (), m_name_status (), m_name_meta (), m_name_
seqspec (), m_name_channel (), m_name_data () { // Empty body } Principal constructor.

Parameters

parent Provides the overall editable-events object that manages the whole set of editable-event.

This function basically adds all of the extra editable_event stuff to a standard event, so that the resulting editable
_event is container-ready.

This function is currently geared only toward support of the SMF 0 channel-splitting feature. Many of the members are not set to useful values when the MIDI file is read, so we don't handle them for now.

Warning

This function does not yet copy the SysEx data. The inclusion of SysEx editable_events was not complete in Seq24, and it is still not complete in Sequencer64. Nor does it currently bother with the links.

Parameters

rhs Provides the editable_event object to be copied.

```
13.8.3.5 ∼editable_event()
```

```
\label{lem:condition} \mbox{virtual seq64::editable\_event::} \sim \mbox{editable\_event ( ) [inline], [virtual]}
```

13.8.4 Member Function Documentation

13.8.4.1 value_to_name()

Parameters

value	The MIDI byte value to look up.	
cat	The category of the MIDI byte. Each category calls a different name array into play.	

Returns

Returns the name associated with the value. If there is no such name, then an empty string is returned.

13.8.4.2 name_to_value()

The string_match() function, which can match abbreviations, case-insensitively, is used to make the string comparisons.

Parameters

name	The string value to look up.
cat	The category of the MIDI byte. Each category calls a different name array into play.

Returns

Returns the value associated with the name. If there is no such value, then SEQ64_END_OF_MIDIBYTE_← TABLE is returned.

13.8.4.3 operator=()

```
13.8.4.4 parent()
const editable_events& seq64::editable_event::parent ( ) const [inline]

13.8.4.5 category() [1/3]
category_t seq64::editable_event::category ( ) const [inline]

13.8.4.6 category() [2/3]
void seq64::editable_event::category (
```

Note that a bad value is translated to the value of category_name.

Parameters

c Provides the category value to set.

category_t c)

13.8.4.7 category_string()

```
const std::string& seq64::editable_event::category_string ( ) const [inline]
```

13.8.4.8 category() [3/3]

Note that a bad value is translated to the value of category_name.

Parameters

name Provides the category name for the category value to set.

13.8.4.9 timestamp_string()

```
\verb|const| std::string& seq64::editable_event::timestamp_string ( ) const [inline]|
```

Plus, we also have to set the string version at the same time.

The format of the string representation is of the format selected by the m_format_timestamp member and is set by the format_timestamp() function.

Parameters

ts | Provides the timestamp in units of MIDI pulses.

The format of the string representation is of the format selected by the m_format_timestamp member and is set by the format_timestamp() function.

Parameters

ts_string | Provides the timestamp in units of MIDI pulses.

```
13.8.4.13 time_as_pulses()
std::string seq64::editable_event::time_as_pulses ( ) [inline]

13.8.4.14 time_as_measures()
std::string seq64::editable_event::time_as_measures ( )
```

Cannot be inlined because of a circular dependency between the editable_event and editable_events classes.

13.8.4.15 time_as_minutes()

```
std::string seq64::editable_event::time_as_minutes ( )
```

Cannot be inlined because of a circular dependency between the editable_event and editable_events classes.

13.8.4.16 set_status_from_string()

Currently, this function handles only the following two messages:

- · category channel message
- · category_system_message

After all of the numbering member items have been set, they are converted and assigned to the string versions via a call to the analyze() function.

Parameters

ts	Provides the time-stamp string of the event.
s	Provides the name of the event, such as "Program Change".
sd0	Provides the string defining the first data byte of the event.
sd1	Provides the string defining the second data byte of the event, if applicable to the event.

13.8.4.17 format_timestamp()

```
std::string seq64::editable_event::format_timestamp ( )
```

The format of the string representation is of the format selected by the m_format_timestamp member.

13.8.4.18 stock_event_string()

```
std::string seq64::editable_event::stock_event_string ( )
```

We get the time-stamp as a string, make sure the event is fully analyzed so that all items and strings are set correctly.

Returns

Returns a human-readable string describing this event.

```
13.8.4.19 status_string()
std::string seq64::editable_event::status_string ( ) const [inline]
13.8.4.20 meta_string()
std::string seq64::editable_event::meta_string ( ) const [inline]
13.8.4.21 seqspec_string()
std::string seq64::editable_event::seqspec_string ( ) const [inline]
13.8.4.22 channel_string()
std::string seq64::editable_event::channel_string ( ) const [inline]
13.8.4.23 data_string()
std::string seq64::editable_event::data_string ( ) const [inline]
13.8.4.24 analyze()
void seq64::editable_event::analyze ( ) [private]
```

Used in the constructors. Some of the setters indirectly set the appropriate string representation, as well.

Category:

This function can figure out if the status byte implies a channel message or a system message, and set the category string as well. However, at this time, detection of Meta events (0xFF) or Proprietary/SeqSpec events (0xFF with 0x2424) doesn't work due to lack of context here (and due to the fact that currently such events are not yet stored in a Sequencer64 sequence/track, and the least-significant-byte gets masked off anyway.)

Status:

We distinguish between channel and system messages, and then one— and two-byte messages, but don't yet distinguish the data values fully.

13.8.5 Field Documentation

13.8.5.1 sm_category_names

```
const editable_event::name_value_t seq64::editable_event::sm_category_names [static]
```

Initializes the array of event/name pairs for the MIDI events categories.

Terminated by an empty string, the latter being the preferred test, for consistency with the other arrays and because 0 is often a legitimate code value.

13.8.5.2 sm_channel_event_names

```
const editable_event::name_value_t seq64::editable_event::sm_channel_event_names [static]
```

Initializes the array of event/name pairs for the channel MIDI events.

We split channel and system messages into two arrays, for semantic reasons and for faster linear lookups.

Terminated by an empty string.

13.8.5.3 sm_system_event_names

```
const editable_event::name_value_t seq64::editable_event::sm_system_event_names [static]
```

Initializes the array of event/name pairs for the system MIDI events.

We split channel and system messages into two arrays, for semantic reasons and for faster linear lookups.

Terminated by an empty string.

13.8.5.4 sm_meta_event_names

```
const editable_event::name_value_t seq64::editable_event::sm_meta_event_names [static]
```

Initializes the array of event/name pairs for all of the Meta events.

Terminated only by the empty string.

13.8.5.5 sm_prop_event_names

```
const editable_event::name_value_t seq64::editable_event::sm_prop_event_names [static]
```

Initializes the array of event/name pairs for all of the seq24/sequencer64-specific events.

Terminated only by the empty string. Note that the numbers reflect the masking off of the high-order bits by 0x242400FF.

```
13.8.5.6 sm_category_arrays
```

```
const editable_event::name_value_t *const seq64::editable_event::sm_category_arrays [static]
```

Contains pointers (references cannot be stored in an array) to the desired array for a given category.

Too bad that an array of references is not possible.

This code could be considered a bit rococo.

```
13.8.5.7 m_parent
```

```
const editable_events& seq64::editable_event::m_parent [private]
```

The container's "children" need to go to their "parent" to get certain items of information.

```
13.8.5.8 m_category
```

```
category_t seq64::editable_event::m_category [private]
```

The category_name value is not set here, since that category is used only for looking up the human-readable form of the category.

```
13.8.5.9 m_name_category
```

```
std::string seq64::editable_event::m_name_category [private]
```

13.8.5.10 m_format_timestamp

```
timestamp_format_t seq64::editable_event::m_format_timestamp [private]
```

The default is to display in timestamp_measures format.

```
13.8.5.11 m_name_timestamp
```

```
\verb|std::string| seq64::editable_event::m_name\_timestamp| [private]|\\
```

13.8.5.12 m_name_status

```
std::string seq64::editable_event::m_name_status [private]
```

It will include the names of the channel messages and the system messages. The latter includes SysEx and Meta messages.

```
13.8.5.13 m_name_meta

std::string seq64::editable_event::m_name_meta [private]

If not applicable, this name will be empty.

13.8.5.14 m_name_seqspec

std::string seq64::editable_event::m_name_seqspec [private]

13.8.5.15 m_name_channel

std::string seq64::editable_event::m_name_channel [private]

13.8.5.16 m_name_data

std::string seq64::editable_event::m_name_data [private]
```

13.9 seq64::editable_events Class Reference

Provides for the management of an ordered collection MIDI editable events.

Public Member Functions

editable_events (sequence &seq, midibpm bpm)

This constructor hooks into the sequence object.

editable_events (const editable_events &rhs)

This copy constructor initializes most of the class members.

editable_events & operator= (const editable_events &rhs)

This principal assignment operator sets most of the class members.

• virtual \sim editable_events ()

This destructor current is a rote virtual function override.

• const midi_timing & timing () const

'Getter' function for member m_midi_parameters

• midipulse string_to_pulses (const std::string &ts_string) const

Calculates the MIDI pulses (divisions) from a string using one of the free functions of the calculations module.

• bool load_events ()

Accesses the sequence's event-list, iterating through it from beginning to end, wrapping each event in the list in an editable event and inserting it into the editable-event container.

• bool save_events ()

Erases the sequence's event container and recreates it using the edited container of editable events.

Events & events ()

'Getter' function for member m_events

· iterator begin ()

'Getter' function for member m_events.begin(), non-constant version.

· const iterator begin () const

'Getter' function for member m_events.begin(), constant version.

· iterator end ()

'Getter' function for member m_events.end(), non-constant version.

· const_iterator end () const

'Getter' function for member m_events.end(), constant version.

· int count () const

Returns the number of events stored in m_events.

• bool add (const event &e)

Adds an event, converted to an editable_event, to the internal event list.

• bool add (const editable_event &e)

Adds an editable event to the internal event list.

bool replace (iterator ie, const editable_event &e)

Provides a wrapper for the iterator form of erase(), which is the only one that the editable events container uses.

void remove (iterator ie)

Provides a wrapper for the iterator form of erase(), which is the only one that sequence uses.

• void clear ()

Provides a wrapper for clear().

• iterator current event () const

'Getter' function for member m_current_event The caller must make sure the iterator is not Events::end().

Static Public Member Functions

• static editable_event & dref (iterator ie)

Dereference access for list or map.

static const editable_event & dref (const_iterator ie)

Dereference const access for list or map.

Private Types

typedef event_list::event_key Key

Types to use to with the multimap implementation.

- typedef std::pair< Key, editable event > EventsPair
- typedef std::multimap< Key, editable_event > Events
- typedef std::multimap< Key, editable_event >::iterator iterator
- typedef std::multimap< Key, editable_event >::const_iterator const_iterator

Private Member Functions

- editable events ()
- void current event (iterator cei)

'Setter' function for member m_current_event

Private Attributes

• Events m_events

Holds the editable_events.

• iterator m_current_event

Points to the current event, which is the event that has just been inserted.

• sequence & m_sequence

Provides a reference to the sequence containing the events to be edited.

midi_timing m_midi_parameters

Holds the current settings for the sequence (and usually for the whole MIDI tune as well).

Friends

· class eventslots

13.9.1 Member Typedef Documentation

```
13.9.1.1 Key
```

```
typedef event_list::event_key seq64::editable_events::Key [private]
```

These typenames are identical to those used in event_list, but of course they are in the editable_events scope instead. See the event_list class.

13.9.1.2 EventsPair

```
typedef std::pair<Key, editable_event> seq64::editable_events::EventsPair [private]
```

13.9.1.3 Events

```
typedef std::multimap<Key, editable_event> seq64::editable_events::Events [private]
```

13.9.1.4 iterator

```
typedef std::multimap<Key, editable_event>::iterator seq64::editable_events::iterator [private]
```

13.9.1.5 const_iterator

```
\label{typedef} $$td::multimap<Key, editable\_event>::const\_iterator seq64::editable\_events::const\_\leftrightarrow iterator [private]
```

13.9.2 Constructor & Destructor Documentation

Parameters

seq	Provides a reference to the sequence object, which provides the events and some of the MIDI timing
	parameters.
bpm	Provides the beats/minute value, which the caller figures out how to get and provides in this parameter.

```
13.9.2.3 editable_events() [3/3]
```

Note that we need to reconstitute the event links here, as well.

Parameters

rhs Provides the editable_events object to be copied.

13.9.2.4 ∼editable_events()

```
\label{lem:virtual} \verb| seq64::editable_events:: \sim \verb| editable_events ( ) [inline], [virtual] |
```

13.9.3 Member Function Documentation

Note that we need to reconstitute the event links here, as well.

Parameters

rhs Provides the editable_events object to be assigned.

Returns

Returns a reference to "this" object, to support the serial assignment of editable_eventss.

13.9.3.2 timing()

```
const midi_timing& seq64::editable_events::timing ( ) const [inline]
```

13.9.3.3 string_to_pulses()

13.9.3.4 load_events()

```
bool seq64::editable_events::load_events ( )
```

Note that the new events will not have valid links (actually, no links). These links are used for associating Note Off events with their respective Note On events. To be consistent, we must take the time to reconstitute these links, using event_list::verify_and_link().

Returns

Returns true if the size of the final editable_event container matches the size of the original events container.

```
13.9.3.5 save_events()
bool seq64::editable_events::save_events ( )
```

Note that the old events are replaced only if the container of editable events is not empty. There are safer ways for the user to erase all the events.

Todo Consider what to do about the sequence::m_is_modified flag.

Returns

Returns true if the size of the final event container matches the size of the original editable_events container.

```
13.9.3.6 events()
Events& seq64::editable_events::events ( ) [inline]
13.9.3.7 begin() [1/2]
iterator seg64::editable_events::begin ( ) [inline]
13.9.3.8 begin() [2/2]
const_iterator seq64::editable_events::begin ( ) const [inline]
13.9.3.9 end() [1/2]
iterator seq64::editable_events::end ( ) [inline]
13.9.3.10 end() [2/2]
const_iterator seq64::editable_events::end ( ) const [inline]
13.9.3.11 dref() [1/2]
static editable_event& seq64::editable_events::dref (
             iterator ie ) [inline], [static]
```

Provides the iterator to the event to which to get a reference.

Parameters

ie Provides the iterator to the event to which to get a reference.

13.9.3.13 count()

```
int seq64::editable_events::count ( ) const [inline]
```

We like returning an integer instead of size_t, and rename the function so nobody is fooled.

Parameters

e Provides the regular event to be added to the list of editable events.

Returns

Returns true if the insertion succeeded, as evidenced by an increment in container size.

For the std::multimap implementation, This is an option if we want to make sure the insertion succeed.

```
std::pair<Events::iterator, bool> result = m_events.insert(p);
return result.second;
```

e Provides the regular event to be added to the list of editable events.

Returns

Returns true if the insertion succeeded, as evidenced by an increment in container size.

Side-effect(s) Sets m_current_event, which can be used right-away in a single-threaded context to get an iterator to the event via the current_event() accessor.

```
13.9.3.16 replace()
bool seq64::editable_events::replace (
             iterator ie,
             const editable_event & e ) [inline]
13.9.3.17 remove()
void seq64::editable_events::remove (
             iterator ie ) [inline]
13.9.3.18 clear()
void seq64::editable_events::clear ( ) [inline]
13.9.3.19 current_event() [1/2]
iterator seq64::editable_events::current_event ( ) const [inline]
13.9.3.20 current_event() [2/2]
void seq64::editable\_events::current\_event (
            iterator cei ) [inline], [private]
Parameters
```

Provide an iterator to the event to set as the current event.

13.9.4 Friends And Related Function Documentation

13.9.4.1 eventslots

friend class eventslots [friend]

13.9.5 Field Documentation

13.9.5.1 m_events

Events seq64::editable_events::m_events [private]

13.9.5.2 m_current_event

```
iterator seq64::editable_events::m_current_event [private]
```

(From this event we can get the current time and other parameters.) If the container were a plain map, we could instead use a key to access it. But we can at least use an iterator, rather than a bare pointer.

13.9.5.3 m_sequence

```
sequence& seq64::editable_events::m_sequence [private]
```

Besides the events, this object also holds the beats/measure, beat-width, and the PPQN value. The beats/minute have to be obtained from the application's perform object, and passed to the editable_events constructor by the caller.

13.9.5.4 m_midi_parameters

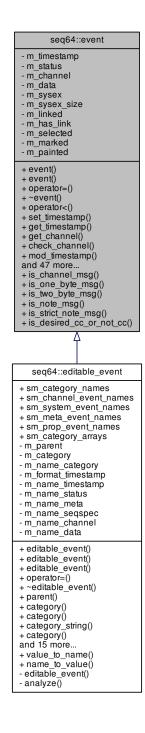
```
midi_timing seq64::editable_events::m_midi_parameters [private]
```

It holds the beats/minute, beats/measure, beat-width, and PPQN values needed to properly convert MIDI pulse timestamps to time and measure values.

13.10 seq64::event Class Reference

Provides events for management of MIDI events.

Inheritance diagram for seq64::event:



Public Types

• typedef std::vector< midibyte > SysexContainer

Provides a type definition for a vector of midibytes.

Public Member Functions

event ()

This constructor simply initializes all of the class members.

· event (const event &rhs)

This copy constructor initializes most of the class members.

event & operator= (const event &rhs)

This principal assignment operator sets most of the class members.

virtual ~event ()

This destructor explicitly deletes m_sysex and sets it to null.

bool operator< (const event &rhsevent) const

If the current timestamp equal the event's timestamp, then this function returns true if the current rank is less than the event's rank.

void set timestamp (midipulse time)

'Setter' function for member m_timestamp

• midipulse get_timestamp () const

'Getter' function for member m timestamp

• midibyte get_channel () const

'Getter' function for member m_channel

· bool check_channel (int channel) const

Checks the channel number to see if the event's channel matches it, or if the event has no channel.

void mod_timestamp (midipulse modtick)

Calculates the value of the current timestamp modulo the given parameter.

• void set_status (midibyte status)

Sets the m_status member to the value of status.

• void set_status (midibyte eventcode, midibyte channel)

This overload is useful when synthesizing events, such as converting a Note On event with a velocity of zero to a Note Off event.

void set_status_keep_channel (midibyte eventcode)

This function is used in recording to preserve the input channel information for deciding what to do with an incoming MIDI event.

void set_channel (midibyte channel)

Sets the channel "nybble", without modifying the status "nybble".

midibyte get_status () const

'Getter' function for member m_status

bool non_cc_match (midibyte status)

Returns true if the event's status is not a control-change, but does match the given status.

• bool cc_match (midibyte st, midibyte cc)

Returns true if the event's status is a control-change that matches the given status, and has a control value matching the given control-change value.

void set_data (midibyte d1)

Clears the most-significant-bit of the d1 parameter, and sets it into the first byte of m_data.

void set data (midibyte d1, midibyte d2)

Clears the most-significant-bit of both parameters, and sets them into the first and second bytes of m_data.

void get_data (midibyte &d0, midibyte &d1) const

Retrieves the two data bytes from m_data[] and copies each into its respective parameter.

void increment_data1 ()

Increments the first data byte (m_data[0]) and clears the most significant bit.

void decrement data1 ()

Decrements the first data byte (m_data[0]) and clears the most significant bit.

• void increment data2 ()

Increments the second data byte (m_data[1]) and clears the most significant bit.

void decrement_data2 ()

Decrements the second data byte (m_data[1]) and clears the most significant bit.

bool append sysex (midibyte *data, int len)

Appends SYSEX data to a new buffer.

bool append_sysex (midibyte data)

An overload for obtaining SYSEX data byte-by-byte.

void restart_sysex ()

Deletes and clears out the SYSEX buffer.

SysexContainer & get_sysex ()

'Getter' function for member m_sysex from stazed, non-const version for use by midibus.

const SysexContainer & get_sysex () const

'Getter' function for member m_sysex from stazed

void set_sysex_size (int len)

'Setter' function for member m_sysex and m_sysex_size from stazed

int get_sysex_size () const

'Getter' function for member m_sysex_size

void link (event *ev)

Sets m_has_link and sets m_link to the provided event pointer.

event * get_linked () const

'Getter' function for member m_linked

• bool is_linked () const

'Getter' function for member m_has_link

· void clear link ()

'Setter' function for member m_has_link and m_linked

• void paint ()

'Setter' function for member m_painted

• void unpaint ()

'Setter' function for member m_painted

• bool is_painted () const

'Getter' function for member m_painted

• void mark ()

'Setter' function for member m_marked

• void unmark ()

'Setter' function for member m_marked

• bool is_marked () const

'Getter' function for member m_marked

• void select ()

'Setter' function for member m_selected

· void unselect ()

'Setter' function for member m_selected

• bool is selected () const

'Getter' function for member m_selected

void make_clock ()

Sets m_status to EVENT_MIDI_CLOCK;.

· midibyte data (int index) const

'Getter' function for member m_data[]

midibyte get_note () const

Assuming m_data[] holds a note, get the note number, which is in the first data byte, m_data[0].

· void set note (midibyte note)

Sets the note number, clearing off the most-significant-bit and assigning it to the first data byte, m_data[0].

• void transpose_note (int tn)

Transpose the note, if possible.

midibyte get_note_velocity () const

'Getter' function for member m_data[1], the note velocity.

void set_note_velocity (int vel)

Sets the note velocity, which is held in the second data byte, and clearing off the most-significant-bit, storing it in m_{data} [1].

• bool is_note_on () const

Check for the Note On value in m_status.

· bool is note off () const

Check for the Note Off value in m_status.

• bool is_note () const

Returns true if m_status is a Note On, Note Off, or Aftertouch message.

bool is note off recorded () const

Some keyboards send Note On with velocity 0 for Note Off, so we provide this function to test that during recording.

· void print () const

Prints out the timestamp, data size, the current status byte, any SYSEX data if present, or the two data bytes for the status byte.

· int get rank () const

This function is used in sorting MIDI status events (e.g.

Static Public Member Functions

static bool is channel msg (midibyte m)

Static test for the channel message/statuse values: Note On, Note Off, Aftertouch, Control Change, Program Change, Channel Pressure, and Pitch Wheel.

static bool is_one_byte_msg (midibyte m)

Static test for channel messages that have only one data byte: Program Change and Channel Pressure.

static bool is two byte msg (midibyte m)

Static test for channel messages that have two data bytes: Note On, Note Off, Control Change, Aftertouch, and Pitch Wheel.

static bool is_note_msg (midibyte m)

Static test for messages that involve notes and velocity: Note On, Note Off, and Aftertouch.

static bool is_strict_note_msg (midibyte m)

Static test for messages that involve notes only: Note On and Note Off.

static bool is_desired_cc_or_not_cc (midibyte m, midibyte cc, midibyte datum)

Static test for channel messages that are either not control-change messages, or are and match the given controller value.

Private Attributes

· midipulse m timestamp

Provides the MIDI timestamp in ticks, otherwise known as the "pulses" in "pulses per quarter note" (PPQN).

• midibyte m_status

This is the status byte without the channel.

· midibyte m channel

In order to be able to handle MIDI channel-splitting of an SMF 0 file, we need to store the channel, even if we override it when playing the MIDI data.

midibyte m_data [SEQ64_MIDI_DATA_BYTE_COUNT]

The two bytes of data for the MIDI event.

• SysexContainer m_sysex

The data buffer for SYSEX messages.

• int m_sysex_size

Gives the size of the SYSEX message.

· event * m linked

This event is used to link Note Ons and Offs together.

• bool m_has_link

Indicates that a link has been made.

· bool m selected

Answers the question "is this event selected in editing.".

· bool m_marked

Answers the question "is this event marked in processing.".

bool m_painted

Answers the question "is this event being painted.".

13.10.1 Detailed Description

A MIDI event consists of 3 bytes:

```
-# Status byte, 1sssnnn, where the sss bits specify the type of
message, and the nnnn bits denote the channel number.
The status byte always starts with 0.
```

- -# The first data byte, 0xxxxxxx, where the data byte always start with 0, and the xxxxxxx values range from 0 to 127.
- -# The second data byte, 0xxxxxxx.

This class may have too many member functions.

13.10.2 Member Typedef Documentation

13.10.2.1 SysexContainer

```
typedef std::vector<midibyte> seq64::event::SysexContainer
```

13.10.3 Constructor & Destructor Documentation

```
13.10.3.1 event() [1/2]
```

seq64::event::event ()

This function is currently geared only toward support of the SMF 0 channel-splitting feature. Many of the members are not set to useful values when the MIDI file is read, so we don't handle them for now.

Note that now events are also copied when creating the editable_events container, so this function is even more important. The event links, for linking Note Off events to their respective Note On events, are dropped. Generally, they will need to be reconstituted by calling the event list::verify and link() function.

Warning

This function does not yet copy the SysEx data. The inclusion of SysEx events was not complete in Seq24, and it is still not complete in Sequencer64. Nor does it currently bother with the links, as noted above.

Parameters

rhs Provides the event object to be copied.

```
13.10.3.3 ~event()

seq64::event::~event ( ) [virtual]
```

The restart_sysex() function does what we need. But now that m_sysex is a vector, no action is needed.

13.10.4 Member Function Documentation

This function is currently geared only toward support of the SMF 0 channel-splitting feature. Many of the member are not set to useful value when the MIDI file is read, so we don't handle them for now.

Warning

This function now copies the SysEx data, but the inclusion of SysEx events was not complete in Seq24, and it is still not complete in Sequencer64. Nor does it currently bother with the link the event might have.

rhs Provides the event object to be assigned.

Returns

Returns a reference to "this" object, to support the serial assignment of events.

13.10.4.2 operator<()

Otherwise, it returns true if the current timestamp is less than the event's timestamp.

Warning

The less-than operator is supposed to support a "strict weak ordering", and is supposed to leave equivalent values in the same order they were before the sort. However, every time we load and save our sample MIDI file, events get reversed. Here are program-changes that get reversed:

```
Save N: 0070: 6E 00 C4 48 00 C4 0C 00 C4 57 00 C4 19 00 C4 26 Save N+1: 0070: 6E 00 C4 26 00 C4 19 00 C4 57 00 C4 0C 00 C4 48

The 0070 is the offset within the versions of the b4uacuse-seq24.midi file.

Because of this mis-feature, and the very slow speed of loading a MIDI file when Sequencer64 is built for debugging, we are exploring using an std::mulitmap instead of an std::list. Search for occurrences of the SEQ64_USE_EVENT_MAP macro. (This actually works better than a list, for loading MIDI event, we have found, but may cause the upper limit of the number of playing sequences to drop a little, due to the overhead of incrementing multimap iterators versus list iterators).
```

Parameters

rhs The object to be compared against.

Returns

Returns true if the time-stamp and "rank" are less than those of the comparison object.

13.10.4.3 set_timestamp()

time Provides the time value, in ticks, to set as the timestamp.

13.10.4.4 get_timestamp()

```
midipulse seq64::event::get_timestamp ( ) const [inline]
```

13.10.4.5 get_channel()

```
midibyte seq64::event::get_channel ( ) const [inline]
```

13.10.4.6 check_channel()

Used in the SMF 0 track-splitting code.

Parameters

channel	The channel to check.
---------	-----------------------

Returns

Returns true if the given channel matches the event's channel.

13.10.4.7 is_channel_msg()

This function requires that the channel data have already been masked off.

Parameters

m The channel status or message byte to be tested, with the channel bits masked off.

We could add an optional boolean to cause the channel nybble to be explicitly cleared.

Returns

Returns true if the byte represents a MIDI channel message.

13.10.4.8 is_one_byte_msg()

The rest of the channel messages have two data bytes. This function requires that the channel data have already been masked off.

Parameters

m The channel status or message byte to be tested, with the channel bits masked off.

We could add an optional boolean to cause the channel nybble to be explicitly cleared.

Returns

Returns true if the byte represents a MIDI channel message that has only one data byte. However, if this function returns false, it might not be a channel message at all, so be careful.

13.10.4.9 is_two_byte_msg()

This function requires that the channel data have already been masked off.

Parameters

m The channel status or message byte to be tested, with the channel bits masked off.

We could add an optional boolean to cause the channel nybble to be explicitly cleared.

Returns

Returns true if the byte represents a MIDI channel message that has two data bytes. However, if this function returns false, it might not be a channel message at all, so be careful.

13.10.4.10 is_note_msg()

This function requires that the channel nybble has already been masked off.

Parameters

m The channel status or message byte to be tested, with the channel bits masked off.

We could add an optional boolean to cause the channel nybble to be explicitly cleared.

Returns

Returns true if the byte represents a MIDI note message.

13.10.4.11 is_strict_note_msg()

Parameters

m The channel status or message byte to be tested, with the channel bits masked off.

Returns

Returns true if the byte represents a MIDI note on/off message.

13.10.4.12 is desired cc or not cc()

Note

The old logic was the first line, but can be simplified to the second line; the third line shows the abstract representation. Also made sure of this using a couple truth tables.

```
(m != EVENT_CONTROL_CHANGE) || (m == EVENT_CONTROL_CHANGE && d == cc)
(m != EVENT_CONTROL_CHANGE) || (d == cc)
a || (! a && b) => a || b
```

```
\param m
    The channel status or message byte to be tested, with the channel
    bits masked off.

\param cc
    The desired cc value, which the datum must match, if the message is
    a control-change message.

\param datum
    The current datum, to be compared to cc, if the message is a
    control-change message.

\return
    Returns true if the message is not a control-change, or if it is
    and the cc and datum parameters match.
```

13.10.4.13 mod_timestamp()

```
void seq64::event::mod_timestamp (
    midipulse modtick ) [inline]
```

Parameters

modtick The tick value to	mod the timestamp against.
---------------------------	----------------------------

Returns

Returns a value ranging from 0 to _mod-1.

If a_status is a channel event, then the channel portion of the status is cleared using a bitwise AND against $EVE \leftarrow NT_CLEAR_CHAN_MASK$.

Found in yet another fork of seq24:

```
// ORL fait de la merde
```

He also provided a very similar routine: set_status_midibus().

Stazed:

The record parameter, if true, does not clear channel portion on record for channel specific recording. The channel portion is cleared in sequence::stream_event() by calling set_status() (a_record = false) after the matching channel is determined. Otherwise, we use a bitwise AND to clear the channel portion of the status. All events will be stored without the channel nybble. This is necessary since the channel is appended by midibus::play() based on the track.

Instead of adding a "record" parameter to set_status(), we provide a more specific function, set_status_keep_channel(), for use in the mastermidibus class.

The status byte, perhaps read from a MIDI file or edited in the sequencer's event editor. Sometime, this byte will have the channel nybble masked off. If that is the case, the eventcode/channel overload of this function is more appropriate.

Parameters

eventcode	The status byte, perhaps read from a MIDI file. This byte is assumed to have already had its low nybble cleared by masking against EVENT_CLEAR_CHAN_MASK.
channel	The channel byte. Combined with the event-code, this makes a valid MIDI "status" byte. This byte
	is assume to have already had its high nybble cleared by masking against
	EVENT_GET_CHAN_MASK.

13.10.4.16 set_status_keep_channel()

It replaces stazed's set_status() with the optional "record" parameter.

Parameters

eventcode	The status byte, generally read from the MIDI buss.
-----------	---

13.10.4.17 set_channel()

It actually just sets the m_channel member. Note that the sequence channel generally overrides this value in the usage of the event.

Parameters

channel	The channel byte to be set.

```
13.10.4.18 get_status()
midibyte seq64::event::get_status ( ) const [inline]
```

13.10.4.19 non_cc_match()

Parameters

status The status to be checked.

13.10.4.20 cc_match()

Parameters

st	The status to be checked.
СС	The control-change value to be checked against the events current "d0" value.

The second byte of data is zeroed. The data bytes are in a two =-byte array member, m_data.

Parameters

d1 The byte value to set as the first data byte.

d1	The first byte value to set.
d2	The second byte value to set.

13.10.4.23 get_data()

Parameters

d0	[out] The return reference for the first byte.
d1	[out] The return reference for the first byte.

13.10.4.24 increment_data1()

```
void seq64::event::increment_data1 ( ) [inline]
```

13.10.4.25 decrement_data1()

```
void seq64::event::decrement_data1 ( ) [inline]
```

13.10.4.26 increment_data2()

```
void seq64::event::increment_data2 ( ) [inline]
```

13.10.4.27 decrement_data2()

We now use a vector instead of an array, so there is no need for reallocation and copying of the current SYSEX data. The data represented by data and dsize is appended to that data buffer.

Parameters

da	ata	Provides the additional SYSEX data. If not provided, nothing is done, and false is returned.
ds	ize	Provides the size of the additional SYSEX data. If not provided, nothing is done.

Returns

Returns false if there was an EVENT_MIDI_SYSEX_END byte in the appended data, or if an error occurred, and the caller needs to stop trying to process the data. We're not quite sure what to do with any extra data remains.

```
13.10.4.29 append_sysex() [2/2]
```

Parameters

data A single MIDI byte of data, assumed to be part of a SYSEX message event.

13.10.4.30 restart_sysex()

```
void seq64::event::restart_sysex ( )
```

(The m_sysex member used to be a pointer.)

```
13.10.4.31 get_sysex() [1/2]
```

```
SysexContainer& seq64::event::get_sysex ( ) [inline]
```

ev

Provides a pointer to the event value to set. If null, then m_has_link is set to false, to guarantee that is_linked() is correct.

```
13.10.4.36 get_linked()
```

```
event* seq64::event::get_linked ( ) const [inline]
```

13.10.4.37 is_linked()

```
bool seq64::event::is_linked ( ) const [inline]
```

13.10.4.38 clear_link()

```
void seq64::event::clear_link ( ) [inline]
```

```
13.10.4.39 paint()
void seq64::event::paint ( ) [inline]
13.10.4.40 unpaint()
void seq64::event::unpaint ( ) [inline]
13.10.4.41 is_painted()
bool seq64::event::is_painted ( ) const [inline]
13.10.4.42 mark()
void seq64::event::mark ( ) [inline]
13.10.4.43 unmark()
void seq64::event::unmark ( ) [inline]
13.10.4.44 is_marked()
bool seq64::event::is_marked ( ) const [inline]
13.10.4.45 select()
void seq64::event::select ( ) [inline]
13.10.4.46 unselect()
void seq64::event::unselect ( ) [inline]
```

```
13.10.4.47 is_selected()
bool seq64::event::is_selected ( ) const [inline]
13.10.4.48 make_clock()
void seq64::event::make_clock ( ) [inline]
13.10.4.49 data()
midibyte seq64::event::data (
             int index ) const [inline]
13.10.4.50 get_note()
midibyte seq64::event::get_note ( ) const [inline]
13.10.4.51 set_note()
void seq64::event::set_note (
             midibyte note ) [inline]
Parameters
 note | Provides the note value to set.
13.10.4.52 transpose_note()
void seq64::event::transpose_note (
```

int tn)

The amount (positive or negative) to transpose a note. If the result is out of range, the transposition is not performed.

13.10.4.55 is_note_on()

```
bool seq64::event::is_note_on ( ) const [inline]
```

Provides the velocity value to set.

Currently assumes that the channel nybble has already been stripped.

Returns

Returns true if m_status is EVENT_NOTE_ON.

```
13.10.4.56 is_note_off()
```

```
bool seq64::event::is_note_off ( ) const [inline]
```

Currently assumes that the channel nybble has already been stripped.

Returns

Returns true if m_status is EVENT_NOTE_OFF.

```
13.10.4.57 is_note()
```

```
bool seq64::event::is_note ( ) const [inline]
```

All of these are notes, associated with a MIDI key value. Uses the static function is_note_msg().

Returns

The return value of is_note_msg() is returned.

13.10.4.58 is_note_off_recorded()

```
bool seq64::event::is_note_off_recorded ( ) const [inline]
```

The channel nybble is masked off before the test.

Returns

Returns true if the event is a Note On event with velocity of 0.

13.10.4.59 print()

```
void seq64::event::print ( ) const
```

13.10.4.60 get_rank()

```
int seq64::event::get_rank ( ) const
```

The ranking, from high to low, is note off, note on, aftertouch, channel pressure, and pitch wheel, control change, and program changes.

note on/off, aftertouch, control change, etc.) The sort order is not determined by the actual status values.

The lower the ranking the more upfront an item comes in the sort order.

Returns

Returns the rank of the current m status byte.

13.10.5 Field Documentation

```
13.10.5.1 m_timestamp
```

```
midipulse seq64::event::m_timestamp [private]
```

13.10.5.2 m_status

```
midibyte seq64::event::m_status [private]
```

The channel is included when recording MIDI, but, once a sequence with a matching channel is found, the channel nybble is cleared for storage. The channel will be added back on the MIDI bus upon playback. The high nibble = type of event; The low nibble = channel. Bit 7 is present in all status bytes.

```
13.10.5.3 m_channel
```

```
midibyte seq64::event::m_channel [private]
```

This member adds another 4 bytes to the event object, most likely.

```
13.10.5.4 m_data
```

```
midibyte seq64::event::m_data[SEQ64_MIDI_DATA_BYTE_COUNT] [private]
```

Remember that the most-significant bit of a data byte is always 0. A one-byte message uses only the 0th index.

```
13.10.5.5 m_sysex
```

```
SysexContainer seq64::event::m_sysex [private]
```

Adapted from Stazed's Seq32 project on GitHub.

```
13.10.5.6 m_sysex_size
```

```
int seq64::event::m_sysex_size [private]
```

Perhaps redundant.

```
13.10.5.7 m_linked
```

```
event* seq64::event::m_linked [private]
```

```
13.10.5.8 m_has_link
```

```
bool seq64::event::m_has_link [private]
```

This item is used [via the get_link() and link() accessors] in the sequence class.

13.10.5.9 m_selected

```
bool seq64::event::m_selected [private]
```

13.10.5.10 m_marked

```
bool seq64::event::m_marked [private]
```

13.10.5.11 m_painted

```
bool seq64::event::m_painted [private]
```

13.11 seq64::event_list::event_key Class Reference

Provides a key value for an event map.

Public Member Functions

• event_key (midipulse tstamp, int rank)

Principal event_key constructor.

• event_key (const event &e)

Event-based constructor.

bool operator< (const event_key &rhs) const

Provides the minimal operator needed to sort events using an event_key.

Private Attributes

• midipulse m_timestamp

The primary key-value for the key.

• int m_rank

The sub-key-value for the key.

13.11.1 Detailed Description

Its types match the m_timestamp and get_rank() function of this event class.

13.11.2 Constructor & Destructor Documentation

Parameters

tstamp	The time-stamp is the primary part of the key. It is the most important key item.
rank	Rank is an arbitrary number used to prioritize events that have the same time-stamp. See the
	event::get_rank() function for more information.

This constructor makes it even easier to create an event_key. Note that the call to event::get_rank() makes a simple calculation based on the status of the event.

Parameters

rhs Provides the event key to be copied.

13.11.3 Member Function Documentation

13.11.3.1 operator<()

Parameters

rhs Provides the event key to be compared against.

Returns

Returns true if the rank and timestamp of the current object are less than those of rhs.

13.11.4 Field Documentation

```
13.11.4.1 m_timestamp

midipulse seq64::event_list::event_key::m_timestamp [private]

13.11.4.2 m_rank

int seq64::event_list::event_key::m_rank [private]
```

13.12 seq64::event_list Class Reference

The event_list class is a receptable for MIDI events.

Data Structures

· class event_key

Provides a key value for an event map.

Public Member Functions

event_list ()

Principal constructor.

event_list (const event_list &a_rhs)

Copy constructor.

event_list & operator= (const event_list &a_rhs)

Principal assignment operator.

∼event_list ()

A rote destructor.

• iterator begin ()

'Getter' function for member m_events.begin(), non-constant version.

• const_iterator begin () const

'Getter' function for member m_events.begin(), constant version.

• iterator end ()

 ${\it 'Getter' function for member m_events.end(), non-constant version.}$

• const_iterator end () const

 ${\it 'Getter' function for member m_events.end(), constant version.}$

• int count () const

Returns the number of events stored in m_events.

• bool empty () const

Returns true if there are no events.

bool add (const event &e)

Adds an event to the internal event list in an optionally sorted manner.

• bool append (const event &e)

Adds an event to the internal event list without sorting.

• void push back (const event &)

The multimap version of this function does nothing.

· bool is modified () const

'Getter' function for member m_is_modified

· void unmodify ()

'Setter' function for member m_is_modified This function may be needed by some of the sequence editors.

• void remove (iterator ie)

Provides a wrapper for the iterator form of erase(), which is the only one that sequence uses.

• void clear ()

Provides a wrapper for clear().

• void merge (event_list &el, bool presort=true)

Provides a merge operation for the event multimap analogous to the merge operation for the event list.

• void sort ()

TEMPORARILY HERE for gdb.

Static Public Member Functions

• static event & dref (iterator ie)

Dereference access for list or map.

static const event & dref (const iterator ie)

Dereference const access for list or map.

Private Types

typedef std::multimap< event_key, event > Events

Types to use to swap between list and multimap implementations.

- typedef std::pair< event_key, event > EventsPair
- typedef std::multimap< event key, event >::iterator iterator
- typedef std::multimap< event_key, event >::const_iterator const_iterator

Private Member Functions

• void link_new ()

Links a new event.

• void clear links ()

Clears all event links and unmarks them all.

void verify and link (midipulse slength)

This function verifies state: all note-ons have an off, and it links note-offs with their note-ons.

bool mark_selected ()

Marks all selected events.

· void mark_out_of_range (midipulse slength)

Marks all events that have a time-stamp that is out of range.

void mark_all ()

Marks all events.

void unmark_all ()

Unmarks all events.

• bool remove_marked ()

Removes marked events.

· void unpaint_all ()

Unpaints all list-events.

• int count_selected_notes () const

Counts the selected note-on events in the event list.

· bool any selected notes () const

Indicates that at least one note is selected.

int count_selected_events (midibyte status, midibyte cc) const

Counts the selected events, with the given status, in the event list.

void select_all ()

Selects all events, unconditionally.

void unselect_all ()

Deselects all events, unconditionally.

· void print () const

Prints a list of the currently-held events.

· const Events & events () const

'Getter' function for member m_events

Private Attributes

· Events m events

This list holds the current pattern/sequence events.

bool m_is_modified

A new flag to indicate if an event was added or removed.

Friends

- · class editable events
- · class midifile
- · class midi_container
- · class midi_splitter
- class sequence

13.12.1 Detailed Description

Two implementations, an std::multimap, and the original, an std::list, are provided for comparison, and are selected at build time, by manually defining the SEQ64_USE_EVENT_MAP macro near the top of this module.

13.12.2 Member Typedef Documentation

13.12.2.1 Events

typedef std::multimap<event_key, event> seq64::event_list::Events [private]

13.12.2.2 EventsPair

```
typedef std::pair<event_key, event> seq64::event_list::EventsPair [private]
```

13.12.2.3 iterator

```
typedef std::multimap<event_key, event>::iterator seq64::event_list::iterator [private]
```

13.12.2.4 const_iterator

```
typedef std::multimap<event_key, event>::const_iterator seq64::event_list::const_iterator
[private]
```

13.12.3 Constructor & Destructor Documentation

Parameters

rhs Provides the event list to be copied.

```
13.12.3.3 ~event_list()
seq64::event_list::~event_list ( )
```

13.12.4 Member Function Documentation

```
13.12.4.1 operator=()
```

Follows the stock rules for such an operator, just assigning member values.

Parameters

rhs Provides the event list to be assigned.

```
13.12.4.2 begin() [1/2]
iterator seq64::event_list::begin ( ) [inline]
13.12.4.3 begin() [2/2]
const_iterator seq64::event_list::begin ( ) const [inline]
13.12.4.4 end() [1/2]
iterator seq64::event_list::end ( ) [inline]
13.12.4.5 end() [2/2]
const_iterator seq64::event_list::end ( ) const [inline]
13.12.4.6 count()
int seq64::event_list::count ( ) const [inline]
We like returning an integer instead of size_t, and rename the function so nobody is fooled.
13.12.4.7 empty()
bool seq64::event_list::empty ( ) const [inline]
return m_events.size() == 0;
13.12.4.8 add()
bool seq64::event_list::add (
```

const event & e) [inline]

Parameters

e Provides the event to be added to the list.

Returns

Returns true. We assume the insertion succeeded, and no longer care about an increment in container size. It's a multimap, so it always inserts, and if we don't have memory left, all bets are off anyway.

13.12.4.9 append()

It is a wrapper, wrapper for insert() or push_front(), with an option to call sort().

The add() function without sorting, useful to speed up the initial container loading into the event-list.

For the std::multimap implementation, This is an option if we want to make sure the insertion succeed.

If the std::list implementation has been built in, then the event list is sorted after the addition. This is a time-consuming operation.

Warning

This pushing (and, in writing the MIDI file, the popping), causes events with identical timestamps to be written in reverse order. Doesn't affect functionality, but it's puzzling until one understands what is happening. That's why we're now preferring to use a multimap as the container.

Parameters

e Provides the event to be added to the list.

Returns

Returns true. We assume the insertion succeeded, and no longer care about an increment in container size. It's a multimap, so it always inserts, and if we don't have memory left, all bets are off anyway.

13.12.4.10 push_back()

Currently, no check on removal is performed. Sets the modified-flag.

Parameters

ie Provides the iterator to the event to be removed.

We have certain constraints to preserve, as the following discussion shows.

For std::list, sequence merges list T into list A by first calling T.sort(), and then A.merge(T). The merge() operation merges T into A by transferring all of its elements, at their respective ordered positions, into A. Both containers must already be ordered.

The merge effectively removes all the elements in T (which becomes empty), and inserts them into their ordered position within container (which expands in size by the number of elements transferred). The operation is performed without constructing nor destroying any element, whether T is an Ivalue or an rvalue, or whether the value-type supports move-construction or not.

Each element of T is inserted at the position that corresponds to its value according to the strict weak ordering defined by operator <. The resulting order of equivalent elements is stable (i.e. equivalent elements preserve the relative order they had before the call, and existing elements precede those equivalent inserted from x). The function does nothing if (&x == this).

For std::multimap, sorting is automatic. However, unless move-construction is supported, merging will be less efficient than for the list version. Also, we need a way to include duplicates of each event, so we need to use a multimap. Once all this setup, merging is really just insertion. And, since sorting isn't needed, the multimap actually turns out to be faster.

Parameters

el	Provides the event list to be merged into the current event list.
presort	If true, the events are presorted. This is a requirement for merging an std::list, but is a no-op for the std::multimap implementation.

13.12.4.16 sort()

```
void seq64::event\_list::sort ( )
```

```
13.12.4.17 dref() [1/2]
```

```
static event& seq64::event_list::dref (
          iterator ie ) [inline], [static]
```

Parameters

ie Provides the iterator to the event to which to get a reference.

```
13.12.4.18 dref() [2/2]
```

Parameters

ie Provides the iterator to the event to which to get a reference.

```
13.12.4.19 link_new()
```

```
void seq64::event_list::link_new ( ) [private]
```

This function checks for a note on, then look for its note off. This function is provided in the event_list because it does not depend on any external data. Also note that any desired thread-safety must be provided by the caller.

13.12.4.20 clear_links()

```
void seq64::event_list::clear_links ( ) [private]
```

13.12.4.21 verify_and_link()

Stazed (seq32):

```
This function now deletes any notes that are \geq= m_length, so any resize or move of notes must modify for wrapping if Note Off is \geq= m_length.
```

Not threadsafe As in most case, the caller will use an automutex to call this function safely.

Parameters

```
slength Provides the length beyond which events will be pruned.
```

13.12.4.22 mark_selected()

```
bool seq64::event_list::mark_selected ( ) [private]
```

Returns

Returns true if there was even one event selected and marked.

13.12.4.23 mark_out_of_range()

Used for killing (pruning) those events not in range. If the current time-stamp is greater than the length, then the event is marked for pruning.

Note

This code was comparing the timestamp as greater than or equal to the sequence length. However, being equal is fine. This may explain why the midifile code would add one tick to the length of the last note when processing the end-of-track.

Parameters

slength Provides the length beyond which events will be pruned.

13.12.4.24 mark_all()

```
void seq64::event_list::mark_all ( ) [private]
```

Not yet used, but might come in handy with the event editor dialog.

13.12.4.25 unmark_all()

```
void seq64::event_list::unmark_all ( ) [private]
```

13.12.4.26 remove_marked()

```
bool seq64::event_list::remove_marked ( ) [private]
```

Note how this function handles removing a value to avoid incrementing a now-invalid iterator.

Threadsafe

Returns

Returns true if at least one event was removed.

13.12.4.27 unpaint_all()

```
void seq64::event_list::unpaint_all ( ) [private]
```

13.12.4.28 count_selected_notes()

```
int seq64::event_list::count_selected_notes ( ) const [private]
```

13.12.4.29 any_selected_notes()

```
bool seq64::event_list::any_selected_notes ( ) const [private]
```

Acts like event_list::count_selected_notes(), but stops after finding a selected note. We could add a flag to count
__selected_notes() to break, I suppose.

Returns

Returns true if at least one note is selected.

13.12.4.30 count_selected_events()

If the event is a control change (CC), then it must also match the given CC value.

Parameters

status	The desired status value to count.
СС	The desired control-change to count. Used only if the status parameter indicates a control-change event.

Returns

Returns the number of selected events.

13.12.4.31 select_all()

```
void seq64::event_list::select_all ( ) [private]
```

13.12.4.32 unselect_all()

```
void seq64::event_list::unselect_all ( ) [private]
```

13.12.4.33 print()

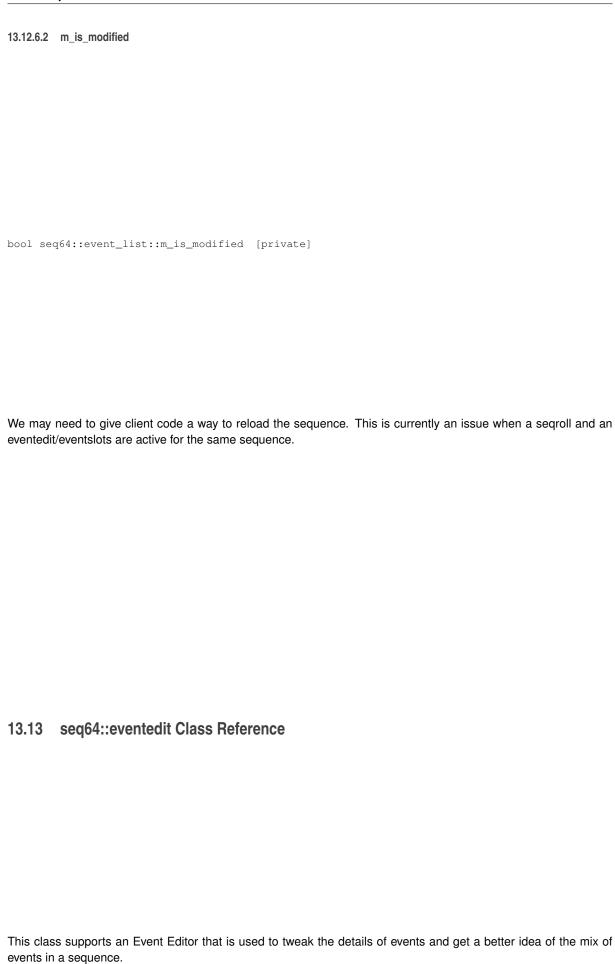
```
void seq64::event_list::print ( ) const [private]
```

13.12.6.1 m_events

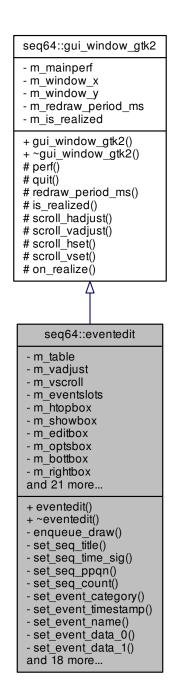
Events seq64::event_list::m_events [private]

```
13.12.4.34 events()
const Events& seq64::event_list::events ( ) const [inline], [private]
13.12.5 Friends And Related Function Documentation
13.12.5.1 editable_events
friend class editable_events [friend]
13.12.5.2 midifile
friend class midifile [friend]
13.12.5.3 midi_container
friend class midi_container [friend]
13.12.5.4 midi_splitter
friend class midi_splitter [friend]
13.12.5.5 sequence
friend class sequence [friend]
13.12.6 Field Documentation
```

Generated by Doxygen



Inheritance diagram for seq64::eventedit:



Public Member Functions

• eventedit (perform &p, sequence &seq)

Principal constructor, has a reference to a perform object.

virtual ∼eventedit ()

This rote constructor does nothing.

Private Member Functions

void enqueue_draw ()

Helper wrapper for calling eventslots::queue_draw().

void set_seq_title (const std::string &title)

Sets m_label_seq_name to the title.

void set_seq_time_sig (const std::string &sig)

Sets m_label_time_sig to the time-signature string.

• void set_seq_ppqn (const std::string &p)

Sets m_label_ppqn to the parts-per-quarter-note string.

void set_seq_count ()

Sets m label ev count to the number-of-events string.

void set_event_category (const std::string &c)

Sets m_label_category to the category string.

void set_event_timestamp (const std::string &ts)

Sets m entry ev timestamp to the time-stamp string.

void set_event_name (const std::string &n)

Sets m_entry_ev_name to the name-of-event string.

void set_event_data_0 (const std::string &d)

Sets m_entry_ev_data_0 to the first data byte string.

void set_event_data_1 (const std::string &d)

Sets m_entry_data_1 to the second data byte string.

void perf modify ()

Provides a way to mark the perform object as modified, when the modified sequence is saved.

• void set_dirty (bool flag=true)

Sets the "modified" status of the user-interface.

void v adjustment (int value)

Sets the parameters for the vertical scroll-bar, using only the value parameter.

void v_adjustment (int value, int lower, int upper)

Sets the parameters for the vertical scroll-bar that is associated with the eventslots event-list user-interface.

void change focus (bool set it=true)

Changes what perform and mainwid see as the "current sequence".

void close_out ()

Handles closing the sequence editor, common code for handle_cancel() and handle_close().

• void handle close ()

Handles closing the sequence editor.

void handle_delete ()

Initiates the deletion of the current editable event.

void handle_insert ()

Initiates the insertion of a new editable event.

• void handle modify ()

Passes the edited fields to the current editable event in the eventslot.

void handle_save ()

Handles saving the edited data back to the original sequence.

• void handle cancel ()

Cancels the edits and closes the dialog box.

void on_realize ()

This callback function calls the base-class on_realize() function.

void on set focus (Widget *focus)

On receiving focus, attempt to tell mainwid that this sequence is now the current sequence.

• bool on_focus_in_event (GdkEventFocus *)

Implements the on-focus event handling.

bool on_focus_out_event (GdkEventFocus *)

Implements the on-unfocus event handling.

bool on_key_press_event (GdkEventKey *ev)

This function is the callback for a key-press event.

· bool on_delete_event (GdkEventAny *event)

Handles an on-delete event.

Private Attributes

• Gtk::Table * m table

A whole horde of GUI elements.

Gtk::Adjustment * m_vadjust

Vertical paging for event list.

• Gtk::VScrollbar * m_vscroll

Vertical scroll for event list.

• eventslots * m eventslots

Drawing area for events.

• Gtk::HBox * m htopbox

Padding at the top of the dialog.

• Gtk::VBox * m_showbox

Area for sequence information.

• Gtk::VBox * m editbox

Text-edits and buttons for data.

• Gtk::VBox * m_optsbox

Reserved for future options.

• Gtk::HBox * m_bottbox

Holds the Save and Close buttons.

• Gtk::VBox * m_rightbox

Used for padding on right side.

• Gtk::Button * m_button_del

"Delete Current Event (*)" button.

• Gtk::Button * m_button_ins

"Insert New Event" button.

• Gtk::Button * m_button_modify

"Modify New Event" button.

• Gtk::Button * m_button_save

"Save to Sequence" button.

• Gtk::Button * m_button_cancel

"Close" button.

• Gtk::Label * m_label_seq_name

Items for the inside of the m_showbox member.

• Gtk::Label * m_label_time_sig

Shows time signature for pattern.

• Gtk::Label * m_label_ppqn

Shows the parts per quarter note.

Gtk::Label * m label channel

Shows channel number of pattern.

• Gtk::Label * m label ev count

Shows the count of pattern events.

```
    Gtk::Label * m_label_spacer

     Spacer for the showbox elements.
• Gtk::Label * m_label_modified
     Shows "[Modified]" if edited.

    Gtk::Label * m_label_category

     Items for the inside of the m_editbox member.
• Gtk::Entry * m_entry_ev_timestamp
```

Text edit for event time-stamp.

• Gtk::Entry * m_entry_ev_name

Text edit for MIDI event name.

• Gtk::Entry * m_entry_ev_data_0

Text edit for first event datum.

• Gtk::Entry * m_entry_ev_data_1

Text edit for second event datum.

• Gtk::Label * m_label_time_fmt

Optsbox item, only "Sequencer64".

• Gtk::Label * m_label_right

Padding at the right of dialog.

· sequence & m_seq

A reference to the sequence being edited, to control its editing flag.

• bool m_have_focus

Indicates that the focus has already been changed to this sequence.

Friends

· class eventslots

Additional Inherited Members

13.13.1 Constructor & Destructor Documentation

13.13.1.1 eventedit()

```
seq64::eventedit::eventedit (
            perform & p,
             sequence & seq )
```

We've reordered the pointer members and put them in the initializer list to make the constructor a bit cleaner.

Adjustment parameters:

```
value
               initial value
                minimum value
lower
                maximum value
step_increment step increment
page_increment page increment
page_size
                page size
```

Table constructor parameters:

rows columns homogenous

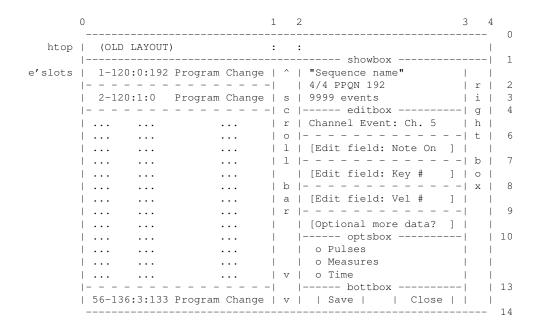
Table attach() parameters:

child widget to add.

left_attach column number to attach left side of a child widget right_attach column number to attach right side of a child widget top_attach row number to attach the top of a child widget bottom_attach row number to attach the bottom of a child widget xoptions properties of the child widget when table resized yoptions same as xoptions, except vertical.

xpadding padding on L and R of widget added to table ypadding amount of padding above and below the child widget

Layout:



Parameters

p	Refers to the main performance object.
seq	Refers to the sequence holding the event data to be edited.

The sequence dirty flags, and this allows the sequence's pattern slot to be updated, which, for example, allows the new experimental in-edit-highlight feature to work. To get the eventedit to also show the in-edit highlighting, we can make the sequence::set_dirty_mp() call. This call does not cause a prompt for saving the file when exiting.

```
13.13.1.2 ~eventedit()
seq64::eventedit::~eventedit ( ) [virtual]
```

We're going to have to run the application through valgrind to make sure that nothing is left behind.

13.13.2 Member Function Documentation

13.13.2.1 enqueue_draw()

```
void seq64::eventedit::enqueue_draw ( ) [private]
```

13.13.2.2 set_seq_title()

Parameters

```
title The name of the sequence.
```

13.13.2.3 set_seq_time_sig()

```
void seq64::eventedit::set_seq_time_sig (
    const std::string & sig ) [private]
```

Parameters

sig | The time signature of the sequence.

13.13.2.4 set_seq_ppqn()

Parameters

p The parts-per-quarter-note string for the sequence.

13.13.2.5 set_seq_count()

```
void seq64::eventedit::set_seq_count ( ) [private]
```

13.13.2.6 set_event_category()

Parameters

c The category string for the current event.

13.13.2.7 set_event_timestamp()

Parameters

ts | The time-stamp string for the current event.

13.13.2.8 set_event_name()

Parameters

n The name-of-event string for the current event.

13.13.2.9 set_event_data_0()

Parameters

d The first data byte string for the current event.

```
13.13.2.10 set_event_data_1()
```

void seq64::eventedit::set_event_data_1 (

```
Parameters
```

d The second data byte string for the current event.

const std::string & d) [private]

```
13.13.2.11 perf_modify()
```

```
void seq64::eventedit::perf_modify ( ) [private]
```

13.13.2.12 set_dirty()

```
void seq64::eventedit::set_dirty (
          bool flag = true ) [private]
```

This includes changing a label and enabling/disabling the Save button.

Parameters

flag If true, the modified status is indicated, otherwise it is cleared.

```
13.13.2.13 v_adjustment() [1/2]
```

This function overload provides a common use case.

Parameters

value The new current value to be indicated by the scroll-bar.

```
13.13.2.14 v_adjustment() [2/2]
```

```
int lower,
int upper ) [private]
```

It keeps the frame scroll-bar in sync with the frame movement actions. Some of the parameters are obtained from the eventslots object:

```
Page size comes from eventslots::line_maximum().Page increment is a little less than the page-size value.
```

Parameters

value	The current value to be indicated by the scroll-bar. It will lie between the lower and upper parameter.
lower	The lowest value to be indicated by the scroll-bar.
upper	The highest value to be indicated by the scroll-bar.

13.13.2.15 change_focus()

Similar to the same function in sequedit.

Parameters

set⊷	If true (the default value), indicates we want focus, otherwise we want to give up focus.
_it	

13.13.2.16 close_out()

```
void seq64::eventedit::close_out ( ) [private]
```

13.13.2.17 handle_close()

```
void seq64::eventedit::handle_close ( ) [private]
```

Simply calls close_out().

13.13.2.18 handle_delete()

```
void seq64::eventedit::handle_delete ( ) [private]
```

```
13.13.2.19 handle_insert()
```

```
void seq64::eventedit::handle_insert ( ) [private]
```

The event's location will be determined by the timestamp and existing events. Note that we have to recalibrate the scroll-bar when we insert/delete events by calling v_adjustment().

13.13.2.20 handle_modify()

```
void seq64::eventedit::handle_modify ( ) [private]
```

Note that there are two cases to worry about. If the timestamp has not changed, then we can simply modify the existing current event in place. Otherwise, we need to delete the old event and insert the new one. But that is done for us by eventslots::modify_current_event().

13.13.2.21 handle_save()

```
void seq64::eventedit::handle_save ( ) [private]
```

The event list in the original sequence is cleared, and the editable events are converted to plain events, and added to the container, one by one.

Todo Could also support writing the events to a new sequence, for added flexibility.

13.13.2.22 handle_cancel()

```
void seq64::eventedit::handle_cancel ( ) [private]
```

In order for removing the current-highlighting in the mainwd or perfedit windows, some of the work of handle_close() needs to be done here as well.

13.13.2.23 on_realize()

```
void seq64::eventedit::on_realize ( ) [private]
```

Then it sets the vertical adjustment to account for the number of events in the eventslot.

13.13.2.24 on_set_focus()

Only works in certain circumstances.

Parameters

focus The widget that has the focus. Merely passed on to gui_window_gtk2's version of this function.

```
13.13.2.25 on_focus_in_event()
```

```
bool seq64::eventedit::on_focus_in_event (
    GdkEventFocus * ) [private]
```

It sets the focus flag and calls change_focus().

```
13.13.2.26 on focus out event()
```

It resets the focus flag and calls change_focus().

13.13.2.27 on_key_press_event()

If the Up or Down arrow is pressed (later, k and j :-), then we tell the eventslots object to move the "current event" highlighting up or down. In Gtkmm, these arrows also cause movement from one edit field to the next, so we disable that process if the event was handled here.

Note that the Delete key is needed for the edit fields. For now, we replace it with the asterisk, which is easy to access from the numeric pad of a keyboard, and allows for rapid deletion. The Insert key also causes confusing effects in the edit fields, so we replaced it by the slash, but that didn't work. Note that the asterisk and slash should not be required in any of the edit fields. HOWEVER, "/" still gets passed the edit fields (!), so you'll just have to click the button to insert an event. Let's try the backslash! No go there, either.

Parameters

ev The key event to process.

Returns

Returns true if the event got handled somewhere along the line.

13.13.2.28 on_delete_event()

It sets the sequence object's editing flag to false, and deletes "this". This function is called if the "Close" ("X") button in the window's title bar is clicked. That is a different action from clicking the Close button.

Returns

Always returns false.

13.13.3 Friends And Related Function Documentation

13.13.3.1 eventslots

```
friend class eventslots [friend]
```

13.13.4 Field Documentation

```
13.13.4.1 m_table
```

```
Gtk::Table* seq64::eventedit::m_table [private]
```

Provides the layout table for UI.

13.13.4.2 m_vadjust

Gtk::Adjustment* seq64::eventedit::m_vadjust [private]

13.13.4.3 m_vscroll

Gtk::VScrollbar* seq64::eventedit::m_vscroll [private]

13.13.4.4 m_eventslots

eventslots* seq64::eventedit::m_eventslots [private]

```
13.13.4.5 m_htopbox
Gtk::HBox* seq64::eventedit::m_htopbox [private]
13.13.4.6 m_showbox
Gtk::VBox* seq64::eventedit::m_showbox [private]
13.13.4.7 m_editbox
Gtk::VBox* seq64::eventedit::m_editbox [private]
13.13.4.8 m_optsbox
Gtk::VBox* seq64::eventedit::m_optsbox [private]
13.13.4.9 m_bottbox
Gtk::HBox* seq64::eventedit::m_bottbox [private]
13.13.4.10 m_rightbox
Gtk::VBox* seq64::eventedit::m_rightbox [private]
13.13.4.11 m_button_del
Gtk::Button* seq64::eventedit::m_button_del [private]
13.13.4.12 m_button_ins
```

Gtk::Button* seq64::eventedit::m_button_ins [private]

```
13.13.4.13 m_button_modify
Gtk::Button* seq64::eventedit::m_button_modify [private]
13.13.4.14 m_button_save
Gtk::Button* seq64::eventedit::m_button_save [private]
13.13.4.15 m_button_cancel
Gtk::Button* seq64::eventedit::m_button_cancel [private]
13.13.4.16 m_label_seq_name
Gtk::Label* seq64::eventedit::m_label_seq_name [private]
Shows the name of the pattern.
13.13.4.17 m_label_time_sig
Gtk::Label* seq64::eventedit::m_label_time_sig [private]
13.13.4.18 m_label_ppqn
Gtk::Label* seq64::eventedit::m_label_ppqn [private]
13.13.4.19 m_label_channel
Gtk::Label* seq64::eventedit::m_label_channel [private]
13.13.4.20 m_label_ev_count
```

Gtk::Label* seq64::eventedit::m_label_ev_count [private]

```
13.13.4.21 m_label_spacer
Gtk::Label* seq64::eventedit::m_label_spacer [private]
13.13.4.22 m_label_modified
Gtk::Label* seq64::eventedit::m_label_modified [private]
13.13.4.23 m_label_category
Gtk::Label* seq64::eventedit::m_label_category [private]
Shows the type of MIDI event.
13.13.4.24 m_entry_ev_timestamp
Gtk::Entry* seq64::eventedit::m_entry_ev_timestamp [private]
13.13.4.25 m_entry_ev_name
Gtk::Entry* seq64::eventedit::m_entry_ev_name [private]
13.13.4.26 m_entry_ev_data_0
Gtk::Entry* seq64::eventedit::m_entry_ev_data_0 [private]
13.13.4.27 m_entry_ev_data_1
Gtk::Entry* seq64::eventedit::m_entry_ev_data_1 [private]
13.13.4.28 m_label_time_fmt
Gtk::Label* seq64::eventedit::m_label_time_fmt [private]
```

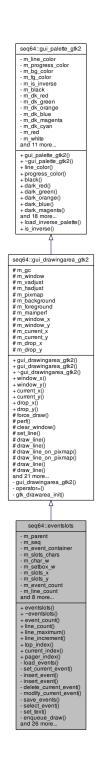
```
13.13.4.29 m_label_right
Gtk::Label* seq64::eventedit::m_label_right [private]
13.13.4.30 m_seq
sequence& seq64::eventedit::m_seq [private]
13.13.4.31 m_have_focus
bool seq64::eventedit::m_have_focus [private]
```

This item is to modify the mainwid and perfedit "edit-sequence" value in order to highlight pattern slot of the pattern/event editor that currently has the user-input focus.

13.14 seq64::eventslots Class Reference

This class implements the left-side list of events in the pattern event-edit window.

Inheritance diagram for seq64::eventslots:



Public Member Functions

- eventslots (perform &p, eventedit &parent, sequence &seq, Gtk::Adjustment &vadjust)

 Principal constructor for this user-interface object.
- virtual ~eventslots ()
 Let's provide a do-nothing virtual destructor.
- int event_count () const

'Getter' function for member m_event_count Returns the number of total events in the sequence represented by the eventslots object.

int line_count () const

'Getter' function for member m_line_count Returns the current number of rows (events) in the eventslots's display.

• int line maximum () const

'Getter' function for member m_line_maximum Returns the maximum number of rows (events) in the eventslots's display.

· int line increment () const

Provides the "page increment" or "line increment" of the frame, This value is the current line-maximum of the frame minus its overlap value.

int top_index () const

'Getter' function for member m_top_index

• int current_index () const

'Getter' function for member m current index

int pager_index () const

'Getter' function for member m_pager_index

Private Member Functions

· bool load_events ()

Grabs the event list from the sequence and uses it to fill the editable-event list.

void set_current_event (const editable_events::iterator ei, int index, bool full_redraw=true)

Set the current event, which is the event that is highlighted.

bool insert_event (const editable_event &edev)

Inserts an event.

 bool insert_event (const std::string &evtimestamp, const std::string &evname, const std::string &evdata0, const std::string &evdata1)

Inserts an event based on the setting provided, which the eventedit object gets from its Entry fields.

• bool delete_current_event ()

Deletes the current event, and makes adjustments due to that deletion.

• bool modify_current_event (const std::string &evtimestamp, const std::string &evname, const std::string &evdata0, const std::string &evdata1)

Modifies the data in the currently-selected event.

• bool save_events ()

Writes the events back to the sequence.

void select_event (int event_index=SEQ64_NULL_EVENT_INDEX, bool full_redraw=true)

Selects and highlights the event that is located in the frame at the given event index.

 void set_text (const std::string &evcategory, const std::string &evtimestamp, const std::string &evname, const std::string &evdata0, const std::string &evdata1)

Sets the text in the parent dialog, eventedit.

• void enqueue_draw ()

Wraps queue_draw().

int convert_y (int y)

Converts a y-value into an event index relative to 0 (the top of the eventslots window/pixmap) and returns it.

• void draw event (editable events::iterator ei, int index)

Draw the given slot/event.

void draw_events ()

Draws all of the events in the current eventslots frame.

· void change vert ()

Change the vertical offset of events.

void page_movement (int new_value)

Adjusts the vertical position of the frame according to the given new scrollbar/vadjust value.

void page_topper (editable_events::iterator newcurrent)

Adjusts the vertical position of the frame according to the given new bottom iterator.

int decrement_top ()

Decrements the top iterator, if possible.

int increment_top ()

Increments the top iterator, if possible.

int decrement_current ()

Decrements the current iterator, if possible.

int increment_current ()

Increments the current iterator, if possible.

• int decrement_bottom ()

Decrements the bottom iterator, if possible.

• int increment_bottom ()

Increments the bottom iterator, if possible.

• void on realize ()

Handles the callback when the window is realized.

bool on_expose_event (GdkEventExpose *ev)

Handles an on-expose event.

bool on_button_press_event (GdkEventButton *ev)

Provides the callback for a button press, and it handles only a left mouse button.

 $\bullet \ \ bool \ on_button_release_event \ (GdkEventButton *ev)\\$

Currently does nothing.

• bool on focus in event (GdkEventFocus *ev)

This callback is an attempt to get keyboard focus into the eventslots pixmap area.

bool on focus out event (GdkEventFocus *ev)

This callback handles an out-of-focus event by resetting the flag HAS_FOCUS.

• bool on scroll event (GdkEventScroll *ev)

Handle the scrolling of the window.

• void on_size_allocate (Gtk::Allocation &)

Handles a size-allocation event.

void on_move_up ()

Move to the previous event.

void on_move_down ()

Move to the next event.

void on_frame_up ()

Move to the previous frame.

• void on frame down ()

Move to the next frame.

void on_frame_home ()

Move to the first frame.

• void on_frame_end ()

Move to the last frame.

Private Attributes

· eventedit & m parent

Provides a link to the eventedit that created this object.

· sequence & m_seq

Provides a reference to the sequence that this dialog is meant to view or modify.

editable_events m_event_container

Holds the editable events for this sequence.

· int m slots chars

Provides the number of the characters in the name box.

· int m char w

Provides the "real" width of a character.

· int m setbox w

Provides the width of the "set number" box.

int m_slots_x

Provides the width of the names box, which is the width of a character for 24 characters.

int m_slots_y

Provides the height of the names box, which is hardwired to 24 pixels.

· int m event count

The current number of events in the edited container.

· int m line count

Counts the number of displayed events, which depends on how many events there are (m_event_count) and the size of the event list (m_line_maximum).

int m_line_maximum

Counts the maximum number of displayed events, which depends on the size of the event list (and thus the size of the dialog box for the event editor).

· int m line overlap

Provides a little overlap for paging through the frame.

· int m top index

The index of the event that is 0th in the visible list of events.

int m_current_index

Indicates the index of the current event within the frame.

· editable_events::iterator m_top_iterator

Provides the top "pointer" to the start of the editable-events section that is being shown in the user-interface.

• editable_events::iterator m_bottom_iterator

Provides the bottom "pointer" to the end of the editable-events section that is being shown in the user-interface.

· editable_events::iterator m_current_iterator

Provides the "pointer" to the event currently in focus.

· int m pager index

Indicates the event index that matches the index value of the vertical pager.

Friends

· class eventedit

Additional Inherited Members

13.14.1 Constructor & Destructor Documentation

```
13.14.1.1 eventslots()
```

```
seq64::eventslots::eventslots (
            perform & p,
             eventedit & parent,
             sequence & seq,
             Gtk::Adjustment & vadjust)
13.14.1.2 ∼eventslots()
virtual seq64::eventslots::~eventslots ( ) [inline], [virtual]
13.14.2 Member Function Documentation
13.14.2.1 event_count()
int seq64::eventslots::event_count ( ) const [inline]
13.14.2.2 line_count()
int seq64::eventslots::line_count ( ) const [inline]
13.14.2.3 line_maximum()
int seq64::eventslots::line_maximum ( ) const [inline]
13.14.2.4 line_increment()
int seq64::eventslots::line_increment ( ) const [inline]
13.14.2.5 top_index()
int seq64::eventslots::top_index ( ) const [inline]
```

13.14.2.6 current_index()

```
int seq64::eventslots::current_index ( ) const [inline]
```

13.14.2.7 pager_index()

```
int seq64::eventslots::pager_index ( ) const [inline]
```

13.14.2.8 load_events()

```
bool seq64::eventslots::load_events ( ) [private]
```

Determines how many events can be shown in the GUI [later] and adjusts the top and bottom editable-event iterators to show the first page of events.

Returns

Returns true if the event iterators were able to be set up as valid.

13.14.2.9 set_current_event()

Note in the snprintf() calls that the first digit is part of the data byte, so that translation is easier.

Parameters

ei	The iterator that points to the event.
index	The index (re 0) of the event, starting at the top line of the frame. It is a frame index, not a container index.
full_redraw	If true (the default) does a full redraw of the frame. Otherwise, only the current event is drawn. Generally, the only time a single event (actually, two adjacent events) is convenient to draw is when using the arrow keys, where the speed of keystroke auto-repeats makes the full-frame update scrolling very flickery and disconcerting.

```
13.14.2.10 insert_event() [1/2]
```

```
bool seq64::eventslots::insert_event (
```

```
const editable_event & edev ) [private]
```

What actually happens here depends if the new event is before the frame, within the frame, or after the frame, based on the timestamp.

If before the frame: To keep the previous events visible, we do not need to increment the iterators (insertion does not affect multimap iterators), but we do need to increment their indices. The contents shown in the frame should not change.

If at the frame top: The new timestamp equals the top timestamp. We don't know exactly where the new event goes in the multimap, but we do have an new event.

If at the frame bottom: TODO

If after the frame: No action needed if the bottom event is actually at the bottom of the frame. But if the frame is not yet filled, we need to increment the bottom iterator, and its index.

Note

Actually, it is far easier to just adjust all the counts and iterators and redraw the screen, as done by the page_topper() function.

Parameters

edev	The event to insert, prebuilt.
------	--------------------------------

Returns

Returns true if the event was inserted.

```
13.14.2.11 insert_event() [2/2]
```

It calls the other insert_event() overload.

Note that we need to qualify the temporary event class object we create below, with the seq64 namespace, otherwise the compiler thinks we're trying to access some Gtkmm thing.

Parameters

evtimestamp	The time-stamp of the new event, as obtained from the event-edit timestamp field.
evname	The type name (status name) of the new event, as obtained from the event-edit event-name
	field.
evdata0	The first data byte of the new event, as obtained from the event-edit data 1 field.
evdata1	The second data byte of the new event, as obtained from the event-edit data 2 field. Used only
	for two-parameter events.

Returns

Returns true if the event was inserted.

13.14.2.12 delete_current_event()

```
bool seq64::eventslots::delete_current_event () [private]
```

To delete the current event, this function moves the current iterator to the next event, deletes the previously-current iterator, adjusts the event count and the bottom iterator, and redraws the pixmap. The exact changes depend upon whether the deleted event was at the top of the visible frame, within the visible frame, or at the bottom the visible frame. Note that only visible events can be the current event, and thus get deleted.

Basically, when an event is deleted, the frame (delimited by the event-index members) stays in place, while the frame iterators move to the previous event. If the top of the frame would move to before the first event, then the frame must shrink.

Top case: If the current iterator is the top (of the frame) iterator, then the top iterator needs to be incremented. The new top event has the same index as the now-gone top event. The index of the bottom event is decremented, since an event before it is now gone. The bottom iterator moves to the next event, which is now at the bottom of the frame. The current event is treated like the top event.

Inside case: If the current iterator is in the middle of the frame, the top iterator and index remain unchanged. The current iterator is incremented, but its index is now the same as the old bottom index. Same for the bottom iterator.

Bottom case: If the current iterator (and bottom iterator) point to the last event in the frame, then both of them need to be decremented. The frame needs to be moved up by one event, so that the current event remains at the bottom (it's just simpler to manage that way).

If there is no event after the bottom of the frame, the iterators that now point to end() must backtrack one event. If the container becomes empty, then everything is invalidated.

Returns

Returns true if the delete was possible. If the container was empty or became empty, then false is returned.

13.14.2.13 modify_current_event()

If the timestamp has changed, however, we can't just modify the event in place. Instead, we finish modifying the event, but tell the caller to delete and reinsert the new event (in its proper new location based on timestamp).

This function always copies the original event, modifiles the copy, deletes the original event, and inserts the "new" event into the editable-event container.

Parameters

evtimestamp	Provides the new event time-stamp as edited by the user.
evname	Provides the event name as edited by the user.
evdata0	Provides the first data byte as edited by the user.
evdata1	Provides the second data byte as edited by the user.

Returns

Returns true simply if the event-count is greater than 0.

13.14.2.14 save_events()

```
bool seq64::eventslots::save_events () [private]
```

Also sets the dirty flag for the sequence, via the sequence::add_event() function, but this doesn't seem to set the perform dirty flag. So now we pass the modification buck to the parent, who passes it to the perform object.

We added a copy_events() function in the sequence class to replace add_event() for the purpose of reconstructing the events container for the sequence. It is locked by a mutex, and so will not draw until all is done, preventing a nasty segfault (all segfaults are nasty).

We create a new plain event container here, and then passing it to the new locked/threadsafe sequence::copy_covents() function that clears the sequence container and copies the events from the parameter container.

Note that this code will operate event if all events were deleted.

Returns

Returns true if the operations succeeded.

13.14.2.15 select_event()

```
void seq64::eventslots::select_event (
    int event_index = SEQ64_NULL_EVENT_INDEX,
    bool full_redraw = true ) [private]
```

The event index is provided by converting the y-coordinate of the mouse pointer into a slot number, and then an event index (actually the slot-distance from the m_top_iterator. Confusing, yes no?

Note that, if the event index is negative, then we just queue up a draw operation, which should paint an empty frame – the event container is empty.

Parameters

event_index	Provides the numeric index of the event in the event frame, or SEQ64_NULL_EVENT if there is
	no event to draw.
full_redraw	Defaulting to true, this parameter can be set to false in some case to reduce the flickering of the
	frame under fast movement.

13.14.2.16 set_text()

Parameters

evcategory	The category of event to be set in the parent.
evtimestamp	The event time-stamp to be set in the parent.
evname	The event name to be set in the parent.
evdata0	The first event data byte to be set in the parent.
evdata1	The second event data byte to be set in the parent.

13.14.2.17 enqueue_draw()

```
void seq64::eventslots::enqueue_draw ( ) [private]
```

13.14.2.18 convert_y()

Parameters

y The y coordinate of the position of the mouse click in the eventslot window/pixmap.

Returns

Returns the index of the event position in the user-interface, which should range from 0 to m_line_count.

13.14.2.19 draw_event()

The slot contains the event details in (so far) one line of text in the box:

| timestamp | event kind | channel | data 0 name + value | data 1 name + value

Currently, this view shows only events that get copied to the sequence's event list. This rules out the following items from the view:

```
    MThd (song header)
    MTrk and Meta TrkEnd (track marker, a sequence has only one track)
    SeqNr (sequence number)
    SeqSpec (but there are three that might appear, see below)
    Meta TrkName
```

The events that are shown in this view are:

```
- One-data-value events:
- Program Change
- Channel Pressure
- Two-data-value events:
- Note Off
- Note On
- Aftertouch
- Control Change
- Pitch Wheel
- Other:
- SysEx events, with partial show of data bytes
- SeqSpec events (TBD):
- Key
- Scale
- Background sequence
```

The index of the event is shown in the editor portion of the eventedit dialog.

```
13.14.2.20 draw_events()

void seq64::eventslots::draw_events ( ) [private]
```

It first clears the whole bitmap to white, so that no artifacts from the previous state of the frame are left behind.

Need to figure out how to calculate the number of displayable events.

```
m_line_maximum = ???
```

13.14.2.21 change_vert()

```
void seq64::eventslots::change_vert ( ) [private]
```

Note that m_vadjust is the Gtk::Adjustment object that the eventedit parent passes to the gui_drawingarea_gtk2 constructor.

The top-event and bottom-event indices (and their corresponding editable-event iterators) delimit the part of the event container that is displayed in the eventslots user-interface. The top-event index starts at 0, and the bottom-event is larger (initially, by 42 slots).

When the scroll-bar thumb moves up or down, we need to change both event indices and both event iterators by the corresponding amount. Luckily, the std::multimap iterator is bidirectional.

Note that we may need to reduce the movement of events to a value less than a page; it can be limited backwards by the value of the top index, and forward by the value of the bottom index.

13.14.2.22 page_movement()

The adjustment is done via movement from the current position.

Do we even need a way to detect excess movement? The scrollbar, if properly set up, should never move the frame too high or too low. Verified by testing.

Parameters

new_value | Provides the new value of the scrollbar position.

13.14.2.23 page_topper()

The adjustment is done "from scratch". We've found page movement to be an insoluable problem in some editing circumstances. So now we move to the inserted event, and make it the top event.

However, always moving an inserted event to the top is a bit annoying. So now we backtrack so that the inserted event is at the bottom.

Parameters

newcurrent Provides the iterator to the event to be shown at the bottom of the frame.

13.14.2.24 decrement_top()

```
int seq64::eventslots::decrement_top ( ) [private]
```

Returns

Returns 0, or SEQ64_NULL_EVENT_INDEX if the iterator could not be decremented.

13.14.2.25 increment_top()

```
int seq64::eventslots::increment_top ( ) [private]
```

Also handles the top-event index, so that the GUI can display the proper event numbers.

Returns

Returns the top index, or SEQ64_NULL_EVENT_INDEX if the iterator could not be incremented, or would increment to the end of the container.

13.14.2.26 decrement_current()

```
int seq64::eventslots::decrement_current ( ) [private]
```

Returns

Returns the decremented index, or SEQ64_NULL_EVENT_INDEX if the iterator could not be decremented. Remember that the index ranges only from 0 to m_line_count-1, and that is enforced here.

13.14.2.27 increment_current()

```
int seq64::eventslots::increment_current ( ) [private]
```

Returns

Returns the incremented index, or SEQ64_NULL_EVENT_INDEX if the iterator could not be incremented. Remember that the index ranges only from 0 to m_line_count-1, and that is enforced here.

13.14.2.28 decrement_bottom()

```
int seq64::eventslots::decrement_bottom ( ) [private]
```

Returns

Returns 0, or SEQ64_NULL_EVENT_INDEX if the iterator could not be decremented.

13.14.2.29 increment_bottom()

```
int seq64::eventslots::increment_bottom ( ) [private]
```

There is an issue in paging down using the scrollbar where, at the bottom of the scrolling, the bottom iterator ends up bad. Not yet sure how this happens, so for now we backtrack one event if this happens.

Returns

Returns the incremented index, or SEQ64 NULL EVENT INDEX if the iterator could not be incremented.

13.14.2.30 on_realize()

```
void seq64::eventslots::on_realize ( ) [private]
```

It first calls the base-class version of on_realize(). Then it allocates any additional resources needed.

13.14.2.31 on_expose_event()

It draws all of the sequences.

13.14.2.32 on_button_press_event()

```
bool seq64::eventslots::on_button_press_event ( {\tt GdkEventButton} \ *\ ev\ ) \quad [{\tt private}]
```

13.14.2.33 on_button_release_event()

```
13.14.2.34 on_focus_in_event()
```

See the same function in the perfroll module.

13.14.2.35 on_focus_out_event()

```
bool seq64::eventslots::on_focus_out_event (
    GdkEventFocus * ev ) [private]
```

13.14.2.36 on_scroll_event()

```
bool seq64::eventslots::on_scroll_event ( {\tt GdkEventScroll * ev ) \quad [private]}
```

13.14.2.37 on_size_allocate()

It first calls the base-class version of this function.

13.14.2.38 on_move_up()

```
void seq64::eventslots::on_move_up ( ) [private]
```

We must scroll up if the event is now before the frame, and should be made the new top event of the frame. Note that this function isn't really an event-response callback. It is called by eventedit::on_key_press_event().

13.14.2.39 on_move_down()

```
void seq64::eventslots::on_move_down ( ) [private]
```

We must scroll down if the event is now after the frame. Note that this function isn't really an event-response callback. It is called byh eventedit::on_key_press_event().

13.14.2.40 on_frame_up()

```
void seq64::eventslots::on_frame_up ( ) [private]
```

```
13.14.2.41 on_frame_down()
void seq64::eventslots::on_frame_down ( ) [private]
13.14.2.42 on_frame_home()
void seq64::eventslots::on_frame_home ( ) [private]
13.14.2.43 on_frame_end()
void seq64::eventslots::on_frame_end ( ) [private]
13.14.3 Friends And Related Function Documentation
13.14.3.1 eventedit
friend class eventedit [friend]
13.14.4 Field Documentation
13.14.4.1 m_parent
eventedit& seq64::eventslots::m_parent [private]
13.14.4.2 m_seq
sequence& seq64::eventslots::m_seq [private]
13.14.4.3 m_event_container
editable_events seq64::eventslots::m_event_container [private]
```

```
13.14.4.4 m_slots_chars
```

```
int seq64::eventslots::m_slots_chars [private]
```

Pretty much hardwired to 64 at present. It helps determine the m_slots_x value (the width of the eventslots list).

```
13.14.4.5 m_char_w
```

```
int seq64::eventslots::m_char_w [private]
```

This value is obtained from a font-renderer accessor function.

```
13.14.4.6 m_setbox_w
```

```
int seq64::eventslots::m_setbox_w [private]
```

This used to be hardwired to 6 * 2 (character-width times two).

13.14.4.7 m_slots_x

```
int seq64::eventslots::m_slots_x [private]
```

13.14.4.8 m_slots_y

```
int seq64::eventslots::m_slots_y [private]
```

This value was once 22 pixels, but we need a little extra room for our new font. This extra room is compatible enough with the old font, as well.

13.14.4.9 m_event_count

```
int seq64::eventslots::m_event_count [private]
```

13.14.4.10 m_line_count

```
int seq64::eventslots::m_line_count [private]
```

```
13.14.4.11 m_line_maximum
```

```
int seq64::eventslots::m_line_maximum [private]
```

13.14.4.12 m_line_overlap

```
int seq64::eventslots::m_line_overlap [private]
```

13.14.4.13 m_top_index

```
int seq64::eventslots::m_top_index [private]
```

It is used in numbering the events that are shown in the event-slot frame. Do not confuse it with m_current_index, which is relative to the frame, not the container-beginning.

13.14.4.14 m_current_index

```
int seq64::eventslots::m_current_index [private]
```

This event will also be pointed to by the m_current_event iterator. Do not confuse it with m_top_index, which is relative to the container-beginning, not the frame.

13.14.4.15 m_top_iterator

```
editable_events::iterator seq64::eventslots::m_top_iterator [private]
```

13.14.4.16 m_bottom_iterator

```
editable_events::iterator seq64::eventslots::m_bottom_iterator [private]
```

13.14.4.17 m_current_iterator

```
editable_events::iterator seq64::eventslots::m_current_iterator [private]
```

```
13.14.4.18 m_pager_index
```

```
int seq64::eventslots::m_pager_index [private]
```

13.15 seq64::font Class Reference

This class provides a wrapper for rendering fonts that are encoded as a 16 x 16 pixmap file in XPM format.

Public Types

```
    enum Color {
        BLACK,
        WHITE,
        BLACK_ON_YELLOW,
        YELLOW_ON_BLACK,
        BLACK_ON_CYAN,
        CYAN_ON_BLACK }
```

A simple enumeration to describe the basic colors used in writing text.

Public Member Functions

• font ()

Rote default constructor, except that it does add 1 to the cf_text_h or co_text_h values to use in m_padded_h.

void init (Glib::RefPtr< Gdk::Window > windo)

Initialization function for a window on which fonts will be drawn.

void render_string_on_drawable (Glib::RefPtr< Gdk::GC > m_gc, int x, int y, Glib::RefPtr< Gdk::Drawable > drawable, const char *str, font::Color col, bool invert=false) const

Draws a text string.

• int char_width () const

'Getter' function for member m font w

• int char_height () const

'Getter' function for member m_font_h

• int padded_height () const

'Getter' function for member m_padded_h

Private Attributes

· bool m use new font

If true, use the new font, which is a little bit more modern looking, and is also thicker, and thus a little easier to see.

• int m_cell_w

Specifies the cell width of the whole character cell.

• int m cell h

Specfies the cell height of the whole character cell.

int m_font_w

Specifies the exact width of a character cell, in pixels.

· int m font h

Specifies the exact height of a character cell, in pixels.

int m_offset

Provides an ad hoc small horizontal or vertical offset for printing strings.

int m_padded_h

Provides a common constant used by much of the drawing code, but only marginally related to the padded character height.

const Glib::RefPtr< Gdk::Pixmap > * m_pixmap

Points to the current pixmap (m_black_pixmap or m_white_pixmap) to use to render a string.

Glib::RefPtr< Gdk::Pixmap > m_black_pixmap

The pixmap in the file src/pixmaps/font_b.xpm is loaded into this object.

Glib::RefPtr< Gdk::Pixmap > m_white_pixmap

The pixmap in the file src/pixmaps/font_w.xpm is loaded into this object.

• Glib::RefPtr< Gdk::Pixmap > m_b_on_y_pixmap

The pixmap in the file src/pixmaps/font_y.xpm is loaded into this object.

Glib::RefPtr< Gdk::Pixmap > m_y_on_b_pixmap

The pixmap in the file src/pixmaps/font_yb.xpm is loaded into this object.

Glib::RefPtr< Gdk::Pixmap > m_b_on_c_pixmap

The pixmap in the file src/pixmaps/cyan_wenfont_y.xpm is loaded into this object.

Glib::RefPtr< Gdk::Pixmap > m_c_on_b_pixmap

The pixmap in the file src/pixmaps/cyan_wenfont_yb.xpm is loaded into this object.

Glib::RefPtr< Gdk::Bitmap > m clip mask

This object is instantiated as a default object.

13.15.1 Member Enumeration Documentation

13.15.1.1 Color

```
enum seq64::font::Color
```

Basically, these two values cause the selection of one or another pixmap (font_b_xpm and font_w_xpm). We've added two more pixmaps to draw black text on a yellow background (font_y.xpm) and yellow text on a black background (font_yb.xpm). Oh, and couple more for cyan and black text-blitting.

Enumerator

BLACK	The first supported color. A black font on a white background.
WHITE	The second supported color. A white font on a black background.
BLACK_ON_YELLOW	A new color, for drawing black text on a yellow background.
YELLOW_ON_BLACK	A new color, for drawing yellow text on a black background.
BLACK_ON_CYAN	A new color, for drawing black text on a cyan background.
CYAN_ON_BLACK	A new color, for drawing cyan text on a black background.

13.15.2 Constructor & Destructor Documentation

```
13.15.2.1 font() seq64::font::font ( )
```

13.15.3 Member Function Documentation

This function loads four pixmaps that contain the characters to be used to draw text strings.

One pixmap has white characters on a black background, one has black characters on a white background, one has yellow characters on a black background, and one has black characters on a yellow background.

Parameters

wp | Provides the windows pointer for the window that holds the color map.

13.15.3.2 render_string_on_drawable()

This function grabs the proper font bitmap, extracts the current character pixmap from it, and slaps it down where it needs to be to render the character in the string.

Parameters

gc	Provides the graphics context for drawing the text using GTK+.
Х	The horizontal location of the text.
У	The vertical location of the text.
a_draw	The drawable object on which to draw the text.
str	The string to draw. Should use a constant string reference instead.
col	The font color to use to draw the string. The supported values are font::BLACK, font::WHITE, font::BLACK_ON_YELLOW, font::YELLOW_ON_BLACK. The actual correct colors are provided by selecting one of four font pixmaps, as described in the init() function.
invert	If true, apply color inversion, if specified.

```
13.15.3.3 char_width()
int seq64::font::char_width ( ) const [inline]
13.15.3.4 char_height()
int seq64::font::char_height ( ) const [inline]
13.15.3.5 padded_height()
int seq64::font::padded_height ( ) const [inline]
13.15.4 Field Documentation
13.15.4.1 m_use_new_font
bool seq64::font::m_use_new_font [private]
13.15.4.2 m_cell_w
int seq64::font::m_cell_w [private]
13.15.4.3 m_cell_h
int seq64::font::m_cell_h [private]
13.15.4.4 m_font_w
int seq64::font::m_font_w [private]
```

Currently defaults to cf_text_w = 6. Note that a lot of stuff depends on this being 6 at present, even with our new,

Generated by Doxygen

slightly wider, font.

```
13.15.4.5 m_font_h
```

```
int seq64::font::m_font_h [private]
```

Currently defaults to cf_text_h = 10. Note that a lot of stuff depends on this being 10 at present, even with our new, slightly wider, font. But some of the drawing code doesn't use the character height, but the padded character height.

13.15.4.6 m_offset

```
int seq64::font::m_offset [private]
```

13.15.4.7 m_padded_h

```
int seq64::font::m_padded_h [private]
```

13.15.4.8 m_pixmap

```
const Glib::RefPtr<Gdk::Pixmap>* seq64::font::m_pixmap [mutable], [private]
```

This member used to be an object, but it's probably a bit faster to just use a pointer (or a reference).

13.15.4.9 m_black_pixmap

```
Glib::RefPtr<Gdk::Pixmap> seq64::font::m_black_pixmap [private]
```

It contains a black font on a white background. The new-style font, if selected, is in the resources/pixmaps/wenfont ← _b.xmp pixmap.

13.15.4.10 m_white_pixmap

```
Glib::RefPtr<Gdk::Pixmap> seq64::font::m_white_pixmap [private]
```

It contains a black font on a white background. The new-style font, if selected, is in the resources/pixmaps/wenfont — _w.xmp pixmap.

13.15.4.11 m_b_on_y_pixmap

```
Glib::RefPtr<Gdk::Pixmap> seq64::font::m_b_on_y_pixmap [private]
```

It contains a black font on a yellow background. The new-style font, if selected, is in the resources/pixmaps/wenfont ← _y.xmp pixmap.

13.15.4.12 m_y_on_b_pixmap

```
Glib::RefPtr<Gdk::Pixmap> seq64::font::m_y_on_b_pixmap [private]
```

It contains a yellow font on a black background. The new-style font, if selected, is resources/pixmaps/wenfont ← _yb.xmp pixmap.

13.15.4.13 m_b_on_c_pixmap

```
Glib::RefPtr<Gdk::Pixmap> seq64::font::m_b_on_c_pixmap [private]
```

It contains a black font on a cyan background. It is available only for the new font-style.

13.15.4.14 m_c_on_b_pixmap

```
Glib::RefPtr<Gdk::Pixmap> seq64::font::m_c_on_b_pixmap [private]
```

It contains a cyan font on a black background. It is available only for the new font-style.

13.15.4.15 m_clip_mask

```
Glib::RefPtr<Gdk::Bitmap> seq64::font::m_clip_mask [private]
```

All we know is it seems to be a requirement for creating a pixmap object from an XMP file.

13.16 seq64::FruityPerfInput Class Reference

Implements the performance input of that certain fruity sequencer that people seem to like.

Inheritance diagram for seq64::FruityPerfInput:

seq64::AbstractPerfInput - m_adding - m_adding_pressed + AbstractPerfInput() # ~AbstractPerfInput() # on button press event() # on_button_release_event() # on_motion_notify_event() # activate_adding() # handle_motion_key() # is_adding() # set_adding() # is_adding_pressed() # set adding pressed() seq64::FruityPerfInput - m_current_x - m_current_y + FruityPerfInput() + on_button_press_event() + on_button_release_event() + on motion notify event() - update mouse pointer() - on_left_button_pressed() - on right button pressed() - activate_adding() - handle_motion_key()

Public Member Functions

• FruityPerfInput ()

Default constructor.

• bool on button press event (GdkEventButton *ev, perfroll &roll)

Handles a button-press event in the Fruity manner.

• bool on_button_release_event (GdkEventButton *ev, perfroll &roll)

Handles a button-release event.

• bool on_motion_notify_event (GdkEventMotion *ev, perfroll &roll)

Handles a Fruity motion-notify event.

Private Member Functions

void update_mouse_pointer (perfroll &roll)

Updates the mouse pointer, implementing a context-sensitive mouse.

• bool on_left_button_pressed (GdkEventButton *ev, perfroll &roll)

Handles the left button of the mouse.

- bool on_right_button_pressed (GdkEventButton *ev, perfroll &roll)

 Handles the right button of the mouse.
- virtual void activate_adding (bool, perfroll &)
- virtual bool handle_motion_key (bool, perfroll &)

Private Attributes

• long m_current_x

The current x value of the mouse.

long m_current_y

The current y value of the mouse.

Friends

· class perfroll

Additional Inherited Members

13.16.1 Constructor & Destructor Documentation

```
13.16.1.1 FruityPerfInput()
```

```
seq64::FruityPerfInput::FruityPerfInput ( ) [inline]
```

13.16.2 Member Function Documentation

13.16.2.1 on_button_press_event()

Parameters

ev	The button-press event to process.
roll	The song editor piano roll that is the "parent" of this class.

Returns

Returns true if a modification occurred.

Implements seq64::AbstractPerfInput.

13.16.2.2 on_button_release_event()

Why is m_adding_pressed modified conditionally when the same modification is then made unconditionally?

Parameters

ev	The button-release event to process.
roll	The song editor piano roll that is the "parent" of this class.

Returns

Returns true if a modification occurred.

Implements seq64::AbstractPerfInput.

13.16.2.3 on_motion_notify_event()

Parameters

ev	The motion-notify event to process.
roll	The song editor piano roll that is the "parent" of this class.

Returns

Returns true if a modification occurred, and sets the perform modified flag based on that result.

Implements seq64::AbstractPerfInput.

13.16.2.4 update_mouse_pointer()

Note that perform::convert_xy() returns its values via side-effects on the last two parameters.

Parameters

```
roll The song editor piano roll that is the "parent" of this class.
```

13.16.2.5 on_left_button_pressed()

It can handle splitting triggers (?), adding notes, and the following clicks to resize the event, or move it, depending on where clicked:

```
    clicked left side: begin a grow/shrink for the left side
    clicked right side: grow/shrink the right side
    clicked in the middle - move it
```

I don't get it, though... all three buttons are handled in the generic button-press callback. Oh, this is just a helper function.

Parameters

ev	The left-button-press event to process.
roll	The song editor piano roll that is the "parent" of this class.

Returns

Now returns true if a modification occurred.

13.16.2.6 on_right_button_pressed()

I don't get it, though... all three buttons are handled in the generic button-press callback. Oh, this is a helper function.

Parameters

ev	The right-button-press event to process.
roll	The song editor piano roll that is the "parent" of this class.

Returns

Returns true if a modification occurred.

13.16.2.7 activate_adding()

Implements seq64::AbstractPerfInput.

13.16.2.8 handle_motion_key()

Implements seq64::AbstractPerfInput.

13.16.3 Friends And Related Function Documentation

13.16.3.1 perfroll

```
friend class perfroll [friend]
```

13.16.4 Field Documentation

13.16.4.1 m_current_x

```
long seq64::FruityPerfInput::m_current_x [private]
```

```
13.16.4.2 m_current_y
```

```
long seq64::FruityPerfInput::m_current_y [private]
```

13.17 seq64::FruitySeqEventInput Struct Reference

This structure implements the interaction methods for the "fruity" mode of operation.

Public Member Functions

FruitySeqEventInput ()

Default constructor.

void update_mouse_pointer (seqevent &ths)

Provides support for a context-sensitive mouse.

bool on_button_press_event (GdkEventButton *ev, seqevent &ths)

Implements the on-button-press event callback.

bool on_button_release_event (GdkEventButton *ev, seqevent &ths)

Implements the on-button-release callback.

bool on_motion_notify_event (GdkEventMotion *ev, seqevent &ths)

Implements the on-motion-notify callback.

Data Fields

• bool m_justselected_one

Indicates that the left mouse button was click to start a selection.

bool m_is_drag_pasting_start

Set to true when the mouse button is pressed and we're starting to drag some notes to move them and paste them to a different location.

bool m_is_drag_pasting

Set to true when the left mouse button is pressed for dragging and pasting, set to false when the mouse button is released to drop the pasted items.

13.17.1 Constructor & Destructor Documentation

13.17.1.1 FruitySeqEventInput()

```
seq64::FruitySeqEventInput::FruitySeqEventInput ( ) [inline]
```

13.17.2 Member Function Documentation

13.17.2.1 update_mouse_pointer()

Parameters

seqev

Provides the sequeent pane (actually a strip on the sequedit window) to update to show the proper mouse cursor (left pointer, center pointer, and pencil).

13.17.2.2 on_button_press_event()

Handles dragging and other actions.

The first thing is to set the values for dragging, then reset the box that holds the dirty redraw spot. If pasting, undo the clipboard, and paste the selected events.

Otherwise, process the mouse actions. The current steps shown below are my initial guesses, to be verified at some point.

- 1. Left button:
 - (a) Click:
 - i. A click and release without a drag, or without a Ctrl-Shift, deselects the events.
 - ii. A direct click on an event selects only that event.
 - (b) Click-drag:
 - i. If events already selected, adds note and length to the selected notes.
 - ii. Otherwise, select the notes and events.
 - iii. If no events selected in the end, undo the selection.
- · Ctrl-left button:
 - TODO.

The opening part of this function matches that of Seq24SeqEventInput :: on_button_press_event().

Parameters

ev	The button event for the press of a mouse button.
seqev	Provides the sequeent strip to be affected by this button event.

Returns

Returns true if a modification was made. It used to return true all the time.

13.17.2.3 on_button_release_event()

Parameters

ev	The button event for the press of a mouse button.
seqev	Provides the sequeent strip to be affected by this button event.

Returns

Returns true if a modification was made. It used to return true all the time.

13.17.2.4 on_motion_notify_event()

Parameters

ev	The button event for the press of a mouse button.
seqev	Provides the sequeent strip to be affected by this button event.

Returns

Returns true if a modification occurred, and sets the perform modified flag based on that result.

13.17.3 Field Documentation

13.17.3.1 m_justselected_one

bool seq64::FruitySeqEventInput::m_justselected_one

13.17.3.2 m_is_drag_pasting_start

bool seq64::FruitySeqEventInput::m_is_drag_pasting_start

13.17.3.3 m_is_drag_pasting

bool seq64::FruitySeqEventInput::m_is_drag_pasting

13.18 seq64::FruitySeqRollInput Class Reference

Implements the fruity mouse interaction paradigm for the seqroll.

Public Member Functions

• FruitySeqRollInput ()

Default constructor.

void update_mouse_pointer (seqroll &ths)

Updates the mouse pointer, implementing a context-sensitive mouse.

• bool on_button_press_event (GdkEventButton *ev, seqroll &ths)

Implements the fruity on-button-press callback.

bool on_button_release_event (GdkEventButton *ev, seqroll &ths)

Implements the fruity handling for the on-button-release event.

bool on motion notify event (GdkEventMotion *ev, segroll &ths)

Implements the fruity handling for the on-motion-notify event.

Private Attributes

· bool m_erase_painting

Set to tru if we hold the right mouse button down (in "fruity" mode) and start to drag the mouse around, erasing notes.

• int m_drag_paste_start_pos [2]

Holds the original position of the mouse when ctrl-left-click-drag is done, and is used to make sure that the action doesn't occur until a movement of at least 6 pixels has occurred, to avoid unintended actions caused by minimal jitter in the user's hands.

13.18.1 Constructor & Destructor Documentation

13.18.1.1 FruitySegRollInput()

```
seq64::FruitySeqRollInput::FruitySeqRollInput ( ) [inline]
```

13.18.2 Member Function Documentation

13.18.2.1 update_mouse_pointer()

Parameters

sroll	Provides the "parent" of this interaction class.
-------	--

13.18.2.2 on button press event()

This function now uses the needs update flag to determine if the perform object should modify().

Parameters

ev	The button event.
sroll	The parent of this "fruity" interaction class.

Returns

Returns the value of needs_update. It used to return only true.

13.18.2.3 on_button_release_event()

Parameters

ev	The button event.
sroll	The parent of this "fruity" interaction class.

Returns

Returns the value of needs_update. It used to return only true.

If in moving mode, adjust for snap and convert deltas into screen coordinates. Since delta_note was from delta_y, it will be flipped (delta_y[0] = note[127], etc.), so we have to adjust.

13.18.2.4 on_motion_notify_event()

Parameters

ev	,	The motion event.
src	oll	The parent of this "fruity" interaction class. (Why not just inherit and save all these indirect accesses to
		the seqroll? Well, that would make it more difficult to change the mode of interation, in the Options
		menu, on the fly.)

Returns

Returns the value of needs_update.

In "fruity" interatction mode, ctrl-left-click-drag on selected note(s) starts a copy/unselect/paste. Doesn't begin the paste until the mouse moves a few pixels, to filter out the unsteady hand.

13.18.3 Field Documentation

13.18.3.1 m_erase_painting

bool seq64::FruitySeqRollInput::m_erase_painting [private]

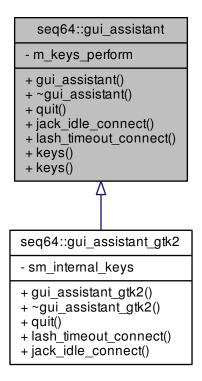
13.18.3.2 m_drag_paste_start_pos

int seq64::FruitySeqRollInput::m_drag_paste_start_pos[2] [private]

13.19 seq64::gui_assistant Class Reference

This class provides an interface for some of the GUI support needed in Sequencer64.

Inheritance diagram for seq64::gui_assistant:



Public Member Functions

• gui_assistant (keys_perform &kp)

This constructor wires in some externally (for now) created objects.

virtual ~gui_assistant ()

Stock base-class implementation of a virtual destructor.

- virtual void quit ()=0
- virtual void jack_idle_connect (jack_assistant &jack)=0
- virtual void lash_timeout_connect (lash *lashobject)=0
- const keys_perform & keys () const

 ${\it 'Getter' function for member m_keys_perform\ The\ const\ getter.}$

keys_perform & keys ()

 ${\it 'Getter' function for member m_keys_perform\ The\ un-const\ getter.}$

Private Attributes

• keys_perform & m_keys_perform

Provides a reference to the app-specific GUI-specific keys_perform-derived object that an application is going to use for handling sequence-control keys.

13.19.1 Detailed Description

It also contain a number of helper objects that all kind of go together; only this assistant object will need to be passed around (by non-GUI code).

13.19.2 Constructor & Destructor Documentation

```
13.19.2.1 gui_assistant()
```

Parameters

 $\varphi \mid$ Provides a set of key codes to be used by the perform object to control patterns and their performance.

```
13.19.2.2 ∼gui_assistant()
```

```
virtual seq64::gui_assistant::~gui_assistant ( ) [inline], [virtual]
```

13.19.3 Member Function Documentation

```
13.19.3.1 quit()
```

```
virtual void seq64::gui_assistant::quit ( ) [pure virtual]
```

Implemented in seq64::gui_assistant_gtk2.

13.19.3.2 jack_idle_connect()

Implemented in seq64::gui_assistant_gtk2.

Implemented in seq64::gui assistant gtk2.

```
13.19.3.4 keys() [1/2]

const keys_perform& seq64::gui_assistant::keys ( ) const [inline]

13.19.3.5 keys() [2/2]

keys_perform& seq64::gui_assistant::keys ( ) [inline]
```

13.19.4 Field Documentation

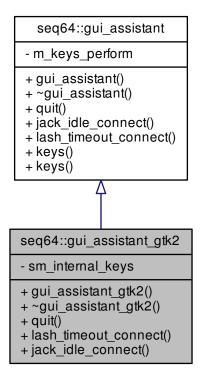
```
13.19.4.1 m_keys_perform
```

keys_perform& seq64::gui_assistant::m_keys_perform [private]

13.20 seq64::gui_assistant_gtk2 Class Reference

This class provides an interface for some of the Gtk/Gdk/Glib support needed in Sequencer64.

Inheritance diagram for seq64::gui_assistant_gtk2:



Public Member Functions

• gui_assistant_gtk2 ()

This class provides an interface for some of the Gtk/Gdk/Glib support needed in Sequencer64.

virtual ~gui_assistant_gtk2 ()

Virtual classes require a virtual destructor.

virtual void quit ()

Calls the Glib Main object's quit() function.

virtual void lash_timeout_connect (lash *lashobject)

Connects the LASH timeout-event callback to the Glib timeout object.

virtual void jack_idle_connect (jack_assistant &jack)

Connects the JACK session-event callback to the Glib idle object.

Static Private Attributes

static keys_perform_gtk2 sm_internal_keys
 Provides a pre-made keys_perform object.

13.20.1 Constructor & Destructor Documentation

```
13.20.1.1 gui_assistant_gtk2()
seq64::gui_assistant_gtk2::gui_assistant_gtk2 ( )
13.20.1.2 ~gui_assistant_gtk2()
virtual seq64::gui_assistant_gtk2::~gui_assistant_gtk2 ( ) [inline], [virtual]
13.20.2 Member Function Documentation
13.20.2.1 quit()
void seq64::gui_assistant_gtk2::quit ( ) [virtual]
Implements seq64::gui_assistant.
13.20.2.2 lash_timeout_connect()
void seq64::gui_assistant_gtk2::lash_timeout_connect (
             lash * lashobject ) [virtual]
The time-out value is set to 250 ms.
Implements seq64::gui_assistant.
13.20.2.3 jack_idle_connect()
```

If JACK session support is not enabled, we might emit a message. This mainly prevents a compiler warning about an unused parameter.

Implements seq64::gui assistant.

void seq64::gui_assistant_gtk2::jack_idle_connect (

jack_assistant & jack) [virtual]

13.20.3 Field Documentation

13.20.3.1 sm_internal_keys

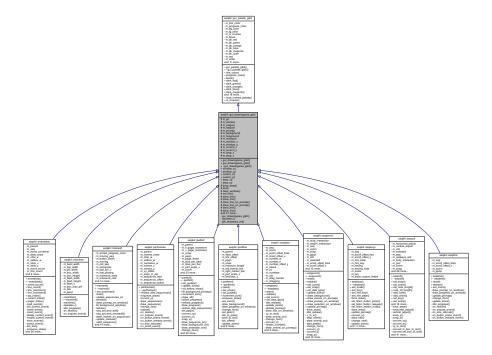
```
keys_perform_gtk2 seq64::gui_assistant_gtk2::sm_internal_keys [static], [private]
```

This object is set into the reference provided in the gui_assistant base class.

13.21 seq64::gui_drawingarea_gtk2 Class Reference

Implements the basic drawing areas of the application.

Inheritance diagram for seq64::gui_drawingarea_gtk2:



Data Structures

struct rect

A small helper structure representing a rectangle.

Public Member Functions

- gui_drawingarea_gtk2 (perform &p, int window_x=0, int window_y=0)
- gui_drawingarea_gtk2 (perform &a_perf, Gtk::Adjustment &a_hadjust, Gtk::Adjustment &a_vadjust, int window_x=0, int window_y=0)

Principal constructor.

virtual ~gui drawingarea gtk2 ()

Perform-only constructor.

Provides a destructor to delete allocated objects.

• int window_x () const

'Getter' function for member m_window_x

• int window_y () const

'Getter' function for member m_window_y

• int current x () const

'Getter' function for member m_current_x

• int current y () const

'Getter' function for member m current y

int drop_x () const

'Getter' function for member m drop x

int drop_y () const

'Getter' function for member m_drop_y

Protected Member Functions

virtual void force_draw ()

Provides a common function for redrawing.

perform & perf ()

'Getter' function for member m_mainperf

void clear window ()

Clears the main window.

• void set_line (Gdk::LineStyle Is, int width=1)

A small wrapper function for readability in line-drawing.

void draw line (int x1, int y1, int x2, int y2)

A small wrapper function to draw a line on the window.

void draw_line (const Color &c, int x1, int y1, int x2, int y2)

A small wrapper function to draw a line on the window after setting the given foreground color.

void draw_line_on_pixmap (int x1, int y1, int x2, int y2)

A small wrapper function to draw a line on the pixmap.

void draw_line_on_pixmap (const Color &c, int x1, int y1, int x2, int y2)

A small wrapper function to draw a line on the pixmap after setting the given foreground color.

• void draw line (Glib::RefPtr< Gdk::Pixmap > &pixmap, int x1, int y1, int x2, int y2)

A small wrapper function to draw a line on any pixmap (not a drawable, though, due to a compiler error after setting the given foreground color.

void draw_line (Glib::RefPtr< Gdk::Pixmap > &pixmap, const Color &c, int x1, int y1, int x2, int y2)

A small wrapper function to draw a line on the pixmap after setting the given foreground color.

void draw line (Glib::RefPtr< Gdk::Drawable > &drawable, int x1, int y1, int x2, int y2)

A small wrapper function to draw a line on any pixmap (not a drawable, though, due to a compiler error after setting the given foreground color.

void draw_line (Glib::RefPtr< Gdk::Drawable > &drawable, const Color &c, int x1, int y1, int x2, int y2)

A small wrapper function to draw a line on the drawable after setting the given foreground color.

• void render_string (int x, int y, const std::string &s, font::Color color, bool invert=false)

A small wrapper function for readability in string-drawing to the window.

void render_string_on_pixmap (int x, int y, const std::string &s, font::Color color, bool invert=false)

A small wrapper function for readability in string-drawing to the pixmap.

void draw_rectangle (int x, int y, int lx, int ly, bool fill=true)

A small wrapper function for readability in box-drawing on the window.

• void draw_rectangle (const Color &c, int x, int y, int lx, int ly, bool fill=true)

A small wrapper function for readability in box-drawing.

• void draw_rectangle (Glib::RefPtr< Gdk::Drawable > &drawable, int x, int y, int lx, int ly, bool fill=true)

A small wrapper function for readability in box-drawing on a "drawable" context, where the foreground color has already been specified.

• void draw_rectangle (Glib::RefPtr< Gdk::Drawable > &drawable, const Color &c, int x, int y, int lx, int ly, bool fill=true)

A small wrapper function for readability in box-drawing on any drawable context.

void draw_rectangle (Glib::RefPtr< Gdk::Pixmap > &pixmap, int x, int y, int lx, int ly, bool fill=true)

A small wrapper function for readability in box-drawing on a "pixmap" context, where the foreground color has already been specified.

void draw_rectangle (Glib::RefPtr< Gdk::Pixmap > &pixmap, const Color &c, int x, int y, int lx, int ly, bool fill=true)

A small wrapper function for readability in box-drawing on any pixmap context.

void draw_rectangle_on_pixmap (int x, int y, int lx, int ly, bool fill=true)

A small wrapper function for readability in box-drawing on the pixmap.

void draw_rectangle_on_pixmap (const Color &c, int x, int y, int lx, int ly, bool fill=true)

A small wrapper function for readability in box-drawing on the pixmap.

void draw normal rectangle on pixmap (int x, int y, int lx, int ly, bool fill=true)

A small wrapper function for readability in box-drawing on the pixmap.

• void draw_drawable (int xsrc, int ysrc, int xdest, int ydest, int width, int height)

Provides the most common use case for redrawing.

void scroll_hadjust (Gtk::Adjustment &hadjust, double step)

This function provides optimization for the on_scroll_event() functions, and should provide support for having the seqedit/seqroll/seqtime/seqdata panes follow the scrollbar, in a future upgrade (now partly in place).

void scroll vadjust (Gtk::Adjustment &vadjust, double step)

This function is the vertical version of the scroll_hadjust() function, intended for adding keystroke vertical scrolling using the Page-Up and Page-Down keys, as a new feature of Sequencer64.

- void scroll hset (Gtk::Adjustment &hadjust, double value)
- void scroll_vset (Gtk::Adjustment &vadjust, double value)
- void set_current_drop_x (int x)

Sets the current x value and the drop x value.

void set_current_drop_y (int y)

Sets the current y value and the drop y value.

• void on_realize ()

For this GTK callback, on realization of window, initialize the shiz.

Protected Attributes

Glib::RefPtr< Gdk::GC > m gc

The graphics context, which is required for ever drawing and rendering operation.

• Glib::RefPtr< Gdk::Window > m window

Provides the default "window".

Gtk::Adjustment & m_vadjust

Provides an object for vertical "adjustments".

· Gtk::Adjustment & m hadjust

Provides an object for horizontal "adjustments".

Glib::RefPtr< Gdk::Pixmap > m_pixmap

Provides the default "pixmap".

Glib::RefPtr< Gdk::Pixmap > m background

Another pixmap, used for backgrounds.

Glib::RefPtr< Gdk::Pixmap > m_foreground

Another pixmap, used for foregrounds.

· perform & m mainperf

A frequent hook into the main perform object.

• int m window x

Window sizes.

• int m_window_y

Window height value.

· int m current x

The x and y value of the current location of the mouse (during dragging?)

· int m_current_y

Current mouse y value.

• int m_drop_x

These values are used when roping and highlighting a bunch of events.

int m_drop_y

Current mouse y-drop value.

Private Member Functions

- gui_drawingarea_gtk2 (const gui_drawingarea_gtk2 &)
- gui_drawingarea_gtk2 & operator= (const gui_drawingarea_gtk2 &)
- void gtk_drawarea_init ()

Does basic initialization for each of the constructors.

Additional Inherited Members

13.21.1 Detailed Description

Note that this class really "isn't" a gui_pallete_gtk2; it should simply "have" one. But that base class must be derived from Gtk::DrawingArea. We don't want to waste some space by using a "has-a" relationship, and also put up with having to access the palette indirectly. So, in this case, we tolerate the less strict implementation.

13.21.2 Constructor & Destructor Documentation

```
13.21.2.3 gui_drawingarea_gtk2() [3/3]
seq64::gui_drawingarea_gtk2::gui_drawingarea_gtk2 (
             perform & a_perf,
             Gtk::Adjustment & a_hadjust,
             Gtk::Adjustment & a_vadjust,
             int window_x = 0,
             int window_y = 0)
13.21.2.4 \simgui_drawingarea_gtk2()
seq64::gui\_drawingarea\_gtk2::\sim gui\_drawingarea\_gtk2 \ (\ ) \quad [virtual]
13.21.3 Member Function Documentation
13.21.3.1 operator=()
gui_drawingarea_gtk2& seq64::gui_drawingarea_gtk2::operator= (
             const gui_drawingarea_gtk2 & ) [private]
13.21.3.2 window_x()
int seq64::gui_drawingarea_gtk2::window_x ( ) const [inline]
13.21.3.3 window_y()
int seq64::gui_drawingarea_gtk2::window_y ( ) const [inline]
13.21.3.4 current_x()
int seq64::gui_drawingarea_gtk2::current_x ( ) const [inline]
```

```
13.21.3.5 current_y()
int seq64::gui_drawingarea_gtk2::current_y ( ) const [inline]
13.21.3.6 drop_x()
int seq64::gui_drawingarea_gtk2::drop_x ( ) const [inline]
13.21.3.7 drop_y()
int seq64::gui_drawingarea_gtk2::drop_y ( ) const [inline]
13.21.3.8 force_draw()
virtual void seq64::gui_drawingarea_gtk2::force_draw ( ) [inline], [protected], [virtual]
This function forces a redraw. Some classes extend this function.
Reimplemented in seq64::seqroll, seq64::seqevent, and seq64::seqkeys.
13.21.3.9 perf()
perform& seq64::gui_drawingarea_gtk2::perf ( ) [inline], [protected]
13.21.3.10 clear_window()
void seq64::gui_drawingarea_gtk2::clear_window ( ) [inline], [protected]
One less need to access m_window directly.
13.21.3.11 set_line()
void seq64::gui_drawingarea_gtk2::set_line (
             Gdk::LineStyle 1s,
             int width = 1 ) [inline], [protected]
```

Sets the attributes of a line to be drawn.

ls	Provides the Gtk-specific line style.	
width	Provides the width of the line to be drawn. It defaults to the most common value, 1.	

13.21.3.12 draw_line() [1/6]

Parameters

x1	The x coordinate of the starting point.
y1	The y coordinate of the starting point.
x2	The x coordinate of the ending point.
y2	The y coordinate of the ending point.

13.21.3.13 draw_line() [2/6]

```
void seq64::gui_drawingarea_gtk2::draw_line (
    const Color & c,
    int x1,
    int y1,
    int x2,
    int y2 ) [protected]
```

Parameters

С	The foreground color in which to draw the line.
x1	The x coordinate of the starting point.
y1	The y coordinate of the starting point.
x2	The x coordinate of the ending point.
<i>y</i> 2	The y coordinate of the ending point.

13.21.3.14 draw_line_on_pixmap() [1/2]

```
void seq64::gui_drawingarea_gtk2::draw_line_on_pixmap (
    int x1,
```

```
int y1, int x2, int y2) [inline], [protected]
```

x1	The x coordinate of the starting point.
y1	The y coordinate of the starting point.
x2	The x coordinate of the ending point.
y2	The y coordinate of the ending point.

13.21.3.15 draw_line_on_pixmap() [2/2]

Parameters

С	The foreground color in which to draw the line.
x1	The x coordinate of the starting point.
y1	The y coordinate of the starting point.
x2	The x coordinate of the ending point.
y2	The y coordinate of the ending point.

13.21.3.16 draw_line() [3/6]

Parameters

pixmap	Provides the Gdk::Pixmap pointer needed to draw the line.
x1	The x coordinate of the starting point.
y1	The y coordinate of the starting point.
x2	The x coordinate of the ending point.
y2	The y coordinate of the ending point.

13.21.3.17 draw_line() [4/6]

Parameters

pixmap	Provides the Gdk::Drawable pointer needed to draw the line.
С	The foreground color in which to draw the line.
x1	The x coordinate of the starting point.
y1	The y coordinate of the starting point.
x2	The x coordinate of the ending point.
y2	The y coordinate of the ending point.

13.21.3.18 draw_line() [5/6]

Parameters

drawable	Provides the Gdk::Drawable pointer needed to draw the line.
x1	The x coordinate of the starting point.
y1	The y coordinate of the starting point.
x2	The x coordinate of the ending point.
y2	The y coordinate of the ending point.

13.21.3.19 draw_line() [6/6]

```
int y1, int x2, int y2) [protected]
```

drawable	Provides the Gdk::Drawable pointer needed to draw the line.
С	The foreground color in which to draw the line.
x1	The x coordinate of the starting point.
y1	The y coordinate of the starting point.
x2	The x coordinate of the ending point.
y2	The y coordinate of the ending point.

13.21.3.20 render_string()

```
void seq64::gui_drawingarea_gtk2::render_string (
    int x,
    int y,
    const std::string & s,
    font::Color color,
    bool invert = false ) [inline], [protected]
```

Parameters

х	The x-coordinate of the origin.
У	The y-coordinate of the origin.
s	The string to be drawn.
color	The color with which to draw the string.
invert	If true, apply color inversion, if active. Defaults to false.

13.21.3.21 render_string_on_pixmap()

Parameters

X	The x-coordinate of the origin.
У	The y-coordinate of the origin.
s	The string to be drawn.
color	The color with which to draw the string.
invert	If true, apply color inversion, if active. Defaults to false.

13.21.3.22 draw_rectangle() [1/6]

```
void seq64::gui_drawingarea_gtk2::draw_rectangle (
    int x,
    int y,
    int lx,
    int ly,
    bool fill = true ) [inline], [protected]
```

Parameters

X	The x-coordinate of the origin.
У	The y-coordinate of the origin.
lx	The width of the box.
ly	The height of the box.
fill	If true, fill the rectangle with the current foreground color, as set by m_gc->set_foreground(color). Defaults
	to true.

13.21.3.23 draw_rectangle() [2/6]

It adds setting the foreground color to the draw_rectangle() function.

Parameters

С	Provides the foreground color to set.
X	The x-coordinate of the origin.
У	The y-coordinate of the origin.
lx	The width of the box.
ly	The height of the box.
fill	If true, fill the rectangle with the current foreground color, as set by m_gc->set_foreground(color). Defaults
	to true.

13.21.3.24 draw_rectangle() [3/6]

```
int x,
int y,
int lx,
int ly,
bool fill = true ) [inline], [protected]
```

drawable	The object on which to draw the rectangle.	
X	The x-coordinate of the origin.	
У	he y-coordinate of the origin.	
lx	The width of the box.	
ly	The height of the box.	
fill	If true, fill the rectangle with the current foreground color, as set by m_gc->set_foreground(color).	
	Defaults to true.	

13.21.3.25 draw_rectangle() [4/6]

It also supports setting the foreground color to the draw_rectangle() function.

We have a number of such functions: for the main window, for the main pixmap, and for any drawing surface. Is the small bit of conciseness worth it?

Parameters

drawable The surface on which to draw the box.	
С	Provides the foreground color to set.
Х	The x-coordinate of the origin.
У	The y-coordinate of the origin.
lx	The width of the box.
ly	The height of the box.
fill	If true, fill the rectangle with the current foreground color, as set by m_gc->set_foreground(color).
	Defaults to true.

```
13.21.3.26 draw_rectangle() [5/6]
```

```
int x,
int y,
int lx,
int ly,
bool fill = true ) [inline], [protected]
```

pixmap	The object on which to draw the rectangle.	
X	The x-coordinate of the origin.	
У	he y-coordinate of the origin.	
lx	The width of the box.	
ly	The height of the box.	
fill	If true, fill the rectangle with the current foreground color, as set by m_gc->set_foreground(color).	
	Defaults to true.	

13.21.3.27 draw_rectangle() [6/6]

```
void seq64::gui_drawingarea_gtk2::draw_rectangle (
    Glib::RefPtr< Gdk::Pixmap > & pixmap,
    const Color & c,
    int x,
    int y,
    int lx,
    int ly,
    bool fill = true ) [protected]
```

It also supports setting the foreground color to the draw_rectangle() function.

We have a number of such functions: for the main window, for the main pixmap, and for any drawing surface. Is the small bit of conciseness worth it?

Parameters

pixmap	The surface on which to draw the box.	
С	Provides the foreground color to set.	
Х	The x-coordinate of the origin.	
У	The y-coordinate of the origin.	
lx	The width of the box.	
ly	The height of the box.	
fill	If true, fill the rectangle with the current foreground color, as set by m_gc->set_foreground(color).	
	Defaults to true.	

```
13.21.3.28 draw_rectangle_on_pixmap() [1/2]
```

```
int y,
int lx,
int ly,
bool fill = true ) [inline], [protected]
```

X	The x-coordinate of the origin.
У	The y-coordinate of the origin.
lx	The width of the box.
ly	The height of the box.
fill	If true, fill the rectangle with the current foreground color, as set by m_gc->set_foreground(color). Defaults
	to true.

13.21.3.29 draw_rectangle_on_pixmap() [2/2]

It adds setting the foreground color to the draw_rectangle() function.

Parameters

С	Provides the foreground color to set.
Х	The x-coordinate of the origin.
У	The y-coordinate of the origin.
lx	The width of the box.
ly	The height of the box.
fill	If true, fill the rectangle with the current foreground color, as set by m_gc->set_foreground(color). Defaults
	to true.

13.21.3.30 draw_normal_rectangle_on_pixmap()

It uses Gtk to get the proper background styling for the rectangle.

X	The x-coordinate of the origin.
У	The y-coordinate of the origin.
lx	The width of the box.
ly	The height of the box.
fill	If true, fill the rectangle with the current foreground color, as set by m_gc->set_foreground(color). Defaults
	to true.

13.21.3.31 draw_drawable()

```
void seq64::gui_drawingarea_gtk2::draw_drawable (
    int xsrc,
    int ysrc,
    int xdest,
    int ydest,
    int width,
    int height ) [inline], [protected]
```

13.21.3.32 scroll_hadjust()

This function is currently duplicated in the gui_drawingarea_gtk2 and gui_window_gtk2 modules.

Parameters

hadjust	Provides a reference to the adjustment object to be adjusted. Do we really need this to be a parameter? Why not just use the m_hadjust member? (Note that this member is not present in the similar gui_window_gtk2 class.)
step	Provides the step value to use for adjusting the horizontal scrollbar. If negative, the adjustment is leftward. If positive, the adjustment is rightward. It can be the value of m_hadjust->get_step_increment(), or provided especially to keep up with the progress bar.

13.21.3.33 scroll_vadjust()

vadjust	Provides a reference to the adjustment object to be adjusted.
step	Provides the step value to use for adjusting the vertical scrollbar. If negative, the adjustment is upward. If positive, the adjustment is downward. It can be the value of m_vadjust->get_step_increment().

13.21.3.34 scroll_hset()

13.21.3.35 scroll_vset()

13.21.3.36 set_current_drop_x()

Parameters

x The x value to be set.

13.21.3.37 set_current_drop_y()

Parameters

y The y value to be set.

13.21.3.38 gtk_drawarea_init()

```
void seq64::gui_drawingarea_gtk2::gtk_drawarea_init ( ) [private]
```

13.21.3.39 on_realize()

```
void seq64::gui_drawingarea_gtk2::on_realize ( ) [protected]
```

It allocates any additional resources that weren't initialized in the constructor.

13.21.4 Field Documentation

13.21.4.1 m_gc

```
Glib::RefPtr<Gdk::GC> seq64::gui_drawingarea_gtk2::m_gc [protected]
```

13.21.4.2 m_window

```
Glib::RefPtr<Gdk::Window> seq64::gui_drawingarea_gtk2::m_window [protected]
```

Wrapper functions with undecorated wrapper names are used for accessing this item. We hope to be able to hide this items completely some day.

13.21.4.3 m_vadjust

```
Gtk::Adjustment& seq64::gui_drawingarea_gtk2::m_vadjust [protected]
```

13.21.4.4 m_hadjust

```
Gtk::Adjustment& seq64::gui_drawingarea_gtk2::m_hadjust [protected]
```

13.21.4.5 m_pixmap

```
Glib::RefPtr<Gdk::Pixmap> seq64::gui_drawingarea_gtk2::m_pixmap [protected]
```

Wrapper functions with undecorated wrapper names are used for accessing this item. We hope to be able to hide this items completely some day.

```
13.21.4.6 m_background
```

```
Glib::RefPtr<Gdk::Pixmap> seq64::gui_drawingarea_gtk2::m_background [protected]
```

Our wrappers still leave this member exposed (giggle).

13.21.4.7 m_foreground

```
Glib::RefPtr<Gdk::Pixmap> seq64::gui_drawingarea_gtk2::m_foreground [protected]
```

Our wrappers still leave this member exposed.

13.21.4.8 m_mainperf

```
perform& seq64::gui_drawingarea_gtk2::m_mainperf [protected]
```

We could move this into yet another base class, since a number of classes don't need it. Probably not worth the effort at this time.

13.21.4.9 m_window_x

```
int seq64::gui_drawingarea_gtk2::m_window_x [protected]
```

Could make this constant, but some windows are resizable. Window width value.

13.21.4.10 m_window_y

```
int seq64::gui_drawingarea_gtk2::m_window_y [protected]
```

13.21.4.11 m_current_x

```
int seq64::gui_drawingarea_gtk2::m_current_x [protected]
```

Current mouse x value.

13.21.4.12 m_current_y

```
int seq64::gui_drawingarea_gtk2::m_current_y [protected]
```

13.21.4.13 m_drop_x

int seq64::gui_drawingarea_gtk2::m_drop_x [protected]

Provides the x and y value of where the dragging started. Current mouse x-drop value.

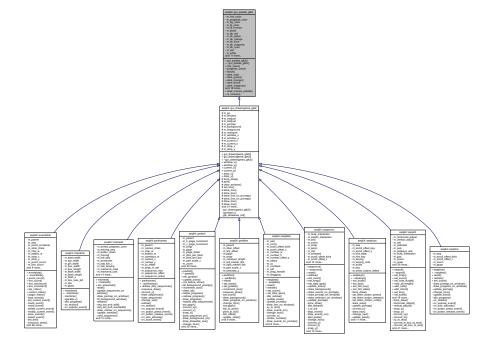
13.21.4.14 m_drop_y

int seq64::gui_drawingarea_gtk2::m_drop_y [protected]

13.22 seq64::gui_palette_gtk2 Class Reference

Implements a stock palette of Gdk::Color elements.

Inheritance diagram for seq64::gui_palette_gtk2:



Public Member Functions

- gui_palette_gtk2 ()
 - Principal constructor.
- ~gui_palette_gtk2 ()

Provides a destructor to delete allocated objects.

• const Color & line_color () const

'Getter' function for member m_line_color Provides an experimental way to change some line colors from black to something else.

• const Color & progress_color () const

'Getter' function for member m_progress_color Provides an experimental way to change the progress line color from black to something else.

· const Color & black () const

'Getter' function for member m_black Although these color getters return static values (if so compiled), these colors are used only in the window and drawing-area classes, so no need to make these functions static.

· const Color & dark red () const

'Getter' function for member m_dk_red

const Color & dark_green () const

'Getter' function for member m_dk_green

const Color & dark_orange () const

'Getter' function for member m_dk_orange

· const Color & dark_blue () const

'Getter' function for member m_dk_blue

const Color & dark_magenta () const

'Getter' function for member m_dk_magenta

const Color & dark_cyan () const

'Getter' function for member m dk cyan

const Color & white () const

'Getter' function for member m_white

const Color & grey () const

'Getter' function for member m_grey

const Color & dark_grey () const

'Getter' function for member m_dk_grey

const Color & light_grey () const

'Getter' function for member m_lt_grey

const Color & red () const

'Getter' function for member m_red

• const Color & orange () const

'Getter' function for member m_orange

· const Color & yellow () const

'Getter' function for member m_yellow

· const Color & green () const

'Getter' function for member m_green

• const Color & blue () const

'Getter' function for member m_blue

const Color & black_paint () const

'Getter' function for member m_blk_paint

· const Color & white paint () const

'Getter' function for member m_wht_paint

const Color & black_key () const

'Getter' function for member m_blk_key

· const Color & white_key () const

'Getter' function for member m_wht_key

const Color & bg_color () const

'Getter' function for member m bg color

void bg_color (const Color &c)

'Setter' function for member m_bg_color

const Color & fg_color () const

'Getter' function for member m_fg_color

void fg_color (const Color &c)

'Setter' function for member m_fg_color

Static Public Member Functions

static void load_inverse_palette (bool inverse=true)

Provides an alternate color palette, somewhat constrained by the colors in the font bitmaps.

• static bool is inverse ()

Indicates if the inverse color palette is loaded.

Protected Types

typedef Gdk::Color Color

Provides a type for the color object.

Private Attributes

· Color m line color

Provides the line color.

· Color m_progress_color

Provides the progress bar color.

Color m_bg_color

The current background color.

Color m_fg_color

The current foreground color.

Static Private Attributes

• static bool m_is_inverse

Flags the presense of the inverse color palette.

• static const Color m_black

Provides the black color.

· static const Color m_dk_red

Provides a blood-red color.

• static const Color m_dk_green

Provides a dark green color.

• static const Color m_dk_orange

Provides a dark orange color.

static const Color m_dk_blue

Provides the dark blue color.

static const Color m_dk_magenta

Provides the dark cyan color.

• static const Color m_dk_cyan

Provides the dark cyan color.

static const Color m_red

Provides the red color.

• static const Color m_white

Provides the white color.

static const Color m_orange

Provides the orange color.

· static const Color m yellow

Provides the yellow color.

static const Color m_green

Provides the green color.

· static const Color m blue

Provides the blue color.

static Color m_grey

Provides the grey color.

static Color m_dk_grey

Provides the dark grey color.

static Color m_lt_grey

Provides the light grey color.

static Color m blk paint

An invertible black color.

static Color m_wht_paint

An invertible white color.

· static Color m blk key

Provides the color of a black key.

· static Color m wht key

Provides the color of a white key.

13.22.1 Detailed Description

Note that this class must be derived from Gtk::DrawingArea (or Gtk::Widget) in order to get access to the get_
default_colormap() function used in the constructor.

13.22.2 Member Typedef Documentation

13.22.2.1 Color

```
typedef Gdk::Color seq64::gui_palette_gtk2::Color [protected]
```

The following uses are made of each color:

- Black. The background color of armed patterns. The color of most lines in the user interface, including the main grid lines. The default color of progress lines and text.
- White. The default background color of just about everything drawn in the application.
- · Grey. The color of minor grid lines and the markers for the currently-selected scale.
- · Dark grey. The color of some grid lines, and the background of a queued pattern slot.
- · Light grey. The color of some grid lines.
- · Red. The optional color of progress bars.
- · Orange. The fill-in color for selected notes and events.
- Dark orange. The color of selected event data lines and the color of the selection box for events to be pasted.

- Yellow. The background of the pattern and name slots for empty patterns. The text color for selected empty pattern slots.
- · Green. Not yet used.
- · Blue. Not yet used.
- Dark cyan. The background color of muted patterns currently in edit, or the pattern that contains the original data for an imported SMF 0 song. The text color of an unmuted pattern currently in edit. These colors apply to the pattern editor and the song editor. The color of the selected background pattern in the song editor.
- · Line color. The generic line color, meant for expansion. Currently black.
- Progress color. The progress line color. Black by default, but can be set to red.
- · Background color. The currently-in-use background color. Can vary a lot when a pixmap is being redrawn.
- Foreground color. The currently-in-use foreground color. Can vary a lot when a pixmap is being redrawn.

13.22.3 Constructor & Destructor Documentation

13.22.3.1 gui_palette_gtk2()

```
seq64::gui_palette_gtk2::gui_palette_gtk2 ( )
```

In the constructor one can only allocate colors; get_window() returns 0 because this window has not yet been realized. Also note that the possible color names that can be used are found in /usr/share/X11/rgb.txt.

13.22.3.2 \sim gui_palette_gtk2()

```
seq64::gui_palette_gtk2::~gui_palette_gtk2 ( )
```

13.22.4 Member Function Documentation

13.22.4.1 load_inverse_palette()

Inverse is not a complete inverse. It is more like a "night" mode. However, there are still some bright colors even in this mode. Some colors, such as the selection color (orange) are the same in either mode.

Parameters

inverse If true, load the alternate palette. Otherwise, load the default palette.

```
13.22.4.2 is_inverse()
static bool seq64::gui_palette_gtk2::is_inverse ( ) [inline], [static]
13.22.4.3 line_color()
const Color& seq64::gui_palette_gtk2::line_color ( ) const [inline]
Might eventually be selectable from the "user" configuration file
13.22.4.4 progress_color()
const Color& seq64::gui_palette_gtk2::progress_color ( ) const [inline]
Now selectable from the "user" configuration file.
13.22.4.5 black()
const Color& seq64::gui_palette_gtk2::black ( ) const [inline]
13.22.4.6 dark_red()
const Color& seq64::gui_palette_gtk2::dark_red ( ) const [inline]
13.22.4.7 dark_green()
const Color& seq64::gui_palette_gtk2::dark_green ( ) const [inline]
13.22.4.8 dark_orange()
const Color& seq64::gui_palette_gtk2::dark_orange ( ) const [inline]
```

```
13.22.4.9 dark_blue()
const Color& seq64::gui_palette_gtk2::dark_blue ( ) const [inline]
13.22.4.10 dark_magenta()
const Color& seq64::gui_palette_gtk2::dark_magenta ( ) const [inline]
13.22.4.11 dark_cyan()
const Color& seq64::gui_palette_gtk2::dark_cyan ( ) const [inline]
13.22.4.12 white()
const Color& seq64::gui_palette_gtk2::white ( ) const [inline]
13.22.4.13 grey()
const Color& seq64::gui_palette_gtk2::grey ( ) const [inline]
13.22.4.14 dark_grey()
const Color& seq64::gui_palette_gtk2::dark_grey ( ) const [inline]
13.22.4.15 light_grey()
const Color& seq64::gui_palette_gtk2::light_grey ( ) const [inline]
13.22.4.16 red()
const Color& seq64::gui_palette_gtk2::red ( ) const [inline]
```

```
13.22.4.17 orange()
const Color& seq64::gui_palette_gtk2::orange ( ) const [inline]
13.22.4.18 yellow()
const Color& seq64::gui_palette_gtk2::yellow ( ) const [inline]
13.22.4.19 green()
const Color& seq64::gui_palette_gtk2::green ( ) const [inline]
13.22.4.20 blue()
const Color& seq64::gui_palette_gtk2::blue ( ) const [inline]
13.22.4.21 black_paint()
const Color& seq64::gui_palette_gtk2::black_paint ( ) const [inline]
13.22.4.22 white_paint()
const Color& seq64::gui_palette_gtk2::white_paint ( ) const [inline]
13.22.4.23 black_key()
const Color& seq64::gui_palette_gtk2::black_key ( ) const [inline]
13.22.4.24 white_key()
const Color& seq64::gui_palette_gtk2::white_key ( ) const [inline]
```

```
13.22.4.25 bg_color() [1/2]
const Color& seq64::gui_palette_gtk2::bg_color ( ) const [inline]
13.22.4.26 bg_color() [2/2]
void seq64::gui_palette_gtk2::bg_color (
            const Color & c ) [inline]
13.22.4.27 fg_color() [1/2]
const Color& seq64::gui_palette_gtk2::fg_color ( ) const [inline]
13.22.4.28 fg_color() [2/2]
void seq64::gui_palette_gtk2::fg_color (
             const Color & c ) [inline]
13.22.5 Field Documentation
13.22.5.1 m_is_inverse
bool seq64::gui_palette_gtk2::m_is_inverse [static], [private]
By default, the inverse color palette is not loaded.
13.22.5.2 m_black
const STATIC_COLOR seq64::gui_palette_gtk2::m_black [static], [private]
13.22.5.3 m_dk_red
const STATIC_COLOR seq64::gui_palette_gtk2::m_dk_red [static], [private]
```

```
13.22.5.4 m_dk_green
```

const STATIC_COLOR seq64::gui_palette_gtk2::m_dk_green [static], [private]

13.22.5.5 m_dk_orange

const STATIC_COLOR seq64::gui_palette_gtk2::m_dk_orange [static], [private]

13.22.5.6 m_dk_blue

const STATIC_COLOR seq64::gui_palette_gtk2::m_dk_blue [static], [private]

13.22.5.7 m_dk_magenta

const STATIC_COLOR seq64::gui_palette_gtk2::m_dk_magenta [static], [private]

13.22.5.8 m_dk_cyan

const STATIC_COLOR seq64::gui_palette_gtk2::m_dk_cyan [static], [private]

13.22.5.9 m_red

const STATIC_COLOR seq64::gui_palette_gtk2::m_red [static], [private]

13.22.5.10 m_white

const STATIC_COLOR seq64::gui_palette_gtk2::m_white [static], [private]

13.22.5.11 m_orange

const STATIC_COLOR seq64::gui_palette_gtk2::m_orange [static], [private]

```
13.22.5.12 m_yellow
const STATIC_COLOR seq64::gui_palette_gtk2::m_yellow [static], [private]
13.22.5.13 m_green
const STATIC_COLOR seq64::gui_palette_gtk2::m_green [static], [private]
13.22.5.14 m_blue
const STATIC_COLOR seq64::gui_palette_gtk2::m_blue [static], [private]
13.22.5.15 m_grey
STATIC_COLOR seq64::gui_palette_gtk2::m_grey [static], [private]
13.22.5.16 m_dk_grey
STATIC_COLOR seq64::gui_palette_gtk2::m_dk_grey [static], [private]
13.22.5.17 m_lt_grey
STATIC_COLOR seq64::gui_palette_gtk2::m_lt_grey [static], [private]
13.22.5.18 m_blk_paint
STATIC_COLOR seq64::gui_palette_gtk2::m_blk_paint [static], [private]
13.22.5.19 m_wht_paint
```

STATIC_COLOR seq64::gui_palette_gtk2::m_wht_paint [static], [private]

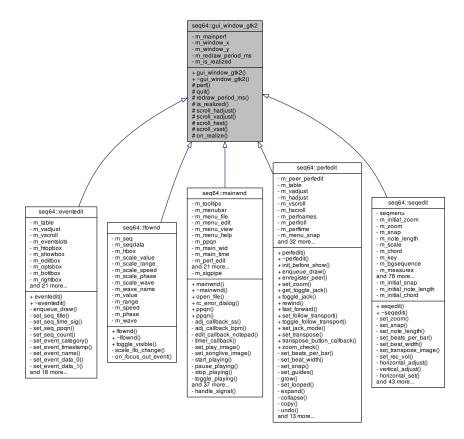
```
13.22.5.20 m_blk_key
STATIC_COLOR seq64::gui_palette_gtk2::m_blk_key [static], [private]
13.22.5.21 m_wht_key
STATIC_COLOR seq64::gui_palette_gtk2::m_wht_key [static], [private]
13.22.5.22 m_line_color
Color seq64::gui_palette_gtk2::m_line_color [private]
13.22.5.23 m_progress_color
Color seq64::gui_palette_gtk2::m_progress_color [private]
13.22.5.24 m_bg_color
Color seq64::gui_palette_gtk2::m_bg_color [private]
13.22.5.25 m_fg_color
```

Color seq64::gui_palette_gtk2::m_fg_color [private]

13.23 seq64::gui_window_gtk2 Class Reference

This class supports a basic interface for Gtk::Window-derived objects.

Inheritance diagram for seq64::gui window gtk2:



Public Member Functions

• gui_window_gtk2 (perform &p, int window_x=0, int window_y=0)

Principal constructor, has a reference to the all-important perform object.

virtual ~gui_window_gtk2 ()

This rote constructor does nothing.

Protected Member Functions

· perform & perf ()

'Getter' function for member m_mainperf

virtual void quit ()

Provides "quit" functionality that WE HAVE OVERLOOKED!!! At some point we need to rectify this situation, probably for the sake of session support.

· int redraw period ms () const

'Getter' function for member m_redraw_period_ms

bool is_realized () const

'Getter' function for member m_is_realized

void scroll_hadjust (Gtk::Adjustment &hadjust, double step)

This function provides optimization for the on_scroll_event() functions, and should provide support for having the seqedit/seqroll/seqtime/seqdata panes follow the scrollbar, in a future upgrade.

void scroll_vadjust (Gtk::Adjustment &vadjust, double step)

This function is the vertical version of scroll_hadjust().

void scroll_hset (Gtk::Adjustment &hadjust, double value)

This function is the horizontal scroll setter.

• void scroll vset (Gtk::Adjustment &vadjust, double value)

This function is the vertical scroll setter.

· void on realize ()

This callback function calls the base-class on_realize() function, and sets the m_is_realized flag.

Private Attributes

· perform & m mainperf

The master object, sort of a sequence buss for all of the sequence.

• int m_window_x

Window sizes.

· int m_window_y

The height of the window.

· int m redraw period ms

Provides the timer period for the eventedit timer, used to determine the rate of redrawing.

· bool m is realized

Indicates if on_realize() has been called.

13.23.1 Constructor & Destructor Documentation

13.23.1.1 gui_window_gtk2()

Note

We've collected the redraw timeouts into a base-class member. Most were valued at c_redraw_ms (40 ms), but mainwind used 25 ms, so beware. We will eventually make this a user-interface parameter.

Parameters

p	Refers to the main performance object.
window←	The width of the window.
_X	
window←	The height of the window.
У	

```
13.23.1.2 ∼gui_window_gtk2()
seq64{::}gui\_window\_gtk2{::}{\sim}gui\_window\_gtk2 \text{ ( ) } [virtual]
13.23.2 Member Function Documentation
13.23.2.1 perf()
perform& seq64::gui_window_gtk2::perf ( ) [inline], [protected]
13.23.2.2 quit()
virtual void seq64::gui_window_gtk2::quit ( ) [inline], [protected], [virtual]
13.23.2.3 redraw_period_ms()
int seq64::gui_window_gtk2::redraw_period_ms () const [inline], [protected]
13.23.2.4 is_realized()
bool seq64::qui_window_qtk2::is_realized ( ) const [inline], [protected]
13.23.2.5 scroll_hadjust()
void seq64::gui_window_gtk2::scroll_hadjust (
             Gtk::Adjustment & hadjust,
             double step ) [protected]
```

This function is currently duplicated in the gui_drawingarea_gtk2 and gui_window_gtk2 modules.

Parameters

hadjust	Provides a reference to the adjustment object to be adjusted.
step	Provides the step value to use for adjusting the horizontal scrollbar. If negative, the adjustment is leftward. If positive, the adjustment is rightward. It can be the value of m_hadjust->get_step_increment(), or provided especially to keep up with the progress bar.

13.23.2.6 scroll_vadjust()

Parameters

vadjust	Provides a reference to the adjustment object to be adjusted.
step	Provides the step value to use for adjusting the vertical scrollbar. If greater than 0, the movement is
	downward. If less than zero, the movement is upward.

13.23.2.7 scroll_hset()

Parameters

h	adjust	Provides a reference to the adjustment object to be set. It is clamped as necessary.
V	alue	Provides the value to use for setting the horizontal scrollbar.

13.23.2.8 scroll_vset()

Parameters

vadjust	Provides a reference to the vertical adjustment object to be set. It is clamped as necessary.
value	Provides the value to use for setting the vertical scrollbar.

13.23.2.9 on_realize()

```
void seq64::gui_window_gtk2::on_realize ( ) [protected]
```

13.23.3 Field Documentation

```
13.23.3.1 m_mainperf
```

```
perform& seq64::gui_window_gtk2::m_mainperf [private]
```

And a whole lot more than that.

```
13.23.3.2 m_window_x
```

```
int seq64::gui_window_gtk2::m_window_x [private]
```

Could make this constant, but some windows are resizable. The width of the window.

```
13.23.3.3 m_window_y
```

```
int seq64::gui_window_gtk2::m_window_y [private]
```

13.23.3.4 m_redraw_period_ms

```
int seq64::gui_window_gtk2::m_redraw_period_ms [private]
```

This is currently hardwired to 40 ms in Linux, and 20 ms in Windows. Note that mainwnd used 25 ms.

13.23.3.5 m_is_realized

```
bool seq64::gui_window_gtk2::m_is_realized [private]
```

In some cases, we don't want to draw in objects that haven't yet appeared, otherwise crashes occur.

13.24 seq64::jack_assistant Class Reference

This class provides the performance mode JACK support.

Public Member Functions

jack_assistant (perform &parent, midibpm bpminute=SEQ64_DEFAULT_BPM, int ppqn=SEQ64_USE_DE
 FAULT_PPQN, int bpm=SEQ64_DEFAULT_BEATS_PER_MEASURE, int beatwidth=SEQ64_DEFAULT_
 BEAT_WIDTH)

This constructor initializes a number of member variables, some of them public!

~jack_assistant ()

The destructor doesn't need to do anything yet.

perform & parent ()

'Getter' function for member m_jack_parent Needed for external callbacks.

· const perform & parent () const

'Getter' function for member m_jack_parent, const version

bool is_running () const

'Getter' function for member m_jack_running

bool is_master () const

'Getter' function for member m_jack_master

int get_ppqn () const

'Getter' function for member m_ppqn

int get_beat_width () const

'Getter' function for member m_beat_width

void set_beat_width (int bw)

'Setter' function for member m_beat_width

int get_beats_per_measure () const

'Getter' function for member m_beats_per_measure

void set_beats_per_measure (int bpm)

'Setter' function for member m_beats_per_measure

midibpm get_beats_per_minute () const

'Getter' function for member m_beats_per_minute

void set_beats_per_minute (midibpm bpminute)

'Setter' function for member m_beats_per_minute For the future, changing the BPM (beats/minute) internally.

jack_transport_state_t transport_state () const

'Getter' function for member m_jack_transport_state

• bool transport_not_starting () const

Returns true if the JACK transport state is not JackTransportStarting.

• bool init ()

Initializes JACK support.

• bool deinit ()

Tears down the JACK infrastructure.

bool session_event ()

Writes the MIDI file named "< jack session dir>-file.mid" using a midifile object, quits if told to by JACK, and can free the JACK session event.

• bool activate ()

Activate JACK here.

· void start ()

If JACK is supported, starts the JACK transport.

void stop ()

If JACK is supported, stops the JACK transport.

void position (bool state, midipulse tick=0)

If JACK is supported and running, sets the position of the transport to the new frame number, frame 0.

• bool output (jack scratchpad &pad)

Performance output function for JACK, called by the perform function of the same name.

void set_ppqn (int ppqn)

'Setter' function for member m_ppqn For the future, changing the PPQN internally.

• double get_jack_tick () const

'Getter' function for member m_jack_tick

const jack position t & get jack pos () const

'Getter' function for member m_jack_pos

void toggle_jack_mode ()

'Setter' function for member m_toggle_jack

void set_jack_mode (bool mode)

'Setter' function for member m_toggle_jack

• bool get_jack_mode () const

'Getter' function for member m_toggle_jack Seems misnamed.

· midipulse get_jack_stop_tick () const

'Getter' function for member m_jack_stop_tick

void set_jack_stop_tick (long tick)

'Setter' function for member m jack stop tick

jack_nframes_t jack_frame_rate () const

'Getter' function for member m_jack_frame_rate

· bool get follow transport () const

'Getter' function for member m_follow_transport

void set follow transport (bool aset)

'Setter' function for member m_follow_transport

void toggle_follow_transport ()

'Setter' function for member m_follow_transport

bool toggle_song_start_mode ()

'Setter' function for member parent().toggle_song_start_mode()

bool song_start_mode () const

'Getter' function for member parent().song_start_mode()

void set_start_from_perfedit (bool start)

'Setter' function for member parent().start from perfedit()

- jack client t * client () const
- const std::string & client_name () const

'Getter' function for member m_jack_client_name

· const std::string & client_uuid () const

'Getter' function for member m_jack_client_uuid

Private Member Functions

void set jack running (bool flag)

'Setter' function for member m_jack_running

· double tick multiplier () const

Convenience function for internal use.

• jack_client_t * client_open (const std::string &clientname)

A member wrapper function for the new free function create_jack_client().

void get_jack_client_info ()

Tries to obtain the best information on the JACK client and the UUID assigned to this client.

void show_position (const jack_position_t &pos) const

Shows a one-line summary of a JACK position structure.

int sync (jack_transport_state_t state=(jack_transport_state_t)(-1))

A helper function for syncing up with JACK parameters.

void set_position (midipulse currenttick)

Provides the code that was effectively commented out in the perform::position_jack() function.

Static Private Member Functions

static bool info message (const std::string &msg)

Common-code for console messages.

static bool error_message (const std::string &msg)

Common-code for error messages.

Private Attributes

perform & m_jack_parent

Provides the perform object that needs this JACK assistant/scratchpad class.

• jack_client_t * m_jack_client

Provides a handle into JACK, so that the application, as a JACK client, can issue commands and retrieve status information from JACK.

std::string m_jack_client_name

A new member to hold the actual name of the client assigned by JACK.

std::string m_jack_client_uuid

A new member to hold the actual UUID of the client assigned by JACK.

• jack_nframes_t m_jack_frame_current

Holds the current frame number obtained from JACK transport, via a call to jack_get_current_transport_frame().

jack_nframes_t m_jack_frame_last

Holds the last frame number we got from JACK, so that progress can be tracked.

• jack position t m jack pos

Provides positioning information on JACK playback.

jack_transport_state_t m_jack_transport_state

Holds the JACK transport state.

• jack_transport_state_t m_jack_transport_state_last

Holds the last JACK transport state.

· double m jack tick

The tick/pulse value derived from the current frame number, the ticks/beat value, the beats/minute value, and the frame rate.

• jack_session_event_t * m_jsession_ev

Provides a kind of handle to the JACK session manager.

bool m_jack_running

Indicates if JACK Sync has been enabled successfully.

bool m_jack_master

Indicates if JACK Sync has been enabled successfully, with the application running as JACK Master.

• jack_nframes_t m_jack_frame_rate

Holds the current frame rate.

• bool m_toggle_jack

Ostensibly a toggle, the functions that access this member are called "jack_mode" functions.

midipulse m_jack_stop_tick

Used in jack_process_callback() to reposition when JACK transport is not rolling or starting.

bool m_follow_transport

TBD.

int m_ppqn

Holds the global PPQN value for the Sequencer64 session.

int m_beats_per_measure

Holds the song's beats/measure value for using in setting JACK position.

· int m beat width

Holds the song's beat width value (denominator of the time signature) for using in setting JACK position.

· midibpm m beats per minute

Holds the song's beats/minute (BPM) value for using in setting JACK position.

Static Private Attributes

• static jack_status_pair_t sm_status_pairs []

Pairs the JACK status bits with human-readable descriptions of each one.

Friends

- int jack_transport_callback (jack_nframes_t nframes, void *arg)
- void jack_shutdown_callback (void *arg)

This callback is to shut down JACK by clearing the jack_assistant :: m_jack_running flag.

 $\bullet \ \ int \ jack_sync_callback \ (jack_transport_state_t \ state, \ jack_position_t \ *pos, \ void \ *arg)\\$

Global functions for JACK support and JACK sessions.

void jack_timebase_callback (jack_transport_state_t state, jack_nframes_t nframes, jack_position_t *pos, int new_pos, void *arg)

The JACK timebase function defined here sets the JACK position structure.

- long get_current_jack_position (void *arg)
- void jack_session_callback (jack_session_event_t *ev, void *arg)

Set the m_jsession_ev (event) value of the perform object.

13.24.1 Constructor & Destructor Documentation

13.24.1.1 jack_assistant()

Note that the perform object currently calls jack_assistant::init(), but that call could be made here instead.

Parameters

parent	Provides a reference to the main perform object that needs to control JACK event.
bpminute	The beats/minute to set up JACK to use (applies to Master setup).
ppqn	The parts-per-quarter-note setting in force for the present tune.
bpmeasure	The beats/measure (time signature numerator) in force for the present tune.
beatwidth	The beat-width (time signature denominator) in force for the present tune.

```
13.24.1.2 \sim jack_assistant()
```

```
seq64::jack_assistant::~jack_assistant ( )
```

The perform object currently calls jack_assistant::deinit(), but that call could be made here instead.

13.24.2 Member Function Documentation

```
13.24.2.1 parent() [1/2]
perform& seq64::jack_assistant::parent ( ) [inline]
13.24.2.2 parent() [2/2]
const perform& seq64::jack_assistant::parent ( ) const [inline]
13.24.2.3 is_running()
bool seq64::jack_assistant::is_running ( ) const [inline]
13.24.2.4 is_master()
bool seq64::jack_assistant::is_master ( ) const [inline]
13.24.2.5 get_ppqn()
int seq64::jack_assistant::get_ppqn ( ) const [inline]
13.24.2.6 get_beat_width()
int seq64::jack_assistant::get_beat_width ( ) const [inline]
13.24.2.7 set_beat_width()
void seq64::jack_assistant::set_beat_width (
             int bw ) [inline]
```

Parameters

bw Provides the beat-width (denominator of the time signature) value to set.

```
13.24.2.8 get_beats_per_measure()
```

```
int seq64::jack_assistant::get_beats_per_measure ( ) const [inline]
```

13.24.2.9 set_beats_per_measure()

Parameters

bpm | Provides the beats/measure (numerator of the time signature) value to set.

13.24.2.10 get_beats_per_minute()

```
midibpm seq64::jack_assistant::get_beats_per_minute ( ) const [inline]
```

13.24.2.11 set_beats_per_minute()

We should consider adding validation. However, perform::set_beats_per_minute() does validate already.

Parameters

bpminute Provides the beats/minute value to set.

13.24.2.12 transport_state()

```
jack_transport_state_t seq64::jack_assistant::transport_state ( ) const [inline]
```

13.24.2.13 transport_not_starting()

```
bool seq64::jack_assistant::transport_not_starting ( ) const [inline]
```

13.24.2.14 init()

```
bool seq64::jack_assistant::init ( )
```

Then we become a new client of the JACK server.

A sync callback is needed for polling of slow-sync clients. But seq24/sequencer64 are not slow-sync clients. We don't really need to be a slow-sync client, as far as we can tell. We can't get JACK working exactly the way it does in seq24 without the callback in place. Plus, it does things important to the setup of JACK. So now this setup is permanent.

Jack transport settings:

```
There are three settings: On, Master, and Master Conditional. Currently, they can all be selected in the user-interface's File / Options / JACK/LASH page. We really want only the proper combinations to be set, for clarity (the user-interface now takes care of this. We need to initialize if any of them are set, and the rc_settings::with_jack() function tells us that.
```

jack_set_process_callback() patch:

```
Implemented first patch from freddix/seq24 GitHub project, to fix JACK transport. One line of code. Well, we added some error-checking. :-) Found some old notes on the Web the this patch really only works (to prevent seq24 freeze) if seq24 is set as JACK Master, or if another client application, such as Qtractor, is running as JACK Master (and then seq24 will apparently follow it).
```

STAZED: The call to jack_timebase_callback() to supply jack with BBT, etc would occasionally fail when the *pos information had zero or some garbage in the pos.frame_rate variable. This would occur when there was a rapid change of frame position by another client... i.e. gjackctl. From the jack API:

"pos address of the position structure for the next cycle; pos->frame will be its frame number. If new_pos is FALSE, this structure contains extended position information from the current cycle. If TRUE, it contains whatever was set by the requester. The timebase_callback's task is to update the extended information here."

The "If TRUE" line seems to be the issue. It seems that qjackctl does not always set pos.frame_rate so we get garbage and some strange BBT calculations that display in qjackctl. So we need to set it here and just use m_\circ} jack_frame_rate for calculations instead of pos.frame_rate.

Returns

Returns true if JACK is now considered to be running (or if it was already running.)

13.24.2.15 deinit()

```
bool seq64::jack_assistant::deinit ( )
```

Returns

Returns the value of m jack running, which should be false.

13.24.2.16 session_event()

```
bool seq64::jack_assistant::session_event ( )
```

ca 2015-07-24 Just a note: The OMA (OpenMandrivaAssociation) patch was already applied to seq24 v.0.9.2. It put quotes around the –file argument. However, the –file option doesn't work, so let's change that line.

```
sequencer64 --file \"${SESSION_DIR}file.mid\" --jack_session_uuid
```

Why are we using a Glib::ustring here? Convenience. But with C++11, we could use a lexical_cast<>. No more ustring, baby! It doesn't really matter; this function can call Gtk::Main::quit(), via the parent's gui().quit() function.

Returns

Always returns false.

13.24.2.17 activate()

```
bool seq64::jack_assistant::activate ( )
```

Returns

Returns true if the m_jack_client pointer is null, which means only that we're not running JACK. Also returns true if the pointer exists and the jack_active() call succeeds.

Side-effect(s) The m jack running and m jack master flags are falsified in jack activate() fails.

13.24.2.18 start()

```
void seq64::jack_assistant::start ( )
```

This function assumes that m_jack_client is not null, if m_jack_running is true.

Found this note in the Hydrogen code:

```
When jack_transport_start() is called, it takes effect from the next processing cycle. The location info from the timebase_master, if there is one, will not be available until the _next_ next cycle. The code must therefore wait one cycle before syncing up with timebase_master.
```

```
13.24.2.19 stop()
void seq64::jack_assistant::stop ( )
```

This function assumes that m_jack_client is not null, if m_jack_running is true.

13.24.2.20 position()

```
void seq64::jack_assistant::position (
    bool songmode,
    midipulse tick = 0 )
```

This new position takes effect in two process cycles. If there are slow-sync clients and the transport is already rolling, it will enter the JackTransportStarting state and begin invoking their sync_callbacks until ready. This function is realtime-safe.

```
http://jackaudio.org/files/docs/html/transport-design.html
```

This position() function is called via perform::position_jack() in the mainwnd, perfedit, perfroll, and seqroll graphical user-interface support objects.

The code that was disabled sets the current tick to 0 or, if state was true, to the leftmost tick (which is probably the position of the L marker). The current tick is then converted to a frame number, and then we locate the transport to that position. We're going to enable this code, but make it dependent on a new boolean parameter that defaults to false, in anticipation of trying it out later.

Stazed:

```
The jack_frame calculation is all that is needed to change JACK position. The BBT calculation can be sent, but will be overridden by the first call to jack_timebase_callback() of any Master set. If no Master is set, then the BBT will display the new position but will not change it, even if the transport is rolling. There is no need to send BBT on position change -- the fact that jack_transport_locate() exists and only uses the frame position is proof that BBT is not needed! Upon further reflection, why not send BBT? Because other programs do not... let's follow convention. The calculation for jack_transport_locate(), works, is simpler, and does not send BBT. The calculation for jack_transport_reposition() will be commented out again. jack_BBT_position() is not necessary to change jack position!
```

Note that there are potentially a couple of divide-by-zero opportunities in this function.

Helgrind complains about a possible data race involving jack_transport_locate() when starting playing.

Parameters

songmode	True if the caller wants to position while in Song mode.
----------	--

Alternate parameter to_left_tick (non-seq32 version):

```
If true, the current tick is set to the leftmost tick, instead of the Oth tick. Now used, but only if relocate is true. One question is, do we want to perform this function if rc().with_jack_transport() is true? Seems like we should be able to do it only if m_jack_master is true.
```

Parameters

tick

If using Song mode for this call then this value is set as the "current tick" value. If it's value is bad (SEQ64_NULL_MIDIPULSE), then this parameter is set to 0 before being used.

13.24.2.21 output()

This code comes from perform::output_func() from seq24.

Note

Follow up on this note found "out there": "Maybe I'm wrong but if I understood correctly, recent jack1 transport no longer goes into Jack_Transport_Starting state before going to Jack_Transport_Rolling (this was deliberately dropped), but seq24 currently needs this to start off with JACK transport." On the other hand, some people have no issues. This may have been due to the lack of m_jack_pos initialization.

Stazed:

Another note about JACK. If another JACK client supplies tempo/BBT different from seq42 (as Master), the perfroll grid will be incorrect. Perfroll uses internal temp/BBT and cannot update on the fly. Even if seq42 could support tempo/BBT changes, all info would have to be available before the transport start, to work. For this reason, the tempo/BBT info will be plugged from the seq42 internal settings here, always. This is the method used by probably all other JACK clients with some sort of time-line. The JACK API indicates that BBT is optional and AFIK, other sequencers only use frame & frame_rate from JACK for internal calculations. The tempo and BBT info is always internal. Also, if there is no Master set, then we would need to plug it here to follow the JACK frame anyways.

Parameters

pad

Provides a JACK scratchpad for sharing certain items between the perform object and the jack_assistant object.

Returns

Returns true if JACK is running.

13.24.2.22 set_ppqn()

We should consider adding validation. But it is used by perform.

```
Parameters
```

```
Provides the PPQN value to set.
 ppqn
13.24.2.23 get_jack_tick()
double seq64::jack_assistant::get_jack_tick ( ) const [inline]
13.24.2.24 get_jack_pos()
const jack_position_t& seq64::jack_assistant::get_jack_pos ( ) const [inline]
13.24.2.25 toggle_jack_mode()
void seq64::jack_assistant::toggle_jack_mode ( ) [inline]
13.24.2.26 set_jack_mode()
void seq64::jack_assistant::set_jack_mode (
             bool mode ) [inline]
13.24.2.27 get_jack_mode()
bool seq64::jack_assistant::get_jack_mode ( ) const [inline]
13.24.2.28 get_jack_stop_tick()
```

midipulse seq64::jack_assistant::get_jack_stop_tick () const [inline]

```
13.24.2.29 set_jack_stop_tick()
void seq64::jack_assistant::set_jack_stop_tick (
             long tick ) [inline]
13.24.2.30 jack_frame_rate()
jack_nframes_t seq64::jack_assistant::jack_frame_rate ( ) const [inline]
13.24.2.31 get_follow_transport()
bool seq64::jack_assistant::get_follow_transport ( ) const [inline]
13.24.2.32 set_follow_transport()
void seq64::jack_assistant::set_follow_transport (
            bool aset ) [inline]
13.24.2.33 toggle_follow_transport()
void seq64::jack_assistant::toggle_follow_transport ( ) [inline]
13.24.2.34 toggle_song_start_mode()
bool seq64::jack_assistant::toggle_song_start_mode ( )
13.24.2.35 song_start_mode()
bool seq64::jack\_assistant::song\_start\_mode ( ) const
```

```
13.24.2.36 set_start_from_perfedit()
\verb"void seq64::jack_assistant::set_start_from_perfedit" (
             bool start )
13.24.2.37 client()
jack_client_t * seq64::jack_assistant::client ( ) const
13.24.2.38 client_name()
const std::string& seq64::jack_assistant::client_name ( ) const [inline]
13.24.2.39 client_uuid()
const std::string& seq64::jack_assistant::client_uuid ( ) const [inline]
13.24.2.40 set_jack_running()
void seq64::jack_assistant::set_jack_running (
              bool flag ) [inline], [private]
Parameters
 flag
       Provides the is-running value to set.
13.24.2.41 tick_multiplier()
```

Returns

Returns the multiplier to convert a JACK tick value according to the PPQN, ticks/beat, and beat-type settings.

double seq64::jack_assistant::tick_multiplier () const [inline], [private]

Should we change 4.0 to a member value? What does it mean?

13.24.2.42 client_open()

Parameters

clientname	Provides the name of the client, used in the call to create_jack_client(). By default, this name is
	the macro SEQ64_PACKAGE (i.e. "sequencer64").

Returns

Returns a pointer to the JACK client if JACK has opened the client connection successfully. Otherwise, a null pointer is returned.

```
13.24.2.43 get_jack_client_info()
void seq64::jack_assistant::get_jack_client_info ( ) [private]
```

Sets m jack client name and m jack client info as side-effects.

```
13.24.2.44 show_position()
```

This function is meant for experimenting and learning.

The fields of this structure are as follows. Only the fields we care about are shown.

```
jack_nframes_t
                       frame_rate:
                                        current frame rate (per second)
    jack_nframes_t frame: frame number, always present
jack_position_bits_t valid: which other fields are valid
JackPositionBBT:
    int32_t
                                        current bar
                        bar:
    int32_t
                        beat:
                                        current beat-within-bar
    int32_t
                        tick:
                                        current tick-within-beat
    double
                        bar_start_tick
    float
                        beats_per_bar: time signature "numerator"
                       beat_type:
                                         time signature "denominator"
    float
    double
                       ticks_per_beat
    double
                        beats_per_minute
JackBBTFrameOffset:
    jack_nframes_t
                        bbt_offset;
                                         frame offset for the BBT fields
Only the most "important" and time-varying fields are shown. The format
output is brief and inscrutable unless you read this format example:
```

```
nnnnn frame B:B:T N/D TPB BPM BBT
            | | | | ------ bbt_offset (frame), even if invalid
 1
         ----- beats_per_minute
         ----- ticks_per_beat (PPQN * 10?)
         | | -----beat_type (denominator)
            -----beats_per_bar (numerator)
          ----- bar : beat : tick
                 ----- frame (number)
     ----- the "valid" bits
```

The "valid" field is shown as bits in the same bit order as shown here, but represented as a five-character string, "nnnnn", n = 0 or 1:

We care most about nnnnn = "00101" in our experiments (the most common output will be "00001"). And we don't worry about non-integer measurements... we truncate them to integers. Change the output format if you want to play with non-Western timings.

Parameters

```
pos The JACK position structure to dump.
```

13.24.2.45 sync()

Sequencer64 is not a slow-sync client (and Stazed support doesn't use it), so that callback is not really needed, but we probably need this sub-function here to start out with the right values for interacting with JACK.

Note the call to jack_transport_query(). This call is *not* is seq24, but seems to be needed in sequencer64 because we put m_jack_pos in the initializer list, which sets all its fields to 0. Seq24 accesses m_jack_pos before it ever gets set, but its fields have values. These values are bogus, but are consistent from run to run on my computer, and allow seq24 to follow another JACK Master, on some computers. It explains why people had different experiences with JACK sync.

If we explicity call jack_transport_query() here, without changing the *state* parameter, then sequencer64 also can follow another JACK Master. (CURRENTLY BUGGY!)

Note that we should consider massaging the following jack_position_t members to set them to 0 (or 0.0) if less than 1.0 or 0.5:

```
bar_start_tick
ticks_per_beat
beats_per_minute
frame_time
next_time
audio frames per video frame
```

Also, why does bbt_offset start at 2128362496?

Parameters

state The JACK transport state to be set.

13.24.2.46 set_position()

We might be able to use it in other functions.

Computing the BBT information from the frame number is relatively simple here, but would become complex if we supported tempo or time signature changes at specific locations in the transport timeline.

```
ticks * 10 = jack ticks;
jack ticks / ticks per beat = num beats;
num beats / beats per minute = num minutes
num minutes * 60 = num seconds
num secords * frame_rate = frame
```

Parameters

	currenttick	Provides the current position to be set.
--	-------------	--

13.24.2.47 info_message()

Adds markers and a newline.

Parameters

```
msg The message to print, sans the newline.
```

Returns

Returns true.

13.24.2.48 error_message()

Adds markers, and sets $m_jack_running$ to false.

Parameters

msg The message to print, sans the newline.

Returns

Returns false for convenience/brevity in setting function return values.

13.24.3 Friends And Related Function Documentation

13.24.3.1 jack_transport_callback

13.24.3.2 jack_shutdown_callback

Parameters

arg Points to the jack_assistant in charge of JACK support for the perform object.

13.24.3.3 jack_sync_callback

This JACK synchronization callback informs the specified perform object of the current state and parameters of JACK.

The transport state will be:

- JackTransportStopped when a new position is requested.
- JackTransportStarting when the transport is waiting to start.
- JackTransportRolling when the timeout has expired, and the position is now a moving target.

This is the slow-sync callback, which the stazed code replaces with jack_transport_callback().

Parameters

state	The JACK Transport state.
pos	The JACK position value.
arg	The pointer to the jack_assistant object. Currently not checked for nullity, nor dynamic-casted.

Returns

Returns 1 if the function works, and 0 if something was wrong.

13.24.3.4 jack timebase callback

```
void jack_timebase_callback (
         jack_transport_state_t state,
         jack_nframes_t nframes,
         jack_position_t * pos,
         int new_pos,
         void * arg ) [friend]
```

The original version of the function worked properly with Hydrogen, but not with Klick. The new code seems to work with both. More testing and clarification is needed. This new code was "discovered" in the source-code for the "SooperLooper" project:

```
http://essej.net/sooperlooper/
```

The first difference with the new code is that it handles the case where the JACK position is moved (new_pos == true). If this is true, and the JackPositionBBT bit is off in pos->valid, then the new BBT value is set.

The seconds set of differences are in the "else" clause. In the new code, it is very simple: calculate the new tick value, back it off by the number of ticks in a beat, and perhaps go to the first beat of the next bar.

In the old code (complex!), the simple BBT adjustment is always made. This changes (perhaps) the beats_per_bar, beat_type, etc. We need to make these settings use the actual global values for beats set for Sequencer64. Then, if transitioning from JackTransportStarting to JackTransportRolling (instead of checking new_pos!), the BBT values (bar, beat, and tick) are finally adjusted. Here are the steps, with old and new steps noted:

```
-# Calculate the "delta" ticks based on the current frame, the
    ticks_per_beat, the beats_per_minute, and the frame_rate. The old
    code saves this in a local, the new code assigns it to pos->tick.
-# Old code: save this delta as a positive value.
-# Figure out the settings and modify bar, beat, tick, and
    bar_start_tick. The old and new code seem to have the same intent,
    but it seems like the new code is faster and also correct.
- Old code: Calculations are made by division and mod
        operations.
- New code: Calculations are made by increments and decrements
    in a while loop.
```

Stazed:

The call to jack_timebase_callback() to supply JACK with BBT, etc. would occasionally fail when the pos information had zero or some garbage in the pos.frame_rate variable. This would occur when there was a rapid change of frame position by another client... i.e. qjackctl. From the JACK APT:

pos address of the position structure for the next cycle; pos->frame will be its frame number. If new_pos is FALSE, this structure contains extended position information from the current cycle. If TRUE, it contains whatever was set by the requester. The timebase_callback's task is to update the extended information here."

The "If TRUE" line seems to be the issue. It seems that qjackctl does not always set pos.frame_rate so we get garbage and some strange BBT calculations that display in qjackctl. So we need to set it here and just use m_jack_frame_rate for calculations instead of pos.frame_rate.

Parameters

state	Indicates the current state of JACK transport.
nframes	The number of JACK frames in the current time period.
pos	Provides the position structure to be filled in, the address of the position structure for the next cycle; pos->frame will be its frame number. If new_pos is FALSE, this structure contains extended position information from the current cycle. If TRUE, it contains whatever was set by the requester. The timebase_callback's task is to update the extended information here.
new_pos	TRUE (non-zero) for a newly requested pos, or for the first cycle after the timebase_callback is defined. This is usually 0 in Sequencer64 at present, and 1 if one, say, presses "rewind" in qjackctl.
arg	Provides the jack_assistant pointer, currently unchecked for nullity.

13.24.3.5 get_current_jack_position

```
long get_current_jack_position (
     void * arg ) [friend]
```

Warning

Currently valgrind flags j->client() as uninitialized.

13.24.3.6 jack_session_callback

```
void jack_session_callback (
          jack_session_event_t * ev,
          void * arg ) [friend]
```

Glib is then used to connect in perform::jack_session_event(). However, the perform object's GUI-support interface is used instead of the following, so that the libseq64 library can be independent of a specific GUI framework:

```
Glib::signal_idle().
    connect(sigc::mem_fun(*jack, &jack_assistant::session_event));
```

Parameters

ev	The JACK event to be set.]
arg	The pointer to the jack_assistant object. Currently not checked for nullity.]

13.24.4 Field Documentation

```
13.24.4.1 sm_status_pairs
```

```
jack_status_pair_t seq64::jack_assistant::sm_status_pairs[] [static], [private]
```

13.24.4.2 m_jack_parent

```
perform& seq64::jack_assistant::m_jack_parent [private]
```

13.24.4.3 m_jack_client

```
jack_client_t* seq64::jack_assistant::m_jack_client [mutable], [private]
```

13.24.4.4 m_jack_client_name

```
std::string seq64::jack_assistant::m_jack_client_name [private]
```

We might show this in the user-interface at some point.

13.24.4.5 m_jack_client_uuid

```
std::string seq64::jack_assistant::m_jack_client_uuid [private]
```

We might show this in the user-interface at some point.

13.24.4.6 m_jack_frame_current

```
jack_nframes_t seq64::jack_assistant::m_jack_frame_current [private]
```

```
13.24.4.7 m_jack_frame_last
```

```
jack_nframes_t seq64::jack_assistant::m_jack_frame_last [private]
```

Also used in incrementing m_jack_tick.

13.24.4.8 m_jack_pos

```
jack_position_t seq64::jack_assistant::m_jack_pos [private]
```

This structure is filled via a call to jack_transport_query(). It holds, among other items, the frame rate (often 48000), the ticks/beat, and the beats/minute.

13.24.4.9 m_jack_transport_state

```
jack_transport_state_t seq64::jack_assistant::m_jack_transport_state [private]
```

Common values are JackTransportStopped, JackTransportRolling, and JackTransportLooping.

13.24.4.10 m_jack_transport_state_last

```
jack_transport_state_t seq64::jack_assistant::m_jack_transport_state_last [private]
```

13.24.4.11 m_jack_tick

```
double seq64::jack_assistant::m_jack_tick [private]
```

13.24.4.12 m_jsession_ev

```
jack_session_event_t* seq64::jack_assistant::m_jsession_ev [private]
```

Used in the session_event() function.

13.24.4.13 m_jack_running

```
bool seq64::jack_assistant::m_jack_running [private]
```

```
13.24.4.14 m_jack_master
```

```
bool seq64::jack_assistant::m_jack_master [private]
```

13.24.4.15 m_jack_frame_rate

```
jack_nframes_t seq64::jack_assistant::m_jack_frame_rate [private]
```

Just in case. QJackCtl does not always set pos.frame_rate, so we get garbage and some strange BBT calculations displayed in qjackctl.

13.24.4.16 m_toggle_jack

```
bool seq64::jack_assistant::m_toggle_jack [private]
```

13.24.4.17 m_jack_stop_tick

```
midipulse seq64::jack_assistant::m_jack_stop_tick [private]
```

Repositions the transport marker.

13.24.4.18 m_follow_transport

```
bool seq64::jack_assistant::m_follow_transport [private]
```

13.24.4.19 m_ppqn

```
int seq64::jack_assistant::m_ppqn [private]
```

It is used for calculating ticks/beat (pulses/beat) and for setting the tick position.

13.24.4.20 m_beats_per_measure

```
int seq64::jack_assistant::m_beats_per_measure [private]
```

13.24.4.21 m_beat_width

```
int seq64::jack_assistant::m_beat_width [private]
```

13.24.4.22 m_beats_per_minute

```
midibpm seq64::jack_assistant::m_beats_per_minute [private]
```

13.25 seq64::jack_scratchpad Class Reference

Provide a temporary structure for passing data and results between a perform and jack_assistant object.

Data Fields

• double js_current_tick

Holds current location.

double js_total_tick

Current location ignoring L/R.

double js_clock_tick

Identical to js_total_tick.

bool js_jack_stopped

Flags perform::inner_stop().

bool js_dumping

Non-JACK playback in progress?

bool js_init_clock

We now have a good JACK lock.

bool js_looping

seqedit loop button is active.

bool js_playback_mode

Song mode (versus live mode).

· double js_ticks_converted

Keeps track of ...?

· double js_ticks_delta

Minor difference in tick.

• double js_ticks_converted_last

Keeps track of position?

long js_delta_tick_frac

More precision for seq24 0.9.3.

13.25.1 Detailed Description

The jack_assistant class already has access to the members of perform, but it needs access to and modification of "local" variables in perform::output_func(). This scratchpad is useful even if JACK support is not enabled.

13.25.2 Field Documentation

```
13.25.2.1 js_current_tick
double seq64::jack_scratchpad::js_current_tick
13.25.2.2 js_total_tick
double seq64::jack_scratchpad::js_total_tick
13.25.2.3 js_clock_tick
double seq64::jack_scratchpad::js_clock_tick
13.25.2.4 js_jack_stopped
bool seq64::jack_scratchpad::js_jack_stopped
13.25.2.5 js_dumping
bool seq64::jack_scratchpad::js_dumping
13.25.2.6 js_init_clock
bool seq64::jack_scratchpad::js_init_clock
```

13.25.2.7 js_looping

bool seq64::jack_scratchpad::js_looping

13.25.2.8 js_playback_mode

bool seq64::jack_scratchpad::js_playback_mode

13.25.2.9 js_ticks_converted

double seq64::jack_scratchpad::js_ticks_converted

13.25.2.10 js ticks delta

double seq64::jack_scratchpad::js_ticks_delta

13.25.2.11 js_ticks_converted_last

 $\verb|double seq64::jack_scratchpad::js_ticks_converted_last|\\$

13.25.2.12 js_delta_tick_frac

 $\verb|long seq64::jack_scratchpad::js_delta_tick_frac|\\$

13.26 seq64::jack_status_pair_t Struct Reference

Provides an internal type to make it easier to display a specific and accurate human-readable message when a JACK operation fails.

Data Fields

- unsigned jf_bit
 - Holds one of the bit-values from jack_status_t, which is defined as an "enum JackStatus" type.
- std::string jf_meaning

Holds a textual description of the corresponding status bit.

13.26.1 Field Documentation

```
13.26.1.1 jf_bit
```

```
unsigned seq64::jack_status_pair_t::jf_bit
```

13.26.1.2 jf_meaning

```
std::string seq64::jack_status_pair_t::jf_meaning
```

13.27 seq64::keybindentry Class Reference

Class for management of application key-bindings.

Inherits Entry.

Public Member Functions

• keybindentry (type t, unsigned int *location_to_write=nullptr, perform *p=nullptr, long s=0)

This constructor initializes the member with values dependent on the value type provided in the first parameter.

void set (unsigned int val)

Gets the key name from the integer value; if there is one, then it is printed into a temporary buffer, otherwise the value is printed into that buffer as is.

virtual bool on_key_press_event (GdkEventKey *event)

Handles a key press by calling set() with the event's key value.

Private Types

enum type { location, events, groups }

Provides the type of keybindings that can be made.

Private Attributes

unsigned int * m_key

Points to the value of the key that is part of this key-binding.

• type m_type

Stores the type of key-binding.

• perform * m_perf

Stores an optional pointer to a perform object.

• long m_slot

Provides an index into a set of group-keys or event-keys.

Friends

class options

13.27.1 Member Enumeration Documentation

13.27.1.1 type

```
enum seq64::keybindentry::type [private]
```

Enumerator

location	Used for handling a keystroke made while a keyboard-options field is active, for selecting a key via the keyboard, and binding to pattern/sequence boxes, we think. It is used in the options class to associate a key with the binding.
events	Used for binding to events.
groups	Used for binding to groups.

13.27.2 Constructor & Destructor Documentation

13.27.2.1 keybindentry()

Usage In options, a pointer to a new key-binding entry is managed by calling keybindentry (keybindentry ::location, &perf→keyname).

Parameters

t	Provides the type of key-binding: location, events, or groups.
location_to_write	The location that holds the value of the key associated with the key-binding. The default value of this parameter is the null pointer.
p	Points to the performance object used with this key-binding. The default value of this parameter is the null pointer.
s	Provides the slot value for this key-binding. The default value of this parameter is zero.

13.27.3 Member Function Documentation

Then we call set_text(buf). The set_width_char() function is then called.

13.27.3.2 on_key_press_event()

This value is used to set the event or key depending on the value of m_type.

Parameters

<i>event</i> Provides the key-press event.
--

Returns

Returns the result of the call to Entry::on_key_press_event().

13.27.4 Friends And Related Function Documentation

13.27.4.1 options

```
friend class options [friend]
```

13.27.5 Field Documentation

```
13.27.5.1 m_key
```

```
unsigned int* seq64::keybindentry::m_key [private]
```

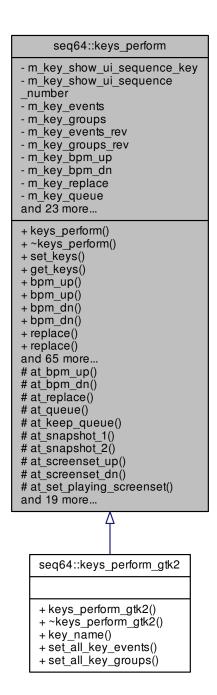
Not yet sure by the address of this key value is needed. It can be a null pointer, as well.

13.27.5.2 m_type type seq64::keybindentry::m_type [private] 13.27.5.3 m_perf perform* seq64::keybindentry::m_perf [private] 13.27.5.4 m_slot long seq64::keybindentry::m_slot [private] (This item should be changed to unsigned int, though.)

13.28 seq64::keys_perform Class Reference

This class supports the performance mode.

Inheritance diagram for seq64::keys_perform:



Public Member Functions

• keys_perform ()

This construction initializes a vast number of member variables, some of them public!

virtual ~keys_perform ()

The destructor sets some running flags to false, signals this condition, then joins the input and output threads if the were launched.

void set_keys (const keys_perform_transfer &kpt)

Copies fields from the transfer structure in this object.

void get_keys (keys_perform_transfer &kpt)

Copies fields from this object into the transfer structure.

unsigned int bpm_up () const

'Getter' function for member m_key_bpm_up

void bpm up (unsigned int x)

'Setter' function for member m_key_bpm_up

unsigned int bpm_dn () const

'Getter' function for member m_key_bpm_dn

void bpm dn (unsigned int x)

'Setter' function for member m_key_bpm_dn

unsigned int replace () const

'Getter' function for member m_key_replace

void replace (unsigned int x)

'Setter' function for member m_key_replace

· unsigned int queue () const

'Getter' function for member m_key_queue

void queue (unsigned int x)

'Setter' function for member m_key_queue

• unsigned int keep_queue () const

'Getter' function for member m_key_keep_queue

• void keep_queue (unsigned int x)

'Setter' function for member m_key_keep_queue

unsigned int snapshot_1 () const

'Getter' function for member m_key_snapshot_1

void snapshot_1 (unsigned int x)

'Setter' function for member m_key_snapshot_1

• unsigned int snapshot_2 () const

'Getter' function for member m_key_snapshot_2

void snapshot_2 (unsigned int x)

'Setter' function for member m_key_snapshot_2

· unsigned int screenset_up () const

'Getter' function for member m_key_screenset_up

void screenset_up (unsigned int x)

'Setter' function for member m_key_screenset_up

unsigned int screenset_dn () const

'Getter' function for member m_key_screenset_dn

void screenset_dn (unsigned int x)

'Setter' function for member m_key_screenset_dn

• unsigned int set_playing_screenset () const

'Getter' function for member m_key_playing_screenset

void set_playing_screenset (unsigned int x)

'Setter' function for member m_key_playing_screenset

• unsigned int group_on () const

'Getter' function for member m_key_group_on

void group_on (unsigned int x)

'Setter' function for member m_key_group_on

• unsigned int group off () const

'Getter' function for member m_key_group_off

void group_off (unsigned int x)

'Setter' function for member m_key_group_off

· unsigned int group_learn () const

'Getter' function for member m key group learn

void group learn (unsigned int x)

'Setter' function for member m_key_group_learn

• unsigned int start () const

'Getter' function for member m_key_start

void start (unsigned int x)

'Setter' function for member m_key_start

• unsigned int pause () const

'Getter' function for member m_key_pause

void pause (unsigned int x)

'Setter' function for member m_key_pause

• unsigned int pattern_edit () const

'Getter' function for member m_key_pattern_edit

void pattern_edit (unsigned int x)

'Setter' function for member m_key_pattern_edit

unsigned int event_edit () const

'Getter' function for member m_key_event_edit

void event_edit (unsigned int x)

'Setter' function for member m_key_event_edit

• unsigned int stop () const

'Getter' function for member m_key_stop

void stop (unsigned int x)

'Setter' function for member m key stop

- unsigned int song mode () const
- void song mode (unsigned int key)
- unsigned int menu_mode () const
- void menu mode (unsigned int key)
- unsigned int follow_transport () const
- void follow_transport (unsigned int key)
- unsigned int fast_forward () const
- void fast_forward (unsigned int key)unsigned int rewind () const
- void rewind (unsigned int key)
- unsigned int pointer position () const
- void pointer position (unsigned int key)
- unsigned int toggle_mutes () const
- void toggle_mutes (unsigned int key)
- unsigned int toggle_jack () const
- · void toggle_jack (unsigned int key)
- unsigned int tap_bpm () const
- void tap_bpm (unsigned int key)
- bool show_ui_sequence_key () const

'Getter' function for member m_key_show_ui_sequency_key

· void show ui sequence key (bool flag)

'Setter' function for member m_key_show_ui_sequency_key

bool show_ui_sequence_number () const

'Getter' function for member m_key_show_ui_sequence_number

void show_ui_sequence_number (bool flag)

'Setter' function for member m_key_show_ui_sequence_key

• SlotMap & get_key_events ()

'Getter' function for member m_key_events

SlotMap & get_key_groups ()

'Getter' function for member m_key_groups

RevSlotMap & get_key_events_rev ()

'Getter' function for member m_key_events_rev

RevSlotMap & get_key_groups_rev ()

'Getter' function for member m_key_groups_rev

unsigned int lookup keyevent key (long segnum)

'Getter' function for member m_key_events_rev[seqnum];

long lookup_keyevent_seq (unsigned int keycode)

'Getter' function for member m key events rev[keycode];

unsigned int lookup_keygroup_key (long groupnum)

'Getter' function for member m_key_events_rev[groupnum];

long lookup_keygroup_group (unsigned int keycode)

'Getter' function for member m_key_events_rev[keycode];

· virtual std::string key_name (unsigned int key) const

Obtains the name of the key.

virtual void set_all_key_events ()

Provides base class functionality.

virtual void set all key groups ()

Provides base class functionality.

void set key event (unsigned int keycode, long sequence slot)

At construction time, this function sets up one keycode and one event slot.

void set_key_group (unsigned int keycode, long group_slot)

At construction time, this function sets up one keycode and one group slot.

Protected Types

typedef std::map< unsigned int, long > SlotMap

This typedef defines a map in which the key is the keycode, that is, the integer value of a keystroke, and the value is the pattern/sequence number or slot.

typedef std::map< long, unsigned int > RevSlotMap

This typedef is like SlotMap, but used for lookup in the other direction.

Protected Member Functions

unsigned int * at_bpm_up ()

The following are tricky ways to get at address of the key and group operation values so that we don't directly expose the members to manipulation.

• unsigned int * at bpm dn ()

'Getter' function for member m_key_bpm_dn

unsigned int * at_replace ()

'Getter' function for member m_key_replace

• unsigned int * at_queue ()

'Getter' function for member m_key_queue

unsigned int * at_keep_queue ()

'Getter' function for member m_key_keep_queue

• unsigned int * at snapshot 1 ()

'Getter' function for member m_key_snapshot_1

• unsigned int * at_snapshot_2 ()

```
'Getter' function for member m_key_snapshot_2
unsigned int * at_screenset_up ()
      'Getter' function for member m_key_screenset_up
unsigned int * at_screenset_dn ()
      'Getter' function for member m_key_screenset_dn

    unsigned int * at_set_playing_screenset ()

      'Getter' function for member m_key_playing_screenset
unsigned int * at_group_on ()
      'Getter' function for member m_key_group_on
unsigned int * at_group_off ()
      'Getter' function for member m_key_group_off

    unsigned int * at group learn ()

      'Getter' function for member m_key_group_learn
• unsigned int * at start ()
      'Getter' function for member m key start
• unsigned int * at_pause ()
      'Getter' function for member m_key_pause
unsigned int * at_song_mode ()
      'Getter' function for member m_key_song_mode
unsigned int * at_toggle_jack ()
      'Getter' function for member m_key_toggle_jack

    unsigned int * at_menu_mode ()

      'Getter' function for member m_key_menu_mode

    unsigned int * at follow transport ()

      'Getter' function for member m_key_follow_transport
unsigned int * at_fast_forward ()
      'Getter' function for member m_key_fast_forward
unsigned int * at_rewind ()
      'Getter' function for member m_key_rewind
unsigned int * at_pointer_position ()
      'Getter' function for member m_key_pointer_position
• unsigned int * at_toggle_mutes ()
      'Getter' function for member m_key_toggle_mutes
unsigned int * at_tap_bpm ()
      'Getter' function for member m_key_tap_bpm
unsigned int * at_pattern_edit ()
      'Getter' function for member m key pattern edit
unsigned int * at_event_edit ()
      'Getter' function for member m key event edit
unsigned int * at_stop ()
      'Getter' function for member m_key_stop
bool * at_show_ui_sequence_key ()
      'Getter' function for member m_key_show_ui_sequence_key
bool * at_show_ui_sequence_number ()
      'Getter' function for member m_key_show_ui_sequence_number
```

Private Attributes

bool m_key_show_ui_sequence_key

If set, shows the shortcut-keys on each filled pattern slot in the main window.

• bool m_key_show_ui_sequence_number

If set, shows the sequence number on each filled pattern and empty pattern slot in the main window.

• SlotMap m_key_events

Holds the mapping of keys to the pattern slots.

SlotMap m_key_groups

Holds the mapping of keys to the mute groups.

RevSlotMap m_key_events_rev

Holds the reverse mapping of the pattern slots to the keys.

RevSlotMap m_key_groups_rev

Holds the reverse mapping of the mute groups to the keys.

unsigned int m_key_bpm_up

Provides key assignments for some key sequencer features.

• unsigned int m_key_bpm_dn

BPM down, semicolon.

• unsigned int m_key_replace

Replace, Ctrl-L.

• unsigned int m_key_queue

Queue, Ctrl-R.

• unsigned int m_key_keep_queue

Keep queue, backslash.

unsigned int m_key_snapshot_1

Snapshot 1, Alt-L.

• unsigned int m_key_snapshot_2

Snapshot 1, Alt-R.

unsigned int m_key_screenset_up

Set up, Right-].

· unsigned int m key screenset dn

Set down, Left-[.

• unsigned int m_key_set_playing_screenset

Set set, Home key.

unsigned int m_key_group_on

Group on, igrave key.

• unsigned int m_key_group_off

Group off, apostrophe!

• unsigned int m_key_group_learn

Group learn, Insert.

• unsigned int m_key_start

Start play, Space key.

unsigned int m_key_pause

Pause play, Period.

• unsigned int m_key_song_mode

Song versus Live mode.

unsigned int m_key_toggle_jack

Toggle JACK connect.

· unsigned int m key menu mode

Menu enabled/disabled.

• unsigned int m_key_follow_transport

Toggle following JACK.

• unsigned int m_key_rewind

Start rewind.

• unsigned int m_key_fast_forward

Start fast-forward.

unsigned int m_key_pointer_position

Set progress to mouse.

unsigned int m_key_toggle_mutes

Toggle all patterns.

• unsigned int m_key_tap_bpm

To tap out the BPM.

• unsigned int m_key_pattern_edit

Show pattern editor.

• unsigned int m_key_event_edit

Show event editor.

• unsigned int m_key_stop

Stop play, Escape.

Friends

- class options
- class perform
- · class optionsfile

13.28.1 Detailed Description

It provides a way a mapping keystrokes to sequencer actions and song settings.

13.28.2 Member Typedef Documentation

13.28.2.1 SlotMap

```
typedef std::map<unsigned int, long> seq64::keys_perform::SlotMap [protected]
```

13.28.2.2 RevSlotMap

```
typedef std::map<long, unsigned int> seq64::keys_perform::RevSlotMap [protected]
```

13.28.3 Constructor & Destructor Documentation

13.28.3.1 keys_perform()

```
seq64::keys_perform::keys_perform ( )
```

13.28.3.2 ∼keys_perform()

```
seq64::keys_perform::~keys_perform ( ) [virtual]
```

Finally, any active patterns/sequences are deleted.

13.28.4 Member Function Documentation

```
13.28.4.1 set_keys()
```

This structure holds all of the key settings from the File / Options / Keyboard tab dialog.

Parameters

kpt

The structure that holds the values of the keys to be used for various purposes in controlling a performance live.

13.28.4.2 get_keys()

Parameters

kpt

The structure that holds the values of the keys to be used for various purposes in controlling a performance live.

```
13.28.4.3 bpm_up() [1/2]
```

```
unsigned int seq64::keys_perform::bpm_up ( ) const [inline]
```

```
13.28.4.4 bpm_up() [2/2]
void seq64::keys_perform::bpm_up (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
13.28.4.5 bpm_dn() [1/2]
unsigned int seq64::keys_perform::bpm_dn ( ) const [inline]
13.28.4.6 bpm_dn() [2/2]
void seq64::keys_perform::bpm_dn (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
13.28.4.7 replace() [1/2]
unsigned int seq64::keys_perform::replace ( ) const [inline]
13.28.4.8 replace() [2/2]
void seq64::keys_perform::replace (
             unsigned int x ) [inline]
Parameters
```

The key value to assign to the operation.

```
13.28.4.9 queue() [1/2]
unsigned int seq64::keys_perform::queue ( ) const [inline]
13.28.4.10 queue() [2/2]
void seq64::keys_perform::queue (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
13.28.4.11 keep_queue() [1/2]
unsigned int seq64::keys_perform::keep_queue ( ) const [inline]
13.28.4.12 keep_queue() [2/2]
void seq64::keys_perform::keep_queue (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
13.28.4.13 snapshot_1() [1/2]
unsigned int seq64::keys_perform::snapshot_1 ( ) const [inline]
13.28.4.14 snapshot_1() [2/2]
void seq64::keys_perform::snapshot_1 (
             unsigned int x ) [inline]
```

```
Parameters
```

```
x The key value to assign to the operation.
```

```
13.28.4.17 screenset_up() [1/2]
```

```
unsigned int seq64::keys_perform::screenset_up ( ) const [inline]
```

```
13.28.4.18 screenset_up() [2/2]
```

The key value to assign to the operation.

Parameters

x The key value to assign to the operation.

```
13.28.4.19 screenset_dn() [1/2]
```

```
unsigned int seq64::keys_perform::screenset_dn ( ) const [inline]
```

```
13.28.4.20 screenset_dn() [2/2]
void seq64::keys_perform::screenset_dn (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
13.28.4.21 set_playing_screenset() [1/2]
unsigned int seq64::keys_perform::set_playing_screenset ( ) const [inline]
13.28.4.22 set_playing_screenset() [2/2]
void seq64::keys_perform::set_playing_screenset (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
13.28.4.23 group_on() [1/2]
unsigned int seq64::keys_perform::group_on ( ) const [inline]
13.28.4.24 group_on() [2/2]
void seq64::keys_perform::group_on (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
```

```
13.28.4.25 group_off() [1/2]
unsigned int seq64::keys_perform::group_off ( ) const [inline]
13.28.4.26 group_off() [2/2]
void seq64::keys_perform::group_off (
             unsigned int x ) [inline]
Parameters
    The key value to assign to the operation.
13.28.4.27 group_learn() [1/2]
unsigned int seq64::keys_perform::group_learn ( ) const [inline]
13.28.4.28 group_learn() [2/2]
void seq64::keys_perform::group_learn (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
13.28.4.29 start() [1/2]
unsigned int seq64::keys_perform::start ( ) const [inline]
13.28.4.30 start() [2/2]
void seq64::keys_perform::start (
             unsigned int x ) [inline]
```

Parameters

x The key value to assign to the operation.

Parameters

x The key value to assign to the operation.

```
13.28.4.35 event_edit() [1/2]
unsigned int seq64::keys_perform::event_edit ( ) const [inline]
```

```
13.28.4.36 event_edit() [2/2]
void seq64::keys_perform::event_edit (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
13.28.4.37 stop() [1/2]
unsigned int seq64::keys_perform::stop ( ) const [inline]
13.28.4.38 stop() [2/2]
void seq64::keys_perform::stop (
             unsigned int x ) [inline]
Parameters
     The key value to assign to the operation.
13.28.4.39 song_mode() [1/2]
unsigned int seq64::keys_perform::song_mode ( ) const [inline]
13.28.4.40 song_mode() [2/2]
void seq64::keys_perform::song_mode (
             unsigned int key ) [inline]
13.28.4.41 menu_mode() [1/2]
unsigned int seq64::keys_perform::menu_mode ( ) const [inline]
```

```
13.28.4.42 menu_mode() [2/2]
void seq64::keys_perform::menu_mode (
             unsigned int key ) [inline]
13.28.4.43 follow_transport() [1/2]
unsigned int seq64::keys_perform::follow_transport ( ) const [inline]
13.28.4.44 follow_transport() [2/2]
void seq64::keys_perform::follow_transport (
             unsigned int key ) [inline]
13.28.4.45 fast_forward() [1/2]
unsigned int seq64::keys_perform::fast_forward ( ) const [inline]
13.28.4.46 fast_forward() [2/2]
void seq64::keys_perform::fast_forward (
             unsigned int key ) [inline]
13.28.4.47 rewind() [1/2]
unsigned int seq64::keys_perform::rewind ( ) const [inline]
13.28.4.48 rewind() [2/2]
void seq64::keys_perform::rewind (
             unsigned int key ) [inline]
```

```
13.28.4.49 pointer_position() [1/2]
unsigned int seq64::keys_perform::pointer_position ( ) const [inline]
13.28.4.50 pointer_position() [2/2]
void seq64::keys_perform::pointer_position (
             unsigned int key ) [inline]
13.28.4.51 toggle_mutes() [1/2]
unsigned int seq64::keys_perform::toggle_mutes ( ) const [inline]
13.28.4.52 toggle_mutes() [2/2]
void seq64::keys_perform::toggle_mutes (
             unsigned int key ) [inline]
13.28.4.53 toggle_jack() [1/2]
unsigned int seq64::keys_perform::toggle_jack ( ) const [inline]
13.28.4.54 toggle_jack() [2/2]
void seq64::keys_perform::toggle_jack (
             unsigned int key ) [inline]
13.28.4.55 tap_bpm() [1/2]
unsigned int seq64::keys_perform::tap_bpm ( ) const [inline]
```

```
13.28.4.56 tap_bpm() [2/2]
void seq64::keys_perform::tap_bpm (
             unsigned int key ) [inline]
13.28.4.57 show_ui_sequence_key() [1/2]
bool seq64::keys_perform::show_ui_sequence_key ( ) const [inline]
Used in mainwid, options, optionsfile, userfile, and perform.
13.28.4.58 show_ui_sequence_key() [2/2]
void seq64::keys_perform::show_ui_sequence_key (
             bool flag ) [inline]
Parameters
 flag
       The flag for showing the sequence key characters in each pattern slot.
13.28.4.59 show_ui_sequence_number() [1/2]
bool seq64::keys_perform::show_ui_sequence_number ( ) const [inline]
Used in mainwid, options, optionsfile, userfile, and perform.
13.28.4.60 show_ui_sequence_number() [2/2]
void seq64::keys_perform::show_ui_sequence_number (
             bool flag ) [inline]
Parameters
 flag
       The flag for showing the sequence number in each pattern slot.
13.28.4.61 get_key_events()
SlotMap& seq64::keys_perform::get_key_events ( ) [inline]
```

```
13.28.4.62 get_key_groups()
SlotMap& seq64::keys_perform::get_key_groups ( ) [inline]
13.28.4.63 get_key_events_rev()
RevSlotMap& seq64::keys_perform::get_key_events_rev ( ) [inline]
13.28.4.64 get_key_groups_rev()
RevSlotMap& seq64::keys_perform::get_key_groups_rev ( ) [inline]
13.28.4.65 lookup_keyevent_key()
unsigned int seq64::keys_perform::lookup_keyevent_key (
             long segnum ) [inline]
Parameters
 segnum
           Provides the sequence number to look up in the reverse key map for patterns/sequences. If the
```

count for this value is 0, then a question mark character is returned. Not checked for maximum!

13.28.4.66 lookup_keyevent_seq()

```
long seq64::keys_perform::lookup_keyevent_seq (
            unsigned int keycode ) [inline]
```

Parameters

keycode

Provides the keycode to look up in the (forward) key map for patterns/sequences. If the count for this value is 0, then a 0 is returned.

13.28.4.67 lookup_keygroup_key()

```
unsigned int seq64::keys_perform::lookup_keygroup_key (
            long groupnum ) [inline]
```

Parameters

Provides the group number to look up in the reverse key map for groups. If the count for this value is 0, then a question mark character is returned.

13.28.4.68 lookup_keygroup_group()

Parameters

keycode

Provides the sequence number to look up in the reverse key map for groups. If the count for this value is 0, then a 0 is returned.

13.28.4.69 key_name()

In gtkmm, this is done via the gdk_keyval_name() function. Here, in the base class, we just provide an easy-to-create string.

Parameters

key Provides the numeric value of the keystroke.

Returns

Returns the name of the key, in the format "Key 0xkkkk".

Reimplemented in seq64::keys_perform_gtk2.

```
13.28.4.70 set_all_key_events()
```

```
virtual void seq64::keys_perform::set_all_key_events ( ) [inline], [virtual]
```

Must be called by the derived-class's override of this function.

Reimplemented in seq64::keys_perform_gtk2.

```
13.28.4.71 set_all_key_groups()
```

```
virtual void seq64::keys_perform::set_all_key_groups ( ) [inline], [virtual]
```

Must be called by the derived-class's override of this function.

Reimplemented in seq64::keys perform gtk2.

13.28.4.72 set_key_event()

It is called 32 times, corresponding the pattern/sequence slots in the Patterns window.

Parameters

keycode	The key to be assigned.
sequence_slot	The perform event slot into which the keycode will be assigned.

13.28.4.73 set_key_group()

```
void seq64::keys_perform::set_key_group (
          unsigned int keycode,
          long group_slot )
```

It is called 32 times, corresponding the pattern/sequence slots in the Patterns window.

Parameters

keycode	The key to be assigned.
group_slot	The perform group slot into which the keycode will be assigned.

13.28.4.74 at_bpm_up()

```
unsigned int* seq64::keys_perform::at_bpm_up ( ) [inline], [protected]
```

They are used in the options module, and, for brevity, are accessed using the PREFKEY_ADDR() macro. 'Getter' function for member $m_{key_bpm_up}$

Address getter for the bpm_up operation.

```
13.28.4.75 at_bpm_dn()
unsigned int* seq64::keys_perform::at_bpm_dn ( ) [inline], [protected]
Address getter for the bpm_dn operation.
13.28.4.76 at_replace()
unsigned int* seq64::keys_perform::at_replace ( ) [inline], [protected]
Address getter for the replace operation.
13.28.4.77 at_queue()
unsigned int* seq64::keys_perform::at_queue ( ) [inline], [protected]
Address getter for the queue operation.
13.28.4.78 at_keep_queue()
unsigned int* seq64::keys_perform::at_keep_queue ( ) [inline], [protected]
Address getter for the keep_queue operation.
13.28.4.79 at_snapshot_1()
unsigned int* seq64::keys_perform::at_snapshot_1 ( ) [inline], [protected]
Address getter for the snapshot_1 operation.
13.28.4.80 at snapshot 2()
unsigned int* seq64::keys_perform::at_snapshot_2 ( ) [inline], [protected]
Address getter for the snapshot 2 operation.
13.28.4.81 at_screenset_up()
unsigned int* seq64::keys_perform::at_screenset_up ( ) [inline], [protected]
Address getter for the screenset_up operation.
13.28.4.82 at_screenset_dn()
unsigned int* seq64::keys_perform::at_screenset_dn ( ) [inline], [protected]
```

Address getter for the screenset_dn operation.

```
13.28.4.83 at_set_playing_screenset()
unsigned int* seq64::keys_perform::at_set_playing_screenset ( ) [inline], [protected]
Address getter for the set_playing_screenset operation.
13.28.4.84 at_group_on()
unsigned int* seq64::keys_perform::at_group_on ( ) [inline], [protected]
Address getter for the group_on operation.
13.28.4.85 at_group_off()
unsigned int* seq64::keys_perform::at_group_off ( ) [inline], [protected]
Address getter for the group_off operation.
13.28.4.86 at_group_learn()
unsigned int* seq64::keys_perform::at_group_learn ( ) [inline], [protected]
Address getter for the group_learn operation.
13.28.4.87 at_start()
unsigned int* seq64::keys\_perform::at\_start ( ) [inline], [protected]
Address getter for the start operation.
13.28.4.88 at_pause()
unsigned int* seq64::keys_perform::at_pause ( ) [inline], [protected]
Address getter for the pause operation.
13.28.4.89 at_song_mode()
unsigned int* seq64::keys_perform::at_song_mode ( ) [inline], [protected]
Address getter for the song-mode operation.
13.28.4.90 at_toggle_jack()
unsigned int* seq64::keys_perform::at_toggle_jack ( ) [inline], [protected]
```

Address getter for the toggle-jack operation.

```
13.28.4.91 at_menu_mode()
unsigned int* seq64::keys_perform::at_menu_mode ( ) [inline], [protected]
Address getter for the menu-mode operation.
13.28.4.92 at_follow_transport()
unsigned int* seq64::keys_perform::at_follow_transport ( ) [inline], [protected]
Address getter for the follow-transport operation.
13.28.4.93 at_fast_forward()
unsigned int* seq64::keys_perform::at_fast_forward ( ) [inline], [protected]
Address getter for the fast-forward operation.
13.28.4.94 at_rewind()
unsigned int* seq64::keys_perform::at_rewind ( ) [inline], [protected]
Address getter for the rewind operation.
13.28.4.95 at_pointer_position()
unsigned int* seq64::keys_perform::at_pointer_position ( ) [inline], [protected]
Address getter for the pointer operation.
13.28.4.96 at_toggle_mutes()
unsigned int* seq64::keys_perform::at_toggle_mutes ( ) [inline], [protected]
Address getter for the toggle-mutes operation.
13.28.4.97 at_tap_bpm()
unsigned int* seq64::keys_perform::at_tap_bpm ( ) [inline], [protected]
Address getter for the tap_bpm operation.
13.28.4.98 at_pattern_edit()
unsigned int* seq64::keys_perform::at_pattern_edit ( ) [inline], [protected]
```

Address getter for the pattern-edit operation.

```
13.28.4.99 at_event_edit()
unsigned int* seq64::keys_perform::at_event_edit ( ) [inline], [protected]
Address getter for the event-edit operation.
13.28.4.100 at_stop()
unsigned int* seq64::keys_perform::at_stop ( ) [inline], [protected]
Address getter for the stop operation.
13.28.4.101 at_show_ui_sequence_key()
bool* seq64::keys_perform::at_show_ui_sequence_key ( ) [inline], [protected]
{\tt Address\ getter\ for\ the\ show\_ui\_sequence\_key\ value}.
13.28.4.102 at_show_ui_sequence_number()
bool* seq64::keys_perform::at_show_ui_sequence_number ( ) [inline], [protected]
Address getter for the show_ui_sequence_number value.
13.28.5 Friends And Related Function Documentation
13.28.5.1 options
friend class options [friend]
13.28.5.2 perform
friend class perform [friend]
13.28.5.3 optionsfile
friend class optionsfile [friend]
```

13.28.6 Field Documentation

```
bool seq64::keys_perform::m_key_show_ui_sequence_key [private]
```

13.28.6.2 m_key_show_ui_sequence_number

13.28.6.1 m_key_show_ui_sequence_key

```
bool seq64::keys_perform::m_key_show_ui_sequence_number [private]
```

Also shows the sequence number as part of the sequence name in the performance window (song editor). Always disabled in legacy mode.

```
13.28.6.3 m_key_events
```

```
SlotMap seq64::keys_perform::m_key_events [private]
```

Do not access directly, use the set/lookup functions declared below.

```
13.28.6.4 m_key_groups
```

```
SlotMap seq64::keys_perform::m_key_groups [private]
```

Do not access directly, use the set/lookup functions declared below.

```
13.28.6.5 m_key_events_rev
```

```
RevSlotMap seq64::keys_perform::m_key_events_rev [private]
```

Do not access directly, use the set/lookup functions declared below.

```
13.28.6.6 m_key_groups_rev
```

```
RevSlotMap seq64::keys_perform::m_key_groups_rev [private]
```

Do not access directly, use the set/lookup functions declared below.

```
13.28.6.7 m_key_bpm_up
```

```
unsigned int seq64::keys_perform::m_key_bpm_up [private]
```

Used in mainwnd, options, optionsfile, perfedit, seqroll, userfile, and perform.

We could instead use the keys_perform_transfer structure instead of all these individual members.BPM up, apostrophe!!!

```
13.28.6.8 m_key_bpm_dn
```

unsigned int seq64::keys_perform::m_key_bpm_dn [private]

13.28.6.9 m_key_replace

unsigned int seq64::keys_perform::m_key_replace [private]

13.28.6.10 m_key_queue

unsigned int seq64::keys_perform::m_key_queue [private]

13.28.6.11 m_key_keep_queue

unsigned int seq64::keys_perform::m_key_keep_queue [private]

13.28.6.12 m_key_snapshot_1

 $unsigned \ int \ seq64{::} keys_perform{::} m_key_snapshot_1 \quad [private] \\$

13.28.6.13 m_key_snapshot_2

unsigned int seq64::keys_perform::m_key_snapshot_2 [private]

13.28.6.14 m_key_screenset_up

 ${\tt unsigned\ int\ seq64::keys_perform::m_key_screenset_up\quad [private]}$

13.28.6.15 m_key_screenset_dn

unsigned int seq64::keys_perform::m_key_screenset_dn [private]

13.28.6.16 m_key_set_playing_screenset unsigned int seq64::keys_perform::m_key_set_playing_screenset [private]

13.28.6.17 m_key_group_on

unsigned int seq64::keys_perform::m_key_group_on [private]

13.28.6.18 m_key_group_off

unsigned int seq64::keys_perform::m_key_group_off [private]

13.28.6.19 m_key_group_learn

unsigned int seq64::keys_perform::m_key_group_learn [private]

13.28.6.20 m_key_start

unsigned int seq64::keys_perform::m_key_start [private]

13.28.6.21 m_key_pause

unsigned int seq64::keys_perform::m_key_pause [private]

13.28.6.22 m_key_song_mode

unsigned int seq64::keys_perform::m_key_song_mode [private]

13.28.6.23 m_key_toggle_jack

unsigned int seq64::keys_perform::m_key_toggle_jack [private]

```
13.28.6.24 m_key_menu_mode
```

unsigned int seq64::keys_perform::m_key_menu_mode [private]

13.28.6.25 m_key_follow_transport

unsigned int seq64::keys_perform::m_key_follow_transport [private]

13.28.6.26 m_key_rewind

unsigned int seq64::keys_perform::m_key_rewind [private]

13.28.6.27 m_key_fast_forward

unsigned int seq64::keys_perform::m_key_fast_forward [private]

13.28.6.28 m_key_pointer_position

unsigned int seq64::keys_perform::m_key_pointer_position [private]

13.28.6.29 m_key_toggle_mutes

 ${\tt unsigned\ int\ seq64::keys_perform::m_key_toggle_mutes} \quad {\tt [private]}$

13.28.6.30 m_key_tap_bpm

unsigned int seq64::keys_perform::m_key_tap_bpm [private]

13.28.6.31 m_key_pattern_edit

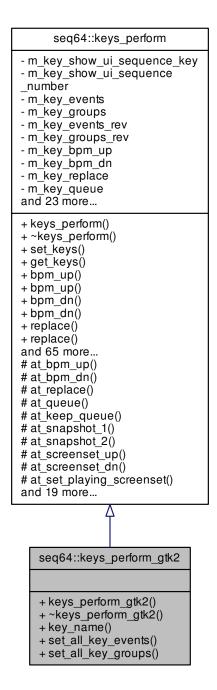
unsigned int seq64::keys_perform::m_key_pattern_edit [private]



This class supports the performance mode.

Generated by Doxygen

Inheritance diagram for seq64::keys_perform_gtk2:



Public Member Functions

keys_perform_gtk2 ()

This construction initializes a vast number of member variables, some of them public!

virtual ~keys_perform_gtk2 ()

A rote virtual destructor.

• virtual std::string key_name (unsigned int key) const

```
virtual void set_all_key_events ()
```

Sets up the keys for arming/unmuting events in the Gtk-2 environment.

• virtual void set_all_key_groups ()

Sets up the keys for group events in the Gtk-2 environment.

Additional Inherited Members

13.29.1 Detailed Description

It has way too many data members, many of the public. Might be ripe for refactoring.

13.29.2 Constructor & Destructor Documentation

```
13.29.2.1 keys_perform_gtk2()
```

```
seq64::keys_perform_gtk2::keys_perform_gtk2 ( )
```

13.29.2.2 \sim keys_perform_gtk2()

```
seq64::keys_perform_gtk2::~keys_perform_gtk2 ( ) [virtual]
```

No action.

13.29.3 Member Function Documentation

```
13.29.3.1 key_name()
```

Reimplemented from seq64::keys perform.

```
13.29.3.2 set_all_key_events()
```

```
void seq64::keys_perform_gtk2::set_all_key_events ( ) [virtual]
```

The base-class function call makes sure the the related lists are cleared before rebuilding them here.

Reimplemented from seq64::keys perform.

```
13.29.3.3 set_all_key_groups()
```

```
void seq64::keys_perform_gtk2::set_all_key_groups ( ) [virtual]
```

The base-class function call makes sure the the related lists are cleared before rebuilding them here.

Reimplemented from seq64::keys_perform.

13.30 seq64::keys_perform_transfer Struct Reference

Provides a data-transfer structure to make it easier to fill in a keys_perform object's members using sscanf().

Data Fields

- unsigned int kpt_bpm_up
- unsigned int kpt_bpm_dn
- unsigned int kpt_screenset_up
- unsigned int kpt_screenset_dn
- unsigned int kpt_set_playing_screenset
- unsigned int kpt_group_on
- unsigned int kpt_group_off
- unsigned int kpt_group_learn
- unsigned int kpt_replace
- unsigned int kpt_queue
- unsigned int kpt keep queue
- unsigned int kpt_snapshot_1
- unsigned int kpt_snapshot_2
- unsigned int kpt_start
- unsigned int kpt_stop
- bool kpt_show_ui_sequence_key
- bool kpt_show_ui_sequence_number
- unsigned int kpt_pattern_edit
- · unsigned int kpt event edit
- · unsigned int kpt tap bpm
- unsigned int kpt_pause
- unsigned int kpt_song_mode
- unsigned int kpt_toggle_jack
- unsigned int kpt_menu_mode
- unsigned int kpt_follow_transport
- unsigned int kpt_fast_forward
- · unsigned int kpt_rewind
- unsigned int kpt_pointer_position
- unsigned int kpt_toggle_mutes

13.30.1 Field Documentation

13.30.1.1 kpt_bpm_up unsigned int seq64::keys_perform_transfer::kpt_bpm_up 13.30.1.2 kpt_bpm_dn unsigned int seq64::keys_perform_transfer::kpt_bpm_dn 13.30.1.3 kpt_screenset_up unsigned int seq64::keys_perform_transfer::kpt_screenset_up 13.30.1.4 kpt_screenset_dn unsigned int $seq64::keys_perform_transfer::kpt_screenset_dn$ 13.30.1.5 kpt_set_playing_screenset unsigned int seq64::keys_perform_transfer::kpt_set_playing_screenset 13.30.1.6 kpt_group_on unsigned int seq64::keys_perform_transfer::kpt_group_on

unsigned int seq64::keys_perform_transfer::kpt_group_off

13.30.1.7 kpt_group_off

13.30.1.8 kpt_group_learn

unsigned int seq64::keys_perform_transfer::kpt_group_learn

13.30.1.9 kpt_replace

unsigned int seq64::keys_perform_transfer::kpt_replace

13.30.1.10 kpt_queue

unsigned int seq64::keys_perform_transfer::kpt_queue

13.30.1.11 kpt_keep_queue

unsigned int seq64::keys_perform_transfer::kpt_keep_queue

13.30.1.12 kpt_snapshot_1

 $unsigned\ int\ seq64{::} keys_perform_transfer{::} kpt_snapshot_1$

13.30.1.13 kpt_snapshot_2

unsigned int seq64::keys_perform_transfer::kpt_snapshot_2

13.30.1.14 kpt_start

unsigned int seq64::keys_perform_transfer::kpt_start

13.30.1.15 kpt_stop

unsigned int seq64::keys_perform_transfer::kpt_stop

13.30.1.16 kpt_show_ui_sequence_key

bool seq64::keys_perform_transfer::kpt_show_ui_sequence_key

13.30.1.17 kpt_show_ui_sequence_number

bool seq64::keys_perform_transfer::kpt_show_ui_sequence_number

13.30.1.18 kpt_pattern_edit

unsigned int seq64::keys_perform_transfer::kpt_pattern_edit

13.30.1.19 kpt_event_edit

unsigned int seq64::keys_perform_transfer::kpt_event_edit

13.30.1.20 kpt_tap_bpm

unsigned int seq64::keys_perform_transfer::kpt_tap_bpm

13.30.1.21 kpt_pause

unsigned int seq64::keys_perform_transfer::kpt_pause

13.30.1.22 kpt_song_mode

unsigned int seq64::keys_perform_transfer::kpt_song_mode

13.30.1.23 kpt_toggle_jack

unsigned int seq64::keys_perform_transfer::kpt_toggle_jack

13.30.1.24 kpt_menu_mode

unsigned int seq64::keys_perform_transfer::kpt_menu_mode

13.30.1.25 kpt_follow_transport

unsigned int seq64::keys_perform_transfer::kpt_follow_transport

13.30.1.26 kpt_fast_forward

unsigned int seq64::keys_perform_transfer::kpt_fast_forward

13.30.1.27 kpt_rewind

unsigned int seq64::keys_perform_transfer::kpt_rewind

13.30.1.28 kpt_pointer_position

unsigned int seq64::keys_perform_transfer::kpt_pointer_position

13.30.1.29 kpt_toggle_mutes

unsigned int seq64::keys_perform_transfer::kpt_toggle_mutes

13.31 seq64::keystroke Class Reference

Encapsulates any practical keystroke.

Public Member Functions

· keystroke ()

The default constructor for class keystroke.

The principal constructor.

keystroke (const keystroke &rhs)

Provides the rote copy constructor.

keystroke & operator= (const keystroke &rhs)

Provides the rote principal assignment operator.

• bool is_press () const

'Getter' function for member m_is_press

• bool is letter (unsigned int ch=SEQ64 KEYSTROKE BAD VALUE) const

'Getter' function for member m_key to test letters, handles ASCII only.

· bool is (unsigned int ch) const

Tests the key value to see if it matches the given character exactly (no case-insensitivity).

• bool is_delete () const

'Getter' function for member m_key to test for a delete-causing key.

• unsigned int key () const

'Getter' function for member m key

void shift_lock ()

If a lower-case letter, a number, or another character on the "main" part of the keyboard, shift the m_key value to upper-case or the character shifted on a standard American keyboard.

• seq_modifier_t modifier () const

'Getter' function for member m modifier

bool mod_control () const

'Getter' function for member m_modifier tested for Ctrl key.

• bool mod_control_shift () const

'Getter' function for member m_modifier tested for Ctrl and Shift key.

• bool mod_super () const

'Getter' function for member m_modifier tested for Mod4/Super/Windows key.

Private Attributes

· bool m_is_press

Determines if the key was a press or a release.

· unsigned int m key

The key that was pressed or released.

• seq_modifier_t m_modifier

The optional modifier value.

13.31.1 Detailed Description

Useful in passing more generic events to non-GUI classes.

13.31.2 Constructor & Destructor Documentation

Parameters

key	The keystroke number of the key that was pressed or released.
press	If true, the keystroke action was a press, otherwise it was a release.
modkey	The modifier key combination that was pressed, if any, in the form of a bit-mask, as defined in the gdk_basic_keys module. Common mask values are SEQ64_SHIFT_MASK, SEQ64_CONTROL_MASK, SEQ64_MOD1_MASK, and SEQ64_MOD4_MASK. If no modifier, this value is SEQ64_NO_MASK.

```
13.31.2.3 keystroke() [3/3]
seq64::keystroke::keystroke (
const keystroke & rhs)
```

Parameters

rhs The object to be copied.

13.31.3 Member Function Documentation

13.31.3.1 operator=()

Parameters

rhs The object to be assigned.

Returns

Returns the reference to the current object, for use in assignment chains.

13.31.3.2 is_press()

```
bool seq64::keystroke::is_press ( ) const [inline]
```

13.31.3.3 is_letter()

Parameters

ch An optional character to test as an ASCII letter.

Returns

If a character is not provided, true is returned if it is an upper or lower-case letter. Otherwise, true is returned if the m_key value matches the character case-insensitively.

Tricky Code

13.31.3.4 is()

Parameters

ch The character to be tested.

Returns

Returns true if m_key == ch.

13.31.3.5 is_delete()

```
bool seq64::keystroke::is_delete ( ) const [inline]
```

```
13.31.3.6 key()
```

```
unsigned int seq64::keystroke::key ( ) const [inline]
```

13.31.3.7 shift_lock()

```
void seq64::keystroke::shift_lock ( )
```

Currently also assumes the ASCII character set.

There's an oddity here: the shift of '2' is the '@' character, but seq24 seems to have treated it like the "" character. Some others were treated the same:

```
Key: 1 2 3 4 5 6 7 8 9 0
Shift: ! @ # $ % ^ & * ( )
Seq24: ! " # $ % & ' ( ) space
```

This function is meant to avoid using the Caps-Lock when picking a group-learn character in the group-learn mode.

13.31.3.8 modifier()

```
seq_modifier_t seq64::keystroke::modifier ( ) const [inline]
```

13.31.3.9 mod_control()

```
bool seq64::keystroke::mod_control ( ) const [inline]
```

13.31.3.10 mod_control_shift()

```
bool seq64::keystroke::mod_control_shift ( ) const [inline]
```

13.31.3.11 mod_super()

```
bool seq64::keystroke::mod_super ( ) const [inline]
```

13.31.4 Field Documentation

```
13.31.4.1 m_is_press
```

bool seq64::keystroke::m_is_press [private]

See the SEQ64 KEYSTROKE PRESS and SEQ64 KEYSTROKE RELEASE readability macros.

```
13.31.4.2 m_key
```

```
unsigned int seq64::keystroke::m_key [private]
```

Generally, the extended ASCII range (0 to 255) is supported. However, Gtk-2.x/3.x will generally support the full gamut of characters defined in the gdk_basic_keys.h module. We define minimum and maximum range macros for keystrokes that are a bit generous.

13.31.4.3 m_modifier

```
seq_modifier_t seq64::keystroke::m_modifier [private]
```

Note that SEQ64 NO MASK is our word for 0, meaning "no modifier".

13.32 seg64::lash Class Reference

This class supports LASH operations, if compiled with LASH support (i.e.

Public Member Functions

lash (perform &p, int argc, char **argv)

This constructor calls lash_extract(), using the command-line arguments, if SEQ64_LASH_SUPPORT is enabled.

· void set alsa client id (int id)

Make ourselves a LASH ALSA client.

• void start ()

Process any LASH events every 250 msec, which is an arbitrarily chosen interval.

• bool process events ()

Process LASH events.

Private Member Functions

• bool init ()

Initializes LASH support, if enabled.

void handle_event (lash_event_t *conf)

Handle a LASH event.

void handle_config (lash_config_t *conf)

Handle a LASH configuration item.

Private Attributes

• perform & m_perform

A hook into the single perform object in the application.

lash_client_t * m_client

Holds the client "handle" returned by the lash_init() function.

lash_args_t * m_lash_args

Holds the command-line arguments used by the lash_init() function.

bool m_is_lash_supported

Indicates if LASH support has been compiled into the library.

13.32.1 Detailed Description

SEQ64_LASH_SUPPORT is defined). All of the ifdef skeleton work is done in this class in such a way that any other part of the code can use this class whether or not lash support is actually built in; the functions will just do nothing.

13.32.2 Constructor & Destructor Documentation

13.32.2.1 lash()

We fixed the crazy usage of argc and argv here and in the client code in the seq24 module.

Parameters

р	The perform object that needs to implement LASH support.
argc	The number of command-line arguments.
argv	The command-line arguments.

13.32.3 Member Function Documentation

13.32.3.1 set_alsa_client_id()

/param id The ALSA client ID to be set.

```
13.32.3.2 start()
```

```
void seq64::lash::start ( )
```

13.32.3.3 process_events()

```
bool seq64::lash::process_events ( )
```

Returns

Always returns true.

13.32.3.4 init()

```
bool seq64::lash::init ( ) [private]
```

Returns

Returns true if the LASH subsystem was able to be initialized, and a LASH client representative (m_client) was allocated.

13.32.3.5 handle_event()

Parameters

ev Provides the event to be handled.

13.32.3.6 handle_config()

Currently incomplete.

conf Provides the configuration item to handle.

13.32.4 Field Documentation

13.32.4.1 m_perform

```
perform& seq64::lash::m_perform [private]
```

13.32.4.2 m_client

```
lash_client_t* seq64::lash::m_client [private]
```

13.32.4.3 m_lash_args

```
lash_args_t* seq64::lash::m_lash_args [private]
```

13.32.4.4 m_is_lash_supported

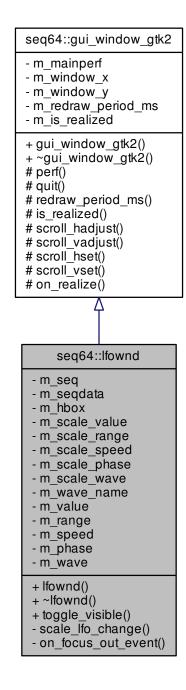
```
bool seq64::lash::m_is_lash_supported [private]
```

Is set to true if SEQ64_LASH_SUPPORT is defined. This variable is not used, but we will keep it around for the possibility of testing LASH support at run time.

13.33 seq64::Ifownd Class Reference

One LFO window class.

Inheritance diagram for seq64::lfownd:



Public Member Functions

• Ifownd (perform &p, sequence &seq, seqdata &sdata)

Constructs the LFO window.

- virtual ∼lfownd ()
- void toggle_visible ()

Private Member Functions

- void scale Ifo change ()
- bool on_focus_out_event (GdkEventFocus *p0)

Private Attributes

• sequence & m_seq

The sequence associated with this window.

• seqdata & m_seqdata

The seqdata associated with this window.

• Gtk::HBox * m_hbox

The main horizontal packing box.

• Gtk::VScale * m_scale_value

Vertical slider for value.

• Gtk::VScale * m_scale_range

Vertical slider for range.

• Gtk::VScale * m_scale_speed

Vertical slider for speed.

• Gtk::VScale * m_scale_phase

Vertical slider for phase.

• Gtk::VScale * m scale wave

Vertical slider for wave type.

• Gtk::Label * m_wave_name

Human readable name for wave type.

• double m_value

Value.

• double m_range

Range.

• double m_speed

Speed.

• double m_phase

Phase.

• wave_type_t m_wave

Wave type.

Additional Inherited Members

13.33.1 Detailed Description

Personally, it seems a bit of a odd duck to be included in Sequencer64, so we're thinking of a better way to manage the data managed by this window.

13.33.2 Constructor & Destructor Documentation

13.33.2.1 Ifownd()

Parameters

р	The performance object, which holds parameters necessary for manipulating events.
seq	The sequence/pattern that is to be affected by the LFO window. It holds the actual MIDI events being modified.
sdata	The data pane/panel of the pattern editor window representing the sequence. We need to tell it to redraw.

13.33.2.2 ∼lfownd()

```
seq64::lfownd::\sim lfownd ( ) [virtual]
```

13.33.3 Member Function Documentation

13.33.3.1 toggle_visible()

```
void seq64::lfownd::toggle_visible ( )
```

13.33.3.2 scale_lfo_change()

```
\label{local_change} \mbox{void seq64::lfownd::scale\_lfo\_change ( ) } \mbox{ [private]}
```

13.33.3.3 on_focus_out_event()

13.33.4 Field Documentation

```
13.33.4.1 m_seq
sequence& seq64::lfownd::m_seq [private]
13.33.4.2 m_seqdata
seqdata& seq64::lfownd::m_seqdata [private]
13.33.4.3 m_hbox
Gtk::HBox* seq64::lfownd::m_hbox [private]
13.33.4.4 m_scale_value
Gtk::VScale* seq64::lfownd::m_scale_value [private]
13.33.4.5 m_scale_range
Gtk::VScale* seq64::lfownd::m_scale_range [private]
13.33.4.6 m_scale_speed
Gtk::VScale* seq64::lfownd::m_scale_speed [private]
13.33.4.7 m_scale_phase
Gtk::VScale* seq64::lfownd::m_scale_phase [private]
```

```
13.33.4.8 m_scale_wave
Gtk::VScale* seq64::lfownd::m_scale_wave [private]
13.33.4.9 m_wave_name
Gtk::Label* seq64::lfownd::m_wave_name [private]
13.33.4.10 m_value
double seq64::lfownd::m_value [private]
13.33.4.11 m_range
double seq64::lfownd::m_range [private]
13.33.4.12 m_speed
double seq64::lfownd::m_speed [private]
13.33.4.13 m_phase
double seq64::lfownd::m_phase [private]
13.33.4.14 m_wave
```

wave_type_t seq64::lfownd::m_wave [private]

13.34 seq64::maintime Class Reference

This class provides the drawing of the progress bar at the top of the main window, along with two "pills" that move in time with the beat and measure.

Inheritance diagram for seg64::maintime:



Public Member Functions

• maintime (perform &p, int ppqn=SEQ64_USE_DEFAULT_PPQN)

This constructor sets up the colors black, white, and grey, and then allocates them.

virtual ∼maintime ()

Let's provide a do-nothing virtual destructor.

Private Member Functions

- · maintime (const maintime &)
- maintime & operator= (const maintime &)
- int idle_progress (midipulse ticks)

This function clears the window, sets the foreground to black, draws the "time" window's rectangle, and then draws a rectangle for noting the progress of the beat, and the progress for a bar.

void on_realize ()

Handles realization of the window.

bool on expose event (GdkEventExpose *ev)

This function merely idles.

Private Attributes

• const int m_beat_width

Provides the divisor for ticks to produce a beat value.

const int m_bar_width

Provides the divisor for ticks to produce a bar value.

const int m_pill_width

Provides the width of the pills, little black squares that show the progress of a beat and a bar (measure).

• const int m box width

The width/length of the rectangle to be drawn inside the maintime window.

• const int m_box_height

The height of the rectangle to be drawn inside the maintime window.

const int m_flash_width

The width/length of the flashing rectangle to be drawn inside the maintime window.

· const int m_flash_height

The height of the flashing rectangle to be drawn inside the maintime window.

· const int m_flash_x

The x value at which a flash should occur.

const int m_box_less_pill

The width/length of the maintime window minus the width of the pill.

· midipulse m tick

Saves the tick value for on_expose_event().

• int m_ppqn

Provides the active PPQN value.

Friends

· class mainwnd

Additional Inherited Members

13.34.1 Detailed Description

We added a lot of members to hold the results of calculations that involve what are essentially constant. This saves CPU time, and maybe a little memory for the code to make those calculations more than once.

13.34.2 Constructor & Destructor Documentation

In the constructor you can only allocate colors; get_window() would return 0 because the windows has not yet been realized.

13.34.2.3 \sim maintime()

```
virtual seq64::maintime::\simmaintime ( ) [inline], [virtual]
```

13.34.3 Member Function Documentation

13.34.3.1 operator=()

13.34.3.2 idle_progress()

Idle hands do the devil's work. We should eventually support some generic coloring for "dark themes". The default coloring is better for "light themes".

ticks Provides the main tick setting. This setting is provided by mainwnd(), in its timer callback.

Returns

Always returns 1 (it used to return "true"!).

13.34.3.3 on_realize()

```
void seq64::maintime::on_realize ( ) [private]
```

It performs the base class's on_realize() function. It then allocates some additional resources: a window, a GC (?), and it clears the window. Then it sets the default size of the window, specified by GUI constructor parameters.

13.34.3.4 on_expose_event()

We don't need the m_tick member, the function works as well if 0 is passed in. We've removed m_tick permanently.

Actually, it might be useful after all, to avoid flickering under JACK transport. Let's put it back for now. (It doesn't help, but we will leave it in, the overhead is small.)

13.34.4 Friends And Related Function Documentation

13.34.4.1 mainwnd

```
friend class mainwnd [friend]
```

13.34.5 Field Documentation

13.34.5.1 m_beat_width

```
const int seq64::maintime::m_beat_width [private]
```

Currently, this value is hardwired to 4, but will eventually be wired up as usr().midi_beat_width().

```
13.34.5.2 m_bar_width
const int seq64::maintime::m_bar_width [private]
Currently, this value is hardwired to 16, but will eventually be wired up as usr().midi_beat_width() * usr().midi_
beats_per_bar().
```

```
13.34.5.3 m_pill_width
```

```
const int seq64::maintime::m_pill_width [private]
```

```
13.34.5.4 m_box_width
```

```
const int seq64::maintime::m_box_width [private]
```

This item absolutely depends on the main window being non-resizable.

```
13.34.5.5 m_box_height
```

```
const int seq64::maintime::m_box_height [private]
```

This item absolutely depends on the main window being non-resizable.

```
13.34.5.6 m_flash_width
```

```
const int seq64::maintime::m_flash_width [private]
```

Just a bit smaller than m_box_width.

```
13.34.5.7 m_flash_height
```

```
const int seq64::maintime::m_flash_height [private]
```

Just a bit smaller than m_box_width.

```
13.34.5.8 m_flash_x
```

```
const int seq64::maintime::m_flash_x [private]
```

```
13.34.5.9 m_box_less_pill
```

```
const int seq64::maintime::m_box_less_pill [private]
```

13.34.5.10 m_tick

```
midipulse seq64::maintime::m_tick [private]
```

It might actually be useful after all. And the overhead is tiny.

13.34.5.11 m_ppqn

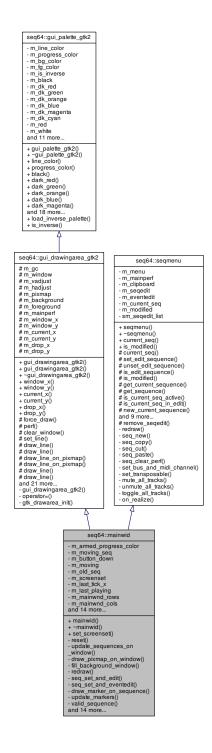
```
int seq64::maintime::m_ppqn [private]
```

While this is effectively a constant for the duration of a tune, it might change as different tunes are loaded.

13.35 seq64::mainwid Class Reference

This class implements the piano roll area of the application.

Inheritance diagram for seq64::mainwid:



Public Member Functions

· mainwid (perform &p)

This constructor sets all of the members.

virtual ∼mainwid ()

A rote destructor.

void set_screenset (int ss, bool setperf=false)

Set the current screen-set.

Private Member Functions

· void reset ()

This function redraws everything and queues up a redraw operation.

void update_sequences_on_window ()

Updates the image of multiple sequencer/pattern slots.

void draw_pixmap_on_window ()

This function queues the blit of pixmap to window.

void fill_background_window ()

This function updates the background window, clearing it.

virtual void redraw (int seq)

This virtual function, overridden from the seqmenu base class, draws the the given pattern/sequence again.

virtual void seg set and edit (int segnum)

Calculates the sequence number based on the screenset and then calls the base-class function to bring up the pattern/sequence editor.

virtual void seg set and eventedit (int segnum)

Calculates the sequence number based on the screenset and then calls the base-class function to bring up the event editor

void draw_marker_on_sequence (int seq, int tick)

Does the actual drawing of one pattern/sequence position marker, a vertical progress bar.

void update markers (int ticks)

Draw the cursors (long vertical bars) on each sequence, so that they follow the playing progress of each sequence in the mainwid (Patterns Panel).

bool valid_sequence (int seq)

Common-code helper function.

void draw_sequence_on_pixmap (int seq)

This function draws a specific pattern/sequence on the pixmap located in the main window of the application, the Patterns Panel.

• void draw_sequences_on_pixmap ()

This function fills the pixmap with sequences.

void draw_sequence_pixmap_on_window (int seq)

This function draws a sequence pixmap in the Patterns Panel.

int seq_from_xy (int x, int y)

Translates XY coordinates in the Patterns Panel to a sequence number.

• int timeout ()

Provides a stock callback, because some kind of callback is needed.

• void calculate_base_sizes (int seq, int &basex, int &basey)

Provides a way to calculate the base x and y size values for the pattern map.

void select_fg_bg_colors (int seqnum)

Picks the foreground and background colors based on the sequence in edit and the SEQ64_EDIT_SEQUENCE_← HIGHLIGHT macro.

• void on realize ()

For this GTK callback, on realization of window, initialize the shiz.

• bool on_expose_event (GdkEventExpose *ev)

Implements the GTK expose event callback.

bool on_button_press_event (GdkEventButton *ev)

Handles a press of a mouse button in one of the sequence/pattern slots.

bool on_button_release_event (GdkEventButton *ev)

Handles a release of a mouse button.

• bool on_motion_notify_event (GdkEventMotion *p0)

Handle the motion of the mouse if a mouse button is down and in another sequence and if the current sequence is not in edit mode.

bool on_focus_in_event (GdkEventFocus *)

Handles an on-focus event.

bool on_focus_out_event (GdkEventFocus *)

Handles an out-of-focus event.

Private Attributes

· Color m_armed_progress_color

Holds the progress color for armed sequences, which have a black background.

• sequence m_moving_seq

Holds a partial copy of the sequence we are moving on the patterns panel.

· bool m button down

Indicates that the mouse button is still down.

bool m moving

Indicates that we are still in the middle of a drag-and-drop operation.

int m_old_seq

Holds the sequence number of a sequence being drag-and-dropped.

· int m screenset

Indicates the current screenset that is visible.

long m_last_tick_x [c_max_sequence]

Holds the last active tick for each sequence, used in erasing the progress bar.

long m_last_playing [c_max_sequence]

Holds the last playing tick for each sequence.

• int m_mainwnd_rows

These values are assigned to the values given by the constants of similar names in globals.h, and we will make them parameters or user-interface configuration items later.

• int m_mainwnd_cols

Number of columns, unused in settings.

• int m_seqarea_x

Roughly with width of the main window.

• int m_seqarea_y

Roughly with height of the main window.

• int m_seqarea_seq_x

To be determined.

int m_seqarea_seq_y

To be determined.

· int m_mainwid_x

To be determined.

int m_mainwid_y

To be determined.

• int m_mainwid_border

Main-window border, unused setting.

int m_mainwid_spacing

Main-window spacing, unused setting.

• int m_text_size_x

Text width, varies with font in use.

· int m_text_size_y

Text height, varies with font in use.

• int m_max_sets

The maximum number of sets, use all over.

• int m_screenset_slots

Provides a convenience variable for avoiding multiplications.

· int m screenset offset

Provides a convenience variable for avoiding multiplications.

• int m_progress_height

Provides the height of the progress bar, to save calculations and for consistency between drawing and erasing the progress bar.

Friends

- · class mainwnd
- void update_mainwid_sequences ()

This global function in the seq64 namespace calls mainwid :: update_sequences_on_window(), if the global mainwid object exists.

Additional Inherited Members

13.35.1 Detailed Description

It inherits from gui_drawingarea_gtk2 to support the font, color, and other GUI functionality, and from seqmenu to support the right-click Edit/New/Cut right-click menu. The friend class and function are for updating the current sequence and for control via the mainwnd object.

13.35.2 Constructor & Destructor Documentation

13.35.2.1 mainwid()

And it asks for a size of c_mainwid_x by c_mainwid_y. It adds GDK masks for button presses, releases, motion, key presses, and focus changes. Also logs a self-referential singleton pointer to use for the current-edit highlighting support.

Parameters

p | Provides the reference to the all-important perform object.

13.35.2.2 \sim mainwid()

```
seq64::mainwid::\sim mainwid ( ) [virtual]
```

13.35.3 Member Function Documentation

13.35.3.1 set_screenset()

The clamping algorithm for the screeset is a bit weird: if less than 0, we set m_screenset to its maximum, and if greater than the maximum, we set it to its minimum. Not sure if this matters.

Note that $m_screenset_slots = m_mainwnd_rows * m_mainwnd_cols$.

We will likely replace this with perform::set_screenset(), which recapitulates the code above completely, whereas perform::set-offset() recapitulates only the line of code immediately above it. However, note that there is a back-and-forth between setting the screenset via perform (using MIDI control) versus the GUI in the mainward class. Probably useful to add a default boolean to prevent circular manipulation.

Parameters

SS	Provides the screen-set number to set.	
setperf	If true, then also call perfrom::set_screenset(). Defaults to false. It might be better if it defaults to true.	

13.35.3.2 reset()

```
void seq64::mainwid::reset ( ) [inline], [private]
```

13.35.3.3 update_sequences_on_window()

```
void seq64::mainwid::update_sequences_on_window ( ) [inline], [private]
```

Used by the friend class mainwnd, but also useful for our new feature to fully highlight the current sequence. Calls reset() if SEQ64_EDIT_SEQUENCE_HIGHLIGHT is defined.

13.35.3.4 draw_pixmap_on_window()

```
void seq64::mainwid::draw_pixmap_on_window ( ) [inline], [private]
```

```
13.35.3.5 fill_background_window()
```

```
void seq64::mainwid::fill_background_window ( ) [inline], [private]
```

13.35.3.6 redraw()

Parameters

segnum Provides the number of the sequence to draw.

Implements seq64::seqmenu.

```
13.35.3.7 seq_set_and_edit()
```

Used with the '=' key selection, by default.

Reimplemented from seq64::seqmenu.

13.35.3.8 seq_set_and_eventedit()

Used with the '-' key selection, by default.

Reimplemented from seq64::seqmenu.

13.35.3.9 draw_marker_on_sequence()

If the sequence has no events, this function doesn't bother drawing a position marker.

Note that, when Sequencer64 first comes up, and perform::is_dirty_main() is called, no sequences exist yet. Also, currently the redraw() is hit when seq_edit() is called, but not when seq_event_edit() is called, which makes the latter not paint the in-edit highlight colors (if enabled). Why?

seqnum Provides the number of the sequence to draw.	
tick	Provides the location to draw the marker. If pause support is compiled in (i.e. no –disable-pause in the configuration), then this parameter is ignored, and is replaced by the sequences' get lask tick()
	value. This causes correct stop/pause/play progress-bar behavior in each pattern slot.

13.35.3.10 update_markers()

Parameters

tick Starting point for drawing the markers.

13.35.3.11 valid_sequence()

Parameters

seqnum	Provides the number of the sequence to validate.

Returns

Returns true if the sequence number is valid for the current m_screenset value.

13.35.3.12 draw_sequence_on_pixmap()

The sequence is drawn only if it is in the current screen set (indicated by m_screenset). Also, we ignore the sequence if it does not exist.

Note

If only the main window is up, then the sequences just play (muted by default) – the progress bars move in each pattern. Gaps in the sequence in the Song (performance) Editor don't change the appearance of the patterns if only the main window is up. But, if the Song Editor window is up, and the song is started using the controls in the Song Editor, then the active patterns are black while playing, and white when gaps in the sequence are encountered. The muting status in the main window is ignored. The muting in the Song (performance) windows is in force. This setup holds for ALSA, but not for JACK transport.

seqnum	Provides the number of the sequence slot that needs to be drawn. It is checked for validity before	
	usage.	

13.35.3.13 draw_sequences_on_pixmap()

```
void seq64::mainwid::draw_sequences_on_pixmap ( ) [private]
```

Please note that draw_sequence_on_pixmap() also draws the empty slots of inactive sequences, so we cannot take shortcuts here.

13.35.3.14 draw_sequence_pixmap_on_window()

The sequence is drawn only if it is in the current screen set (indicated by m_screenset. This function is used when dragging a pattern from one pattern-slot to another pattern-slot.

We have to add 1 pixel to the y height in order to avoid leaving behind a line at the bottom of an empty pattern-slot.

Parameters

seqnum	Provides the number of the sequence to draw.
--------	--

13.35.3.15 seq_from_xy()

Parameters

	Χ	Provides the x coordinate.
Ì	V	Provides the y coordinate.

Returns

Returns -1 if the sequence number cannot be calculated.

13.35.3.16 timeout()

```
int seq64::mainwid::timeout ( ) [private]
```

Todo We should use this callback to display the current time in the playback.

Returns

Always returns true.

13.35.3.17 calculate_base_sizes()

The values are returned as side-effects.

Parameters

		seqnum	Provides the number of the sequence to calculate.
	out	basex	A return parameter for the x coordinate of the base size.
ĺ	out	basey	A return parameter for the y coordinate of the base size.

13.35.3.18 select_fg_bg_colors()

13.35.3.19 on_realize()

```
void seq64::mainwid::on_realize ( ) [private]
```

It allocates any additional resources that weren't initialized in the constructor.

This function used to call font::init(), and was the only place where the font::init() function was called. The init() function gets a color-map from the window. We need a more fool-proof was to do this!

13.35.3.20 on_expose_event()

ev The expose event.

Returns

Always returns true.

13.35.3.21 on_button_press_event()

If the press is a single left-click, and no Ctrl key is pressed, then this function grabs the focus, calculates the pattern/sequence over which the button press occurred, and sets the m_button_down flag if it is over a pattern. In the release event callback, this then causes the sequence arming/muting to be toggled.

If the press is a single Ctrl-left-click, this function brings up the New or Edit menu. The New menu is brought up if the grid slot is empty, and the Edit menu otherwise. Another way to bring up the same functionality is described in the next paragraph.

If the press is a double-click, it first acts just like two single-clicks (which might confuse the user at first, because it toggles the mute state twice). Then it brings up the Edit menu for the sequence. This new behavior is closer to what users have come to expect from a double-click. I miss the double-click when running seq24.

We also try to handle a Ctrl-double-click as a signal to do an event edit, instead of a sequence edit. The event editor provides a way to look at all events in detail, without having to select the type of event to see. However, this doesn't work, the event is treated like a ctrl-single-click. And we use the Alt key to enable window movement or resizing in our window manager, so that's out.

Parameters

ev Provides the parameters of the button event.

Returns

Always returns true.

13.35.3.22 on_button_release_event()

This event is a lot more complex than a press. The left button toggles playback status. The right button brings up a popup menu. If the slot is empty, then a "New" popup is presented, otherwise an "Edit" and selection popup is presented.

Also now implements the new "toggle all other patterns" action, initiated via Shift-Left-Click.

ev Provides the parameters of the button event.

Returns

Always returns true.

Tried disabling the setting of the current sequence; it completely disables drag-n-drop. But leaving it in removes the current-sequence highlighting, which otherwise is fine. So we do it only if moving a pattern (drag-and-drop).

```
13.35.3.23 on_motion_notify_event()
```

This function moves the selected pattern to another pattern slot. The perform::delete_sequence() function sets the perform modification flag.

Parameters

ev Provides the parameters of the button event.

Returns

Always returns true.

13.35.3.24 on_focus_in_event()

Just sets the Gtk::HAS_FOCUS flag.

Returns

Always returns false.

13.35.3.25 on_focus_out_event()

Just unsets the Gtk::HAS_FOCUS flag.

Returns

Always returns false.

13.35.4 Friends And Related Function Documentation

13.35.4.1 mainwnd

```
friend class mainwnd [friend]
```

13.35.4.2 update_mainwid_sequences

```
void update_mainwid_sequences ( ) [friend]
```

It is used by other objects that can modify the currently-edited sequence shown in the mainwid (main window).

13.35.5 Field Documentation

13.35.5.1 m_armed_progress_color

```
Color seq64::mainwid::m_armed_progress_color [private]
```

If the progress color is black(), we want to change it to white for unmuted patterns.

```
13.35.5.2 m_moving_seq
```

```
sequence seq64::mainwid::m_moving_seq [private]
```

The assignment is made by sequence::partial_copy(), which behaves like the legacy seq24 code.

13.35.5.3 m_button_down

```
bool seq64::mainwid::m_button_down [private]
```

Used in the drag-and-drop functionality.

13.35.5.4 m_moving

```
bool seq64::mainwid::m_moving [private]
```

```
13.35.5.5 m_old_seq
int seq64::mainwid::m_old_seq [private]
13.35.5.6 m_screenset
int seq64::mainwid::m_screenset [private]
13.35.5.7 m_last_tick_x
long seq64::mainwid::m_last_tick_x[c_max_sequence] [private]
13.35.5.8 m_last_playing
long seq64::mainwid::m_last_playing[c_max_sequence] [private]
This doesn't seem to be used anywhere, even though values are logged, so it is macroed out.
13.35.5.9 m_mainwnd_rows
int seq64::mainwid::m_mainwnd_rows [private]
Some of them already have counterparts in the user_settings class. Number of rows, unused part of settings.
13.35.5.10 m_mainwnd_cols
int seq64::mainwid::m_mainwnd_cols [private]
13.35.5.11 m_seqarea_x
int seq64::mainwid::m_seqarea_x [private]
13.35.5.12 m_seqarea_y
```

int seq64::mainwid::m_seqarea_y [private]

```
13.35.5.13 m_seqarea_seq_x
int seq64::mainwid::m_seqarea_seq_x [private]
13.35.5.14 m_seqarea_seq_y
int seq64::mainwid::m_seqarea_seq_y [private]
13.35.5.15 m_mainwid_x
int seq64::mainwid::m_mainwid_x [private]
13.35.5.16 m_mainwid_y
int seq64::mainwid::m_mainwid_y [private]
13.35.5.17 m_mainwid_border
int seq64::mainwid::m_mainwid_border [private]
13.35.5.18 m_mainwid_spacing
int seq64::mainwid::m_mainwid_spacing [private]
13.35.5.19 m_text_size_x
int seq64::mainwid::m_text_size_x [private]
13.35.5.20 m_text_size_y
```

int seq64::mainwid::m_text_size_y [private]

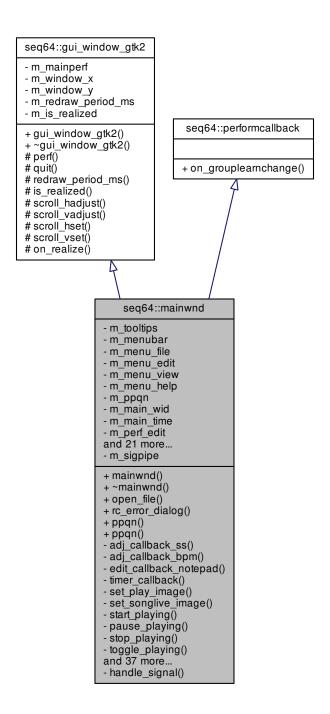
```
13.35.5.21 m_max_sets
int seq64::mainwid::m_max_sets [private]
13.35.5.22 m_screenset_slots
int seq64::mainwid::m_screenset_slots [private]
It is equal to m_mainwnd_rows * m_mainwnd_cols.
13.35.5.23 m_screenset_offset
int seq64::mainwid::m_screenset_offset [private]
It is equally to m\_screenset\_slots*m\_screenset.
13.35.5.24 m_progress_height
```

13.36 seq64::mainwnd Class Reference

int seq64::mainwid::m_progress_height [private]

This class implements the functionality of the main window of the application, except for the Patterns Panel functionality, which is implemented in the mainwid class.

Inheritance diagram for seq64::mainwnd:



Public Member Functions

- mainwnd (perform &p, bool allowperf2=true, int ppqn=SEQ64_USE_DEFAULT_PPQN)
 - The constructor the main window of the application.
- virtual ∼mainwnd ()

This destructor must explicitly delete some allocated resources.

• void open_file (const std::string &filename)

Opens and parses (reads) a MIDI file.

void rc_error_dialog (const std::string &message)

Tells the user that the "rc" file is erroneous.

• int ppqn () const

'Getter' function for member m_ppqn

• void ppqn (int ppqn)

'Setter' function for member m_ppqn We can't set the PPQN value when the mainwnd is created, we have to do it later, using this function.

Private Member Functions

• void adj callback ss ()

This function is the callback for adjusting the screen-set value.

void adj_callback_bpm ()

This function is the callback for adjusting the BPM value.

void edit_callback_notepad ()

A callback function for handling an edit to the screen-set notepad.

bool timer_callback ()

This function is the GTK timer callback, used to draw our current time and BPM on_events (the main window).

void set play image (bool isrunning)

Changes the image used for the pause/play button.

void set songlive image (bool issong)

Changes the image used for the song/live mode button.

void start_playing ()

Starts playing of the song.

void pause_playing ()

Pauses the playing of the song, leaving the progress bar where it stopped.

• void stop_playing ()

Stops the playing of the song.

void toggle_playing ()

Reverses the state of playback.

• void learn_toggle ()

Toggle the group-learn status.

void open_performance_edit ()

Opens the Performance Editor (Song Editor).

• void open_performance_edit_2 ()

Opens the second Performance Editor (Song Editor).

• void enregister_perfedits ()

This function brings together the two perfedit objects, so that they can tell each other when to queue up a draw operation.

void sequence_key (int seq)

Use the sequence key to toggle the playing of an active pattern in the current screen-set.

• void apply_song_transpose ()

Apply full song transposition, if enabled.

void update_window_title ()

Updates the title shown in the title bar of the window.

void toLower (std::string &)

Converts a string to lower-case letters.

• void file new ()

A callback function for the File / New menu entry.

```
· void file_open ()
      A callback function for the File / Open menu entry.
• void file save ()
      A callback function for the File / Save menu entry.
void set_song_mute (perform::mute_op_t op)
      Sets the song-mute mode.
• void file_import_dialog ()
      Presents a file dialog to import a MIDI file.

    void options_dialog ()

      Opens the File / Options dialog.
· void jack_dialog ()
      Opens the File / Options dialog to show only the JACK page.

    void about_dialog ()

      Presents a Help / About dialog.

    void build info dialog ()

      Presents a Help / Build Info dialog.
int query_save_changes ()
      Queries the user to save the changes made while the application was running.
void new_open_error_dialog ()
      Tells the user to close all the edit windows first.

    void file_save_as (bool do_export=false)

      A callback function for the File / Save As menu entry.

    void file exit ()

      A callback function for the File / Exit menu entry.
void new_file ()
      Actually does the work of setting up for a new file.
• bool save file ()
      Saves the current state in a MIDI file.
· void choose_file ()
      Creates a file-chooser dialog.
• bool is_save ()
      If the data is modified, then the user is queried, and the file is save if okayed.
• bool install_signal_handlers ()
      Installs the signal handlers and pipe code.

    bool signal action (Glib::IOCondition condition)

      Handles saving or exiting actions when signalled.
· bool edit field has focus () const
      Check if one of the edit fields (BPM spinbutton, screenset spinbutton, or the Name field) has focus.

    void populate_menu_file ()

      Populates the File menu: File menu items; their accelerator keys; and their hot keys.

    void populate menu edit ()

      Populates the Edit menu: Edit menu items; their accelerator keys; and their hot keys.
void populate_menu_help ()
      Populates the Help menu.
void populate_menu_view ()
      Populates the View menu: View menu items and their hot keys.

    bool on_delete_event (GdkEventAny *ev)

      This callback function handles a delete event from ...?

    bool on key press event (GdkEventKey *ev)

      Handles a key press event.

    bool on_key_release_event (GdkEventKey *ev)
```

Handles a key release event.

• void on_realize ()

We are trying to work around an apparent Gtk+ bug (which occurs on my 64-bit Debian Sid laptop, but not on my 32-bit Debian Jessie laptop) that causes Sequencer64 to freeze, emitting Gtk errors, if one tries to access the main menu via Alt-F, Alt-E, etc.

virtual void on grouplearnchange (bool state)

Notification handler for learn mode toggle.

Static Private Member Functions

static void handle_signal (int sig)

This function is the handler for system signals (SIGUSR1, SIGINT...) It writes a message to the pipe and leaves as soon as possible.

Private Attributes

• Gtk::Tooltips * m_tooltips

A repository for tooltips.

• Gtk::MenuBar * m menubar

Theses objects support the menu and its sub-menus.

• Gtk::Menu * m menu file

The File menu entry.

• Gtk::Menu * m_menu_edit

The (new) Edit menu entry.

Gtk::Menu * m_menu_view

The View menu entry.

• Gtk::Menu * m menu help

The Help menu entry.

• int m_ppqn

Saves the PPQN value obtained from the MIDI file (or the default value, the global ppqn, if $SEQ64_USE_DEFAUL \leftarrow T_PPQN$ was specified in reading the MIDI file.

• mainwid * m_main_wid

The biggest sub-components of mainwnd.

• maintime * m_main_time

Is this the bar at the top that shows moving squares, also known as "pills"? Why yes, it is.

• perfedit * m_perf_edit

A pointer to the first song/performance editor.

perfedit * m_perf_edit_2

A pointer to an optional second song/performance editor.

• options * m_options

A pointer to the program options.

Gdk::Cursor m_main_cursor

Mouse cursor?

Gtk::Image * m image play

Provides a pointer to hold the images for the pause/play button.

Gtk::Button * m_button_learn

This button is the learn button, otherwise known as the "L" button.

• Gtk::Button * m button stop

Implements the red square stop button.

Gtk::Button * m_button_play

Implements the green triangle play button.

Gtk::Button * m button perfedit

The button for bringing up the Song Editor (Performance Editor).

• Gtk::Button * m button jack

Sets and indicates the current mode of Sequencer64: JACK, Master, and ALSA.

Gtk::Adjustment * m adjust bpm

The spin/adjustment controls for the BPM (beats-per-minute) value.

• Gtk::SpinButton * m_spinbutton_bpm

BPM spin-button object.

• Gtk::Adjustment * m adjust ss

The spin/adjustment controls for the screenset value.

• Gtk::SpinButton * m_spinbutton_ss

Screenset adjustment.

Gtk::Adjustment * m adjust load offset

The spin/adjustment controls for the load offset value.

Gtk::SpinButton * m_spinbutton_load_offset

Spin button for import.

• Gtk::Entry * m entry notes

This item provides user-interface access to the screenset notepad editor.

• bool m_is_running

Holds the current status of running, for use in display the play versus pause icon.

• sigc::connection m_timeout_connect

Provides a timeout handler.

• bool m_menu_mode

Indicates if the menu bar is to be greyed out or not.

bool m_call_seq_edit

Indicates that this object is in a mode where the usual mute/unmute keystroke will instead bring up the pattern slot for editing.

· bool m call seg eventedit

Indicates that this object is in a mode where the usual mute/unmute keystroke will instead bring up the pattern slot for event-editing.

Static Private Attributes

• static int m sigpipe [2]

This small array holds the "handles" for the pipes need to intercept the system signals SIGINT and SIGUSR1, so that the application shuts down gracefully when aborted.

Additional Inherited Members

13.36.1 Constructor & Destructor Documentation

13.36.1.1 mainwnd()

This constructor is way too large; it would be nicer to provide a number of well-named initialization functions.

р	Refers to the main performance object.
allowperf2	Indicates if a second perfedit window should be created. This is currently a run-time option, selectable in the "user" configuration file.
ppqn	An optional PPQN value to use in the song.

Todo Offload most of the work into an initialization function like options does; make the perform parameter a reference; valgrind flags m_tooltips as lost data, but if we try to manage it ourselves, many more leaks occur.

Top panel items, including the logo (updated for the new version of this application) and the "timeline" progress bar.

```
13.36.1.2 \simmainwnd() seq64::mainwnd::\simmainwnd ( ) [virtual]
```

13.36.2 Member Function Documentation

We leave the ppqn parameter set to the SEQ64_USE_DEFAULT for now, to preserve the legacy behavior of using the global ppqn, and scaling the running time against the PPQN read from the MIDI file. Later, we can provide a value like 0, that will certainly be changed by reading the MIDI file.

We don't need to specify the "oldformat" or "global sequence" parameters of the midifile constructor when reading the MIDI file, since reading handles both the old and new formats, dealing with new constructs only if they are present in the file.

Parameters

fn Provides the file-name for the MIDI file to be opened.

13.36.2.2 rc_error_dialog() void seq64::mainwnd::rc_error_dialog (

We can't yet display the specific error, except in a terminal window.

const std::string & message)

message Provides the error message returned by the configuration file.

Its sets the screen-set value in the Performance/Song window, the Patterns, and something about setting the text based on a screen-set notepad from the Performance/Song window. We let the perform object keep track of modifications.

```
13.36.2.7 adj_callback_bpm()

void seq64::mainwnd::adj_callback_bpm ( ) [private]

Let the perform object keep track of modifications.

13.36.2.8 edit_callback_notepad()
```

void seq64::mainwnd::edit_callback_notepad () [private]

Let the perform object keep track of modifications.

13.36.2.9 timer_callback()

```
bool seq64::mainwnd::timer_callback ( ) [private]
```

It also supports the ALSA pause functionality.

Note

When Sequencer64 first starts up, and no MIDI tune is loaded, the call to mainwid::update_markers() leads to trying to do some work on sequences that don't yet exist. Also, if a sequence is changed by the event editor, we get a crash; need to find out how sequence away with the changes.

Returns

Always returns true.

13.36.2.10 set_play_image()

Parameters

isrunning	If true, set the image to the "Pause" icon, since playback is running. Otherwise, set it to the "Play"	
	button, since playback is not running.	

13.36.2.11 set_songlive_image()

Parameters

issong If true, set the image to the "Song" icon. Otherwise, set it to the "Live" button.

13.36.2.12 start_playing()

```
void seq64::mainwnd::start_playing ( ) [private]
```

An accessor to perform::start_playing(). This function is actually a callback for the pause/play button. Now very similar to perfedit::start_playing(), except that the implicit songmode == false parameter is used here.

We still need to see if pause_key() is workable with Stazed JACK support in force. Doesn't pause at present.

13.36.2.13 pause_playing()

```
void seq64::mainwnd::pause_playing ( ) [private]
```

Currently, it is just the same as stop playing(), but we will get it to work.

13.36.2.14 stop_playing()

```
void seq64::mainwnd::stop_playing ( ) [private]
```

An accessor to perform's stop_playing() function. Also calls the mainwid::update_sequences_on_window() function. Not sure that we need this call, since the slots seem to update anyway. But we've noticed that, with this call in place, hitting the Stop button causes a subtle change in the appearance of the first non-empty pattern of the "allofarow.mid" file.

After the Stop button is pushed (in ALSA mode), then the Space key ("start") doesn't work properly. The song starts, then quickly stops. It doesn't matter if update_sequences_on_window() is called or not. This happens even in seq24! This bug has proven incredibly difficult to track down, still working on it.

13.36.2.15 toggle_playing()

```
void seq64::mainwnd::toggle_playing ( ) [private]
```

Meant only to be called when the "Play" button is pressed, if the pause feature has been compiled into the application.

13.36.2.16 learn_toggle()

```
void seq64::mainwnd::learn_toggle ( ) [inline], [private]
```

Simply forwards the call to perform::learn_toggle().

13.36.2.17 open_performance_edit()

```
void seq64::mainwnd::open_performance_edit ( ) [private]
```

We will let perform keep track of modifications, and not just set an is-modified flag just because we opened the song editor. We're going to centralize the modification flag in the perform object, and see if it can work.

13.36.2.18 open_performance_edit_2()

```
void seq64::mainwnd::open_performance_edit_2 ( ) [private]
```

Experiment: open a second one and see what happens. It works, but one needs to tell the other to redraw if a change is made.

```
13.36.2.19 enregister_perfedits()
void seq64::mainwnd::enregister_perfedits ( ) [private]
13.36.2.20 sequence_key()
void seq64::mainwnd::sequence_key (
              int seq ) [inline], [private]
13.36.2.21 apply_song_transpose()
void seq64::mainwnd::apply_song_transpose ( ) [private]
Then reset the perfedit transpose setting to 0.
13.36.2.22 update_window_title()
void seq64::mainwnd::update_window_title ( ) [private]
Note that the name of the application is obtained by the "(SEQ64_PACKAGE)" construction.
The format of the caption bar is the name of the package/application, followed by the file-specification (shortened if
necessary so that the name of the file itself can be seen), ending with the PPQN value in parentheses.
13.36.2.23 toLower()
void seq64::mainwnd::toLower (
              std::string & s ) [private]
13.36.2.24 file_new()
void seq64::mainwnd::file_new ( ) [inline], [private]
13.36.2.25 file_open()
```

void seq64::mainwnd::file_open () [inline], [private]

Note that every track of the MIDI file will be imported, even if the track is only a label track (without any MIDI events), or a very long track.

The main difference between the Open operation and the Import operation seems to be that the latter can read MIDI files into a screen-set greater than screen-set 0. No, that's not true, so far. No matter what the current screen-set setting, the import is appended after the current data in screen-set 0. Then, if it overflows that screen-set, the overflow goes into the next screen-set.

It might be nice to have the option of importing a MIDI file into a specific screen-set, for better organization, as well as being able to offset the sequence number.

Also, it is important to note that perf().clear_all() is not called by this routine, as we are merely adding to what might already be there.

```
13.36.2.29 options_dialog()
void seq64::mainwnd::options_dialog ( ) [private]

13.36.2.30 jack_dialog()
void seq64::mainwnd::jack_dialog ( ) [private]
13.36.2.31 about_dialog()
```

void seq64::mainwnd::about_dialog () [private]

I (Chris) took the liberty of tacking my name at the end, and hope to have done eventually enough work to warrant having it there. Hmmmmm....

13.36.2.32 build_info_dialog()

13.36.2.35 file_save_as()

```
void seq64::mainwnd::build_info_dialog ( ) [private]
```

It is similar to the "--version" option on the command line. The AboutDialog doesn't seem to have a way to left-align the text, so we're trying the MessageDialog.

```
13.36.2.33 query_save_changes()
int seq64::mainwnd::query_save_changes ( ) [private]

13.36.2.34 new_open_error_dialog()

void seq64::mainwnd::new_open_error_dialog ( ) [private]
```

bool do_export = false) [private]

Please note that Sequencer64 will not adopt the "c_seq32_midi" type of file, because it already saves its files in a format that other sequencers should be able to read.

Stazed on the intent of the export functionality:

void seq64::mainwnd::file_save_as (

The original intent was to be able to play an exported song in something like TiMIDIty. After I completed things I realized that there could be an editing benefit as well. I like to record from my MIDI keyboard, improvised to a drum beat, on a long sequence (64 measures). Some is junk, but there are usually parts that I can use. In original seq24, to cut out the good or bad stuff, you would have to search the sequence by listening, then cut and move or copy and paste to a new sequence. It could be done but was always tedious. The paste box for the sequence sometimes made it difficult to find the correct note location, measure, and beat. Also, on a long sequence, you need to zoom out to see the copy location as it played, but zoom in for the precise paste location. In addition if you wanted to change the measure of the notes, it became a trial and error of copy/paste, listen, move, listen, move....

With the added Song editor feature of split trigger to mouse and copy paste trigger to mouse, you can now do all the editing from the song editor. Listen to the sequence, cut out the good or bad parts and reassemble. Move or copy all good trigger parts to the left start and delete all the bad stuff. Now you can use the song export to create the new sequence. Just mute all other tracks and export. Re-import and the new cleaned sequence is already done. Also I use it for importing drum beats from a single '32/'42 file that contains dozens of different styles with intros and endings. I like to sync two instances of '32 or '42 together with jack, then play/experiment with the different beats. If I find something I like, create the song trigger for the part I like in the drum file, export and import.

I actually do not use the song export for anything but editing.

Note that the split trigger variant of Stazed, where it doesn't just split the section in half, is not yet implemented (2016-08-05).

Parameters

do_export	If true, then just write out the file and don't change the name of the current file based on the	
	file-name the user selected. The default value of this parameter is false.	

```
13.36.2.36 file_exit()
void seq64::mainwnd::file_exit ( ) [private]

13.36.2.37 new_file()
```

void seq64::mainwnd::new_file () [private]

Not sure that we need to clear the modified flag here, especially since it is now centralizeed in the perform object. Let perf().clear_all() handle it now.

```
13.36.2.38 save_file()
bool seq64::mainwnd::save_file ( ) [private]
```

Here we specify the current value of m_ppqn, which was set when reading the MIDI file. We also let midifile tell the perform that saving worked, so that the "is modified" flag can be cleared. The midifile class is already a friend of perform.

```
13.36.2.39 choose_file()
void seq64::mainwnd::choose_file ( ) [private]
```

Change Note layk 2016-10-11 Issue #43 Added filters for upper-case MIDI-file extensions.

```
13.36.2.40 is_save()
bool seq64::mainwnd::is_save ( ) [private]

13.36.2.41 install_signal_handlers()
```

bool seq64::mainwnd::install_signal_handlers () [private]

13.36.2.42 signal_action()

Returns

Returns true if the signalling was able to be completed, even if it was an unexpected signal.

13.36.2.43 edit_field_has_focus()

```
bool seq64::mainwnd::edit_field_has_focus ( ) const [private]
```

Returns

Returns true if one of the three editable/modifiable fields has the keyboard focus.

13.36.2.44 populate_menu_file()

```
void seq64::mainwnd::populate_menu_file ( ) [private]
```

Provided to make the constructor more readable and manageable.

```
13.36.2.45 populate_menu_edit()
```

```
void seq64::mainwnd::populate_menu_edit ( ) [private]
```

Provided to make the constructor more readable and manageable.

```
13.36.2.46 populate_menu_help()
```

```
void seq64::mainwnd::populate_menu_help ( ) [private]
```

Provided to make the constructor more readable and manageable.

13.36.2.47 populate_menu_view()

```
void seq64::mainwnd::populate_menu_view ( ) [private]
```

It repeats the song editor edit command, just to help those whose muscle memory is already seq32-oriented. Provided to make the constructor more readable and manageable. View menu items and their hot keys.

13.36.2.48 on_delete_event()

Any changed data is saved. If the pattern is playing, then it is stopped. We now use perform::is_pattern_playing().

13.36.2.49 on_key_press_event()

It also handles the control-key and modifier-key combinations matching the entries in its list of if statements.

Also, we now effectively press the CAPS LOCK key for the user if in group-learn mode, via the keystroke::shift_lock() function.

13.36.2.50 on_key_release_event()

Is this worth turning into a switch statement? Or offloading to a perform member function? The latter. Also, we now effectively press the CAPS LOCK key for the user if in group-learn mode. The function that does this is keystroke ::shift_lock().

Todo Test this functionality in old and new application.

Returns

Always returns false. This matches seq24 behavior.

13.36.2.51 on_realize()

```
void seq64::mainwnd::on_realize ( ) [private]
```

without first moving the mouse to the main window. Weird with a beard!

13.36.2.52 on_grouplearnchange()

This handler responds to a learn-mode change from perf().

Reimplemented from seq64::performcallback.

13.36.3 Field Documentation

```
13.36.3.1 m_sigpipe
int seq64::mainwnd::m_sigpipe [static], [private]
This static member provides a couple of pipes for signalling/messaging.
13.36.3.2 m_tooltips
Gtk::Tooltips* seq64::mainwnd::m_tooltips [private]
13.36.3.3 m_menubar
Gtk::MenuBar* seq64::mainwnd::m_menubar [private]
The whole menu bar.
13.36.3.4 m_menu_file
Gtk::Menu* seq64::mainwnd::m_menu_file [private]
13.36.3.5 m_menu_edit
Gtk::Menu* seq64::mainwnd::m_menu_edit [private]
13.36.3.6 m_menu_view
Gtk::Menu* seq64::mainwnd::m_menu_view [private]
13.36.3.7 m_menu_help
Gtk::Menu* seq64::mainwnd::m_menu_help [private]
```

```
13.36.3.8 m_ppqn
```

```
int seq64::mainwnd::m_ppqn [private]
```

We need it early here to be able to pass it along to child objects.

```
13.36.3.9 m_main_wid
```

```
mainwid* seq64::mainwnd::m_main_wid [private]
```

The first is the Patterns Panel, which the mainwid helps implement. We end up sharing this object with perfedit, perfnames, and sequent in order to allow the sequent object to notify the mainwid (indirectly) of the currently-edited sequence.

```
13.36.3.10 m_main_time
```

```
maintime* seq64::mainwnd::m_main_time [private]
```

```
13.36.3.11 m_perf_edit
```

```
perfedit* seq64::mainwnd::m_perf_edit [private]
```

```
13.36.3.12 m_perf_edit_2
```

```
perfedit* seq64::mainwnd::m_perf_edit_2 [private]
```

The second makes it easy to line up two different patterns that cannot be seen together on one performance editor.

```
13.36.3.13 m_options
```

```
options* seq64::mainwnd::m_options [private]
```

13.36.3.14 m_main_cursor

```
Gdk::Cursor seq64::mainwnd::m_main_cursor [private]
```

```
13.36.3.15 m_image_play
Gtk::Image* seq64::mainwnd::m_image_play [private]
13.36.3.16 m_button_learn
Gtk::Button* seq64::mainwnd::m_button_learn [private]
13.36.3.17 m_button_stop
Gtk::Button* seq64::mainwnd::m_button_stop [private]
13.36.3.18 m_button_play
Gtk::Button* seq64::mainwnd::m_button_play [private]
If configured to support pause, it also supports the pause pixmap and functionality.
13.36.3.19 m_button_perfedit
Gtk::Button* seq64::mainwnd::m_button_perfedit [private]
13.36.3.20 m button jack
Gtk::Button* seq64::mainwnd::m_button_jack [private]
13.36.3.21 m_adjust_bpm
Gtk::Adjustment* seq64::mainwnd::m_adjust_bpm [private]
BPM adjustment object.
13.36.3.22 m_spinbutton_bpm
Gtk::SpinButton* seq64::mainwnd::m_spinbutton_bpm [private]
```

```
13.36.3.23 m_adjust_ss
```

Gtk::Adjustment* seq64::mainwnd::m_adjust_ss [private]

Screenset adjustment.

13.36.3.24 m_spinbutton_ss

Gtk::SpinButton* seq64::mainwnd::m_spinbutton_ss [private]

13.36.3.25 m_adjust_load_offset

```
Gtk::Adjustment* seq64::mainwnd::m_adjust_load_offset [private]
```

These controls are used in the File / Import dialog to change where the imported file will be loaded in the sequences space, which ranges from 0 to 1024 in blocks of 32 patterns.Load number for import.

13.36.3.26 m_spinbutton_load_offset

Gtk::SpinButton* seq64::mainwnd::m_spinbutton_load_offset [private]

13.36.3.27 m_entry_notes

```
Gtk::Entry* seq64::mainwnd::m_entry_notes [private]
```

This is just a long text-edit field that can be used to enter a long name or a short description of the current screenset.

13.36.3.28 m_is_running

bool seq64::mainwnd::m_is_running [private]

13.36.3.29 m_timeout_connect

sigc::connection seq64::mainwnd::m_timeout_connect [private]

13.36.3.30 m_menu_mode

bool seq64::mainwnd::m_menu_mode [private]

This is a "stazed" feature that might be generally useful.

13.36.3.31 m_call_seq_edit

bool seq64::mainwnd::m_call_seq_edit [private]

Currently, the hard-wired key for this function is the equals key.

13.36.3.32 m_call_seq_eventedit

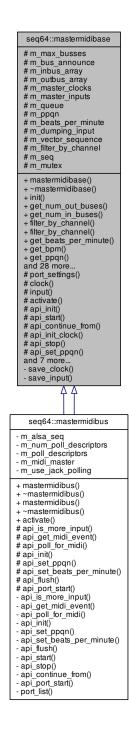
bool seq64::mainwnd::m_call_seq_eventedit [private]

Currently, the hard-wired key for this function is the minus key.

13.37 seq64::mastermidibase Class Reference

The class that "supervises" all of the midibus objects?

Inheritance diagram for seq64::mastermidibase:



Public Member Functions

- mastermidibase (int ppqn=SEQ64_USE_DEFAULT_PPQN, midibpm bpm=SEQ64_DEFAULT_BPM)
 - The mastermidibase default constructor fills the array with our busses.
- virtual ∼mastermidibase ()

The virtual destructor deletes all of the output busses, clears out the ALSA events, stops and frees the queue, and closes ALSA for this application.

virtual void init (int ppqn, midibpm bpm)

Initialize the mastermidibus using the implementation-specific API function.

• int get_num_out_buses () const

'Getter' function for member m_num_out_buses

int get_num_in_buses () const

'Getter' function for member m_num_in_buses

· bool filter by channel () const

'Getter' function for member m_filter_by_channel

void filter_by_channel (bool flag)

'Setter' function for member m_filter_by_channel

• midibpm get_beats_per_minute () const

'Getter' function for member m_beats_per_minute

midibpm get_bpm () const

'Getter' function for member m_beats_per_minute This is a second version.

int get_ppqn () const

'Getter' function for member m_ppqn

• bool is_dumping () const

'Getter' function for member m dumping input

• sequence * get_sequence () const

'Getter' function for member m_seq

· void start ()

Starts all of the configured output busses up to m_num_out_buses.

void stop ()

Stops each of the MIDI output busses.

void port_start (int client, int port)

Start the given MIDI port.

void port_exit (int client, int port)

Turn off the given port for the given client.

void play (bussbyte bus, event *e24, midibyte channel)

Handle the playing of MIDI events on the MIDI buss given by the parameter, as long as it is a legal buss number.

void continue_from (midipulse tick)

Gets the MIDI output busses running again.

void init_clock (midipulse tick)

Initializes the clock of each of the MIDI output busses.

void set_clock (midipulse tick)

Generates the MIDI clock for each of the output busses.

void sysex (event *event)

Handle the sending of SYSEX events.

• void print ()

Print some information about the available MIDI input and output busses.

• void flush ()

Flushes our local queue events out The implementation-specific API function is called.

void set_sequence_input (bool state, sequence *seq)

Set the input sequence object, and set the $m_dumping_input$ value to the given state.

void dump midi input (event in)

This function augments the recording functionality by looking for a sequence that has a matching channel number, logging the event to that sequence, and then immediately exiting.

• bool initialize buses ()

Initializes all fo the busses in the input and output buss arrays.

· std::string get midi out bus name (bussbyte bus)

Get the MIDI output buss name for the given (legal) buss number.

• std::string get_midi_in_bus_name (bussbyte bus)

Get the MIDI input buss name for the given (legal) buss number.

int poll_for_midi ()

Initiate a poll() on the existing poll descriptors.

bool is_more_input ()

Test the sequencer to see if any more input is pending.

bool get midi event (event *in)

Grab a MIDI event via the currently-selected MIDI API.

bool set_clock (bussbyte bus, clock_e clock_type)

Set the clock for the given (legal) buss number.

bool set input (bussbyte bus, bool inputing)

Set the status of the given input buss, if a legal buss number.

bool get_input (bussbyte bus)

Get the input for the given (legal) buss number.

• bool is_input_system_port (bussbyte bus)

Get the system-buss status for the given (legal) buss number.

clock_e get_clock (bussbyte bus)

Gets the clock setting for the given (legal) buss number.

void set_ppqn (int ppqn)

Set the PPQN value (parts per quarter note).

void set_beats_per_minute (midibpm bpm)

Set the BPM value (beats per minute).

Protected Member Functions

- void port_settings (const std::vector< clock_e > &clocks, const std::vector< bool > &inputs)
- clock_e clock (int bus)
- · bool input (int bus)
- virtual bool activate ()

Initializes and ctivates the busses, in a partly API-dependent manner.

- virtual void api_init (int ppqn, midibpm bpm)=0
- virtual void api start ()

Provides MIDI API-specific functionality for the start() function.

virtual void api_continue_from (midipulse)

Provides MIDI API-specific functionality for the continue_from() function.

virtual void api_init_clock (midipulse)

Provides MIDI API-specific functionality for the init_clock() function.

virtual void api_stop ()

Provides MIDI API-specific functionality for the stop() function.

virtual void api_set_ppqn (int)

Provides MIDI API-specific functionality for the set_ppqn() function.

virtual void api_set_beats_per_minute (midibpm)

Provides MIDI API-specific functionality for the set_beats_per_minute() function.

virtual void api_flush ()

Provides MIDI API-specific functionality for the flush() function.

virtual void api_clock ()

Provides MIDI API-specific functionality for the clock() function.

- virtual void api_port_start (int, int)
- virtual bool api_is_more_input ()=0
- virtual bool api get midi event (event *inev)=0
- virtual int api_poll_for_midi ()=0

Protected Attributes

• int m max busses

The maximum number of busses supported.

• midibus * m bus announce

MIDI buss announcer?

busarray m_inbus_array

Encapsulates information about the input busses.

busarray m_outbus_array

Encapsulates information about the output busses.

std::vector < clock_e > m_master_clocks

Saves the clock settings obtained from the "rc" (options) file so that they can be loaded into the mastermidibus once it is created.

• std::vector< bool > m_master_inputs

Saves the input settings obtained from the "[midi-input] section of the "rc" (options) file, so that they can be loaded into the mastermidibus once it is created.

int m_queue

The ID of the MIDI queue.

• int m_ppqn

Resolution in parts per quarter note.

· midibpm m beats per minute

BPM (beats per minute).

• bool m_dumping_input

For dumping MIDI input to a sequence for recording.

std::vector < sequence * > m vector sequence

Used for the new "stazed" feature of filtering MIDI channels so that a sequence gets only the channels meant for it.

bool m_filter_by_channel

If true, the m_vector_sequence container is used to divert incoming data to the sequence that has the channel it is meant for.

• sequence * m_seq

Points to the sequence object.

• mutex m_mutex

The locking mutex.

Private Member Functions

- bool save_clock (bussbyte bus, clock_e clock)
- · bool save_input (bussbyte bus, bool inputing)

Saves the input status (as selected in the MIDI Input tab).

Friends

- · class perform
- · class midi_alsa_info

13.37.1 Constructor & Destructor Documentation

13.37.1.1 mastermidibase()

```
seq64::mastermidibase::mastermidibase (
    int ppqn = SEQ64_USE_DEFAULT_PPQN,
    midibpm bpm = SEQ64_DEFAULT_BPM )
```

Parameters

ppqn	Provides the PPQN value for this object. However, in most cases, the default, SEQ64_USE_DEFAULT_PPQN should be specified. Then the caller of this constructor should call mastermidibase::set_ppqn() to set up the proper PPQN value.
bpm	Provides the beats per minute value, which defaults to c_beats_per_minute.

13.37.1.2 \sim mastermidibase()

```
seq64::mastermidibase::\sim mastermidibase ( ) [virtual]
```

Valgrind indicates we might have issues caused by the following functions:

```
- snd_config_hook_load()
- snd_config_update_r() via snd_seq_open()
- _dl_init() and other GNU function
- init_gtkmm_internals() [version 2.4]
```

13.37.2 Member Function Documentation

13.37.2.1 init()

A return value would be nice.

Parameters

ppqn	The PPQN value to which to initialize the master MIDI buss.
bpm	The beats/minute value to which to initialize the master MIDI buss.

13.37.2.2 get_num_out_buses()

```
int seq64::mastermidibase::get_num_out_buses ( ) const [inline]
```

13.37.2.3 get_num_in_buses()

```
int seq64::mastermidibase::get_num_in_buses ( ) const [inline]
```

```
13.37.2.4 filter_by_channel() [1/2]
bool seq64::mastermidibase::filter_by_channel ( ) const [inline]
13.37.2.5 filter_by_channel() [2/2]
void seq64::mastermidibase::filter_by_channel (
            bool flag ) [inline]
13.37.2.6 get_beats_per_minute()
midibpm seq64::mastermidibase::get_beats_per_minute ( ) const [inline]
13.37.2.7 get_bpm()
midibpm seq64::mastermidibase::get_bpm ( ) const [inline]
13.37.2.8 get_ppqn()
int seq64::mastermidibase::get_ppqn ( ) const [inline]
13.37.2.9 is_dumping()
bool seq64::mastermidibase::is_dumping ( ) const [inline]
13.37.2.10 get_sequence()
sequence* seq64::mastermidibase::get_sequence ( ) const [inline]
```

```
13.37.2.11 start()
void seq64::mastermidibase::start ( )
```

Calls the implementation-specific API function for starting.

Threadsafe

```
13.37.2.12 stop()
void seq64::mastermidibase::stop ( )
```

Then calls the implementation-specific API function to finalize the stoppage. (See the ALSA implementation in the seq_alsamidi library, for example. It is the original Sequencer64 implementation.)

Threadsafe

This function is called by api_get_midi_event() when the ALSA event SND_SEQ_EVENT_PORT_START is received. Unlike port_exit(), the port_start() function does rely on API-specific code, so we do need to create a virtual api_port_start() function to implement the port-start event.

Threadsafe Quite a lot is done during the lock for the ALSA implimentation.

Parameters

client Provides the client number, which		Provides the client number, which is actually an ALSA concept.
	port	Provides the client port, which is actually an ALSA concept.

```
13.37.2.14 port_exit()
```

Both the input and output busses for the given client are stopped: that is, set to inactive.

This function is called by api_get_midi_event() when the ALSA event SND_SEQ_EVENT_PORT_EXIT is received. Since port_exit() has no direct API-specific code in it, we do not need to create a virtual api_port_exit() function to implement the port-exit event.

Threadsafe

Parameters

clier	The client to be matched and acted on. This value is actually an ALSA concept.	
port	The port to be acted on. Both parameter must be matched before the buss is made inactive. This value is actually an ALSA concept.	

13.37.2.15 play()

```
void seq64::mastermidibase::play (
          bussbyte bus,
          event * e24,
          midibyte channel )
```

There's currently no implementation-specific API function here.

Threadsafe

Parameters

bus The buss to start play on. Ooh, we just noticed that value should be checked be	
e24	The seq24 event to play on the buss. For speed, we don't bother to check the pointer.
channel	The channel on which to play the event.

13.37.2.16 continue_from()

This function calls the implementation-specific API function, and then calls midibus::continue_from() for all of the MIDI output busses.

Threadsafe

Parameters

```
tick Provides the tick value to continue from.
```

13.37.2.17 init_clock()

Calls the implementation-specific API function, and then calls midibus::init_clock() for each of the MIDI output busses.

Threadsafe

Parameters

tick Provides the tick value with which to initialize the buss clock.

Also calls the api_clock() function, which does nothing for the original ALSA implementation and the PortMidi implementation.

Threadsafe

Parameters

tick Provides the tick value with which to set the buss clock.

13.37.2.19 sysex()

The event is sent to all MIDI output busses. Then flush() is called.

There's currently no implementation-specific API function for this call.

Threadsafe

Parameters

ev Provides the event pointer to be set.

13.37.2.20 print()

```
void seq64::mastermidibase::print ( )
```

```
13.37.2.21 flush()
```

```
void seq64::mastermidibase::flush ( )
```

For example, ALSA provides a function to "drain" the output.

Threadsafe

```
13.37.2.22 set_sequence_input()
```

The portmidi version only sets m_seq and m_dumping_input, but it seems like all the code below would apply to any mastermidibus.

Threadsafe

Parameters

sta	ė	Provides the dumping-input (recording) state to be set.
sec	1	Provides the sequence object to be logged as the mastermidibase's sequence. Can also be used to set
		a null pointer, to disable the sequence setting.

13.37.2.23 dump_midi_input()

Parameters

ev The event that was recorded, passed as a copy.

13.37.2.24 initialize_buses()

```
bool seq64::mastermidibase::initialize_buses ( )
```

Returns

Returns true if both busses were successfully initialized.

13.37.2.25 get_midi_out_bus_name()

This function is used for display purposes, and is also written to the options ("rc") file.

This function adds the retrieval of client and port numbers that are not needed in the portmidi implementation, but seem generally useful to support in all implementations.

Also, if the client name is already part of the port name, as in "client:client port 0", then we remove the "client:" portion to make the listing look cleaner.

Parameters

bus Provides the output buss number. Checked before usage. Actually should now be an index number

Returns

Returns the buss name as a standard C++ string. Also contains an indication that the buss is disconnected or unconnected. If the buss number is illegal, this string is empty.

13.37.2.26 get_midi_in_bus_name()

This function adds the retrieval of client and port numbers that are not needed in the portmidi implementation, but seem generally useful to support in all implementations.

Parameters

bus	Provides the input buss number.

Returns

Returns the buss name as a standard C++ string. Also contains an indication that the buss is disconnected or unconnected.

13.37.2.27 poll_for_midi()

```
int seq64::mastermidibase::poll_for_midi ( )
```

This base-class implementation could be made identical to portmidi's poll_for_midi() function, maybe. But currently it is better just call the implementation-specific API function.

Do we need to use a mutex lock? NO! It causes a deadlock!!!

Returns

Returns the result of the poll, or 0 if the API is not supported.

13.37.2.28 is_more_input()

```
bool seq64::mastermidibase::is_more_input ( )
```

Calls the implementation-specific API function.

Note that the ALSA implementation calls a single "input-pending" function, while the PortMidi implementation loops through all of the input midibus objects, calling the poll for midi() function of each.

Threadsafe

Returns

Returns true if ALSA is supported, and the returned size is greater than 0, or false otherwise.

13.37.2.29 get_midi_event()

Parameters

ev The event to be set based on the found input event.

```
13.37.2.30 set_clock() [2/2]
```

The legality checks are a little loose, however.

There's currently no implementation-specific API function here.

Threadsafe

Parameters

bus	The buss to start play on. Checked before usage.	
clocktype	The type of clock to be set, either "off", "pos", or "mod", as noted in the midibus_common module.	

13.37.2.31 set_input()

```
bool seq64::mastermidibase::set_input (
          bussbyte bus,
          bool inputing)
```

Why is another buss-count constant, and a global one at that, being used? And I thought there was only one input buss anyway! Well, there is only one ALSA input buss, but more can be used with JACK, apparently.

There's currently no implementation-specific API function here.

Threadsafe

Parameters

bus	Provides the buss number.
inputing	True if the input bus will be inputting MIDI data.

Returns

Returns true if the input buss array item could be set and then saved into the status container.

13.37.2.32 get_input()

```
bool seq64::mastermidibase::get_input (
          bussbyte bus )
```

There's currently no implementation-specific API function here.

Parameters

bus Provides the buss no	umber.
--------------------------	--------

Returns

Returns the value of the busarray::get_input(bus) call.

13.37.2.33 is_input_system_port()

Parameters

bus Provides the buss number.

Returns

Returns the value of the busarray::get_input(bus) call.

```
13.37.2.34 get_clock()
clock_e seq64::mastermidibase::get_clock (
```

bussbyte bus)

There's currently no implementation-specific API function here.

Parameters

bus Provides the buss number to read. Checked before usage.

Returns

If the buss number is legal, and the buss is active, then its clock setting is returned. Otherwise, e_clock_off is returned.

13.37.2.35 set_ppqn()

Then call the implementation-specific API function to complete the PPQN setting.

Threadsafe

Parameters

ppqn The PPQN value to be set.

13.37.2.36 set_beats_per_minute()

Then call the implementation-specific API function to complete the BPM setting.

Threadsafe

Parameters

```
bpm | Provides the beats-per-minute value to set.
```

```
13.37.2.37 port_settings()
```

13.37.2.38 clock()

13.37.2.39 input()

13.37.2.40 activate()

```
virtual bool seq64::mastermidibase::activate ( ) [inline], [protected], [virtual]
```

Currently re-implemented only in the rtmidi JACK API.

Reimplemented in seq64::mastermidibus.

```
13.37.2.41 api_init()
```

Implemented in seq64::mastermidibus, and seq64::mastermidibus.

```
13.37.2.42 api_start()
virtual void seq64::mastermidibase::api_start ( ) [inline], [protected], [virtual]
Reimplemented in seq64::mastermidibus.
13.37.2.43 api_continue_from()
virtual void seq64::mastermidibase::api_continue_from (
             midipulse ) [inline], [protected], [virtual]
Reimplemented in seq64::mastermidibus.
13.37.2.44 api_init_clock()
virtual void seq64::mastermidibase::api_init_clock (
             midipulse ) [inline], [protected], [virtual]
13.37.2.45 api_stop()
virtual void seq64::mastermidibase::api_stop ( ) [inline], [protected], [virtual]
Reimplemented in seq64::mastermidibus.
13.37.2.46 api_set_ppqn()
virtual void seq64::mastermidibase::api_set_ppqn (
            int ) [inline], [protected], [virtual]
Reimplemented in seq64::mastermidibus, and seq64::mastermidibus.
13.37.2.47 api_set_beats_per_minute()
virtual void seq64::mastermidibase::api_set_beats_per_minute (
             midibpm ) [inline], [protected], [virtual]
```

Generated by Doxygen

Reimplemented in seq64::mastermidibus, and seq64::mastermidibus.

```
13.37.2.48 api_flush()
virtual void seq64::mastermidibase::api_flush ( ) [inline], [protected], [virtual]
Reimplemented in seq64::mastermidibus, and seq64::mastermidibus.
13.37.2.49 api_clock()
virtual void seq64::mastermidibase::api_clock ( ) [inline], [protected], [virtual]
13.37.2.50 api_port_start()
virtual void seq64::mastermidibase::api_port_start (
             int ) [inline], [protected], [virtual]
Reimplemented in seq64::mastermidibus.
13.37.2.51 api_is_more_input()
virtual bool seq64::mastermidibase::api_is_more_input ( ) [protected], [pure virtual]
Implemented in seq64::mastermidibus, and seq64::mastermidibus.
13.37.2.52 api_get_midi_event()
virtual bool seq64::mastermidibase::api_get_midi_event (
             event * inev ) [protected], [pure virtual]
Implemented in seq64::mastermidibus, and seq64::mastermidibus.
13.37.2.53 api_poll_for_midi()
virtual int seq64::mastermidibase::api_poll_for_midi ( ) [protected], [pure virtual]
```

Implemented in seq64::mastermidibus, and seq64::mastermidibus.

13.37.2.54 save_clock()

```
bool seq64::mastermidibase::save_clock (
          bussbyte bus,
          clock_e clock ) [private]
```

13.37.2.55 save_input()

Now, we were checking this bus number against the size of the vector as gotten from the perform object, which it got the from the "rc" file's [midi-input] section. However, the "rc" file won't necessarily match what is on the system now. So we might have to adjust.

Do we also have to adjust the perform's vector?

Parameters

bus	Provides the buss number.
inputing	True if the input bus will be inputting MIDI data.

Returns

Returns true, always.

13.37.3 Friends And Related Function Documentation

13.37.3.1 perform

```
friend class perform [friend]
```

13.37.3.2 midi_alsa_info

```
friend class midi_alsa_info [friend]
```

13.37.4 Field Documentation

```
13.37.4.1 m_max_busses
int seq64::mastermidibase::m_max_busses [protected]
Set to c_max_busses (SEQ64_DEFAULT_BUSS_MAX = 32) for now.
13.37.4.2 m_bus_announce
midibus* seq64::mastermidibase::m_bus_announce [protected]
13.37.4.3 m_inbus_array
busarray seq64::mastermidibase::m_inbus_array [protected]
13.37.4.4 m_outbus_array
busarray seq64::mastermidibase::m_outbus_array [protected]
13.37.4.5 m_master_clocks
std::vector<clock_e> seq64::mastermidibase::m_master_clocks [protected]
13.37.4.6 m_master_inputs
std::vector<bool> seq64::mastermidibase::m_master_inputs [protected]
However, these items will be modified if the actual enumerated input ports do not match the port read from the "rc"
13.37.4.7 m_queue
int seq64::mastermidibase::m_queue [protected]
```

```
13.37.4.8 m_ppqn
```

```
int seq64::mastermidibase::m_ppqn [protected]
```

13.37.4.9 m_beats_per_minute

```
midibpm seq64::mastermidibase::m_beats_per_minute [protected]
```

We had to lengthen this name; way too easy to confuse it with "bpm" for "beats per measure".

13.37.4.10 m_dumping_input

```
bool seq64::mastermidibase::m_dumping_input [protected]
```

This value is set to true when a sequence editor window is open and the user has clicked the "record MIDI" or "thru MIDI" button.

13.37.4.11 m_vector_sequence

```
std::vector<sequence *> seq64::mastermidibase::m_vector_sequence [protected]
```

We want to make this a run-time, non-legacy option.

```
13.37.4.12 m_filter_by_channel
```

```
bool seq64::mastermidibase::m_filter_by_channel [protected]
```

13.37.4.13 m_seq

```
sequence* seq64::mastermidibase::m_seq [protected]
```

13.37.4.14 m_mutex

```
mutex seq64::mastermidibase::m_mutex [protected]
```

This object is passed to an automutex object that lends exception-safety to the mutex locking.

13.38 seq64::mastermidibus Class Reference

The class that "supervises" all of the midibus objects.

Inheritance diagram for seq64::mastermidibus:

```
seq64::mastermidibase
           # m_max_busses
# m_bus_announce
# m_inbus_array
# m_outbus_array
# m_master_clocks
         #m_master_clocks
#m_master_inputs
#m_queue
#m_ppqn
#m_beats_per_minute
#m_dumping_input
#m_vector_sequence
#m_filter_by_channel
#m_seq
#m_mutex
              + mastermidibase()
+ ~mastermidibase()
+ init()
       + ~mastermidibase()
+ init()
+ get_num_out_buses()
+ get_num_in_buses()
+ get_py_channel()
+ filter_by_channel()
+ get_beats_per_minute()
+ get_ppm()
+ get_ppm()
+ get_ppm()
and 28 more...
# port_settings()
# clock()
# input()
# api_init()
# api_init()
# api_start()
# api_init_clock()
# api_stop()
# api_stop()
# api_stop()
# api_set_ppqn()
and 7 more...
- save_clock()
- save_input()
                  seq64::mastermidibus
     - m_alsa_seq
- m_num_poll_descriptors
- m_poll_descriptors
- m_midi_master
     - m_use_jack_polling
    + mastermidibus()
  + ~mastermidibus()
+ mastermidibus()
+ ~mastermidibus()
+ mastermidibus()
+ activate()
# api_is_more input()
# api_get_midi_event()
# api_get_midi_event()
# api_poll_for_midi()
# api_set_ppqn()
# api_set_beats_per_minute()
# api_set_beats_per_minute()
# api_port_start()
- api_set_more input()
- api_get_midi_event()
- api_poll_for_midi()
- api_set_ppqn()
- api_set_ppqn()
- api_set_ppqn()
- api_start()
- api_start()
- api_start()
- api_continue_from()
- api_continue_from()
- api_port_start()
- port_list()
```

Public Member Functions

• mastermidibus (int ppqn=SEQ64_USE_DEFAULT_PPQN, midibpm bpm=SEQ64_DEFAULT_BPM)

The base-class constructor fills the array for our busses.

virtual ~mastermidibus ()

The destructor deletes all of the output busses, and terminates the Windows MIDI manager.

- mastermidibus (int ppqn=SEQ64 USE DEFAULT PPQN, midibpm bpm=SEQ64 DEFAULT BPM)
- virtual ∼mastermidibus ()
- virtual bool activate ()

Activates the mastermidibase code and the rtmidi_info object via its api_connect() function.

Protected Member Functions

- virtual bool api_is_more_input ()
- virtual bool api_get_midi_event (event *in)
- virtual int api poll for midi ()
- virtual void api_init (int ppqn, midibpm bpm)
- virtual void api set ppqn (int p)

Provides MIDI API-specific functionality for the set_ppqn() function.

virtual void api_set_beats_per_minute (midibpm b)

Provides MIDI API-specific functionality for the set_beats_per_minute() function.

- virtual void api_flush ()
- · virtual void api port start (mastermidibus &masterbus, int bus, int port)

Private Member Functions

• virtual bool api_is_more_input ()

Test the ALSA sequencer to see if any more input is pending.

virtual bool api get midi event (event *in)

Grab a MIDI event.

virtual int api_poll_for_midi ()

Initiate a poll() on the existing poll descriptors.

virtual void api_init (int ppqn, midibpm bpm)

Initializes the RtMidi implementation.

- virtual void api_set_ppqn (int ppqn)
- virtual void api_set_beats_per_minute (midibpm bpm)
- virtual void api_flush ()
- virtual void api start ()
- virtual void api_stop ()
- virtual void api continue from (midipulse tick)
- virtual void api_port_start (int client, int port)
- void port_list (const std::string &tag)

Shows a list of discovered ports in debug mode.

Private Attributes

snd_seq_t * m_alsa_seq

The ALSA sequencer client handle.

• int m_num_poll_descriptors

The number of descriptors for polling.

struct pollfd * m_poll_descriptors

Points to the list of descriptors for polling.

rtmidi_info m_midi_master

Holds the basic MIDI input and output information for later re-use in the construction of midibus objects.

bool m_use_jack_polling

Indicates we are running with JACK MIDI enabled, and need to use each port's ability to poll for and get MIDI events, rather than use ALSA's method of calling functions from the "MIDI master" object.

Friends

- class midi_alsa_info
- · class midi_jack_info

Additional Inherited Members

13.38.1 Detailed Description

This implementation uses the PortMidi library, which supports Linux and Windows, but not JACK or Mac OSX.

13.38.2 Constructor & Destructor Documentation

```
13.38.2.1 mastermidibus() [1/2]

seq64::mastermidibus::mastermidibus (
    int ppqn = SEQ64_USE_DEFAULT_PPQN,
    midibpm bpm = SEQ64_DEFAULT_BPM )
```

Parameters

ppqn	Provides the PPQN value for this object. However, in most cases, the default value,
	SEQ64_USE_DEFAULT_PPQN should be specified.
bpm	Provides the beats per minute value, which defaults to c_beats_per_minute.

13.38.3 Member Function Documentation

```
13.38.3.1 api_is_more_input() [1/2]
bool seq64::mastermidibus::api_is_more_input ( ) [private], [virtual]
Similar to api_poll_for_midi(), except it is threadsafe. We got some cleanup to do!
```

Threadsafe

Implements seq64::mastermidibase.

For the ALSA implementation, this call

Threadsafe

Implements seq64::mastermidibase.

```
13.38.3.3 api_poll_for_midi() [1/2]
int seq64::mastermidibus::api_poll_for_midi ( ) [private], [virtual]
```

This is a primitive poll, which exits when some data is obtained.

Implements seq64::mastermidibase.

Two different styles are supported. If the –manual-alsa-ports option is in force, then 16 virtual output ports and one virtual input port are created. They are given names that make it clear which application (seq64) has set them up. They are not connected to anything. The user will have to use a connection GUI (such as qjackctl) or a session manager to make the connections.

Otherwise, the system MIDI input and output ports are scanned (via the rtmidi_info member) and passed to the midibus constructor calls. For every MIDI input port found on the system, this function creates a corresponding output port, and connects to the system MIDI input. For example, for an input port found called "qmidiarp:in 1", we want to create a "shadow" output port called "seq64:qmidiarp in 1".

For every MIDI output found on the system this function creates a corresponding input port, and connects it to the system MIDI output. For For example, for an output port found called "qmidiarp:out 1", we want to create a "shadow" input port called "seq64:qmidiarp out 1".

This code creates a midibus in the conventional manner. Then the busarray::add() function makes a new businfo object with the desired "output" and "isvirtual" parameters; the businfo object then decides whether to call init_in(), init_out(), init_in_sub(), or init_out_sub().

Are these good conventions, or potentially confusing to users? They match what the legacy seq24 and sequencer64 do for ALSA.

Parameters

ppqn	Provides the (possibly new) value of PPQN to set. ALSA has a function that sets its idea of the PPQN. JACK, as far as we know, does not.
bpm	Provides the (possibly new) value of BPM (beats per minute) to set. ALSA has a function that sets its idea of the BPM. JACK, as far as we know, does not.

Implements seq64::mastermidibase.

Reimplemented from seq64::mastermidibase.

Reimplemented from seq64::mastermidibase.

```
13.38.3.7 api_flush() [1/2]
virtual void seq64::mastermidibus::api_flush ( ) [private], [virtual]
```

Reimplemented from seq64::mastermidibase.

```
13.38.3.8 api_start()
virtual void seq64::mastermidibus::api_start ( ) [private], [virtual]
```

Reimplemented from seq64::mastermidibase.

```
13.38.3.9 api_stop()
virtual void seq64::mastermidibus::api_stop ( ) [private], [virtual]
Reimplemented from seq64::mastermidibase.
13.38.3.10 api_continue_from()
virtual void seq64::mastermidibus::api_continue_from (
             midipulse tick ) [private], [virtual]
Reimplemented from seq64::mastermidibase.
13.38.3.11 api_port_start() [1/2]
virtual void seq64::mastermidibus::api_port_start (
             int client,
             int port ) [private], [virtual]
Reimplemented from seq64::mastermidibase.
13.38.3.12 activate()
bool seq64::mastermidibus::activate ( ) [virtual]
Reimplemented from seq64::mastermidibase.
13.38.3.13 api_is_more_input() [2/2]
virtual bool seq64::mastermidibus::api_is_more_input ( ) [protected], [virtual]
Implements seq64::mastermidibase.
13.38.3.14 api_get_midi_event() [2/2]
virtual bool seq64::mastermidibus::api_get_midi_event (
             event * in ) [protected], [virtual]
```

Implements seq64::mastermidibase.

```
13.38.3.15 api_poll_for_midi() [2/2]
virtual int seq64::mastermidibus::api_poll_for_midi ( ) [protected], [virtual]
Implements seq64::mastermidibase.
13.38.3.16 api_init() [2/2]
virtual void seq64::mastermidibus::api_init (
             int ppqn,
             midibpm bpm ) [protected], [virtual]
Implements seq64::mastermidibase.
13.38.3.17 api_set_ppqn() [2/2]
virtual void seq64::mastermidibus::api_set_ppqn (
             int p ) [inline], [protected], [virtual]
Reimplemented from seq64::mastermidibase.
13.38.3.18 api_set_beats_per_minute() [2/2]
virtual void seq64::mastermidibus::api_set_beats_per_minute (
             midibpm b ) [inline], [protected], [virtual]
Reimplemented from seq64::mastermidibase.
13.38.3.19 api_flush() [2/2]
virtual void seq64::mastermidibus::api_flush ( ) [inline], [protected], [virtual]
Reimplemented from seq64::mastermidibase.
13.38.3.20 api_port_start() [2/2]
virtual void seq64::mastermidibus::api_port_start (
             mastermidibus & masterbus,
             int bus,
             int port ) [inline], [protected], [virtual]
13.38.3.21 port_list()
void seq64::mastermidibus::port_list (
             const std::string & tag ) [private]
```

Parameters

tag | Provides a string label indicate the context of this list.

13.38.4 Friends And Related Function Documentation

13.38.4.1 midi_alsa_info

friend class midi_alsa_info [friend]

13.38.4.2 midi_jack_info

friend class midi_jack_info [friend]

13.38.5 Field Documentation

13.38.5.1 m_alsa_seq

 $\verb| snd_seq_t* seq64::mastermidibus::m_alsa_seq [private]|\\$

13.38.5.2 m_num_poll_descriptors

int seq64::mastermidibus::m_num_poll_descriptors [private]

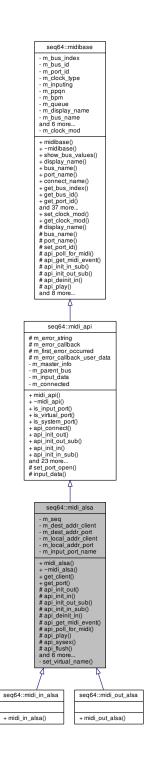
13.38.5.3 m_poll_descriptors

struct pollfd* seq64::mastermidibus::m_poll_descriptors [private]

13.38.5.4 m_midi_master rtmidi_info seq64::mastermidibus::m_midi_master [private] 13.38.5.5 m_use_jack_polling bool seq64::mastermidibus::m_use_jack_polling [private] 13.39 seq64::midi_alsa Class Reference

This class implements the ALSA version of the midi_api.

Inheritance diagram for seq64::midi_alsa:



Public Member Functions

• midi_alsa (midibus &parentbus, midi_info &masterinfo)

Provides a constructor with client number, port number, ALSA sequencer support, name of client, name of port, etc., mostly contained within an already-initialized midi_info object.

virtual ∼midi_alsa ()

A rote empty destructor.

· virtual int get_client () const

'Getter' function for member m_dest_addr_client The address of client.

virtual int get_port () const

'Getter' function for member m_dest_addr_port Can we replace it with get_port_id()?

Protected Member Functions

• virtual bool api_init_out ()

Initialize the MIDI output port.

virtual bool api_init_in ()

Initialize the MIDI input port.

virtual bool api_init_out_sub ()

Initialize the output in a different way.

virtual bool api_init_in_sub ()

Initialize the output in a different way?

virtual bool api_deinit_in ()

Deinitialize the MIDI input.

virtual bool api_get_midi_event (event *)

ALSA get MIDI events via the midi_alsa_info object at present.

virtual int api poll for midi ()

This function is supposed to poll for MIDI data, but the current ALSA implementation DOES NOT USE THIS FUN← CTION.

virtual void api_play (event *e24, midibyte channel)

This play() function takes a native event, encodes it to an ALSA MIDI sequencer event, sets the broadcasting to the subscribers, sets the direct-passing mode to send the event without queueing, and puts it in the queue.

virtual void api_sysex (event *e24)

Takes a native SYSEX event, encodes it to an ALSA event, and then puts it in the queue.

virtual void api_flush ()

Flushes our local queue events out into ALSA.

- virtual void api_continue_from (midipulse tick, midipulse beats)
- virtual void api_start ()

This function gets the MIDI clock a-runnin', if the clock type is not e_clock_off.

virtual void api_stop ()

Stop the MIDI buss.

· virtual void api clock (midipulse tick)

Generates the MIDI clock, starting at the given tick value.

- virtual void api_set_ppqn (int ppqn)
- virtual void api_set_beats_per_minute (midibpm bpm)

Set the BPM value (beats per minute).

Private Member Functions

• bool set virtual name (int portid, const std::string &portname)

Gets information directly from ALSA.

Private Attributes

```
• snd_seq_t *const m_seq
```

ALSA sequencer client handle.

· const int m_dest_addr_client

Destination address of client.

• const int m_dest_addr_port

Destination port of client.

· const int m_local_addr_client

Local address of client.

• int m_local_addr_port

Local port of client.

• const std::string m_input_port_name

Holds the port name for the ALSA MIDI input port.

Additional Inherited Members

13.39.1 Constructor & Destructor Documentation

```
13.39.1.1 midi_alsa()
```

This constructor is the only one that is used for the MIDI input and output busses, whether the [manual-alsa-ports] option is in force or not. The actual setup of a normal or virtual port is done in the api_*_init_*() routines.

Also used for the announce buss, and in the mastermidi_alsa::port_start() function. There's currently some overlap between local/dest client and port numbers and the buss and port numbers of the new midibase interface.

Also, note that the optionsfile module uses the master buss to get the buss names when it writes the file.

Parameters

parentbus	Provides much of the infor about this ALSA buss.
masterinfo	Provides the information about the desired port, and more.

```
13.39.1.2 \simmidi_alsa() seq64::midi_alsa::\simmidi_alsa ( ) [virtual]
```

13.39.2 Member Function Documentation

```
13.39.2.1 get_client()
virtual int seq64::midi_alsa::get_client ( ) const [inline], [virtual]
Can we replace it with get_client_id()?

13.39.2.2 get_port()
virtual int seq64::midi_alsa::get_port ( ) const [inline], [virtual]

13.39.2.3 api_init_out()
bool seq64::midi_alsa::api_init_out ( ) [protected], [virtual]
```

This initialization is done when the "manual ALSA ports" option is not in force.

This initialization is like the "open_port()" function of the RtMidi library, with the addition of the snd_seq_connect_to() call involving the local and destination ports.

Tricky Code One important thing to note is that this output port is initialized with the SND_SEQ_PORT_CAP_R ← EAD flag, which means this is really an input port. We connect this input port with a system output port that was discovered. This is backwards of the way RtMidi does it.

Returns

Returns true unless setting up ALSA MIDI failed in some way.

Implements seq64::midi_api.

```
13.39.2.4 api_init_in()
bool seq64::midi_alsa::api_init_in ( ) [protected], [virtual]
```

Subscription handlers:

In ALSA library, subscription is done via snd_seq_subscribe_port() function. It takes the argument of snd_seq_\circ
port_subscribe_t record pointer. Suppose that you have a client which will receive data from a MIDI input device.
The source and destination addresses are like the below:

```
snd_seq_addr_t sender, dest;
sender.client = MIDI_input_client;
sender.port = MIDI_input_port;
dest.client = my_client;
dest.port = my_port;

To set these values as the connection call like this.

snd_seq_port_subscribe_t *subs;
snd_seq_port_subscribe_alloca(&subs);
snd_seq_port_subscribe_set_dest(subs, &sender);
snd_seq_port_subscribe_set_dest(subs, &dest);
snd_seq_subscribe_port(handle, subs);
```

Tricky Code One important thing to note is that this input port is initialized with the SND_SEQ_PORT_CAP_W ← RITE flag, which means this is really an output port. We connect this output port with a system input port that was discovered. This is backwards of the way RtMidi does it.

Returns

Returns true unless setting up ALSA MIDI failed in some way.

Implements seq64::midi api.

```
13.39.2.5 api_init_out_sub()
bool seq64::midi_alsa::api_init_out_sub ( ) [protected], [virtual]
```

This version of initialization is used by mastermidi_alsa in the "manual ALSA ports" clause. This code is also very similar to the same function in the midibus::api_init_out_sub() function of midibus::api_init_out_sub().

Returns

Returns true unless setting up the ALSA port failed in some way.

Implements seq64::midi_api.

```
13.39.2.6 api_init_in_sub()
bool seq64::midi_alsa::api_init_in_sub ( ) [protected], [virtual]
```

Returns

Returns true unless setting up the ALSA port failed in some way.

Implements seq64::midi_api.

```
13.39.2.7 api_deinit_in()
bool seq64::midi_alsa::api_deinit_in ( ) [protected], [virtual]
```

Set the input and the output ports. The destination port is actually our local port.

Returns

Returns true, unless an error occurs.

Implements seq64::midi_api.

```
13.39.2.8 api_get_midi_event()
```

Implements seq64::midi_api.

```
13.39.2.9 api_poll_for_midi()
```

```
int seq64::midi_alsa::api_poll_for_midi ( ) [protected], [virtual]
```

TODO? See seq_alsamidi's mastermidibus::api_poll_for_midi(). Right now we'd need to forward this call to midi← _alsa_info.

```
return poll(m_poll_descriptors, m_num_poll_descriptors, 1000);
```

This kills startup:

```
return master_info().api_poll_for_midi();
```

Returns

Always returns 0.

Implements seq64::midi_api.

13.39.2.10 api_play()

Threadsafe

Parameters

e24	The event to be played on this bus. For speed, we don't bother to check the pointer.
channel	The channel of the playback.

Implements seq64::midi_api.

13.39.2.11 api_sysex()

Parameters

e24 The event to be handled.

Implements seq64::midi_api.

```
13.39.2.12 api_flush()
void seq64::midi_alsa::api_flush ( ) [protected], [virtual]
This is also a midi_alsa_info function.
Implements seq64::midi_api.
13.39.2.13 api_continue_from()
void seq64::midi_alsa::api_continue_from (
              midipulse tick,
              midipulse beats ) [protected], [virtual]
Implements seq64::midi_api.
13.39.2.14 api_start()
void seq64::midi_alsa::api_start ( ) [protected], [virtual]
Implements seq64::midi_api.
13.39.2.15 api_stop()
void seq64::midi_alsa::api_stop ( ) [protected], [virtual]
Implements seq64::midi_api.
13.39.2.16 api_clock()
void seq64::midi_alsa::api_clock (
             midipulse tick ) [protected], [virtual]
Threadsafe
Parameters
 tick Provides the starting tick, unused in the ASLA implementation.
```

Implements seq64::midi_api.

```
13.39.2.17 api_set_ppqn()
```

Implements seq64::midi_api.

13.39.2.18 api_set_beats_per_minute()

This is done by creating an ALSA tempo structure, adding tempo information to it, and then setting the ALSA sequencer object with this information.

We fill the ALSA tempo structure (snd_seq_queue_tempo_t) with the current tempo information, set the BPM value, put it in the tempo structure, and give the tempo value to the ALSA queue.

Note

Consider using snd_seq_change_queue_tempo() here if the ALSA queue has already been started. It's arguments would be m_alsa_seq, m_queue, tempo (microseconds), and null.

Threadsafe

Parameters

bpm Provides the beats-per-minute value to set.

Implements seq64::midi_api.

13.39.2.19 set_virtual_name()

The problem this function solves is that the midibus constructor for a virtual ALSA port doesn't not have all of the information it needs at that point. Here, we can get this information and get the actual data we need to rename the port to something accurate. Same as the seq_alsamidi version of this function.

Returns

Returns true if all of the information could be obtained. If false is returned, then the caller should not use the side-effects.

Side-effect(s) Passes back the values found.

13.39.3 Field Documentation

```
13.39.3.1 m_seq
snd_seq_t* const seq64::midi_alsa::m_seq [private]
13.39.3.2 m_dest_addr_client
const int seq64::midi_alsa::m_dest_addr_client [private]
Could potentially be replaced by midibase::m_bus_id.
13.39.3.3 m_dest_addr_port
const int seq64::midi_alsa::m_dest_addr_port [private]
Could potentially be replaced by midibase::m port id.
13.39.3.4 m_local_addr_client
const int seq64::midi_alsa::m_local_addr_client [private]
13.39.3.5 m_local_addr_port
int seq64::midi_alsa::m_local_addr_port [private]
13.39.3.6 m_input_port_name
```

const std::string seq64::midi_alsa::m_input_port_name [private]

It is derived from the (optionally configured) official client name for the application with the word "in" appended.

13.40 seq64::midi_alsa_info Class Reference

The class for handling ALSA MIDI input.

Inheritance diagram for seq64::midi_alsa_info:

seq64::midi_info # m error string - m_midi_mode_input - m_input - m_output - m_bus_container - m_global_queue - m_midi_handle - m_app_name - m_ppqn - m_bpm + midi_info() + ~midi_info() + midi_mode() + midi_mode() + midi_handle() + input_ports() + output_ports() + full_port_count() + clear() + app_name() and 23 more... # add_bus() # global_queue() # midi_handle() # bus_container() - nc_midi_port_info() - ref_midi_port_info() seq64::midi_alsa_info - m alsa seq - m_num_poll_descriptors - m_poll_descriptors - sm_input_caps - sm_output_caps + midi_alsa_info() + ~midi_alsa_info() + seq() + api_get_midi_event() + api_poll_for_midi() + api_set_ppqn() + api_set_beats_per_minute() + api_port_start() + api_flush() - get_all_port_info()

Public Member Functions

midi_alsa_info (const std::string &appname, int ppqn=SEQ64_DEFAULT_PPQN, midibpm bpm=SEQ64_←
DEFAULT_BPM)

Principal constructor.

virtual ~midi_alsa_info ()

Destructor.

• snd_seq_t * seq ()

'Getter' function for member m_alsa_seq This is the platform-specific version of midi_handle().

virtual bool api_get_midi_event (event *inev)

Grab a MIDI event.

• virtual int api_poll_for_midi ()

Polls for any ALSA MIDI information using a timeout value of 1000 milliseconds.

virtual void api_set_ppqn (int p)

Sets the PPQN numeric value, then makes ALSA calls to set up the PPQ tempo.

virtual void api_set_beats_per_minute (midibpm b)

Sets the BPM numeric value, then makes ALSA calls to set up the BPM tempo.

virtual void api_port_start (mastermidibus &masterbus, int bus, int port)

Start the given ALSA MIDI port.

virtual void api_flush ()

Flushes our local queue events out into ALSA.

Private Member Functions

virtual int get_all_port_info ()

Gets information on ALL ports, putting input data into one midi_info container, and putting output data into another container.

Private Attributes

• snd_seq_t * m_alsa_seq

Holds the ALSA sequencer client pointer so that it can be used by the midibus objects.

• int m_num_poll_descriptors

The number of descriptors for polling.

struct pollfd * m poll descriptors

Points to the list of descriptors for polling.

Static Private Attributes

• static unsigned sm_input_caps

Flags that denote queries for input (read) ports.

• static unsigned sm_output_caps

Flags that denote queries for output (write) ports.

Additional Inherited Members

13.40.1 Constructor & Destructor Documentation

```
13.40.1.1 midi_alsa_info()
```

Parameters

appname	Provides the name of the application.
ppqn	Provides the PPQN value needed by this object.
bpm	Provides the beats/minute value needed by this object.

```
13.40.1.2 \simmidi_alsa_info() seq64::midi_alsa_info::\simmidi_alsa_info ( ) [virtual]
```

Closes a connection if it exists, shuts down the input thread, and then cleans up any API resources in use.

13.40.2 Member Function Documentation

First, a rather large buffer is allocated on the stack to hold the MIDI event data. Next, if the –alsa-manual-ports option is not in force, then we check to see if the event is a port-start, port-exit, or port-change event, and we process it, and are done.

Otherwise, we create a "MIDI event parser" and decode the MIDI event.

Parameters

inev	The event to be set based on the found input event.
------	---

Returns

This function returns false if we are not using virtual/manual ports and the event is an ALSA port-start, port-exit, or port-change event. It also returns false if there is no event to decode. Otherwise, it returns true.

We will only get EVENT_SYSEX on the first packet of MIDI data; the rest we have to poll for. SysEx processing is currently disabled.

Implements seq64::midi_info.

```
13.40.2.3 api_poll_for_midi()
int seq64::midi_alsa_info::api_poll_for_midi ( ) [virtual]
```

Returns

Returns the result of the call to poll() on the global ALSA poll descriptors.

Implements seq64::midi_info.

```
13.40.2.4 api_set_ppqn()
```

Parameters

```
p The desired new PPQN value to set.
```

Reimplemented from seq64::midi_info.

```
13.40.2.5 api_set_beats_per_minute()
```

Parameters

```
b The desired new BPM value to set.
```

Reimplemented from seq64::midi_info.

```
13.40.2.6 api_port_start()
```

```
int bus,
int port ) [virtual]
```

This function is called by api_get_midi_event() when an ALSA event SND_SEQ_EVENT_PORT_START is received.

- · Get the API's client and port information.
- · Do some capability checks.
- Find the client/port combination among the set of input/output busses. If it exists and is not active, then mark it as a replacement. If it is not a replacement, it will increment the number of input/output busses.

We can simplify this code a bit by using elements already present in midi_alsa_info.

Parameters

masterbus	Provides the object that is need to get access to the busses that need to be started.
bus	Provides the ALSA bus/client number.
port	Provides the ALSA client port.

Reimplemented from seq64::midi_info.

```
13.40.2.7 api_flush()
```

```
void seq64::midi_alsa_info::api_flush ( ) [virtual]
```

This is also a midi_alsa function.

Implements seq64::midi info.

```
13.40.2.8 get_all_port_info()
```

```
int seq64::midi_alsa_info::get_all_port_info ( ) [private], [virtual]
```

For ALSA input, the first item added is the ALSA MIDI system "announce" buss. It has the client:port value of "0:1", denoted by the ALSA macros SND_SEQ_CLIENT_SYSTEM:SND_SEQ_PORT_SYSTEM_ANNOUNCE.

Returns

Returns the total number of ports found. For an ALSA setup, finding no ALSA ports can be considered an error. However, finding no ports for other APIS may be fine. So, we set the result to -1 to flag a true error.

Implements seq64::midi_info.

13.40.3 Field Documentation

13.40.3.1 sm_input_caps

```
unsigned seq64::midi_alsa_info::sm_input_caps [static], [private]
```

13.40.3.2 sm_output_caps

```
unsigned seq64::midi_alsa_info::sm_output_caps [static], [private]
```

13.40.3.3 m_alsa_seq

```
snd_seq_t* seq64::midi_alsa_info::m_alsa_seq [private]
```

This is actually an opaque pointer; there is no way to get the actual fields in this structure; they can only be accessed through functions in the ALSA API.

13.40.3.4 m_num_poll_descriptors

```
int seq64::midi_alsa_info::m_num_poll_descriptors [private]
```

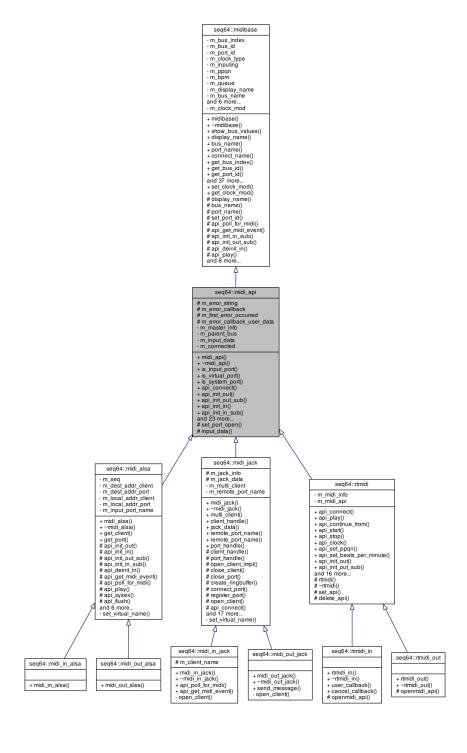
13.40.3.5 m_poll_descriptors

struct pollfd* seq64::midi_alsa_info::m_poll_descriptors [private]

13.41 seq64::midi_api Class Reference

Subclasses of midi_in_api and midi_out_api contain all API- and OS-specific code necessary to fully implement the rtmidi API.

Inheritance diagram for seq64::midi_api:



Public Member Functions

midi_api (midibus &parentbus, midi_info &masterinfo)

Principle constructor.

virtual ~midi_api ()

Destructor, needed because it is virtual.

- bool is_input_port () const
- · bool is virtual port () const
- · bool is_system_port () const
- · virtual bool api_connect ()

No code; only midi_jack overrides this function at present.

- virtual bool api init out ()=0
- virtual bool api_init_out_sub ()=0
- virtual bool api_init_in ()=0
- virtual bool api init in sub ()=0
- virtual bool api_deinit_in ()=0
- virtual bool api_get_midi_event (event *)=0
- virtual int api poll for midi ()=0
- virtual void api_play (event *e24, midibyte channel)=0
- virtual void api_sysex (event *e24)=0
- virtual void api continue from (midipulse tick, midipulse beats)=0
- virtual void api_start ()=0
- virtual void api stop ()=0
- virtual void api_flush ()=0
- virtual void api_clock (midipulse tick)=0
- virtual void api set ppqn (int ppqn)=0
- virtual void api set beats per minute (midibpm bpm)=0
- virtual std::string api get bus name ()
- virtual std::string api_get_port_name ()
- bool is_port_open () const

'Getter' function for member m_connected

midi_info & master_info ()

'Getter' function for member m_master_info

· const midi_info & master_info () const

'Getter' function for member m_master_info The const version.

midibus & parent_bus ()

'Getter' function for member m_parent_bus

• const midibus & parent_bus () const

'Getter' function for member m_parent_bus The const version.

void master_midi_mode (bool input)

 ${\it 'Getter' function for member m_master_info.midi_mode()}\ {\it This function makes it a bit simpler on the caller.}$

void error (rterror::Type type, const std::string &errorstring)

Provides an error handler that can support an error callback.

void user_callback (rtmidi_callback_t callback, void *userdata)

Wires in a MIDI input callback function.

• void cancel callback ()

Removes the MIDI input callback and some items related to it.

Protected Member Functions

void set_port_open ()

'Setter' function for member m_connected

rtmidi_in_data * input_data ()

'Getter' function for member &m_input_data

Protected Attributes

• std::string m_error_string

Holds the last error message, if in force.

rterror_callback m_error_callback

Holds the error callback function pointer, if any.

· bool m_first_error_occurred

Indicates that the first error has happened.

void * m_error_callback_user_data

Holds data needed by the error-callback.

Private Attributes

· midi info & m master info

Contains information about the ports (system or client) enumerated by the API.

• midibus & m_parent_bus

Contains a reference to the parent midibus/midibase object.

· rtmidi in data m input data

Although this really is useful only for MIDI input objects, the split of the midi_api is not as convenient for re-use as is the split for derived classes like midi_in_jack/midi_out_jack.

· bool m_connected

Set to true if the port was opened, activated, and connected without issue.

Additional Inherited Members

13.41.1 Detailed Description

Note that midi_in_api and midi_out_api are abstract base classes and cannot be explicitly instantiated. rtmidi_in and rtmidi_out will create instances of a midi_in_api or midi_out api subclass.

13.41.2 Constructor & Destructor Documentation

```
13.41.2.1 midi_api()
```

seq64::midi_api::~midi_api () [virtual]

13.41.3 Member Function Documentation

```
13.41.3.1 is_input_port()
bool seq64::midi_api::is_input_port ( ) const
13.41.3.2 is_virtual_port()
bool seq64::midi_api::is_virtual_port ( ) const
13.41.3.3 is_system_port()
bool seq64::midi_api::is_system_port ( ) const
13.41.3.4 api_connect()
virtual bool seq64::midi_api::api_connect ( ) [inline], [virtual]
Reimplemented in seq64::midi_jack, and seq64::rtmidi.
13.41.3.5 api_init_out()
virtual bool seq64::midi_api::api_init_out ( ) [pure virtual]
Implements seq64::midibase.
Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
13.41.3.6 api_init_out_sub()
virtual bool seq64::midi_api::api_init_out_sub ( ) [pure virtual]
Reimplemented from seq64::midibase.
Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
```

```
13.41.3.7 api_init_in()
virtual bool seq64::midi_api::api_init_in ( ) [pure virtual]
Implements seq64::midibase.
Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
13.41.3.8 api_init_in_sub()
virtual bool seq64::midi_api::api_init_in_sub ( ) [pure virtual]
Reimplemented from seq64::midibase.
Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
13.41.3.9 api_deinit_in()
virtual bool seq64::midi_api::api_deinit_in ( ) [pure virtual]
Reimplemented from seq64::midibase.
Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
13.41.3.10 api_get_midi_event()
virtual bool seq64::midi_api::api_get_midi_event (
              event * ) [pure virtual]
Reimplemented from seq64::midibase.
Implemented in seq64::midi_in_jack, seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
13.41.3.11 api_poll_for_midi()
virtual int seq64::midi_api::api_poll_for_midi ( ) [pure virtual]
```

Implemented in seq64::midi_in_jack, seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.

Generated by Doxygen

Reimplemented from seq64::midibase.

```
13.41.3.12 api_play()
```

Implements seq64::midibase.

Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.

```
13.41.3.13 api_sysex()
```

Reimplemented from seq64::midibase.

Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.

```
13.41.3.14 api_continue_from()
```

Implements seq64::midibase.

Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.

```
13.41.3.15 api_start()
```

```
virtual void seq64::midi_api::api_start ( ) [pure virtual]
```

Implements seq64::midibase.

Implemented in seq64::midi jack, seq64::midi alsa, and seq64::rtmidi.

```
13.41.3.16 api_stop()
```

```
virtual void seq64::midi_api::api_stop ( ) [pure virtual]
```

Implements seq64::midibase.

Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.

```
13.41.3.17 api_flush()
virtual void seq64::midi_api::api_flush ( ) [pure virtual]
Reimplemented from seq64::midibase.
Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
13.41.3.18 api_clock()
virtual void seq64::midi_api::api_clock (
             midipulse tick ) [pure virtual]
Implements seq64::midibase.
Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
13.41.3.19 api_set_ppqn()
virtual void seq64::midi_api::api_set_ppqn (
              int ppqn ) [pure virtual]
Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
13.41.3.20 api_set_beats_per_minute()
virtual void seq64::midi_api::api_set_beats_per_minute (
              midibpm bpm ) [pure virtual]
Implemented in seq64::midi_jack, seq64::midi_alsa, and seq64::rtmidi.
13.41.3.21 api_get_bus_name()
virtual std::string seq64::midi_api::api_get_bus_name ( ) [inline], [virtual]
13.41.3.22 api_get_port_name()
virtual std::string seq64::midi_api::api_get_port_name () [inline], [virtual]
Reimplemented in seq64::midi_jack.
```

```
13.41.3.23 is_port_open()
bool seq64::midi_api::is_port_open ( ) const [inline]
13.41.3.24 master_info() [1/2]
midi_info& seq64::midi_api::master_info ( ) [inline]
13.41.3.25 master_info() [2/2]
const midi_info& seq64::midi_api::master_info ( ) const [inline]
13.41.3.26 parent_bus() [1/2]
midibus& seq64::midi_api::parent_bus ( ) [inline]
13.41.3.27 parent_bus() [2/2]
const midibus& seq64::midi_api::parent_bus ( ) const [inline]
13.41.3.28 master_midi_mode()
void seq64::midi_api::master_midi_mode (
             bool input )
13.41.3.29 error()
void seq64::midi_api::error (
             rterror::Type type,
             const std::string & errorstring )
Exceptions
```

If \mid the error is not just a warning, then an rterror object is thrown.

Parameters

type	The type of the error.
errorstring	The error message, which gets copied if this is the first error.

13.41.3.30 user_callback()

We moved it into the base class, trading convenience for the chance of confusion.

Parameters

callback	Provides the callback function.
userdata	Provides the user data needed by the callback function.

13.41.3.31 cancel_callback()

```
void seq64::midi_api::cancel_callback ( )
```

We moved it into the base class, trading convenience for the chance of confusion.

```
13.41.3.32 set_port_open()
```

```
void seq64::midi_api::set_port_open ( ) [inline], [protected]
```

13.41.3.33 input_data()

```
rtmidi_in_data* seq64::midi_api::input_data ( ) [inline], [protected]
```

13.41.4 Field Documentation

13.41.4.1 m_master_info

```
midi_info& seq64::midi_api::m_master_info [private]
```

```
13.41.4.2 m_parent_bus
```

```
midibus& seq64::midi_api::m_parent_bus [private]
```

This object is needed to get parameters that are peculiar to the port as it is actually set up, rather than information from the midi_info object.

```
13.41.4.3 m_input_data
```

```
rtmidi_in_data seq64::midi_api::m_input_data [private]
```

13.41.4.4 m_connected

```
bool seq64::midi_api::m_connected [private]
```

13.41.4.5 m_error_string

```
std::string seq64::midi_api::m_error_string [protected]
```

This is an original RtMidi concept.

13.41.4.6 m_error_callback

```
rterror_callback seq64::midi_api::m_error_callback [protected]
```

This is an original RtMidi concept.

13.41.4.7 m_first_error_occurred

```
bool seq64::midi_api::m_first_error_occurred [protected]
```

This is an original RtMidi concept. I have to confess I am not sure how it is/should be used, yet.

13.41.4.8 m_error_callback_user_data

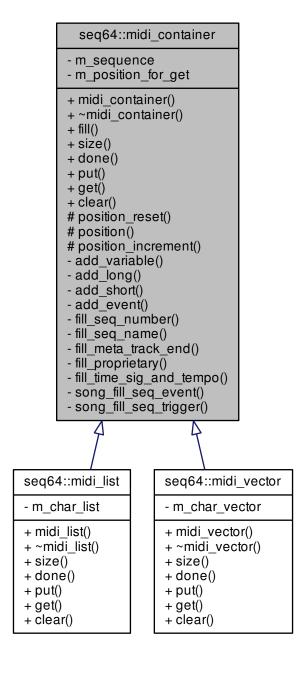
```
void* seq64::midi_api::m_error_callback_user_data [protected]
```

This is an original RtMidi concept. I have to confess I am not sure how it is/should be used, yet.

13.42 seq64::midi_container Class Reference

This class is the abstract base class for a container of MIDI track information.

Inheritance diagram for seq64::midi container:



Public Member Functions

midi_container (sequence &seq)

Fills in the few members of this class.

virtual ∼midi container ()

A rote constructor needed for a base class.

void fill (int tracknumber, const perform &p)

This function fills the given track (sequence) with MIDI data from the current sequence, preparatory to writing it to a file.

• virtual std::size_t size () const

Returns the size of the container, in midibytes.

· virtual bool done () const

Instead of checking for the size of the container when "emptying" it [see the midifile::write() function], use this function, which is overridden to match the type of container being used.

virtual void put (midibyte b)=0

Provides a way to add a MIDI byte into the container.

• virtual midibyte get () const =0

Provide a way to get the next byte from the container.

virtual void clear ()=0

Provides a way to clear the container.

Protected Member Functions

· unsigned int position_reset () const

'Setter' function for member m_position_for_get Sets the position to 0 and then returns that value.

· unsigned int position () const

'Getter' function for member m_position_for_get Returns the current position.

void position_increment () const

'Getter' function for member m_position_for_get Increments the current position.

Private Member Functions

• void add_variable (midipulse v)

This function masks off the lower 8 bits of the long parameter, then shifts it right 7, and, if there are still set bits, it encodes it into the buffer in reverse order.

void add long (midipulse x)

Adds a long value (a MIDI pulse/tick value) to the container.

void add_short (midishort x)

Adds a short value (two bytes) to the container.

void add_event (const event &e, midipulse deltatime)

Adds an event to the container.

void fill_seq_number (int seq)

Fills in the sequence number.

void fill seq name (const std::string &name)

Fills in the sequence name.

- void fill_meta_track_end (midipulse deltatime)
- void fill proprietary ()

Fills in the Sequencer64-specific information for the current sequence: The MIDI buss number, the time-signature, and the MIDI channel.

void fill_time_sig_and_tempo (const perform &p)

Fill in the time-signature and tempo information.

midipulse song_fill_seq_event (const trigger &trig, midipulse prev_timestamp)

Fills in sequence events based on the trigger and events in the sequence associated with this midi container.

void song_fill_seq_trigger (const trigger &trig, midipulse len, midipulse prev_timestamp)

Fills in the trigger for the whole sequence.

Private Attributes

• sequence & m_sequence

Provide a hook into a sequence so that we can exchange data with a sequence object.

• unsigned int m_position_for_get

Provides the position in the container when making a series of get() calls on the container.

Friends

· class midifile

13.42.1 Detailed Description

It is the base class for midi_list and midi_vector.

13.42.2 Constructor & Destructor Documentation

```
13.42.2.1 midi_container()
```

Parameters

seq Provides a reference to the sequence/track for which this container holds MIDI data.

```
13.42.2.2 \simmidi_container()
```

```
\label{lem:container::} \verb|virtual seq64::midi_container::|| \verb|container|| ( ) [inline], [virtual] |
```

13.42.3 Member Function Documentation

```
13.42.3.1 fill()
```

```
void seq64::midi_container::fill (
    int tracknumber,
    const perform & p )
```

Note that some of the events might not come out in the same order they were stored in (we see that with program-change events). This function replaces sequence::fill_list().

Now, for sequence 0, an alternate format for writing the sequencer number chunk is "FF 00 00". But that format can only occur in the first track, and the rest of the tracks then don't need a sequence number, since it is assumed to increment. This application doesn't use that shortcut.

Stazed:

```
The "stazed" (seq32) code implements a function like this one using a function sequence::fill_proprietary_list() that we don't need for our implementation... it is part of our midi_container::fill() function.
```

Triggers:

```
Triggers are added by first calling add_variable(0), which is needed because why? Then 0xFF 0x7F is written, followed by the length value, which is the number of triggers at 3 long integers per trigger, plus the 4-byte code for triggers, c_triggers_new = 0x24240008.
```

Not threadsafe The sequence object bound to this container needs to provide the locking mechanism when calling this function.

Parameters

tracknumber	Provides the track number. This number is masked into the track information.
р	The performance object that will hold some of the parameters needed when filling the MIDI
	container.

To allow other sequencers to read Seq24/Sequencer64 files, we should provide the Time Signature and Tempo meta events, in the 0th (first) track (sequence). These events must precede any "real" MIDI events. They are not included if the legacy-format option is in force.

```
13.42.3.2 size()
```

```
virtual std::size_t seq64::midi_container::size ( ) const [inline], [virtual]
```

Must be overridden in the derived class, though not pure.

Reimplemented in seq64::midi_list, and seq64::midi_vector.

13.42.3.3 done()

```
virtual bool seq64::midi_container::done ( ) const [inline], [virtual]
```

Reimplemented in seq64::midi_vector, and seq64::midi_list.

```
13.42.3.4 put()
virtual void seq64::midi_container::put (
              midibyte b ) [pure virtual]
The original seq24 container used an std::list and a push_front operation.
Implemented in seq64::midi_vector, and seq64::midi_list.
13.42.3.5 get()
virtual midibyte seq64::midi_container::get ( ) const [pure virtual]
It also increments m_position_for_get.
Implemented in seq64::midi_vector, and seq64::midi_list.
13.42.3.6 clear()
virtual void seq64::midi_container::clear ( ) [pure virtual]
Implemented in seq64::midi_vector, and seq64::midi_list.
13.42.3.7 position_reset()
unsigned int seq64::midi_container::position_reset ( ) const [inline], [protected]
13.42.3.8 position()
unsigned int seq64::midi_container::position ( ) const [inline], [protected]
13.42.3.9 position_increment()
void seq64::midi_container::position_increment ( ) const [inline], [protected]
13.42.3.10 add_variable()
void seq64::midi_container::add_variable (
              midipulse v) [private]
```

This function "replaces" sequence::add_list_var().

Parameters

ν The data value to be added to the current event in the MIDI container.

13.42.3.11 add_long()

What is the difference between this function and add_list_var()? This function "replaces" sequence::add_long_list(). This was a *global* internal function called addLongList(). Let's at least make it a private member now, and hew to the naming conventions of this class.

Parameters

x Provides the timestamp (pulse value) to be added to the container.

13.42.3.12 add_short()

Parameters

x Provides the timestamp (pulse value) to be added to the container.

13.42.3.13 add_event()

If the sequence's MIDI channel is EVENT_NULL_CHANNEL == 0xFF, then it is the copy of an SMF 0 sequence that the midi_splitter created. We want to be able to save it along with the other tracks, but won't be able to read it back if all the channels are bad. So we just use the channel from the event.

13.42.3.14 fill_seq_number()

Writes 0xFF 0x00 0x02, and then the number. This function is used in the new midifile::write_song() function, which should be ready to go by the time you're reading this.

Compare this function to the beginning of midi_container::fill().

Parameters

seq The sequence/track number to write.

13.42.3.15 fill_seq_name()

Writes 0xFF 0x03, and then the track name. This function is used in the new midifile::write_song() function, which should be ready to go by the time you're reading this.

Compare this function to the beginning of midi_container::fill().

Parameters

name

The sequence/track name to set. We could get this item from m_sequence, but the parameter allows the flexibility to change the name.

13.42.3.16 fill_meta_track_end()

13.42.3.17 fill_proprietary()

```
void seq64::midi_container::fill_proprietary ( ) [private]
```

Then, if we're not using the legacy output format, we add the "events" for the musical key, musical scale, and the background sequence for the current sequence. Finally, if tranpose support has been compiled into the program, we add that information as well. New feature: save more sequence-specific values, if not legacy format and not saved globally. We use a single byte for the key and scale, and a long for the background sequence. We save these values only if they are different from the defaults; in most cases they will have been left alone by the user. We save per-sequence values here only if the global-background-sequence feature is not in force.

For the new "transposable" flag (tagged by the value c_transpose) we really only care about saving the value of "false", because otherwise we can assume the value is true for the given sequence, and save space by not saving it... generally only drum patterns will not be transposable.

However, for now, write it anyway for consistency with Seq32.

13.42.3.18 fill_time_sig_and_tempo()

This function is used only for the first track, The sizes of these meta events are defined as SEQ64_TIME_TEMP← O_SIZE. However, we do not have to add that value in, as it is already counted in the intrinsic size of the container.

We now make sure that the proper values are part of the perform object for usage in this particular track. For export, we cannot guarantee that the first (0th) track/sequence is exportable.

Parameters

p Provides the performance object from which we get some global MIDI parameters.

13.42.3.19 song_fill_seq_event()

Parameters

trig	The current trigger to be processed.
prev_timestamp	The time-stamp of the previous event.

Returns

The next time-stamp value is returned.

13.42.3.20 song_fill_seq_trigger()

For a song-performance, there will be only one trigger, covering the beginning to the end of the fully unlooped track.

Parameters

trig	The current trigger to be processed.
length	Provides the total length of the sequence.
prev_timestamp	The time-stamp of the previous event, which is actually the first event.

13.42.4 Friends And Related Function Documentation

13.42.4.1 midifile

```
friend class midifile [friend]
```

13.42.5 Field Documentation

```
13.42.5.1 m_sequence
sequence& seq64::midi_container::m_sequence [private]

13.42.5.2 m_position_for_get
unsigned int seq64::midi_container::m_position_for_get [mutable], [private]
```

13.43 seq64::midi_control Class Reference

This class (formerly a struct) contains the control information for sequences that make up a live set.

Public Types

```
enum action {
action_toggle,
action_on,
action_off }
```

Public Member Functions

• midi_control ()

This default constructor creates a "zero" object.

- · bool active () const
- · bool inverse_active () const
- int status () const
- int data () const
- int min_value () const
- int max_value () const
- void set (int values[6])

Not so sure if this really saves trouble for the caller.

• void set (midibyte values[6])

Not so sure if this really saves trouble for the caller.

bool match (midibyte status, midibyte data) const

Handles a common check in the perform module.

• bool in_range (midibyte data) const

Handles a common check in the perform module.

Private Attributes

· bool m active

Provides the value for active.

• bool m_inverse_active

Provides the value for inverse-active.

· int m status

Provides the value for the status.

· int m data

Provides the value for the data.

· int m_min_value

Provides the minimum value for the controller.

int m_max_value

Provides the value for the controller.

13.43.1 Detailed Description

Note that, although we've converted this to a full-fledged class, the ordering of variables and the data arrays used to fill them is very significant. See the midifile and optionsfile modules.

The perform module sets up the three following arrays for each of the MIDI controls that can be defined in the "rc" file:

```
m_midi_cc_toggle[]
m_midi_cc_on[]
m_midi_cc_off[]
```

These three arrays are specified in the "rc" by a line like the following:

where n ranges from 0 to 73 or 83. Lines 0 to 31 provide controller values for the "pattern group", one line for each of the 32 pattern slots. Lines 32 to 63 provide controller values for the "mute in group", one line for each of the 32 pattern slots. The rest of the lines provide entries for control of: BPM up, BPM down, Screen-set up, Screen-set down, Mod Replaces, Mod Snapshot, Mod Queue, Mod gmute (group mute), Mod glearn (group learn), and Screen-set Play. Additional controls are currently in the works.

In each of the bracketed sections, the values correspond to the members in this order: m_active , $m_inverse_active$, m_status , m_data , m_min_value , and m_max_value .

Why are the status, data, and min/max values long? A character or midibyte would be enough. We'll fix that later, once we have tested this stuff. We do need to convert them from long to int, though, and do that in the scanning and output done by optionsfile.

13.43.2 Member Enumeration Documentation

13.43.2.1 action

```
enum seq64::midi_control::action
```

Enumerator

action_toggle	Provides the kind of MIDI control event found, used in the new perform::handle_midi_control_ex() function. Toggles the status of the given control. For the "playback" status, indicates the "pause" functionality.
action_on	Turns on the status of the given control. For the "playback" status, indicates the "start" functionality.
action_off	Turns off the status of the given control. For the "playback" status, indicates the "stop" functionality.

13.43.3 Constructor & Destructor Documentation

13.43.3.1 midi_control()

```
seq64::midi_control::midi_control ( ) [inline]
```

Every member is either false or zero.

13.43.4 Member Function Documentation

13.43.4.1 active()

```
bool seq64::midi_control::active ( ) const [inline]
```

13.43.4.2 inverse_active()

```
bool seq64::midi_control::inverse_active ( ) const [inline]
```

13.43.4.3 status()

```
int seq64::midi_control::status ( ) const [inline]
```

13.43.4.4 data()

```
int seq64::midi_control::data ( ) const [inline]
```

13.43.4.5 min_value()

```
int seq64::midi_control::min_value ( ) const [inline]
```

13.43.4.6 max_value()

```
int seq64::midi_control::max_value ( ) const [inline]
```

13.43.4.7 set() [1/2]

```
void seq64::midi_control::set (
            int values[6] ) [inline]
```

It fits in with the big-ass sscanf() call in optionsfile.

Parameters

values

Provides the six values, in an integer array, to set into the members in this order: m_active, m_inverse_active, m_status, m_data, m_min_value, and m_max_value.

13.43.4.8 set() [2/2]

```
void seg64::midi_control::set (
            midibyte values[6] ) [inline]
```

It fits in with the usage in midifile.

Parameters

values Provides the six values, in a byte array, to set into the members in this order: m_active, m_inverse_active, m_status, m_data, m_min_value, and m_max_value.

13.43.4.9 match()

Parameters

status	Provides the status byte, which is checked against m_status.		
data	Provides the data byte, which is checked against m_data.		

13.43.4.10 in_range()

13.43.5 Field Documentation

13.43.5.1 m_active

```
bool seq64::midi_control::m_active [private]
```

13.43.5.2 m_inverse_active

```
bool seq64::midi_control::m_inverse_active [private]
```

13.43.5.3 m_status

```
int seq64::midi_control::m_status [private]
```

13.43.5.4 m_data

```
int seq64::midi_control::m_data [private]
```



Generated by Doxygen

This class implements the ALSA version of a MIDI input object.

Inheritance diagram for seq64::midi_in_alsa:



Public Member Functions

• midi_in_alsa (midibus &parentbus, midi_info &masterinfo)

ALSA MIDI input normal port or virtual port constructor.

Additional Inherited Members

13.44.1 Constructor & Destructor Documentation

```
13.44.1.1 midi_in_alsa()
```

The kind of port is determine by which port-initialization function the mastermidibus calls.

13.45 seq64::midi_in_jack Class Reference

The class for handling JACK MIDI input.

Inheritance diagram for seq64::midi_in_jack:



Public Member Functions

• midi_in_jack (midibus &parentbus, midi_info &masterinfo)

Principal constructor.

virtual ~midi_in_jack ()

Destructor.

• virtual int api_poll_for_midi ()

Checks the rtmidi_in_data queue for the number of items in the queue.

virtual bool api_get_midi_event (event *)

Gets a MIDI event.

Protected Attributes

• std::string m_client_name

Private Member Functions

• virtual bool open_client ()

This function is virtual, so we don't call it in the constructor, using open_client_impl() directly instead.

Additional Inherited Members

13.45.1 Constructor & Destructor Documentation

```
13.45.1.1 midi_in_jack()
```

For Sequencer64, we don't current need to create a midi_in_jack object; all that is needed is created via the api_\(-\) init_in*() functions. Also, this constructor still needs to do something with queue size.

Parameters

parentbus	Provides the buss object that determines buss-specific parameters of this class.
masterinfo	Provides information about the JACK system as found on this machine.

```
13.45.1.2 \simmidi_in_jack()
```

```
seq64::midi_in_jack::~midi_in_jack ( ) [virtual]
```

Currently the base class closes the port, closes the JACK client, and cleans up the API data structure.

13.45.2 Member Function Documentation

```
13.45.2.1 api_poll_for_midi()
```

```
int seq64::midi_in_jack::api_poll_for_midi ( ) [virtual]
```

WE MAY NEED LOCKING.

Returns

Returns the value of rtindata->queue().count(), unless the caller is using an rtmidi callback function, in which case 0 is always returned.

Reimplemented from seg64::midi jack.

13.45.2.2 api_get_midi_event()

This implementation gets a midi_message off the front of the queue and converts it to an sequence64 event.

Parameters

inev Provides the destination for the MIDI event.

Returns

Returns true if a MIDI event was obtained, indicating that the return parameter can be used.

We will only get EVENT_SYSEX on the first packet of MIDI data; the rest we have to poll for. SysEx processing is currently disabled.

Reimplemented from seq64::midi_jack.

13.45.2.3 open_client()

```
virtual bool seq64::midi_in_jack::open_client ( ) [inline], [private], [virtual]
```

This function replaces the RtMidi function "connect()".

Implements seq64::midi_jack.

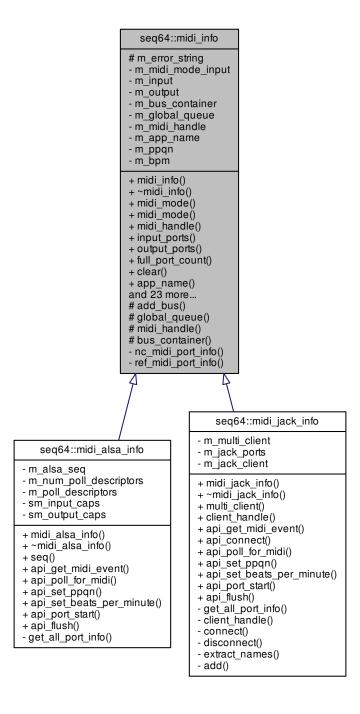
13.45.3 Field Documentation

13.45.3.1 m_client_name

std::string seq64::midi_in_jack::m_client_name [protected]

13.46 seq64::midi_info Class Reference

The class for holding basic information on the MIDI input and output ports currently present in the system. Inheritance diagram for seg64::midi info:



Public Member Functions

midi_info (const std::string &appname, int ppqn=SEQ64_DEFAULT_PPQN, midibpm bpm=SEQ64_DEFA
 ULT_BPM)

Principal constructor.

- virtual ~midi_info ()
- bool midi_mode () const

'Getter' function for member m_midi_mode_input

void midi mode (bool flag)

'Setter' function for member m_midi_mode_input

void * midi handle ()

'Getter' function for member m_midi_handle

• midi_port_info & input_ports ()

'Getter' function for member m_input

midi_port_info & output_ports ()

'Getter' function for member m_output

· int full port count () const

'Getter' function for member Total port count.

- void clear ()
- const std::string & app_name () const

'Getter' function for member m_app_name

• int ppqn () const

'Getter' function for member m_ppqn, simple version, also see api_set_ppqn().

• midibpm bpm () const

'Getter' function for member m_bpm, simple version, also see api_set_beats_per_minute().

virtual void api_set_ppqn (int p)

Special setter.

virtual void api_set_beats_per_minute (midibpm b)

Special setter.

• virtual void api_port_start (mastermidibus &, int, int)

An ALSA-specific function at the moment.

- virtual bool api_get_midi_event (event *inev)=0
- virtual int api_poll_for_midi ()=0
- virtual void api_flush ()=0
- · virtual bool api_connect ()

Used only in the midi_jack_info class.

- virtual int get port count () const
- · virtual int get_bus_id (int index) const
- virtual std::string get_bus_name (int index) const
- · virtual int get port id (int index) const
- virtual std::string get_port_name (int index) const
- virtual bool get_input (int index) const
- virtual bool get_virtual (int index) const
- virtual bool get_system (int index) const
- virtual int queue_number (int index) const
- std::string connect_name (int index) const
- std::string port_list () const

Generates a string listing all of the ports present in the port container.

- int global_queue () const
- void error (rterror::Type type, const std::string &errorstring)

A basic error reporting function for midi info classes.

• virtual int get_all_port_info ()=0

Protected Member Functions

void add_bus (const midibus *m)

Adds the midibus to a quick list of all ports for use in the api_connect() call in mastermidibus.

void global_queue (int q)

'Setter' function for member m_global_queue

• void midi_handle (void *h)

'Setter' function for member m_midi_handle

std::vector< midibus * > & bus container ()

'Getter' function for member m_bus_container

Protected Attributes

std::string m_error_string

Error string for the midi_info interface.

Private Member Functions

• const midi_port_info & nc_midi_port_info () const

'Getter' function for member m_input or m_output Used for retrieving values from the input or output containers.

midi_port_info & ref_midi_port_info ()

'Getter' function for member m_input or m_output

Private Attributes

· bool m midi mode input

Indicates which mode we're in, input or output.

midi_port_info m_input

Holds data on the ALSA/JACK/Core/WinMM inputs.

· midi port info m output

Holds data on the ALSA/JACK/Core/WinMM outputs.

std::vector< midibus * > m_bus_container

Holds pointers to the ports that were created, so that, after activation, we can call the connect_port() function on those that are not virtual.

• int m_global_queue

The ID of the ALSA MIDI queue.

void * m_midi_handle

Provides a handle to the main ALSA or JACK implementation object.

• const std::string m_app_name

Holds this value for passing along, to reduce the number of arguments needed.

int m_ppqn

Hold this value for passing along to some ports that get created.

• midibpm m_bpm

Hold this value for passing along to some ports that get created.

Friends

· class rtmidi_info

13.46.1.1 midi_info()

13.46.1 Constructor & Destructor Documentation

```
seq64::midi_info::midi_info (
            const std::string & appname,
             int ppqn = SEQ64_DEFAULT_PPQN,
             midibpm bpm = SEQ64_DEFAULT_BPM )
13.46.1.2 ∼midi_info()
virtual seq64::midi_info::~midi_info ( ) [inline], [virtual]
13.46.2 Member Function Documentation
13.46.2.1 midi_mode() [1/2]
bool seq64::midi_info::midi_mode ( ) const [inline]
13.46.2.2 midi_mode() [2/2]
void seq64::midi_info::midi_mode (
             bool flag ) [inline]
13.46.2.3 midi_handle() [1/2]
void* seq64::midi_info::midi_handle ( ) [inline]
13.46.2.4 input_ports()
midi_port_info& seq64::midi_info::input_ports ( ) [inline]
```

```
13.46.2.5 output_ports()
midi_port_info& seq64::midi_info::output_ports ( ) [inline]
13.46.2.6 full_port_count()
int seq64::midi_info::full_port_count ( ) const [inline]
13.46.2.7 clear()
void seq64::midi_info::clear ( ) [inline]
13.46.2.8 app_name()
const std::string& seq64::midi_info::app_name ( ) const [inline]
13.46.2.9 ppqn()
int seq64::midi_info::ppqn ( ) const [inline]
13.46.2.10 bpm()
midibpm seq64::midi_info::bpm ( ) const [inline]
13.46.2.11 api_set_ppqn()
virtual void seq64::midi_info::api_set_ppqn (
             int p ) [inline], [virtual]
Reimplemented in seq64::midi_jack_info, and seq64::midi_alsa_info.
```

```
13.46.2.12 api_set_beats_per_minute()
virtual void seq64::midi_info::api_set_beats_per_minute (
             midibpm b ) [inline], [virtual]
Reimplemented in seq64::midi_jack_info, and seq64::midi_alsa_info.
13.46.2.13 api_port_start()
virtual void seq64::midi_info::api_port_start (
             mastermidibus & ,
              int ) [inline], [virtual]
Reimplemented in seq64::midi_jack_info, and seq64::midi_alsa_info.
13.46.2.14 api_get_midi_event()
virtual bool seq64::midi_info::api_get_midi_event (
             event * inev ) [pure virtual]
Implemented in seq64::midi jack info, and seq64::midi alsa info.
13.46.2.15 api_poll_for_midi()
virtual int seq64::midi_info::api_poll_for_midi ( ) [pure virtual]
Implemented in seq64::midi_jack_info, and seq64::midi_alsa_info.
13.46.2.16 api_flush()
virtual void seq64::midi_info::api_flush ( ) [pure virtual]
Implemented in seq64::midi_jack_info, and seq64::midi_alsa_info.
13.46.2.17 api_connect()
virtual bool seq64::midi_info::api_connect ( ) [inline], [virtual]
```

Reimplemented in seq64::midi_jack_info.

```
13.46.2.18 get_port_count()
virtual int seq64::midi_info::get_port_count ( ) const [inline], [virtual]
13.46.2.19 get_bus_id()
virtual int seq64::midi_info::get_bus_id (
            int index ) const [inline], [virtual]
13.46.2.20 get_bus_name()
virtual std::string seq64::midi_info::get_bus_name (
            int index ) const [inline], [virtual]
13.46.2.21 get_port_id()
virtual int seq64::midi_info::get_port_id (
            int index ) const [inline], [virtual]
13.46.2.22 get_port_name()
virtual std::string seq64::midi_info::get_port_name (
             int index ) const [inline], [virtual]
13.46.2.23 get_input()
virtual bool seq64::midi_info::get_input (
            int index ) const [inline], [virtual]
13.46.2.24 get_virtual()
virtual bool seq64::midi_info::get_virtual (
            int index ) const [inline], [virtual]
```

```
13.46.2.25 get_system()
```

13.46.2.26 queue_number()

13.46.2.27 connect_name()

13.46.2.28 port_list()

```
std::string seq64::midi_info::port_list ( ) const
```

Useful for debugging and probing.

Returns

Returns a multi-line ASCII string enumerating all of the ports.

Provides an error handler.

Unlike the midi_api version, it cannot support an error callback.

Exceptions

```
If the error is not just a warning, then an rterror object is thrown.
```

Parameters

type		The type of the error.
errorstr	ing	The error message, which gets copied if this is the first error.

```
13.46.2.31 get_all_port_info()

virtual int seq64::midi_info::get_all_port_info ( ) [pure virtual]

Implemented in seq64::midi_jack_info, and seq64::midi_alsa_info.
```

```
13.46.2.32 add_bus()
```

We could add the midibus pointer to the midi_port_info structure, but that information is strictly for representing data obtained by querying the system via the selected API.

```
13.46.2.36 nc_midi_port_info()
```

const midi_port_info& seq64::midi_info::nc_midi_port_info () const [inline], [private]

The caller must insure the proper container by calling the midi_mode() function with the value of true (SEQ64_MI ← DI_INPUT_PORT) or false (SEQ64_MIDI_OUTPUT_PORT) first. Ugly stuff. I hate it.

```
13.46.2.37 ref_midi_port_info()
```

```
midi_port_info& seq64::midi_info::ref_midi_port_info ( ) [inline], [private]
```

13.46.3 Friends And Related Function Documentation

```
13.46.3.1 rtmidi_info
```

```
friend class rtmidi_info [friend]
```

13.46.4 Field Documentation

```
13.46.4.1 m_midi_mode_input
```

```
bool seq64::midi_info::m_midi_mode_input [private]
```

We have to pick the mode we need to be in with the set_mode() function before we do a series of operations. This clumsy two-step is needed in order to preserve the midi_api interface.

```
13.46.4.2 m_input
```

```
midi_port_info seq64::midi_info::m_input [private]
```

13.46.4.3 m_output

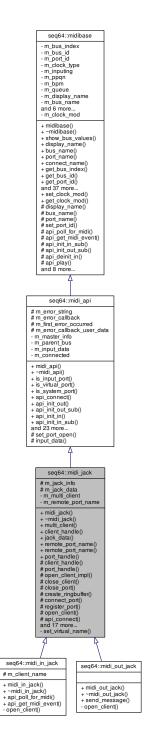
```
midi_port_info seq64::midi_info::m_output [private]
```

```
13.46.4.4 m_bus_container
std::vector<midibus *> seq64::midi_info::m_bus_container [private]
See the add_bus() and bus_container() member functions.
13.46.4.5 m_global_queue
int seq64::midi_info::m_global_queue [private]
13.46.4.6 m_midi_handle
void* seq64::midi_info::m_midi_handle [private]
Created by the class derived from midi_info.
13.46.4.7 m_app_name
const std::string seq64::midi_info::m_app_name [private]
This value is the main application name as determined at ./configure time.
13.46.4.8 m_ppqn
int seq64::midi_info::m_ppqn [private]
Some APIs can use this value.
13.46.4.9 m_bpm
midibpm seq64::midi_info::m_bpm [private]
Some APIs can use this value.
13.46.4.10 m_error_string
std::string seq64::midi_info::m_error_string [protected]
```

13.47 seq64::midi_jack Class Reference

This class implements with JACK version of the midi_alsa object.

Inheritance diagram for seq64::midi_jack:



Public Member Functions

• midi_jack (midibus &parentbus, midi_info &masterinfo)

We still need to figure out if we want a "master" client handle, or a handle to each port.

virtual ~midi_jack ()

This could be a rote empty destructor if we offload this destruction to the midi_jack_data structure.

· bool multi client () const

'Getter' function for member m_multi_client

jack_client_t * client_handle ()

'Getter' function for member m_jack_client This is the platform-specific version of midi_handle().

midi jack data & jack data ()

'Getter' function for member m_jack_data

const std::string & remote_port_name () const

'Getter' function for member m remote port name

void remote port name (const std::string &s)

'Setter' function for member m_remote_port_name

jack_port_t * port_handle ()

'Getter' function for member m_jack_port This is the platform-specific version of midi_handle().

Protected Member Functions

void client handle (jack client t *handle)

'Setter' function for member m_jack_data.m_jack_client

void port_handle (jack_port_t *handle)

'Setter' function for member m_jack_data.m_jack_port

bool open_client_impl (bool input)

Opens input or output JACK clients, sets up the input or output callback, and actives the JACK client.

· void close client ()

Closes the JACK client handle.

void close_port ()

Closes the MIDI port by calling jack_port_unregister() and nullifying the port pointer.

bool create ringbuffer (size t rbsize)

Creates the JACK ring-buffers.

• bool connect_port (bool input, const std::string &sourceportname, const std::string &destportname)

Connects two named JACK ports.

bool register port (bool input, const std::string &portname)

Registers a named JACK port.

- virtual bool open_client ()=0
- virtual bool api_connect ()

Assumes that the port has already been registered, and that JACK activation has already occurred.

virtual bool api_init_out ()

Initialize the MIDI output port.

• virtual bool api init in ()

This function is called when we are processing the list of system input ports.

- virtual bool api_init_out_sub ()
- virtual bool api_init_in_sub ()

Initializes a virtual/manual input port.

• virtual bool api deinit in ()

We could define these in the opposite order.

- virtual bool api_get_midi_event (event *)
- virtual int api_poll_for_midi ()
- virtual void api_play (event *e24, midibyte channel)

We could push the bytes of the event into a midibyte vector, as done in send_message().

- virtual void api_sysex (event *e24)
- virtual void api_flush ()

It seems like JACK doesn't have the concept of flushing event.

virtual void api_continue_from (midipulse tick, midipulse beats)

jack_transport_locate(), jack_transport_reposition(), or something else?

• virtual void api_start ()

Starts this JACK client.

virtual void api stop ()

Starts this JACK client.

- · virtual void api clock (midipulse tick)
- virtual void api_set_ppqn (int ppqn)
- virtual void api_set_beats_per_minute (midibpm bpm)
- virtual std::string api_get_port_name ()

Gets the name of the current port via jack_port_name().

Protected Attributes

· midi_jack_info & m_jack_info

This reference is needed in order for this midi_jack object to add itself to the main midi_jack_info list when running in single-JACK client mode.

• midi_jack_data m_jack_data

Holds the data needed for JACK processing.

Private Member Functions

• bool set_virtual_name (int portid, const std::string &portname)

Gets information directly from JACK.

Private Attributes

• bool m_multi_client

Set to true if each JACK port should be its own client.

• std::string m_remote_port_name

Preserves the original name of the remote port, so it can be used later for connection.

Friends

· class midi_jack_info

Additional Inherited Members

13.47.1 Constructor & Destructor Documentation

13.47.1.1 midi_jack()

Same for the JACK data item. Currently, we provide the m_multi_client member so that we can experiment, but "multi-client" mode is currently incompletely implemented.

Note that this constructor also adds its object to the midi_jack_info port list, so that the JACK callback functions can iterate through all of the JACK ports in use by this application, performing work on them.

Parameters

parentbus	Provides a reference to the midibus that represents this object.
masterinfo	Provides a reference to the midi_jack_info object that may provide extra informatino that is needed by this port. Too many entities!

```
13.47.1.2 ~midi_jack()
seq64::midi_jack::~midi_jack ( ) [virtual]
```

However, other than the initialization, that structure is currently "dumb".

13.47.2 Member Function Documentation

```
13.47.2.1 multi_client()
bool seq64::midi_jack::multi_client ( ) const [inline]

13.47.2.2 client_handle() [1/2]

jack_client_t* seq64::midi_jack::client_handle ( ) [inline]

13.47.2.3 jack_data()
```

midi_jack_data& seq64::midi_jack::jack_data () [inline]

```
13.47.2.4 remote_port_name() [1/2]
const std::string& seq64::midi_jack::remote_port_name ( ) const [inline]
```

This code is combined from the former versions of the midi_in_jack::connect() and midi_out_jack::connect() functions for better readability and re-use in the input and output open_client() functions.

For input, it connects the MIDI input port. The following calls are made:

```
- jack_client_open(), to initialize JACK client.
- jack_set_process_callback(), to set jack_process_rtmidi_input() or
jack_process_rtmidi_output().
```

Note that jack_activate() is no longer called here.

For output, connects the MIDI output port. The following calls are made:

```
    jack_ringbuffer_create(), called twice, to initialize the output ringbuffers
    jack_client_open(), to initialize JACK client
    jack_set_process_callback(), to set jack_process_inpu()
```

Note that jack_activate() is no longer called here.

If the midi_jack_data client member is already set, this function returns immediately. Only one client needs to be open for each midi_jack object.

Let's replace JackNullOption with JackNoStartServer. We might also want to OR in the JackUseExactName option.

The former name of this function was a bit of a misnomer, since it does not actually call jack_connect(). That call is made in other functions.

Which "client" name? Let's start with the full name, connect_name(). Is UUID an output-only, input-only option, or both?

```
const char * name = master_info().get_bus_name(get_bus_index()).c_str();
const char * name = master_info().get_port_name(get_bus_index()).c_str();
```

Parameters

input True if an input connection is to be made, and false if an output connection is to be made.

```
13.47.2.10 close_client()
```

```
void seq64::midi_jack::close_client ( ) [protected]
```

13.47.2.11 close_port()

```
void seq64::midi_jack::close_port ( ) [protected]
```

13.47.2.12 create_ringbuffer()

13.47.2.13 connect_port()

First, we register the local port. If this is nominally a local input port, it is really doing output, and this is the source-port name. If this is nominally a local output port, it is really accepting input, and this is the destination-port name.

This code is disabled for now because the order of JACK setup calls that works is

```
- jack_port_register()
- jack_activate()
- jack_connect()
```

So we have to break this up.

Parameters

input	Indicates true if the port to register and connect is an input port, and false if the port is an output port. Useful macros for readability: SEQ64_MIDI_INPUT_PORT and SEQ64_MIDI_OUTPUT_PORT.
srcportname	Provides the destination port-name for the connection. For input, this should be the name associated with the JACK client handle; it is the port that gets registered. For of the port should be the full name of the port that was enumerated at start-up. The JackPortFlags of the source port must include JackPortIsOutput.
destportname	For input, this should be full name of port that was enumerated at start-up. For output, this

Returns

If the jack_connect() call succeeds, true is returned. If the port is a virtual (manual) port, then it is not connected, and true is returned without any action.

13.47.2.14 register_port()

We made this function to encapsulate some otherwise cut-and-paste functionality.

For jack_port_register(), the port-name must be the actual port-name, not the full-port name ("busname:portname").

Note that the buffer size of non-built-in port type is 0, and so it is ignored.

Tricky Code If we are registering an input port, this means that we got the input port from the system. In order to connect to that port, we have to register as an output port, even though the application calls it in input port (midi
_in_jack). Confusing, but the same thing was implicit in the ALSA implementation, and so we have to apply that same setup here.

Parameters

input	Indicates true if the port to register input port, and false if the port is an output port. Two macros can be used for this purpose: SEQ64_MIDI_INPUT_PORT and SEQ64_MIDI_OUTPUT_PORT.
portname	Provides the local name of the port. This is the full name ("clientname:portname"), but the "portname" part is extracted to fit the requirements of the jack_port_register() function. There is an issue here when a2jmidid is running. We may see a client name of "seq64", but a port name of "a2j Midi Through [1] capture: Midi Through Port-0", which as a colon in it. What to do? Just not extract the port name from the portname parameter. If we have an issue here, we'll ahve to fix it in the caller.

13.47.2.15 open_client()

```
virtual bool seq64::midi_jack::open_client ( ) [protected], [pure virtual]
Implemented in seq64::midi_out_jack, and seq64::midi_in_jack.
```

13.47.2.16 api_connect()

```
bool seq64::midi_jack::api_connect ( ) [protected], [virtual]
```

Returns

Returns true if all steps of the connection succeeded.

Reimplemented from seq64::midi_api.

```
13.47.2.17 api_init_out()
```

```
bool seq64::midi_jack::api_init_out ( ) [protected], [virtual]
```

This initialization is done when the "manual ports" option is not in force. This code is basically what was done by midi_out_jack::open_port() in RtMidi.

For jack_connect(), the first port-name is the source port, and the source port must include the JackPortIsOutput flag. The second port-name is the destination port, and the destination port must include the JackPortIsInput flag. For this function, the source/output port is this port, and the destination/input is....

Note that connect_port() [which calls jack_connect()] cannot usefully be called until jack_activate() has been called.

Returns

Returns true unless setting up JACK MIDI failed in some way.

Implements seq64::midi_api.

```
13.47.2.18 api_init_in()
```

```
bool seq64::midi_jack::api_init_in ( ) [protected], [virtual]
```

We want to create an output port of a similar name, but with the application as client, and connect it to this sytem input port. We want to follow the model we got from seq24, rather than the RtMidi model, so that we do not have to rework (and probably break) the seq24 model.

We can't use the API port name here at this time, because it comes up empty. It comes up empty because we haven't yet registered the ports, including the source ports. So we register it first; connect_port() will not try to register it again.

Based on the comments in the jack.txt note file, here is what we need to do:

```
-# open_client_impl(INPUT).
    -# Get port name via master_info().connect_name().
    -# Call jack_client_open() with or without a UUID.
    -# Call jack_set_process_callback() for input/output.
    -# Call jack_activate(). Premature?
-# register_port(INPUT...). The flag is JackPortIsInput.
-# connect_port(INPUT...). Call jack_connect(srcport, destport).
```

Unlike the corresponding virtual port, this input port is actually an output port.

Returns

Returns true if the function was successful, and sets the flag indicating that the port is open.

Implements seq64::midi_api.

```
13.47.2.19 api_init_out_sub()
bool seq64::midi_jack::api_init_out_sub ( ) [protected], [virtual]
Implements seq64::midi_api.
13.47.2.20 api_init_in_sub()
bool seq64::midi_jack::api_init_in_sub ( ) [protected], [virtual]
Returns
     Returns true if all steps of the initialization succeeded.
Implements seq64::midi_api.
13.47.2.21 api_deinit_in()
bool seq64::midi_jack::api_deinit_in ( ) [protected], [virtual]
Implements seq64::midi_api.
13.47.2.22 api_get_midi_event()
virtual bool seq64::midi_jack::api_get_midi_event (
              event * ) [inline], [protected], [virtual]
Returns
     Returns false, since this is an input function that is implemented fully only by midi_in_jack.
Implements seq64::midi_api.
Reimplemented in seq64::midi_in_jack.
```

```
13.47.2.23 api_poll_for_midi()
```

```
virtual int seq64::midi_jack::api_poll_for_midi ( ) [inline], [protected], [virtual]
```

Returns

Returns 0, since this is an input function that is implemented fully only by midi_in_jack.

Implements seq64::midi_api.

Reimplemented in seq64::midi_in_jack.

```
13.47.2.24 api_play()
```

The ALSA code (seq_alsamidi/src/midibus.cpp) sticks the event bytes in an array, which might be a little faster than using push_back(), but let's try the vector first. The rtmidi code here is from midi_out_jack::send_message().

Implements seq64::midi_api.

```
13.47.2.25 api_sysex()
```

Todo Flesh out this routine.

Implements seq64::midi_api.

```
13.47.2.26 api_flush()
```

```
void seq64::midi_jack::api_flush ( ) [protected], [virtual]
```

Implements seq64::midi_api.

```
13.47.2.27 api_continue_from()
```

What is used by jack_assistant?

Implements seq64::midi_api.

```
13.47.2.28 api_start()
```

```
void seq64::midi_jack::api_start ( ) [protected], [virtual]
```

Note that the jack_assistant code (which implements JACK transport) checks if JACK is running, but a check of the JACK client handle here should be enough.

Implements seq64::midi_api.

```
13.47.2.29 api_stop()
```

```
void seq64::midi_jack::api_stop ( ) [protected], [virtual]
```

Note that the jack_assistant code (which implements JACK transport) checks if JACK is running, but a check of the JACK client handle here should be enough.

Implements seq64::midi_api.

```
13.47.2.30 api_clock()
```

Implements seq64::midi_api.

```
13.47.2.31 api_set_ppqn()
```

Implements seq64::midi_api.

13.47.2.32 api_set_beats_per_minute()

Implements seq64::midi_api.

13.47.2.33 api_get_port_name()

```
std::string seq64::midi_jack::api_get_port_name ( ) [protected], [virtual]
```

This is different from get_port_name(index), which simply looks up the port-name in the attached midi_info object.

Returns

Returns the full port name ("clientname:portname") if the port has already been opened/registered; otherwise an empty string is returned.

Reimplemented from seq64::midi_api.

13.47.2.34 set_virtual_name()

The problem this function solves is that the midibus constructor for a virtual JACK port doesn't not have all of the information it needs at that point. Here, we can get this information and get the actual data we need to rename the port to something accurate.

```
const char * pname = jack_port_name(const jack_port_t *);
```

Returns

Returns true if all of the information could be obtained. If false is returned, then the caller should not use the side-effects.

Side-effect(s) Passes back the values found.

13.47.3 Friends And Related Function Documentation

13.47.3.1 midi_jack_info

```
friend class midi_jack_info [friend]
```

13.47.4 Field Documentation

13.47.4.1 m_multi_client

```
bool seq64::midi_jack::m_multi_client [private]
```

In this case, the functions api_init_in(), api_init_out(), api_init_in_sub(), and api_init_out_sub() need to open their own JACK client. Otherwise, they will use the JACK client created in the midi_jack_info class.

13.47.4.2 m_remote_port_name

```
std::string seq64::midi_jack::m_remote_port_name [private]
```

13.47.4.3 m_jack_info

```
midi_jack_info& seq64::midi_jack::m_jack_info [protected]
```

13.47.4.4 m_jack_data

```
midi_jack_data seq64::midi_jack::m_jack_data [protected]
```

Please do not confuse this item with the m_midi_handle of the midi_api base class. This object holds a JACK-client pointer and a JACK-port pointer.

13.48 seq64::midi_jack_data Struct Reference

Contains the JACK MIDI API data as a kind of scratchpad for this object.

Public Member Functions

• midi_jack_data ()

Constructor midi_jack_data

~midi_jack_data ()

This destructor currently does nothing.

• bool valid_buffer () const

Tests that the buffer is good.

Data Fields

• jack_client_t * m_jack_client

Holds the JACK sequencer client pointer so that it can be used by the midibus objects.

jack_port_t * m_jack_port

Holds the JACK port information of the JACK client.

• jack_ringbuffer_t * m_jack_buffsize

Holds the size of data for communicating between the client ring-buffer and the JACK port's internal buffer.

• jack_ringbuffer_t * m_jack_buffmessage

Holds the data for communicating between the client ring-buffer and the JACK port's internal buffer.

• jack_time_t m_jack_lasttime

The last time-stamp obtained.

rtmidi_in_data * m_jack_rtmidiin

Holds special data peculiar to the client and its MIDI input processing.

13.48.1 Detailed Description

This guy needs a constructor taking parameters for an rtmidi_in_data pointer.

13.48.2 Constructor & Destructor Documentation

```
13.48.2.1 midi_jack_data()
seq64::midi_jack_data::midi_jack_data ( ) [inline]

13.48.2.2 ~midi_jack_data()
seq64::midi_jack_data::~midi_jack_data ( ) [inline]
```

We rely on the enclosing class to close out the things that it created.

13.48.3 Member Function Documentation

```
13.48.3.1 valid_buffer()
bool seq64::midi_jack_data::valid_buffer ( ) const [inline]
```

13.48.4 Field Documentation

```
13.48.4.1 m_jack_client
```

```
jack_client_t* seq64::midi_jack_data::m_jack_client
```

This is actually an opaque pointer; there is no way to get the actual fields in this structure; they can only be accessed through functions in the JACK API. Note that it is also stored as a void pointer in midi_info::m_midi_handle. This item can either be the single JACK client created by the midi_jack_info object, or a JACK client created by the midi_jack object in the "multi-client" mode (which is not yet complete or usable).

```
13.48.4.2 m_jack_port
```

```
jack_port_t* seq64::midi_jack_data::m_jack_port
```

13.48.4.3 m_jack_buffsize

```
jack_ringbuffer_t* seq64::midi_jack_data::m_jack_buffsize
```

13.48.4.4 m_jack_buffmessage

```
jack_ringbuffer_t* seq64::midi_jack_data::m_jack_buffmessage
```

13.48.4.5 m_jack_lasttime

```
jack_time_t seq64::midi_jack_data::m_jack_lasttime
```

Use for calculating the delta time, I would imagine.

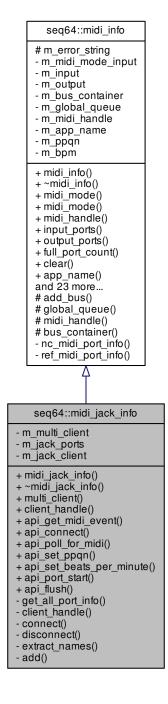
13.48.4.6 m_jack_rtmidiin

```
rtmidi_in_data* seq64::midi_jack_data::m_jack_rtmidiin
```

13.49 seq64::midi_jack_info Class Reference

The class for handling JACK MIDI port enumeration.

Inheritance diagram for seq64::midi_jack_info:



Public Member Functions

midi_jack_info (const std::string &appname, int ppqn=SEQ64_DEFAULT_PPQN, midibpm bpm=SEQ64_←
DEFAULT_BPM)

Principal constructor.

• virtual \sim midi_jack_info ()

Destructor.

· bool multi client () const

'Getter' function for member m_multi_client

• jack client t * client handle ()

'Getter' function for member m_jack_client This is the platform-specific version of midi_handle().

virtual bool api_get_midi_event (event *inev)

Grab a MIDI event.

virtual bool api_connect ()

Sets up all of the ports, represented by midibus objects, that have been created.

virtual int api_poll_for_midi ()

MUCH TO DO!

virtual void api_set_ppqn (int p)

Sets the PPQN numeric value, then makes JACK calls to set up the PPQ tempo.

virtual void api_set_beats_per_minute (midibpm b)

Sets the BPM numeric value, then makes JACK calls to set up the BPM tempo.

virtual void api_port_start (mastermidibus &masterbus, int bus, int port)

Start the given JACK MIDI port.

virtual void api_flush ()

Flushes our local queue events out into JACK.

Private Member Functions

• virtual int get all port info ()

Gets information on ALL ports, putting input data into one midi_info container, and putting output data into another midi_info container.

void client_handle (jack_client_t *j)

'Getter' function for member m_jack_client This is the platform-specific version of midi_handle().

jack_client_t * connect ()

Local JACK connection for enumerating the ports.

• void disconnect ()

The opposite of connect().

• void extract_names (const std::string &fullname, std::string &clientname, std::string &portname)

Extracts the two names from the JACK port-name format, "clientname:portname".

• bool add (midi_jack &mj)

Adds a pointer to a JACK port.

Private Attributes

bool m_multi_client

Set to true if each JACK port should be its own client.

std::vector< midi_jack * > m_jack_ports

Holds the port data.

jack_client_t * m_jack_client

Holds the JACK sequencer client pointer so that it can be used by the midibus objects.

Friends

- · class midi_jack
- int jack_process_io (jack_nframes_t nframes, void *arg)

Provides a JACK callback function that uses the callbacks defined in the midi_jack module.

Additional Inherited Members

13.49.1 Constructor & Destructor Documentation

13.49.1.1 midi_jack_info()

Note the m_multi_client member. We may want each JACK port to have its own client, as in the original RtMidi implementation.

Parameters

appname	Provides the name of the application.
ppqn	Provides the desired value of the PPQN (pulses per quarter note).
bpm	Provides the desired value of the BPM (beats per minute).

```
13.49.1.2 \sim midi_jack_info()
```

```
seq64::midi_jack_info::~midi_jack_info ( ) [virtual]
```

Deactivates (disconnects and closes) any ports maintained by the JACK client, then closes the JACK client, shuts down the input thread, and then cleans up any API resources in use.

13.49.2 Member Function Documentation

13.49.2.1 multi_client()

```
bool seq64::midi_jack_info::multi_client ( ) const [inline]
```

Parameters

inev

The event to be set based on the found input event. We should make this value a reference someday. Not used here.

Returns

Always returns false. Will eventually delete this function.

Implements seq64::midi_info.

```
13.49.2.4 api_connect()
bool seq64::midi_jack_info::api_connect ( ) [virtual]
```

If multi-client usage has been specified, each non-virtual port that has been set up (with its own JACK client pointer) is activated, and then connected to its corresponding remote system port.

Otherwise, the main JACK client is activated, and then all non-virtual ports are simply connected.

Each JACK port's midi_jack::api_connect() function decides, based on multi-client status, whether or not to activate before making the connection.

Returns

Returns true if activation succeeds.

Reimplemented from seq64::midi_info.

Parameters

```
p The desired new PPQN value to set.
```

Reimplemented from seq64::midi_info.

13.49.2.7 api_set_beats_per_minute()

Parameters

```
b The desired new BPM value to set.
```

Reimplemented from seq64::midi_info.

13.49.2.8 api_port_start()

This function is called by api_get_midi_event() when an JACK event SND_SEQ_EVENT_PORT_START is received.

- · Get the API's client and port information.
- · Do some capability checks.
- Find the client/port combination among the set of input/output busses. If it exists and is not active, then mark it as a replacement. If it is not a replacement, it will increment the number of input/output busses.

We can simplify this code a bit by using elements already present in midi_jack_info.

Parameters

masterbus	Provides the object needed to get access to the array of input and output buss objects.
bus	Provides the JACK bus/client number.
port	Provides the JACK client port.

Reimplemented from seq64::midi_info.

```
13.49.2.9 api_flush()
```

```
void seq64::midi_jack_info::api_flush ( ) [virtual]
```

This is also a midi_jack function.

Implements seq64::midi_info.

```
13.49.2.10 get_all_port_info()
```

```
int seq64::midi_jack_info::get_all_port_info ( ) [private], [virtual]
```

Tricky Code When we want to connect to a system input port, we want to use an output port to do that. When we want to connect to a system output port, we want to use an input port to do that. Therefore, we search for the *opposite* kind of port.

If in multi-client mode, then this function disconnects the JACK client afterward. At this point, we have got all the data we need, and are not providing a client to each JACK port we create.

If there is no system input port, or no system output port, then we add a virtual port of that type so that the application has something to work with.

Note that, at some pointer, we ought to consider how to deal with transitory system JACK clients and ports, and adjust for it. A kind of miniature form of session management. Also, don't forget about the usefulness of jack_get _ port_by_id() and jack_get_port_by_name().

Error handling:

Not having any JACK input ports present isn't necessarily an error. There may not be any, and there may still be at least one output port.

```
m_error_string = func_message("no JACK input ports available");
error(rterror::WARNING, m_error_string);
```

Also, if there are none, we try to make a virtual port so that the application has something to work with. The only issue is the client number. Currently all virtual ports we create have a client number of 0.

JackPortIsPhysical:

If this flag is added, then only ports corresponding to a physical device are get detected and connected. This might be a useful option to add at a later date.

Returns

Returns the total number of ports found. Note that 0 ports is not necessarily an error; there may be no JACK apps running with exposed ports. If there is no JACK client, then -1 is returned.

Implements seq64::midi_info.

Note that this name will be used for normal ports, so we make sure it reflects the application name.

Note that this function does not call jack_connect(). We need to add a call to jack_on_shutdown() to set up a shutdown callback. We also need to wait on the activation call until we have registered all the ports. Then we (actually the mastermidibus) can call the api_connect() function to activate this JACK client and connect all the ports.

jack_activate(result);

```
13.49.2.13 disconnect()
```

```
void seq64::midi_jack_info::disconnect ( ) [private]
```

13.49.2.14 extract_names()

13.49.2.15 add()

13.49.3 Friends And Related Function Documentation

```
13.49.3.1 midi_jack
```

```
friend class midi_jack [friend]
```

13.49.3.2 jack_process_io

```
int jack_process_io (
          jack_nframes_t nframes,
          void * arg ) [friend]
```

13.49.4 Field Documentation

13.49.4.1 m_multi_client

```
bool seq64::midi_jack_info::m_multi_client [private]
```

In this case, the functions api_init_in(), api_init_out(), api_init_in_sub(), and api_init_out_sub() need to open their own JACK client. Otherwise, they will use the JACK client created here. And this class will have to close out its own client so it will not persist in the JACK client list (e.g. in QJackCtl).

13.49.4.2 m_jack_ports

```
std::vector<midi_jack *> seq64::midi_jack_info::m_jack_ports [private]
```

Not for use with the multi-client option. This list is iterated in the input and output portions of the JACK process callback.

13.49.4.3 m_jack_client

```
jack_client_t* seq64::midi_jack_info::m_jack_client [private]
```

This is actually an opaque pointer; there is no way to get the actual fields in this structure; they can only be accessed through functions in the JACK API. Note that it is also stored as a void pointer in midi_info::m_midi_handle.

13.50 seq64::midi_list Class Reference

This class is the std::list implementation of the midi_container.

Inheritance diagram for seq64::midi_list:



Public Member Functions

• midi_list (sequence &seq)

This constructor fills in the members.

virtual ∼midi_list ()

A rote constructor needed for a base class.

• virtual std::size_t size () const

Returns the size of the container, in midibytes.

• virtual bool done () const

For popping data from the MIDI list, we are done when the container is empty.

• virtual void put (midibyte b)

Provides a way to add a MIDI byte into the list.

• virtual midibyte get () const

Provide a way to get the next byte from the container.

· virtual void clear ()

Provides a way to clear the container.

Private Types

• typedef std::list< midibyte > CharList

Provides the type of this container.

Private Attributes

· CharList m_char_list

The container itself.

Additional Inherited Members

13.50.1 Member Typedef Documentation

```
13.50.1.1 CharList
```

```
typedef std::list<midibyte> seq64::midi_list::CharList [private]
```

This type is basically the same as the midifile::m_char_list container in the midifile module.

13.50.2 Constructor & Destructor Documentation

```
13.50.2.1 midi_list()
```

Parameters

seq The sequence/track object that is using this container.

```
13.50.2.2 ~midi_list()

virtual seq64::midi_list::~midi_list ( ) [inline], [virtual]

13.50.3 Member Function Documentation

13.50.3.1 size()

virtual std::size_t seq64::midi_list::size ( ) const [inline], [virtual]

Reimplemented from seq64::midi_container.

13.50.3.2 done()

virtual bool seq64::midi_list::done ( ) const [inline], [virtual]

Reimplemented from seq64::midi_container.
```

The original seq24 list used an std::list and a push_front operation.

midibyte b) [inline], [virtual]

Implements seq64::midi_container.

virtual void seq64::midi_list::put (

```
13.50.3.4 get()
virtual midibyte seq64::midi_list::get ( ) const [inline], [virtual]
```

In this implementation, m_position_for_get is not used. The elements of the container are popped off backward! This modifies the character list, so it has to be mutable.

Implements seq64::midi_container.

```
13.50.3.5 clear()
```

```
virtual void seq64::midi_list::clear ( ) [inline], [virtual]
```

Implements seq64::midi container.

13.50.4 Field Documentation

```
13.50.4.1 m_char_list
```

```
CharList seq64::midi_list::m_char_list [mutable], [private]
```

It has to be mutable because the const-function get() actually modifies the container when getting a byte.

13.51 seq64::midi_measures Class Reference

Provides a data structure to hold the numeric equivalent of the measures string "measures:beats:divisions" ("m:b↔:d").

Public Member Functions

• midi_measures ()

Default constructor for midi_measures.

midi_measures (int measures, int beats, int divisions)

Principal constructor for midi_measures.

• int measures () const

'Getter' function for member m_measures

void measures (int m)

'Setter' function for member m_measures

• int beats () const

'Getter' function for member m_beats

void beats (int b)

'Setter' function for member m_beats

• int divisions () const

'Getter' function for member m_divisions

• void divisions (int d)

'Setter' function for member m_divisions

Private Attributes

• int m measures

The integral number of measures in the measures-based time.

int m_beats

The integral number of beats in the measures-based time.

· int m_divisions

The integral number of divisions/pulses in the measures-based time.

13.51.1 Detailed Description

More commonly known as "bars:beats:ticks", or "BBT".

13.51.2 Constructor & Destructor Documentation

Parameters

measures	Copied into the m_measures member.
beats	Copied into the m_beats member.
divisions	Copied into the m_divisions member.

13.51.3 Member Function Documentation

Parameters

m The value to which to set the number of measures. We can add validation later.

Parameters

b The value to which to set the number of beats. We can add validation later.

Parameters

d The value to which to set the number of divisions. We can add validation later.

13.51.4 Field Documentation

13.51.4.1 m_measures

int seq64::midi_measures::m_measures [private]

13.51.4.2 m_beats

```
int seq64::midi_measures::m_beats [private]
```

13.51.4.3 m_divisions

```
int seq64::midi_measures::m_divisions [private]
```

There are two possible translations of the two bytes of a division. If the top bit of the 16 bits is 0, then the time division is in "ticks per beat" (or "pulses per quarter note"). If the top bit is 1, then the time division is in "frames per second". This member deals only with the ticks/beat definition.

13.52 seq64::midi_message Class Reference

Provides a handy capsule for a MIDI message, based on the std::vector<unsigned char> data type from the RtMidi project.

Public Types

typedef std::vector< midibyte > container
 Holds the data of the MIDI message.

Public Member Functions

- midi_message ()
 - Constructs an empty MIDI message.
- midibyte operator[] (int i) const
- midibyte & at (int i)
- · const midibyte & at (int i) const
- const char * array () const
- · int count () const
- · bool empty () const
- void push (midibyte b)
- · double timestamp () const
- void timestamp (double t)

Private Attributes

• container m_bytes

Holds the event status and data bytes.

• double m_timestamp

Holds the (optional) timestamp of the MIDI message.

13.52.1 Detailed Description

Please note that the ALSA module in sequencer64's rtmidi infrastructure uses the seq64::event rather than the seq64::midi_message object. For the moment, we will translate between them until we have the interactions between the old and new modules under control.

13.52.2 Member Typedef Documentation

13.52.2.1 container

```
typedef std::vector<midibyte> seq64::midi_message::container
```

Callers should use midi_message::container rather than using the vector directly. Bytes are added by the push() function, and are safely accessed (with bounds-checking) by operator [].

13.52.3 Constructor & Destructor Documentation

```
13.52.3.1 midi_message()
seq64::midi_message::midi_message ( )
```

13.52.4 Member Function Documentation

13.52.4.1 operator[]()

int i) [inline]

midibyte& seq64::midi_message::at (

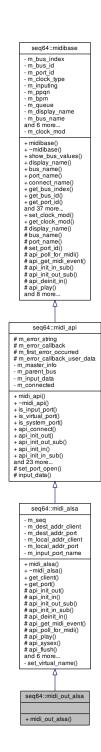
```
13.52.4.3 at() [2/2]
const midibyte& seq64::midi_message::at (
            int i ) const [inline]
13.52.4.4 array()
const char* seq64::midi_message::array ( ) const [inline]
13.52.4.5 count()
int seq64::midi_message::count ( ) const [inline]
13.52.4.6 empty()
bool seq64::midi_message::empty ( ) const [inline]
13.52.4.7 push()
void seq64::midi_message::push (
           midibyte b ) [inline]
13.52.4.8 timestamp() [1/2]
double seq64::midi_message::timestamp ( ) const [inline]
13.52.4.9 timestamp() [2/2]
void seq64::midi_message::timestamp (
            double t ) [inline]
```

13.52.5 Field Documentation



This class implements the ALSA version of a MIDI output object.

Inheritance diagram for seq64::midi_out_alsa:



Public Member Functions

• midi_out_alsa (midibus &parentbus, midi_info &masterinfo)

ALSA MIDI output normal port or virtual port constructor.

Additional Inherited Members

13.53.1 Constructor & Destructor Documentation

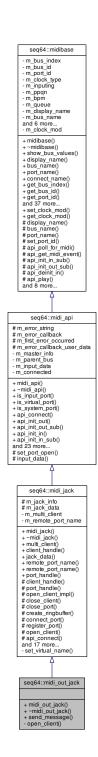
13.53.1.1 midi_out_alsa()

The kind of port is determine by which port-initialization function the mastermidibus calls.

13.54 seq64::midi_out_jack Class Reference

The JACK MIDI output API class.

Inheritance diagram for seq64::midi_out_jack:



Public Member Functions

midi_out_jack (midibus &parentbus, midi_info &masterinfo)

Principal constructor.

• virtual \sim midi_out_jack ()

Destructor.

virtual bool send_message (const midi_message::container &message)

Sends a JACK MIDI output message.

Private Member Functions

virtual bool open_client ()

This function is virtual, so we don't call it in the constructor, using open_client_impl() directly instead.

Additional Inherited Members

13.54.1 Constructor & Destructor Documentation

For Sequencer64, we don't current need to create a midi_out_jack object; all that is needed is created via the api_init_out*() functions.

Parameters

parentbus	Provides the buss object that determines buss-specific parameters of this class.
masterinfo	Provides information about the JACK system as found on this machine.

```
13.54.1.2 ~midi_out_jack()
seq64::midi_out_jack::~midi_out_jack ( ) [virtual]
```

Currently the base class closes the port, closes the JACK client, and cleans up the API data structure.

13.54.2 Member Function Documentation

```
13.54.2.1 send_message()
```

It writes the full message size and the message itself to the JACK ring buffer.

Parameters

message	Provides the vector of message bytes to send.
---------	---

Returns

Returns true if the buffer message and buffer size seem to be written correctly.

13.54.2.2 open_client()

```
virtual bool seq64::midi_out_jack::open_client ( ) [inline], [private], [virtual]
```

This function replaces the RtMidi function "connect()".

Implements seq64::midi jack.

13.55 seq64::midi_port_info Class Reference

A class for holding port information.

Data Structures

struct port_info_t

Hold the information for a single port.

Public Member Functions

• midi_port_info ()

Principal constructor.

 void add (int clientnumber, const std::string &clientname, int portnumber, const std::string &portname, bool makevirtual, bool makesystem, bool makeinput, int queuenumber=SEQ64_BAD_QUEUE_ID)

Adds a set of port information to the port container.

• void add (const midibus *m)

Adds values from a midibus (actually a midibase-derived class).

• void clear ()

This function is useful in replacing the discovered system ports with the manual/virtual ports added in "manual" mode.

- int get_port_count () const
- int get_bus_id (int index) const
- std::string get bus name (int index) const
- int get_port_id (int index) const
- std::string get_port_name (int index) const
- bool get_input (int index) const
- bool get virtual (int index) const
- bool get_system (int index) const
- int get_queue_number (int index) const
- std::string connect_name (int index) const

Provides the bus name and port name in canonical JACK format: "busname:portname".

Private Attributes

• int m_port_count

Holds the number of ports counted.

std::vector< port_info_t > m_port_container

Holds information on all of the ports that were "scanned".

13.55.1 Constructor & Destructor Documentation

```
13.55.1.1 midi_port_info()
seq64::midi_port_info::midi_port_info ( )
```

13.55.2 Member Function Documentation

```
void seq64::midi_port_info::add (
    int clientnumber,
    const std::string & clientname,
    int portnumber,
    const std::string & portname,
    bool makevirtual,
    bool makesystem,
    bool makeinput,
    int queuenumber = SEQ64_BAD_QUEUE_ID )
```

Parameters

clientnumber	Provides the client or buss number for the port. This is a value like
clientname	Provides the system or user-supplied name for the client or buss.
portnumber	Provides the port number, usually re 0.
portname	Provides the system or user-supplied name for the port.
makevirtual	If the system currently has no input or output port available, then we want to create a virtual port so that the application has something to work with.
makesystem	In some systems, we need to create and activate a system port, such as a timer port or an ALSA announce port. For all other ports, this value is false.
makeinput	Indicates if the port is an input port or an output port.
queuenumber	Provides the optional queue number, if applicable. For example, the sequencer64 application grabs the client number (normally valued at 1) from the ALSA subsystem.

```
13.55.2.2 add() [2/2]
void seq64::midi_port_info::add (
            const midibus * m)
13.55.2.3 clear()
void seq64::midi_port_info::clear ( ) [inline]
13.55.2.4 get_port_count()
int seq64::midi_port_info::get_port_count ( ) const [inline]
13.55.2.5 get_bus_id()
int seq64::midi_port_info::get_bus_id (
             int index ) const [inline]
13.55.2.6 get_bus_name()
std::string seq64::midi_port_info::get_bus_name (
            int index ) const [inline]
13.55.2.7 get_port_id()
int seq64::midi_port_info::get_port_id (
           int index ) const [inline]
13.55.2.8 get_port_name()
std::string seq64::midi_port_info::get_port_name (
            int index ) const [inline]
```

13.55.2.9 get_input()

13.55.2.10 get_virtual()

13.55.2.11 get_system()

13.55.2.12 get_queue_number()

13.55.2.13 connect_name()

This function is basically the same as midibase::connect_name() function. If either the bus name or port name are empty, then an empty string is returned.

13.55.3 Field Documentation

13.55.3.1 m_port_count

```
int seq64::midi_port_info::m_port_count [private]
```

13.55.3.2 m_port_container

std::vector<port_info_t> seq64::midi_port_info::m_port_container [private]

13.56 seq64::midi_queue Class Reference

Provides a queue of midi_message structures.

Public Member Functions

• midi queue ()

Default constructor.

~midi_queue ()

Destructor.

· bool empty () const

'Getter' function for member m_size == 0

• int count () const

'Getter' function for member m_size == 0

- · bool full () const
- bool add (const midi_message &mmsg)

As long as we haven't reached our queue size limit, push the message.

• void pop ()

Pops, so to speak, the front message out of the queue, effectively throwing it away.

• midi_message pop_front ()

Pops a copy of the front message.

• void allocate (unsigned queuesize=SEQ64_DEFAULT_QUEUE_SIZE)

This would be better off as a constructor operation.

void deallocate ()

This would be better off as a destructor operation.

• const midi_message & front () const

'Getter' function for member m_ring[m_front]

Private Attributes

- unsigned m_front
- unsigned m_back
- unsigned m size
- unsigned m ring size
- midi_message * m_ring

13.56.1 Detailed Description

This entity used to be a plain structure nested in the midi_in_api class. We made it a class to encapsulate some common operations to save a burden on the callers.

13.56.2 Constructor & Destructor Documentation

```
13.56.2.1 midi_queue()
seq64::midi_queue::midi_queue ( )
13.56.2.2 \simmidi_queue()
seq64::midi_queue::\sim midi_queue ( )
13.56.3 Member Function Documentation
13.56.3.1 empty()
bool seq64::midi_queue::empty ( ) const [inline]
13.56.3.2 count()
int seq64::midi_queue::count ( ) const [inline]
13.56.3.3 full()
bool seq64::midi_queue::full ( ) const [inline]
Returns
     Returns true if the queue size is at maximum.
13.56.3.4 add()
bool seq64::midi\_queue::add (
             const midi_message & mmsg )
```

```
13.56.3.5 pop()

void seq64::midi_queue::pop ( )

One useful call sequence is:

   midi_message latest = queue.front();
   queue.pop();

An alternative is to use the pop_front() function instead.

13.56.3.6 pop_front()
```

Could be a little inefficient, since a couple of copies are made, and we cannot use return-code optimization.

Perhaps at some point we could use move semantics?

midi_message seq64::midi_queue::pop_front ()

Returns

Returns a copy of the message that was in front before the popping. If the queue is empty, an empty (all zeros) message is returned. Can be checked with the midi_message::empty() function.

const midi_message& seq64::midi_queue::front () const [inline]

13.56.4 Field Documentation

```
13.56.4.1 m_front
unsigned seq64::midi_queue::m_front [private]
13.56.4.2 m_back
unsigned seq64::midi_queue::m_back [private]
13.56.4.3 m_size
unsigned seq64::midi_queue::m_size [private]
13.56.4.4 m_ring_size
unsigned seq64::midi_queue::m_ring_size [private]
13.56.4.5 m_ring
midi_message* seq64::midi_queue::m_ring [private]
```

13.57 seq64::midi_splitter Class Reference

This class handles the parsing and writing of MIDI files.

Public Member Functions

• midi_splitter (int ppqn=SEQ64_USE_DEFAULT_PPQN)

Principal constructor.

~midi_splitter ()

A rote destructor.

• bool log main sequence (sequence &seq, int seqnum)

Logs the main sequence (an SMF 0 track) for later usage in splitting the track.

• void initialize ()

Resets the SMF 0 support variables in preparation for parsing a new MIDI file.

void increment (int channel)

Processes a channel number by raising its flag in the m_smf0_channels[] array.

• bool split (perform &p, int screenset)

This function splits an SMF 0 file, splitting all of the channels in the sequence out into separate sequences, and adding each to the perform object.

• int ppqn () const

'Getter' function for member m_ppqn Provides a way to get the actual value of PPQN used in processing the sequences when parse() was called.

· int count () const

'Getter' function for member m_smf0_channels_count

Private Member Functions

• bool split_channel (const sequence &main_seq, sequence *seq, int channel)

This function splits the given sequence into new sequences, one for each channel found in the SMF 0 track.

Private Attributes

int m_ppqn

Provides the current value of the PPQN, which used to be constant and is now only the macro DEFAULT_PPQN.

• bool m_use_default_ppqn

Indicates that the default PPQN is in force.

· int m_smf0_channels_count

Provides support for SMF 0, indicates how many channels were found in the file in a single sequence.

bool m_smf0_channels [16]

Provides support for SMF 0, holds a bool value that indicates the occurrence of a given channel.

• sequence * m_smf0_main_sequence

Provides support for SMF 0, points to the initial SMF 0 sequence, from which the single-channel sequences will be created.

• int m_smf0_seq_number

Provides support for SMF 0, holds the prospective sequence number of the main (SMF 0) sequence.

13.57.1 Detailed Description

In addition to the standard MIDI tracks, it also handles some "private" or "proprietary" tracks specific to Seq24. It does not, however, handle SYSEX events.

13.57.2 Constructor & Destructor Documentation

```
13.57.2.1 midi_splitter()
```

Parameters

ppqn

Provides the initial value of the PPQN setting. It is handled differently for parsing (reading) versus writing the MIDI file.

- · Reading.
 - If set to SEQ64_USE_DEFAULT_PPQN, the legacy application behavior is used. The
 m_ppqn member is set to the default PPQN, DEFAULT_PPQN. The value read from the
 MIDI file, ppqn, is then use to scale the running-time of the sequence relative to
 DEFAULT_PPQN.
 - Otherwise, m_ppqn is set to the value read from the MIDI file. No scaling is done. Since the value gets written, specify ppqn as 0, an obviously bogus value, to get this behavior.
- Writing. This value is written to the MIDI file in the header chunk of the song. Note that the caller
 must query for the PPQN set during parsing, and pass it to the constructor when preparing to
 write the file. See how it is done in the mainwand class.

```
13.57.2.2 ∼midi_splitter()
```

```
seq64::midi_splitter::~midi_splitter ( )
```

13.57.3 Member Function Documentation

13.57.3.1 log_main_sequence()

/param seq The main sequence to be logged.

/param segnum The sequence number of the main sequence.

/return Returns true if the main sequence's address was logged, and false if it was already logged.

13.57.3.2 initialize()

```
void seq64::midi_splitter::initialize ( )
```

13.57.3.3 increment()

If it is the first entry for that channel, m_smf0_channels_count is incremented. We won't check the channel number, to save time, until someday we segfault :-D

Parameters

|--|

13.57.3.4 split()

Lastly, it adds the SMF 0 track as the last track; the user can then examine it before removing it. Is this worth the effort?

There is a little oddity, in that, if the SMF 0 track has events for only one channel, this code will still create a new sequence, as well as the main sequence. Not sure if this is worth extra code to just change the channels on the main sequence and put it into the correct track for the one channel it contains. In fact, we just want to keep it in pattern slot number 16, to keep it out of the way.

Parameters

p	Provides a reference to the perform object into which sequences/tracks are to be added.
screense	The screen-set offset to be used when loading a sequence (track) from the file.

Returns

Returns true if the parsing succeeded. Returns false if no SMF 0 main sequence was logged.

13.57.3.5 ppqn()

```
int seq64::midi_splitter::ppqn ( ) const [inline]
```

The PPQN will be either the global ppqn (legacy behavior) or the value read from the file, depending on the ppqn parameter passed to the midi_splitter constructor.

13.57.3.6 count()

```
int seq64::midi_splitter::count ( ) const [inline]
```

13.57.3.7 split_channel()

Note that the events that are read from the MIDI file have delta times. Sequencer64 converts these delta times to cumulative times. We need to preserve that here. Conversion back to delta times is needed only when saving the sequences to a file. This is done in midi container::fill().

We have to accumulate the delta times in order to be able to set the length of the sequence in pulses.

Luckily, we don't have to worry about copying triggers, since the imported SMF 0 track won't have any Seq24/← Sequencer24 triggers.

It doesn't set the sequence number of the sequence; that is set when the sequence is added to the perform object.

Parameters

main_seq	This parameter is the whole SMF 0 track that was read from the MIDI file. It contains all of the channel data that needs to be split into separate sequences.
s	Provides the new sequence that needs to have its settings made, and all of the selected channel events added to it.
channel	Provides the MIDI channel number (re 0) that marks the channel data the needs to be extracted and added to the new sequence.

Returns

Returns true if at least one event got added. If none were added, the caller should delete the sequence object represented by parameter *s*.

13.57.4 Field Documentation

13.57.4.1 m_ppqn

```
int seq64::midi_splitter::m_ppqn [private]
```

13.57.4.2 m_use_default_ppqn

```
bool seq64::midi_splitter::m_use_default_ppqn [private]
```

```
13.57.4.3 m_smf0_channels_count
```

```
int seq64::midi_splitter::m_smf0_channels_count [private]
```

SMF 1 file parsing will only warn about more than one channel found in a given sequence.

13.57.4.4 m smf0 channels

```
bool seq64::midi_splitter::m_smf0_channels[16] [private]
```

Obviously, we don't have to worry about multiple MIDI busses.

13.57.4.5 m_smf0_main_sequence

```
sequence* seq64::midi_splitter::m_smf0_main_sequence [private]
```

13.57.4.6 m_smf0_seq_number

```
int seq64::midi_splitter::m_smf0_seq_number [private]
```

We want to be able to add that sequence last, for easier and cleaner removal of that sequence by the user.

13.58 seq64::midi_timing Class Reference

We anticipate the need to have a small structure holding the parameters needed to calculate MIDI times within an arbitrary song.

Public Member Functions

• midi_timing ()

Defaults constructor for midi_timing.

midi_timing (midibpm bpminute, int bpmeasure, int beatwidth, int ppqn)

Principal constructor for midi_timing.

• midibpm beats_per_minute () const

'Getter' function for member m_beats_per_minute

void beats_per_minute (midibpm b)

 ${\it 'Setter' function for member m_beats_per_minute}$

• int beats_per_measure () const

'Getter' function for member m_beats_per_measure

void beats_per_measure (int b)

'Setter' function for member m_beats_per_measure

• int beat_width () const

'Getter' function for member m_beats_per_beat_width

void beat_width (int bw)

'Setter' function for member m_beats_per_beat_width

• int ppqn () const

'Getter' function for member m_ppqn

• void ppqn (int p)

'Setter' function for member m_ppqn

Private Attributes

• midibpm m_beats_per_minute

This value should match the BPM value selected when editing the song.

• int m_beats_per_measure

This value should match the numerator value selected when editing the sequence.

• int m_beat_width

This value should match the denominator value selected when editing the sequence.

• int m_ppqn

This value provides the precision of the MIDI song.

13.58.1 Detailed Description

Although Seq24/Sequencer64 currently are heavily dependent on hard-wired values, that will be rectified eventually, so let us get ready for it.

13.58.2 Constructor & Destructor Documentation

Parameters

bpminute	Copied into the m_beats_per_minute member.
bpmeasure	Copied into the m_beats_per_measure member.
beatwidth	Copied into the m_beat_width member.
ppqn	Copied into the m_ppqn member.

13.58.3 Member Function Documentation

```
13.58.3.1 beats_per_minute() [1/2]
midibpm seq64::midi_timing::beats_per_minute ( ) const [inline]
13.58.3.2 beats_per_minute() [2/2]
void seq64::midi_timing::beats_per_minute (
              midibpm b ) [inline]
Parameters
     The value to which to set the number of beats/minute. We can add validation later.
13.58.3.3 beats_per_measure() [1/2]
int seq64::midi_timing::beats_per_measure ( ) const [inline]
13.58.3.4 beats_per_measure() [2/2]
void seq64::midi_timing::beats_per_measure (
             int b ) [inline]
Parameters
     The value to which to set the number of beats/measure. We can add validation later.
13.58.3.5 beat_width() [1/2]
int seq64::midi_timing::beat_width ( ) const [inline]
13.58.3.6 beat_width() [2/2]
void seq64::midi_timing::beat_width (
             int bw ) [inline]
```

Parameters

bw

The value to which to set the number of beats in the denominator of the time signature. We can add validation later.

Parameters

p The value to which to set the PPQN member. We can add validation later.

13.58.4 Field Documentation

```
13.58.4.1 m_beats_per_minute
```

```
midibpm seq64::midi_timing::m_beats_per_minute [private]
```

This value is most commonly set to 120, but is also read from the MIDI file. This value is needed if one want to calculate durations in true time units such as seconds, but is not needed to calculate the number of pulses/ticks/divisions.

```
13.58.4.2 m_beats_per_measure
```

```
int seq64::midi_timing::m_beats_per_measure [private]
```

This value is most commonly set to 4.

13.58.4.3 m_beat_width

```
int seq64::midi_timing::m_beat_width [private]
```

This value is most commonly set to 4, meaning that the fundamental beat unit is the quarter note.



Inheritance diagram for seq64::midi_vector:



Public Member Functions

• midi_vector (sequence &seq)

This constructor fills in the members of this class.

virtual ∼midi_vector ()

A rote constructor needed for a base class.

• virtual std::size_t size () const

• virtual bool done () const

For iterating through the data in the MIDI vector, we are done when we've gotten the last element of the container.

• virtual void put (midibyte b)

Provides a way to add a MIDI byte into the list.

• virtual midibyte get () const

Provide a way to get the next byte from the container.

· virtual void clear ()

Provides a way to clear the container.

Private Types

typedef std::vector< midibyte > CharVector

Provides the type of this container.

Private Attributes

· CharVector m char vector

The container itself.

Additional Inherited Members

13.59.1 Member Typedef Documentation

```
13.59.1.1 CharVector
```

```
typedef std::vector<midibyte> seq64::midi_vector::CharVector [private]
```

13.59.2 Constructor & Destructor Documentation

```
13.59.2.1 midi_vector()
```

Parameters

provides a reference to the sequence/track for which this container holds MIDI data.

```
13.59.2.2 ~midi_vector()
```

virtual seq64::midi_vector::~midi_vector () [inline], [virtual]

13.59.3 Member Function Documentation

```
13.59.3.1 size()

virtual std::size_t seq64::midi_vector::size ( ) const [inline], [virtual]
```

Returns

Returns the size of the container, in midibytes.

Reimplemented from seq64::midi_container.

```
13.59.3.2 done()
virtual bool seq64::midi_vector::done ( ) const [inline], [virtual]
```

Returns

Returns true if the position is greater than or equal to the size of the character vector.

 $Reimplemented \ from \ seq 64:: midi_container.$

The original seq24 list used an std::list and a push_front operation.

Parameters

b Provides the MIDI byte to push_back() into the character vector.

Implements seq64::midi_container.

```
13.59.3.4 get()
```

```
virtual midibyte seq64::midi_vector::get ( ) const [inline], [virtual]
```

In this implementation, m_position_for_get is used. As a side-effect, the position value is incremented.

Returns

Returns the next byte in the character vector.

Implements seq64::midi_container.

13.59.3.5 clear()

```
virtual void seq64::midi_vector::clear ( ) [inline], [virtual]
```

Implements seq64::midi_container.

13.59.4 Field Documentation

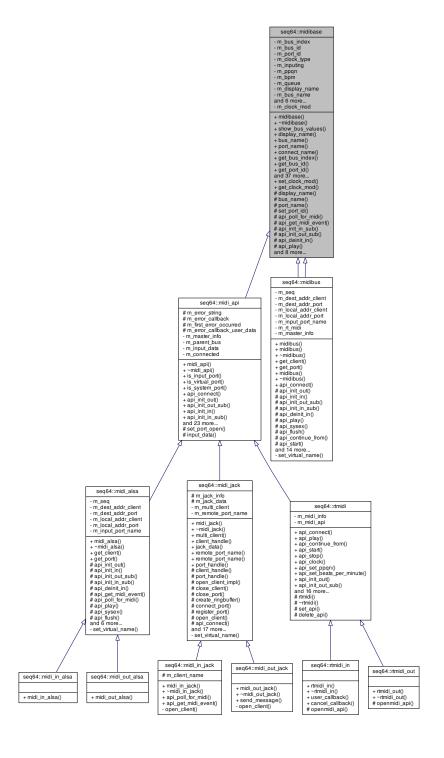
```
13.59.4.1 m_char_vector
```

```
CharVector seq64::midi_vector::m_char_vector [private]
```

13.60 seq64::midibase Class Reference

This class implements with ALSA version of the midibase object.

Inheritance diagram for seq64::midibase:



Public Member Functions

midibase (const std::string &appname, const std::string &busname="", const std::string &portname="", int index=0, int bus_id=SEQ64_NO_BUS, int port_id=SEQ64_NO_PORT, int queue=SEQ64_NO_QUEUE, int ppqn=SEQ64_USE_DEFAULT_PPQN, midibpm bpm=SEQ64_DEFAULT_BPM, bool makevirtual=false, bool isinput=false, bool makesystem=false)

Creates a normal MIDI port, which will correspond to an existing system MIDI port, such as one provided by Timidity or a running JACK application, or a virtual port, which has a name made up by the application.

virtual ∼midibase ()

A rote empty destructor.

• void show_bus_values ()

Shows most midibase members.

const std::string & display_name () const

'Getter' function for member m_display_name

const std::string & bus_name () const

'Getter' function for member m_bus_name

const std::string & port_name () const

'Getter' function for member m_port_name

std::string connect_name () const

'Getter' function for member m_bus_name and m_port_name Concatenates the bus and port names into a string of the form "busname:portname".

• int get_bus_index () const

'Getter' function for member m_bus_index

int get_bus_id () const

'Getter' function for member m_bus_id

int get_port_id () const

'Getter' function for member m_port_id

• int ppqn () const

'Getter' function for member m_ppqn;

· midibpm bpm () const

'Getter' function for member m bpm;

bool match (int bus, int port)

Checks if the given parameters match the current bus and port numbers.

bool is_virtual_port () const

'Getter' function for member m is virtual port

void is_virtual_port (bool flag)

'Setter' function for member m_is_virtual_port This function is needed in the rtmidi library to set the is-virtual flag in the api_init_*_sub() functions, so that midi_alsa, midi_jack (and any other additional APIs that end up supported by our heavily-refactored rtmidi library), as well as the original midibus, can know that they represent a virtual port.

• bool is_input_port () const

'Getter' function for member m_is_input_port

bool is_output_port () const

'Getter' function for member! m is input port

void is_input_port (bool flag)

'Setter' function for member m_is_input_port

bool is_system_port () const

'Getter' function for member m_is_system_port

void set_system_port_flag ()

'Setter' function for member m_is_system_port Can only set it to true.

void set_clock (clock_e clocktype)

'Setter' function for member m_clock_type

• clock_e get_clock () const

'Getter' function for member m_clock_type

void set clock status (clock e clocktype)

'Setter' function for member m_clock_type

bool get_input () const

'Getter' function for member m_inputing

void set_input_status (bool flag)

'Setter' function for member m_inputing

int queue_number () const

'Getter' function for member m_queue

void set_bus_id (int id)

'Setter' function for member m_bus_id Useful for setting the buss ID when using the rtmidi_info object to create a list of busses and ports.

• void set_name (const std::string &appname, const std::string &busname, const std::string &portname)

Sets the name of the buss by assembling the name components obtained from the system in a straightforward manner:

• void set_alt_name (const std::string &appname, const std::string &busname, const std::string &portname)

Sets the name of the buss in a different way.

void set_multi_name (const std::string &appname, const std::string &localbusname, const std::string &remoteportname)

Sets the name of the buss in yet another different way, suitable for the multiclient mode of some APIs (such as JACK).

• int poll_for_midi ()

Polls for MIDI events.

bool get midi event (event *inev)

Obtains a MIDI event.

· bool init_out ()

Initialize the MIDI output port.

• bool init_in ()

Initialize the MIDI input port.

• bool deinit in ()

Deinitialize the MIDI input.

bool init_out_sub ()

Initialize the output in a different way?

• bool init_in_sub ()

Initialize the output in a different way?

void play (event *e24, midibyte channel)

This play() function takes a native event, encodes it to a MIDI sequencer event, sets the broadcasting to the subscribers, sets the direct-passing mode to send the event without queueing, and puts it in the queue.

void sysex (event *e24)

Takes a native SYSEX event, encodes it to an ALSA event, and then puts it in the queue.

· void flush ()

Flushes our local queue events out into ALSA.

• void start ()

This function gets the MIDI clock a-runnin', if the clock type is not e_clock_off.

• void stop ()

Stop the MIDI buss.

void clock (midipulse tick)

Generates the MIDI clock, starting at the given tick value.

· void continue_from (midipulse tick)

Continue from the given tick.

void init_clock (midipulse tick)

Initialize the clock, continuing from the given tick.

• void print ()

Prints m_name.

bool set_input (bool inputing)

Set status to of "inputting" to the given value.

Static Public Member Functions

static void set_clock_mod (int clockmod)

Set the clock mod to the given value, if legal.

static int get_clock_mod ()

Get the clock mod value.

Protected Member Functions

void display_name (const std::string &name)

'Setter' function for member m_display_name

void bus_name (const std::string &name)

'Setter' function for member m_bus_name

void port_name (const std::string &name)

'Setter' function for member m_port_name

void set_port_id (int id)

'Setter' function for member m_port_id Useful for setting the port ID when using the rtmidi_info object to inspect and create a list of busses and ports.

• virtual int api_poll_for_midi ()

Now defined in the ALSA implementation, and used by mastermidibus.

virtual bool api_get_midi_event (event *inev)

Used in the JACK implementation.

• virtual bool api_init_in_sub ()

Not defined in the PortMidi implementation.

virtual bool api_init_out_sub ()

Not defined in the PortMidi implementation.

virtual bool api_deinit_in ()

Not defined in the PortMidi implementation.

- virtual void api_play (event *e24, midibyte channel)=0
- virtual void api_sysex (event *)

Handles implementation details for SysEx messages.

• virtual void api flush ()

Handles implementation details for the flush() function.

- virtual bool api_init_in ()=0
- virtual bool api_init_out ()=0
- virtual void api_continue_from (midipulse tick, midipulse beats)=0
- virtual void api_start ()=0
- virtual void api_stop ()=0
- virtual void api_clock (midipulse tick)=0

Private Attributes

· const int m bus index

Provides the index of the midibase object in either the input list or the output list.

int m_bus_id

The buss ID of the midibase object.

• int m port id

The port ID of the midibase object.

clock_e m_clock_type

The type of clock to use.

bool m_inputing

This flag indicates if an input bus has been selected for action as an input device (such as a MIDI controller).

• int m_ppqn

Provides the PPQN value in force, currently a constant.

• midibpm m_bpm

Provides the PPQN value in force, currently a constant.

• int m_queue

Another ID of the MIDI queue? This is an implementation-dependent value.

• std::string m_display_name

Holds the full display name of the bus, index, ID numbers, and item names.

• std::string m_bus_name

The name of the MIDI buss.

• std::string m_port_name

The name of the MIDI port.

• midipulse m_lasttick

The last (most recent? final?) tick.

• bool m_is_virtual_port

Indicates if the port is to be a virtual port.

• bool m_is_input_port

Indicates if the port is to be an input (versus output) port.

bool m_is_system_port

Indicates if the port is a system port.

• mutex m_mutex

Locking mutex.

Static Private Attributes

• static int m_clock_mod

This is another name for "16 * 4".

Friends

· class mastermidibus

The master MIDI bus sets up the buss.

13.60.1 Constructor & Destructor Documentation

13.60.1.1 midibase()

Provides a constructor with client number, port number, name of client, name of port.

This constructor is the one that seems to be the one that is used for the MIDI input and output busses, when the [manual-alsa-ports] option is *not* in force. Also used for the announce buss, and in the mastermidibase::port_start() function.

Parameters

appname	Provides the the name of the application. The derived class will determine this name.
busname	Provides the ALSA client name or the MIDI subsystem name (e.g. "TiMidity"). If empty, a name will be assembled by the derived class at port-setup time.
portname	Provides the port name. This item defaults to empty, which means the port name should be obtained via the API, or be assembled by the derived class at port-setup time.
index	Provides the ordinal of this buss/port, mostly for display purposes.
bus_id Indicates the port ID. Defaults to SEQ64_NO_PORT.	
If SEQ64_NO_PORT, the derived class will get the port ID at port-setup time.	
queue	
Provides the PPQN value. Defaults to SEQ64_USE_DEFAULT_PPQN.	
bpm	Provides the BPM value. Defaults to SEQ64_DEFAULT_BPM.
makevirtual	Indicates that the port represented by this object is to be virtual. Defaults to false. This could also be set via the init_in(), init_out(), init_in_sub(), or init_out_sub() routines. Doing it here seems okay.
isinput	Indicates that this midibus represents and input port, as opposed to an output port.
makesystem	Indicates that the port represented by this object is a system port. Currently true only for ALSA system ports (timer or announce ports).

```
13.60.1.2 \sim midibase()
seq64::midibase::\sim midibase ( ) [virtual]
13.60.2 Member Function Documentation
13.60.2.1 show_bus_values()
void seq64::midibase::show_bus_values ( )
13.60.2.2 display_name() [1/2]
const std::string& seq64::midibase::display_name ( ) const [inline]
13.60.2.3 bus_name() [1/2]
const std::string& seq64::midibase::bus_name ( ) const [inline]
13.60.2.4 port_name() [1/2]
const std::string& seq64::midibase::port_name ( ) const [inline]
13.60.2.5 connect_name()
std::string seq64::midibase::connect_name ( ) const
If either name is empty, an empty string is returned.
13.60.2.6 get_bus_index()
int seq64::midibase::get_bus_index ( ) const [inline]
```

```
13.60.2.7 get_bus_id()
int seq64::midibase::get_bus_id ( ) const [inline]
13.60.2.8 get_port_id()
int seq64::midibase::get_port_id ( ) const [inline]
13.60.2.9 ppqn()
int seq64::midibase::ppqn ( ) const [inline]
13.60.2.10 bpm()
midibpm seq64::midibase::bpm ( ) const [inline]
13.60.2.11 match()
bool seq64::midibase::match (
             int bus,
             int port ) [inline]
13.60.2.12 is_virtual_port() [1/2]
bool seq64::midibase::is_virtual_port ( ) const [inline]
13.60.2.13 is_virtual_port() [2/2]
void seq64::midibase::is_virtual_port (
           bool flag ) [inline]
```

```
13.60.2.14 is_input_port() [1/2]
bool seq64::midibase::is_input_port ( ) const [inline]
13.60.2.15 is_output_port()
bool seq64::midibase::is_output_port ( ) const [inline]
13.60.2.16 is_input_port() [2/2]
void seq64::midibase::is_input_port (
            bool flag ) [inline]
13.60.2.17 is_system_port()
bool seq64::midibase::is_system_port ( ) const [inline]
13.60.2.18 set_system_port_flag()
void seq64::midibase::set_system_port_flag ( ) [inline]
13.60.2.19 set_clock()
void seq64::midibase::set_clock (
             clock_e clocktype ) [inline]
Parameters
 clocktype
            The value used to set the clock-type.
13.60.2.20 get_clock()
clock_e seq64::midibase::get_clock ( ) const [inline]
```

```
13.60.2.21 set_clock_status()
void seq64::midibase::set_clock_status (
            clock_e clocktype ) [inline]
13.60.2.22 get_input()
bool seq64::midibase::get_input ( ) const [inline]
13.60.2.23 set_input_status()
void seq64::midibase::set_input_status (
            bool flag ) [inline]
13.60.2.24 queue_number()
int seq64::midibase::queue_number ( ) const [inline]
13.60.2.25 set_bus_id()
void seq64::midibase::set_bus_id (
             int id ) [inline]
```

Would be protected, but midi_alsa needs to change this value to reflect the user-client ID actually assigned by ALSA. (That value ranges from 128 to 191.)

13.60.2.26 set_name()

[0] 128:2 seq64:seq64 port 2

Parameters

appname	This is the name of the client, or application. Not to be confused with the ALSA client-name, which is actually a buss or subsystem name.
busname	Provides the name of the sub-system, such as "Midi Through" or "TiMidity".
	Provides the name of the port. In ALSA, this is something like "busname port X".
Generated by Do	xygen

13.60.2.27 set alt name()

If the port is virtual, this function just calls set_name(). Otherwise, it reassembles the name so that it refers to a port found on the system, but modified to make it a unique application port. For example:

```
[0] 128:0 yoshimi:midi in
```

is transformed to this:

```
[0] 128:0 seg64:yoshimi midi in
```

As a side-effect, the "short" portname is changed, from (for example) "midi in" to "yoshimi midi in".

Parameters

appname	This is the name of the client, or application. Not to be confused with the ALSA/JACK client-name, which is actually a buss or subsystem name.
busname	Provides the name of the sub-system, such as "Midi Through", "TiMidity", or "seq64".
portname	Provides the name of the port. In JACK, this should be the full port name, such as "qmidiarp:in".

13.60.2.28 set_multi_name()

If the port is virtual, this function just calls set_name(). Otherwise, it reassembles the name so that it refers to a port found on the system, but modified to make it a unique client port. For example:

```
[0] 128:0 yoshimi:midi in
```

is transformed to this:

```
[0] 128:0 seq64-yoshimi:midi in
```

The name in the latter is the original buss name, "seq64" plus the remote port's buss name (extracted from the long name), plus the remote port's short port name (extracted from the long name).

Internal parameter:

Parameters

appname	This is the name of the client, or application. Not to be confused with the ALSA client-name, which is actually a buss or subsystem name.
localbusname	Provides the name of the sub-system, such as "Midi Through", "TiMidity", "yoshimi", or "seq64". It is assumed this parameter has already been set properly.
remoteportname	Provides the name of the port. In JACK, this should be the long port name, such as "qmidiarp:in" or "yoshimi:midi in". It is assumed this parameter has already been set properly.

13.60.2.29 set_clock_mod()

Parameters

	clockmod	If this value is not equal to 0, it is used to set the static member m_clock_mod.
--	----------	---

13.60.2.30 get_clock_mod()

```
static int seq64::midibase::get_clock_mod ( ) [inline], [static]
```

13.60.2.31 poll_for_midi()

```
int seq64::midibase::poll_for_midi ( )
```

Returns

Returns a value greater than 0 if MIDI events are available. Otherwise 0 is returned, or -1 for some APIs (ALSA) when an internal error occurs.

13.60.2.32 get_midi_event()

Parameters

Returns

Returns true if an event was found, thus making the return parameter useful.

```
13.60.2.33 init_out()
bool seq64::midibase::init_out ( )
```

Returns

Returns true unless setting up MIDI failed in some way.

```
13.60.2.34 init_in()
bool seq64::midibase::init_in ( )
```

Returns

Returns true unless setting up MIDI failed in some way.

```
13.60.2.35 deinit_in()
bool seq64::midibase::deinit_in ( )
```

Set the input and the output ports. The destination port is actually our local port.

Returns

Returns true, unless an error occurs.

```
13.60.2.36 init_out_sub()
bool seq64::midibase::init_out_sub ( )
```

Returns

Returns true unless setting up the ALSA port failed in some way.

```
13.60.2.37 init_in_sub()
```

```
bool seq64::midibase::init_in_sub ( )
```

Returns

Returns true unless setting up the ALSA port failed in some way.

13.60.2.38 play()

Threadsafe

Parameters

e24	The event to be played on this bus. For speed, we don't bother to check the pointer.
channel	The channel of the playback.

13.60.2.39 sysex()

Parameters

e24 The event to be ha

13.60.2.40 flush()

```
void seq64::midibase::flush ( )
```

13.60.2.41 start()

```
void seq64::midibase::start ( )
```

13.60.2.42 stop()

```
void seq64::midibase::stop ( )
```

13.60.2.43 clock()

Threadsafe

Parameters

tick Provides the starting tick.

13.60.2.44 continue_from()

Tell the device that we are going to start at a certain position (starting_tick). If there is anything left, then wait for next beat (16th note) to start clocking.

Parameters

```
tick The continuing tick.
```

13.60.2.45 init_clock()

This function doesn't depend upon the MIDI API in use.

Parameters

tick The starting tick.

```
13.60.2.46 print()
void seq64::midibase::print ( )

13.60.2.47 set_input()
```

bool seq64::midibase::set_input (

bool inputing)

If the parameter is true, then init_in() is called; otherwise, deinit_in() is called.

Parameters

inputing	The inputing value to set. For input system ports, it is always set to true, no matter how it is
	configured in the "rc" file.

```
13.60.2.52 api_poll_for_midi()
```

```
virtual int seq64::midibase::api_poll_for_midi ( ) [inline], [protected], [virtual]
```

Also used in the JACK implementation.

Reimplemented in seq64::midi_in_jack, seq64::midi_jack, seq64::midi_alsa, seq64::rtmidi, seq64::midi_api, and seq64::midibus.

13.60.2.53 api_get_midi_event()

Reimplemented in seq64::midi_in_jack, seq64::midi_jack, seq64::midi_alsa, seq64::midi_api, seq64::rtmidi, and seq64::midibus.

13.60.2.54 api_init_in_sub()

```
virtual bool seq64::midibase::api_init_in_sub () [inline], [protected], [virtual]
```

Reimplemented in seq64::midi_jack, seq64::midibus, seq64::midi_alsa, seq64::midi_api, seq64::rtmidi, and seq64::midibus.

13.60.2.55 api_init_out_sub()

```
virtual bool seq64::midibase::api_init_out_sub ( ) [inline], [protected], [virtual]
```

 $Reimplemented \ in \ seq64::midi_jack, \ seq64::midi_alsa, \ seq64::midi_api, \ seq64::rtmidi, \ and \ seq64::midibus.$

13.60.2.56 api_deinit_in()

```
virtual bool seq64::midibase::api_deinit_in ( ) [inline], [protected], [virtual]
```

Reimplemented in seq64::midi_jack, seq64::midibus, seq64::midi_alsa, seq64::midi_api, seq64::rtmidi, and seq64::midibus.

```
13.60.2.57 api_play()
```

Implemented in seq64::midi_jack, seq64::midibus, seq64::midi_alsa, seq64::midi_api, seq64::midibus, and seq64
::rtmidi.

```
13.60.2.58 api_sysex()
```

The e24 parameter, the SysEx event pointer, is unused here.

Reimplemented in seq64::midi_jack, seq64::midi_alsa, seq64::midi_alsa, seq64::midi_api.

```
13.60.2.59 api_flush()
```

```
virtual void seq64::midibase::api_flush ( ) [inline], [protected], [virtual]
```

Reimplemented in seq64::midi_jack, seq64::midi_alsa, seq64::midi_alsa, seq64::midi_api.

```
13.60.2.60 api_init_in()
```

```
virtual bool seq64::midibase::api_init_in ( ) [protected], [pure virtual]
```

Implemented in seq64::midi_jack, seq64::midibus, seq64::midi_alsa, seq64::midi_api, seq64::rtmidi, and seq64-::midibus.

```
13.60.2.61 api_init_out()
```

```
virtual bool seq64::midibase::api_init_out ( ) [protected], [pure virtual]
```

Implemented in seq64::midi_jack, seq64::midibus, seq64::midi_alsa, seq64::midi_api, seq64::rtmidi, and seq64::midibus.

13.60.2.62 api_continue_from()

Implemented in seq64::midi_jack, seq64::midibus, seq64::midi_alsa, seq64::midi_api, seq64::midibus, and seq64
::rtmidi.

13.60.2.63 api_start()

```
virtual void seq64::midibase::api_start ( ) [protected], [pure virtual]
```

Implemented in seq64::midi_jack, seq64::midibus, seq64::midi_alsa, seq64::midi_api, seq64::midibus, and seq64
::rtmidi.

13.60.2.64 api_stop()

```
virtual void seq64::midibase::api_stop ( ) [protected], [pure virtual]
```

Implemented in seq64::midi_jack, seq64::midibus, seq64::midi_alsa, seq64::midi_api, seq64::midibus, and seq64
::rtmidi.

13.60.2.65 api_clock()

Implemented in seq64::midi_jack, seq64::midibus, seq64::midi_alsa, seq64::midi_api, seq64::midibus, and seq64
::rtmidi.

13.60.3 Friends And Related Function Documentation

13.60.3.1 mastermidibus

friend class mastermidibus [friend]

13.60.4 Field Documentation

```
13.60.4.1 m_clock_mod
int seq64::midibase::m_clock_mod [static], [private]
Initialize this static member.
13.60.4.2 m_bus_index
const int seq64::midibase::m_bus_index [private]
Otherwise, it is currently -1.
13.60.4.3 m_bus_id
int seq64::midibase::m_bus_id [private]
For example, on one system the IDs are 14 (MIDI Through), 128 (TiMidity), and 129 (Yoshimi).
13.60.4.4 m_port_id
int seq64::midibase::m_port_id [private]
13.60.4.5 m_clock_type
clock_e seq64::midibase::m_clock_type [private]
13.60.4.6 m_inputing
bool seq64::midibase::m_inputing [private]
It is turned on if the user selects the port in the Options / MIDI Input tab.
13.60.4.7 m_ppqn
int seq64::midibase::m_ppqn [private]
Some APIs can control or use this value.
13.60.4.8 m_bpm
midibpm seq64::midibase::m_bpm [private]
```

Some APIs can control or use this value.

```
13.60.4.9 m_queue
```

```
int seq64::midibase::m_queue [private]
```

For ALSA, it is the ALSA queue number. For PortMidi, this is the old "m_pm_num" value. For RtMidi, it is not currently used.

```
13.60.4.10 m_display_name
```

```
std::string seq64::midibase::m_display_name [private]
```

Assembled by the set name() function.

```
13.60.4.11 m_bus_name
```

```
std::string seq64::midibase::m_bus_name [private]
```

This should be something like a major device name or the name of a subsystem such as Timidity.

```
13.60.4.12 m_port_name
```

```
std::string seq64::midibase::m_port_name [private]
```

This should be the name of a specific device or port on a major device.

```
13.60.4.13 m lasttick
```

```
midipulse seq64::midibase::m_lasttick [private]
```

```
13.60.4.14 m_is_virtual_port
```

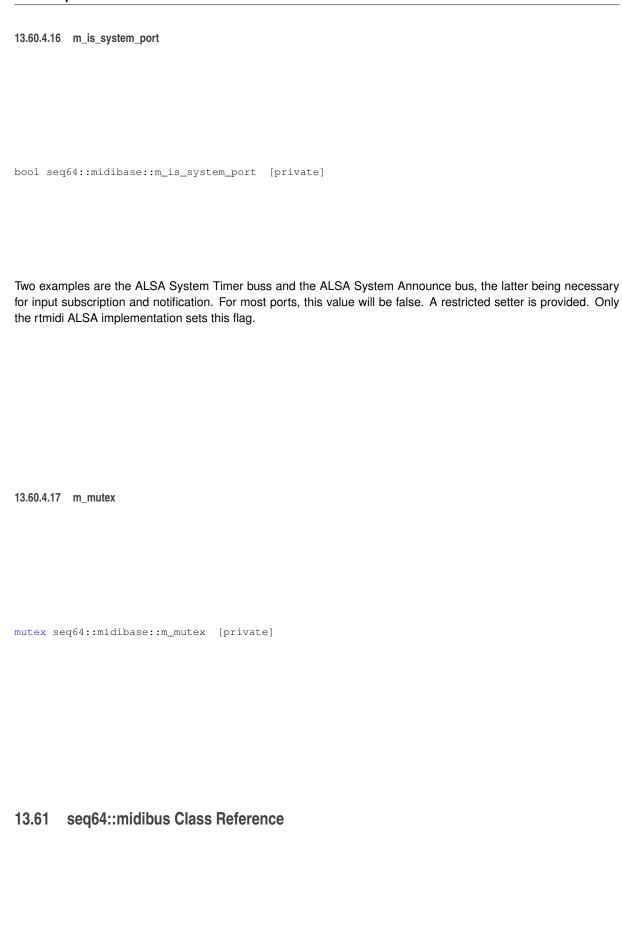
```
bool seq64::midibase::m_is_virtual_port [private]
```

The default is to create a system port (true).

```
13.60.4.15 m_is_input_port
```

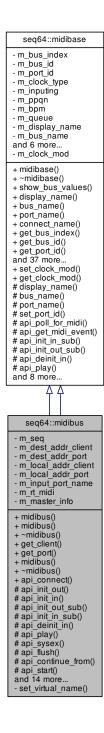
```
bool seq64::midibase::m_is_input_port [private]
```

It matters when we are creating the name of the port, where we don't want an input virtual port to have the same name as an output virtual port... one of them will fail.



This class implements with ALSA version of the midibus object.

Inheritance diagram for seq64::midibus:



Public Member Functions

- midibus (int localclient, int destclient, int destport, snd_seq_t *seq, const std::string &client_name, const std::string &port_name, int index, int queue, int ppqn=SEQ64_USE_DEFAULT_PPQN, midibpm bpm=SE←Q64_DEFAULT_BPM)
- midibus (int localclient, snd_seq_t *seq, int index, int bus_id, int queue, int ppqn=SEQ64_USE_DEFAULT
 —PPQN, midibpm bpm=SEQ64_DEFAULT_BPM)

```
    virtual ~midibus ()
```

The destructor closes out the RtMidi MIDI infrastructure.

virtual int get client () const

'Getter' function for member m_dest_addr_client The address of client.

virtual int get_port () const

'Getter' function for member m_dest_addr_port

midibus (rtmidi_info &rt, int index, bool makevirtual=SEQ64_MIDI_NORMAL_PORT, bool isinput=SEQ64_←
MIDI_OUTPUT_PORT, int bussoverride=SEQ64_NO_BUS, bool makesystem=false)

Normal-port and virtual-port constructor.

- virtual ∼midibus ()
- virtual bool api_connect ()

Connects to another port.

Protected Member Functions

```
    virtual bool api_init_out ()
```

Initializes the MIDI output port.

virtual bool api_init_in ()

Initializes the MIDI input port.

virtual bool api_init_out_sub ()

Initializes the MIDI virtual output port.

virtual bool api_init_in_sub ()

Initializes the MIDI virtual input port.

virtual bool api_deinit_in ()

Forwards the de-initialization call to the API object that implements it.

virtual void api_play (event *e24, midibyte channel)

Takes a native event, and encodes to a Windows message, and writes it to the queue.

- virtual void api_sysex (event *e24)
- virtual void api_flush ()
- virtual void api_continue_from (midipulse tick, midipulse beats)

Continue from the given tick.

virtual void api start ()

Sets the MIDI clock a-runnin', if the clock type is not e_clock_off.

virtual void api_stop ()

Stops the MIDI clock, if the clock-type is not e clock off.

virtual void api_clock (midipulse tick)

Generates MIDI clock.

- virtual bool api_init_in ()
- virtual bool api_init_in_sub ()
- virtual bool api init out ()
- virtual bool api_init_out_sub ()
- virtual bool api deinit in ()
- virtual bool api_get_midi_event (event *inev)

Gets a MIDI event.

• virtual int api_poll_for_midi ()

Polls for MIDI events.

- virtual void api_continue_from (midipulse tick, midipulse beats)
- virtual void api start ()
- virtual void api_stop ()
- virtual void api_clock (midipulse tick)
- virtual void api_play (event *e24, midibyte channel)

Private Member Functions

• bool set_virtual_name (int portid, const std::string &portname)

Private Attributes

```
 snd seq t *const m seq
```

ALSA sequencer client handle.

· const int m_dest_addr_client

Destination address of client.

• const int m_dest_addr_port

Destination port of client.

· const int m_local_addr_client

Local address of client.

int m_local_addr_port

Local port of client.

• const std::string m_input_port_name

Holds the port name for the ALSA MIDI input port.

rtmidi * m_rt_midi

The RtMidi API interface object this midibus will be creating and then using.

· rtmidi_info & m_master_info

For Sequencer64, the ALSA model used requires that all the midibus objects use the same ASLA sequencer "handle".

Friends

· class mastermidibus

The master MIDI bus sets up the buss.

Additional Inherited Members

13.61.1 Detailed Description

This class implements with rtmidi version of the midibus object.

13.61.2 Constructor & Destructor Documentation

```
13.61.2.2 midibus() [2/3]
seq64::midibus::midibus (
            int localclient,
             snd_seq_t * seq,
             int index,
             int bus_id,
             int queue,
             int ppqn = SEQ64_USE_DEFAULT_PPQN,
             midibpm bpm = SEQ64_DEFAULT_BPM )
13.61.2.3 ∼midibus() [1/2]
seq64::midibus::\sim midibus ( ) [virtual]
13.61.2.4 midibus() [3/3]
seq64::midibus::midibus (
             rtmidi_info & rt,
             int index,
             bool makevirtual = SEQ64_MIDI_NORMAL_PORT,
             bool isinput = SEQ64_MIDI_OUTPUT_PORT,
             int bussoverride = SEQ64_NO_BUS,
             bool makesystem = false)
```

rt	Provides the rtmidi_info object to use to obtain the client ID (buss ID), port ID, and port name, as obtained via calls to the ALSA, JACK, Core MIDI, or Windows MM subsystems. We need it to provide the single ALSA "handle" needed in the in Sequencer64 buss model, where the master MIDI buss provides it to be used by all the MIDI buss objects.
index	This is the index into the rtmidi object, and is used to get the desired client and port information. It is an index into the info container held by the rtmidi object.
makevirtual	Indicates that the port is virtual, as opposed to normal.
isinput	Indicates that the port is an input port, as opposed to an output port.
bussoverride	Optional buss ID, if not equal to the index parameter.
makesystem	Indicates that the port is also a system port (i.e. always present).

```
13.61.2.5 \simmidibus() [2/2] virtual seq64::midibus::\simmidibus ( ) [virtual]
```

13.61.3 Member Function Documentation

```
13.61.3.1 get_client()
virtual int seq64::midibus::get_client ( ) const [inline], [virtual]

13.61.3.2 get_port()
virtual int seq64::midibus::get_port ( ) const [inline], [virtual]

13.61.3.3 api_init_out() [1/2]

bool seq64::midibus::api_init_out ( ) [protected], [virtual]
```

Currently, we use the default values for the rtmidi API, the queue number, and the queue size.

Returns

Returns true if the output port was successfully opened.

Implements seq64::midibase.

```
13.61.3.4 api_init_in() [1/2]
bool seq64::midibus::api_init_in ( ) [protected], [virtual]
```

Returns

Returns true if the input port was successfully opened.

Implements seq64::midibase.

```
13.61.3.5 api_init_out_sub() [1/2]
bool seq64::midibus::api_init_out_sub ( ) [protected], [virtual]
```

Returns

Returns true if the output port was successfully opened.

Reimplemented from seq64::midibase.

```
13.61.3.6 api_init_in_sub() [1/2]
bool seq64::midibus::api_init_in_sub ( ) [protected], [virtual]
```

Returns

Returns true if the input port was successfully opened.

Reimplemented from seq64::midibase.

```
13.61.3.7 api_deinit_in() [1/2]
bool seq64::midibus::api_deinit_in ( ) [protected], [virtual]
```

We don't bother checking the m_rt_midi pointer. If it is null, it is the programmer's fault.

Returns

Returns the result of the m_rt_midi->api_deinit_in() call.

Reimplemented from seq64::midibase.

It fills a small byte buffer, sets the MIDI channel, make a message of it, and writes the message.

Again, DO WE NEED to distinguish between input and output here?

Note that we're doing a double-forwarding here, which may lower throughput.

e24	The MIDI event to play.
channel	The channel on which to play the event.

Implements seq64::midibase.

Reimplemented from seq64::midibase.

```
13.61.3.10 api_flush()
virtual void seq64::midibus::api_flush ( ) [protected], [virtual]
```

Reimplemented from seq64::midibase.

This function implements only the RtMidi-specific code.

Note that, unlike in PortMidi, here we do not deal with zeroing the event timestamp.

Parameters

tick	The tick to continue from; unused in the RtMidi API implementation.
beats	The calculated beats. This calculation is made in the midibase::continue_from() function.

Implements seq64::midibase.

```
13.61.3.12 api_start() [1/2]
void seq64::midibus::api_start ( ) [protected], [virtual]
```

This function is called by midibase::start(). No timestamp handling.

Implements seq64::midibase.

```
13.61.3.13 api_stop() [1/2]
void seq64::midibus::api_stop ( ) [protected], [virtual]
```

This function is called by midibase::stop(). No timestamp handling.

Implements seq64::midibase.

This function is called by midibase::clock(). No timestamp handling.

Parameters

tick | The clock tick value, not used in the API implementation of this function for RtMidi.

Implements seq64::midibase.

```
13.61.3.15 set_virtual_name()
```

13.61.3.16 api_connect()

```
bool seq64::midibus::api_connect ( ) [virtual]
```

If the port is an input port, but is not configured (by the user or the "rc" configuration file), then it is not connected, and this is not an error. Output ports are always connected.

Note that the api_connect() function will errprint is own errors. But if we expected to be able to connect, and have a null rt_midi pointer, then this is a reported error.

Returns

Returns true if the connection was made succuessfully, or there was no need to make the connection.

```
13.61.3.17 api_init_in() [2/2]
virtual bool seq64::midibus::api_init_in ( ) [protected], [virtual]
Implements seq64::midibase.
13.61.3.18 api_init_in_sub() [2/2]
virtual bool seq64::midibus::api_init_in_sub ( ) [protected], [virtual]
Reimplemented from seq64::midibase.
13.61.3.19 api_init_out() [2/2]
virtual bool seq64::midibus::api_init_out ( ) [protected], [virtual]
Implements seq64::midibase.
13.61.3.20 api_init_out_sub() [2/2]
virtual bool seq64::midibus::api_init_out_sub ( ) [protected], [virtual]
Reimplemented from seq64::midibase.
13.61.3.21 api_deinit_in() [2/2]
virtual bool seq64::midibus::api_deinit_in ( ) [protected], [virtual]
Reimplemented from seq64::midibase.
13.61.3.22 api_get_midi_event()
bool seq64::midibus::api_get_midi_event (
             event * inev ) [protected], [virtual]
Parameters
```

The location to deposit the MIDI event data.

Returns

Returns true if a MIDI event was obtained.

Reimplemented from seq64::midibase.

```
13.61.3.23 api_poll_for_midi()
int seq64::midibus::api_poll_for_midi ( ) [protected], [virtual]
```

This is the API implementation for RtMidi.

This should work only for input busses, so we need to insure this at some point. Currently, this is the domain of the master bus. We also should make this routine just check the input queue size and then read the queue. Note that the ALSA handle checks incoming MIDI events and either passes them to the callback function or pushes them onto the input queue.

Returns

Returns 0 if the polling succeeded, and 1 if it failed. If the buss hasn't been initialized, it has a null m_rt_midi pointer, and will return 0. This can happen normally when a MIDI input port is configured to be disabled.

Reimplemented from seq64::midibase.

```
13.61.3.25 api_start() [2/2]

virtual void seq64::midibus::api_start ( ) [protected], [virtual]

Implements seq64::midibase.

13.61.3.26 api_stop() [2/2]

virtual void seq64::midibus::api_stop ( ) [protected], [virtual]
```

Implements seq64::midibase.

13.61.4 Friends And Related Function Documentation

13.61.4.1 mastermidibus

```
mastermidibus [friend]
```

The master MIDI bus sets up the buss, so it gets access to private details.

13.61.5 Field Documentation

```
13.61.5.1 m_seq
snd_seq_t* const seq64::midibus::m_seq [private]

13.61.5.2 m_dest_addr_client
const int seq64::midibus::m_dest_addr_client [private]
```

Could potentially be replaced by midibase::m_bus_id.

```
13.61.5.3 m_dest_addr_port
const int seq64::midibus::m_dest_addr_port [private]
Could potentially be replaced by midibase::m_port_id.
13.61.5.4 m_local_addr_client
const int seq64::midibus::m_local_addr_client [private]
13.61.5.5 m_local_addr_port
int seq64::midibus::m_local_addr_port [private]
13.61.5.6 m_input_port_name
const std::string seq64::midibus::m_input_port_name [private]
It is derived from the (optionally configured) official client name for the application with the word "in" appended.
13.61.5.7 m rt midi
rtmidi* seq64::midibus::m_rt_midi [private]
13.61.5.8 m_master_info
rtmidi_info& seq64::midibus::m_master_info [private]
```

The rtmidi_info object used for enumerating the ports is a good place to get this handle. It is an extension of the legacy RtMidi interface.

13.62 seq64::midifile Class Reference

This class handles the parsing and writing of MIDI files.

Public Member Functions

 midifile (const std::string &name, int ppqn=SEQ64_USE_DEFAULT_PPQN, bool oldformat=false, bool globalbgs=true)

Principal constructor.

∼midifile ()

A rote destructor.

bool parse (perform &p, int a_screen_set=0)

This function opens a binary MIDI file and parses it into sequences and other application objects.

• bool write (perform &p)

Write the whole MIDI data and Seg24 information out to the file.

- bool write song (perform &p)
- · const std::string & error_message () const

'Getter' function for member m_error_message

• bool error_is_fatal () const

'Getter' function for member m_error_is_fatal

• int ppqn () const

'Getter' function for member m_ppqn Provides a way to get the actual value of PPQN used in processing the sequences when parse() was called.

Private Member Functions

• bool parse_smf_0 (perform &p, int screenset)

This function parses an SMF 0 binary MIDI file as if it were an SMF 1 file, then, if more than one MIDI channel was encountered in the sequence, splits all of the channels in the sequence out into separate sequences.

bool parse_smf_1 (perform &p, int screenset, bool is_smf0=false)

This function parses an SMF 1 binary MIDI file; it is basically the original seq24 midifile::parse() function.

midilong parse_prop_header (int file_size)

Parse the proprietary header, figuring out if it is the new format, or the legacy format, for sequencer-specific data.

• bool parse proprietary track (perform &a perf, int file size)

After all of the conventional MIDI tracks are read, we're now at the "proprietary" Seq24 data section, which describes the various features that Seq24 supports.

• int pow2 (int logbase2)

Internal function for simple calculation of a power of 2 without a lot of math.

• bool checklen (midilong len, midibyte type)

Internal function to check for and report a bad length value.

void add_trigger (sequence &seq, midishort ppqn)

Internal function to make the parser easier to read.

midilong read_long ()

Reads 4 bytes of data using read_byte().

midishort read_short ()

Reads 2 bytes of data using read_byte().

• midibyte read_byte ()

Reads 1 byte of data directly from the m_data vector, incrementing m_pos after doing so.

midilong read_varinum ()

Read a MIDI Variable-Length Value (VLV), which has a variable number of bytes.

void write long (midilong value)

Writes 4 bytes, each extracted from the long value and shifted rightward down to byte size, using the write_byte() function.

• void write_short (midishort value)

Writes 2 bytes, each extracted from the long value and shifted rightward down to byte size, using the write_byte() function.

void read_byte_array (midibyte *b, int len)

A helper function to simplify reading midi_control data from the MIDI file.

• void write byte (midibyte c)

Writes 1 byte.

· void write varinum (midilong)

Writes a MIDI Variable-Length Value (VLV), which has a variable number of bytes.

void write_track_name (const std::string &trackname)

Writes out a track name.

std::string read_track_name ()

Reads the track name.

void write seq number (midishort seqnum)

Writes out a sequence number.

• int read_seq_number ()

Reads the sequence number.

void write track end ()

Writes out the end-of-track marker.

· bool write header (int numtracks)

We want to write:

void write_prop_header (midilong tag, long len)

Writes a "proprietary" (SeqSpec) Seq24 footer header in either the new MIDI-compliant format, or the legacy Seq24 format.

bool write_proprietary_track (perform &a_perf)

Writes out the final proprietary/SeqSpec section, using the new format if the legacy format is not in force.

· long varinum_size (long len) const

Calculates the length of a variable length value.

• long prop_item_size (long datalen) const

Calculates the size of a proprietary item, as written by the write_prop_header() function, plus whatever is called to write the data.

• long track_name_size (const std::string &trackname) const

Calculates the size of a trackname and the meta event that specifies it.

void errdump (const std::string &msg)

Helper function to emit more useful error messages.

void errdump (const std::string &msg, unsigned long p)

Helper function to emit more useful error messages for erroneous long values.

- void write_track (const midi_vector &lst)
- long seq_number_size () const

Returns the size of a sequence-number event, which is always 5 bytes, plus one byte for the delta time that precedes it.

· long track end size () const

Returns the size of a track-end event, which is always 3 bytes.

bool is_sysex_special_id (midibyte ch)

Check for special SysEx ID byte.

Private Attributes

• mutex m_mutex

Provides locking for the sequence.

• int m_file_size

Holds the size of the MIDI file.

• std::string m_error_message

Holds the last error message, useful for trouble-shooting without having Sequencer64 running in a console window.

· bool m error is fatal

Indicates if the error should be considered fatal.

· bool m_disable_reported

Indicates that file reading has already been disabled (due to serious errors), so don't complain about it anymore.

• int m pos

Holds the position in the MIDI file.

· const std::string m_name

The unchanging name of the MIDI file.

std::vector< midibyte > m_data

This vector of characters holds our MIDI data.

• std::list< midibyte > m char list

Provides a list of characters.

bool m_new_format

Use the new format for the proprietary footer section of the Seq24 MIDI file.

• bool m_global_bgsequence

Indicates to store the new key, scale, and background sequence in the global, "proprietary" section of the MIDI song.

int m_ppqn

Provides the current value of the PPQN, which used to be constant and is now only the macro DEFAULT_PPQN.

• bool m_use_default_ppqn

Indicates that the default PPQN is in force.

• midi_splitter m_smf0_splitter

Provides support for SMF 0.

13.62.1 Detailed Description

In addition to the standard MIDI tracks, it also handles some "private" or "proprietary" tracks specific to Seq24. It does not, however, handle SYSEX events.

13.62.2 Constructor & Destructor Documentation

13.62.2.1 midifile()

name	Provides the name of the MIDI file to be read or written.
ppqn	Provides the initial value of the PPQN setting. It is handled differently for parsing (reading) versus writing the MIDI file.
	Reading.
	 If set to SEQ64_USE_DEFAULT_PPQN, the legacy application behavior is used. The m_ppqn member is set to the default PPQN, DEFAULT_PPQN. The value read from the MIDI file, ppqn, is then use to scale the running-time of the sequence relative to DEFAULT_PPQN.
	 Otherwise, m_ppqn is set to the value read from the MIDI file. No scaling is done. Since the value gets written, specify ppqn as 0, an obviously bogus value, to get this behavior.
	 Writing. This value is written to the MIDI file in the header chunk of the song. Note that the caller must query for the PPQN set during parsing, and pass it to the constructor when preparing to write the file. See how it is done in the mainwand class.
oldformat	If true, write out the MIDI file using the old Seq24 format, instead of the new MIDI-compliant sequencer-specific format, for the seq24-specific SeqSpec tags defined in the globals module. This option is false by default. Note that this option is only used in writing; reading can handle either format transparently.
globalbgs	If true, write any non-default values of the key, scale, and background sequence to the global "proprietary" section of the MIDI file, instead of to each sequence. Note that this option is only used in writing; reading can handle either format transparently.

13.62.2.2 ∼midifile()

```
seq64::midifile::\sim midifile ()
```

13.62.3 Member Function Documentation

13.62.3.1 parse()

In addition to the standard MIDI track data in a normal track, Seq24/Sequencer64 adds four sequencer-specific events just before the end of the track:

```
c_triggers_new: SeqSpec FF 7F 1C 24 24 00 08 00 00 ...
c_midibus: SeqSpec FF 7F 05 24 24 00 01 00
c_timesig: SeqSpec FF 7F 06 24 24 00 06 04 04
c_midich: SeqSpec FF 7F 05 24 24 00 02 06
```

Note that only Sequencer64 adds "FF 7F len" to the SeqSpec data.

Standard MIDI provides for port and channel specification meta events, but they are apparently considered obsolete:

```
Obsolete meta-event: Replacement:
MIDI port (buss): FF 21 01 po Device (port) name: FF 09 len text
MIDI channel: FF 20 01 ch
```

What do other applications use for specifying port/channel?

Note the is-modified flag: We now assume that the perform object is starting from scratch when parsing. But we let mainwnd tell the perform object when to clear everything with perform::clear_all(). The mainwnd does this for a new file, opening a file, but not for a file import, which might be done simply to add more MIDI tracks to the current composition. So, if parsing succeeds, all we want to do is make sure the flag is set. Parsing a file successfully is not always a modification of the setup. For instance, the first read of a MIDI file should start clean, not dirty.

SysEx notes:

Some files (e.g. Dixie04.mid) do not always encode System Exclusive messages properly for a MIDI file. Instead of a varinum length value, they are followed by extended IDs (0x7D, 0x7E, or 0x7F).

We've covered some of those cases by disabling access to m_data if the position passes the size of the file, but we want try to bypass these odd cases properly. So we look ahead for one of these special values.

Currently, Sequencer64, like Se24, handles SysEx message only to the extend of passing them via MIDI Thru. We hope to improve on that capability.

Parameters

р	Provides a reference to the perform object into which sequences/tracks are to be added.
screenset	The screen-set offset to be used when loading a sequence (track) from the file. This value ranges from -31 to 0 to +31 (32 is the maximum screen-set available in Seq24). This offset is added to the sequence number read in for the sequence, to place it elsewhere in the imported tune, and locate it in a specific screen-set. If this parameter is non-zero, then we will assume that the perform data is dirty.

Returns

Returns true if the parsing succeeded. Note that the error status is saved in m_error_is_fatal, and a message (to display later) is saved in m_error_message.

13.62.3.2 write()

```
bool seq64::midifile::write ( perform \& p )
```

Also see the write_song() function, for exporting to standard MIDI.

Seq24 reverses the order of some events, due to popping from its container. Not an issue here.

p | Provides the object that will contain and manage the entire performance.

Returns

Returns true if the write operations succeeded.

The PPQN will be either the global ppqn (legacy behavior) or the value read from the file, depending on the ppqn parameter passed to the midifile constructor.

int seq64::midifile::ppqn () const [inline]

The original sequence remains in place, in sequence slot 16 (the 17th slot). The user is responsible for deleting it if it is not needed.

р	Provides a reference to the perform object into which sequences/tracks are to be added.
screenset	The screen-set offset to be used when loading a sequence (track) from the file.

Returns

Returns true if the parsing succeeded.

13.62.3.8 parse_smf_1()

It assumes the file-data has already been read into memory. It also assumes that the ID, track-length, and format have already been read.

If the MIDI file contains both proprietary (c_timesig) and MIDI type 0x58 then it came from seq42 or seq32 (Stazed versions). In this case the MIDI type is parsed first (because it is listed first) then it gets overwritten by the proprietary, above.

Parameters

р	Provides a reference to the perform object into which sequences/tracks are to be added.
screenset	The screen-set offset to be used when loading a sequence (track) from the file.
is_smf0	True if we detected that the MIDI file is in SMF 0 format.

Returns

Returns true if the parsing succeeded.

13.62.3.9 parse_prop_header()

The new format creates a final track chunk, starting with "MTrk". Then comes the delta-time (here, 0), and the event. An event is a MIDI event, a SysEx event, or a Meta event.

A MIDI Sequencer Specific meta message includes either a delta time or absolute time, and the MIDI Sequencer Specific event encoded as follows:

```
0x00 0xFF 0x7F length data
```

For convenience, this function first checks the amount of file data left. If enough, then it reads a long value. If the value starts with 0×00 0xFF 0x7F, then that is a SeqSpec event, which signals usage of the new Sequencer64 "proprietary" format. Otherwise, it is probably the old format, and the long value is a control tag $(0 \times 242400 \text{nn})$, which can be returned immedidately.

If it is the new format, we back up to the FF, then get the next byte, which should be a 7F. If so, then we read the length (a variable length value) of the data, and then read the long value, which should be the control tag, which, again, is returned by this function.

Note

Most sequencers seem to be tolerant of both the lack of an "MTrk" marker and of the presence of an unwrapped control tag, and so can handle both the old and new formats of the final proprietary track.

Parameters

file_size	The size of the data file. This value is compared against the member m_pos (the position inside
	m_data[]), to make sure there is enough data left to process.

Returns

Returns the control-tag value found. These are the values, such as c_midich, found in the globals module, that indicate the type of sequencer-specific data that comes next. If there is not enough data to process, then 0 is returned.

13.62.3.10 parse_proprietary_track()

It consists of series of tags:

```
c_midictrl
        c_midiclocks
        c notes
        c_bpmtag (beats per minute)
        c mutearoups
        c_musickey (new, added if usr() global_seq_feature() is true)
        c_musicscale (ditto)
        c_backsequence (ditto)
(There are more tags defined in the globals module, but they are not
used in this function. This doesn't quite make sense, as there are
also some "triggers" values, and we're pretty sure the application
uses them. Oh, it turns out that they are set up by actions performed on
each sequence, and are stored as sequencer-specific ("SeqSpec") data with
each track's data as held in the MIDI container for the track. See the
midi_container module for more information.)
The format is (1) tag ID; (2) length of data; (3) the data.
```

First, we separate out this function for a little more clarity. Then we added code to handle reading both the legacy Seq24 format and the new, MIDI-compliant format. Note that even the new format is not quite correct, since it doesn't handle a MIDI manufacturer's ID, making it a single byte that is part of the data. But it does have the "MTrk" marker and track name, so that must be processed for the new format.

Now, in our "midicvt" project, we have a test MIDI file, b4uacuse-non-mtrk.midi that is good, except for having a tag "MUnk" instead of "MTrk". We should consider being more permissive, if possible. Otherwise, though, the only penality is that the "proprietary" chunk is completely skipped.

Extra precision BPM:

Based on a request for two decimals of precision in beats-per-minute, we now save a scaled version of BPM. Our supported range of BPM is SEQ64_MINIMUM_BPM = 1 to SEQ64_MAXIMUM_BPM = 600. If this range is encountered, the value is read as is. If greater than this range (actually, we use 999 as the limit), then we divide the number by 1000 to get the actual BPM, which can thus have more precision than the old integer value allowed. Obviously, when saving, we will multiply by 1000 to encode the BPM.

Parameters

p	The performance object that is being set via the incoming MIDI file.
file_size	The file size as determined in the parse() function.

There are also implicit parameters, with the m pos and m new format member variables.

13.62.3.11 pow2()

Use for calculating the denominator of a time signature.

Parameters

logbase2	Provides the power to which 2 is to be raised. This integer is probably only rarely greater than 4
	(which represents a denominator of 16).

Returns

Returns 2 raised to the logbase2 power.

13.62.3.12 checklen()

A length of zero is now considered legal, but a "warning" message is shown. The largest value allowed within a MIDI file is 0x0FFFFFF. This limit is set to allow variable-length quantities to be manipulated as 32-bit integers.

ſ	len	The length value to be checked, and it should be greater than 0. However, we have seen files with
		zero-length events, such as Lyric events (0x05).
ſ	type	The type of meta event. Used for displaying an error.

Returns

Returns true if the length parameter is valid. This now means it is simply less than 0x0FFFFFFF.

13.62.3.13 add_trigger()

Handles only c_triggers_new values, not the old c_triggers value. If m_ppqn isn't set to the default value, then we must scale these triggers accordingly, just as is done for the MIDI events.

Parameters

seq	Provides the sequence to which the trigger is to be added.
ppqn	Provides the ppqn value to use to scale the tick values if m_use_default_ppqn is true. If 0, the ppqn value is not used.

13.62.3.14 read_long()

```
midilong seq64::midifile::read_long ( ) [private]
```

Warning

This code looks endian-dependent and integer-size dependent.

Returns

Returns the four bytes, shifted appropriately and added together, most-significant byte first, to sum to a long value.

13.62.3.15 read_short()

```
midishort seq64::midifile::read_short ( ) [private]
```

Returns

Returns the two bytes, shifted appropriately and added together, most-significant byte first, to sum to a short value.

13.62.3.16 read_byte()

```
midibyte seq64::midifile::read_byte ( ) [private]
```

Returns

Returns the byte that was read. Returns 0 if there was an error, though there's no way for the caller to determine if this is an error or a good value.

13.62.3.17 read_varinum()

```
midilong seq64::midifile::read_varinum ( ) [private]
```

This function reads the bytes while bit 7 is set in each byte. Bit 7 is a continuation bit. See write_varinum() for more information.

Returns

Returns the accumulated values as a single number.

13.62.3.18 write_long()

Warning

This code looks endian-dependent.

Parameters

x The long value to be written to the MIDI file.

13.62.3.19 write_short()

Warning

This code looks endian-dependent.

x The short value to be written to the MIDI file.

13.62.3.20 read_byte_array()

Parameters

b	The byte array to receive the data.
len	The number of bytes in the array, and to be read.

13.62.3.21 write_byte()

The byte is written to the m_char_list member, using a call to push_back().

Parameters

```
c The MIDI byte to be "written".
```

13.62.3.22 write_varinum()

A MIDI file Variable Length Value is stored in bytes. Each byte has two parts: 7 bits of data and 1 continuation bit. The highest-order bit is set to 1 if there is another byte of the number to follow. The highest-order bit is set to 0 if this byte is the last byte in the VLV.

To recreate a number represented by a VLV, first you remove the continuation bit and then concatenate the leftover bits into a single number.

To generate a VLV from a given number, break the number up into 7 bit units and then apply the correct continuation bit to each byte.

In theory, you could have a very long VLV number which was quite large; however, in the standard MIDI file specification, the maximum length of a VLV value is 5 bytes, and the number it represents can not be larger than 4 bytes.

Here are some common cases:

```
    Numbers between 0 and 127 (0x7F) are represented by a single byte.
    0x80 is represented as "0x81 0x00".
    0x0FFFFFFFF (the largest number) is represented as "0xFF 0xFF 0xFF".
```

Also see the varinum_size() function.

Parameters

value The long value to be encoded as a MIDI varinum, and written to the MIDI file.

13.62.3.23 write_track_name()

Note that we have to precede this "event" with a delta time value, set to 0. The format of the output is "0x00 0xFF 0x03 len track-name-bytes".

Parameters

trackname Provides the name of the track to be written to the MIDI file.

13.62.3.24 read_track_name()

```
std::string seq64::midifile::read_track_name ( ) [private]
```

Meant only for usage in the proprietary/SeqSpec footer track, in the new file format.

Returns

Returns the track name, or an empty string if there was a problem.

13.62.3.25 write_seq_number()

The format is "00 FF 00 02 ss ss", where "02" is actually the constant length of the data. We have to precede these values with a 0 delta time, of course.

Now, for sequence 0, an alternate format is "FF 00 00". But that format can only occur in the first track, and the rest of the tracks then don't need a sequence number, since it is assumed to increment. Our application doesn't bother with that shortcut.

seqnum The s	equence number to write.
--------------	--------------------------

13.62.3.26 read_seq_number()

```
int seq64::midifile::read_seq_number ( ) [private]
```

Meant only for usage in the proprietary/SeqSpec footer track, in the new file format.

Returns

Returns the sequence number found, or -1 if it was not found.

13.62.3.27 write_track_end()

```
void seq64::midifile::write_track_end ( ) [private]
```

13.62.3.28 write_header()

- 0x4D54726B. The track tag "MTrk". The MIDI spec requires that software can skip over non-standard chunks. "Prop"? Would require a fix to midicvt.
- 0xaabbccdd. The length of the track. This needs to be calculated somehow.
- 0x00. A zero delta time.
- 0x7f7f. Sequence number, a special value, well out of normal range.
- · The name of the track:
 - "Seq24-Spec"
 - "Sequencer64-S"

Then follows the proprietary/SeqSpec data, written in the normal manner. Finally, tack on the track-end meta-event.

Components of final track size:

```
-# Delta time. 1 byte, always 0x00.

-# Sequence number. 5 bytes. OPTIONAL. We won't write it.

-# Track name. 3 + 10 or 3 + 15

-# Series of proprietary/SeqSpec specs:

-# Prop header:

-# If legacy format, 4 bytes.

-# Otherwise, 2 bytes + varinum_size(length) + 4 bytes.

-# Length of the prop data.

-# Track End. 3 bytes.
```

13.62.3.29 write_prop_header()

This function does not write the data. It replaces calls such as "write_long(c_midich)" in the proprietary secton of write().

The legacy format just writes the control tag (0x242400xx). The new format writes 0x00 0xFF 0x7F len 0x242400xx; the first 0x00 is the delta time.

In the new format, the 0x24 is a kind of "manufacturer ID". At http://www.midi.org/techspecs/manid. \leftarrow php we see that most manufacturer IDs start with 0x00, and are thus three bytes long, or start with codes at 0x40 and above. Similary, this site shows that no manufacturer uses 0x24:

http://sequence15.blogspot.com/2008/12/midi-manufacturer-ids.html

Warning

Currently, the manufacturer ID is not handled; it is part of the data, which can be misleading in programs that analyze MIDI files.

Parameters

control_tag	Determines the type of sequencer-specific section to be written. It should be one of the value in	
	the globals module, such as c_midibus or c_mutegroups.	
data_length	The amount of data that will be written. This parameter does not count the length of the header	
	itself.	

13.62.3.30 write_proprietary_track()

The first thing to do, for the new format only, is calculate the length of this big section of data. This was quite tricky; we tweaked and adjusted until the midicvt program handled the whole new-format file without emitting any errors.

Here's the basics of what Seq24 did for writing the data in this part of the file:

```
-# Write the c_midictrl value, then write a 0. To us, this looks like
no one wrote any code to write this data. And yet, the parsing
code can handles a non-zero value, which is the number of sequences
as a long value, not a byte. So shouldn't we write 4 bytes, not
one? Yes, indeed, we made a mistake. However, we should be
writing out the full data set as well. But not even Seq24 does
that! Perhaps they decided it was best kept in the "rc"
configuration file.
-# MORE TO COME.
```

p | Provides the object that will contain and manage the entire performance.

Returns

Always returns true. No efficient way to check all of the writes that can happen. Might revisit this issue if some bug crops up.

13.62.3.31 varinum_size()

This function is needed when calculating the length of a track. Note that it handles only the following situations:

```
https://en.wikipedia.org/wiki/Variable-length_quantity
```

This restriction allows the calculation to be simple and fast.

```
1 byte: 0x00 to 0x7F
2 bytes: 0x80 to 0x3FFF
3 bytes: 0x4000 to 0x001FFFFF
4 bytes: 0x200000 to 0x0FFFFFFF
```

Parameters

len The long value whose length, when encoded as a MIDI varinum, is to be found.

Returns

Returns values as noted above. Anything beyond that range returns 0.

13.62.3.32 prop_item_size()

If using the new format, the length includes the sum of sequencer-specific tag (0xFF 0x7F) and the size of the variable-length value. Then, for legacy and new format, 4 bytes are added for the Seq24 MIDI control value, and then the data length is added.

	data_length	Provides the data length value to be encoded.
--	-------------	---

Returns

Returns the length of the item size, including the delta time, meta bytes, length byes, the control tag, and the data-length itself.

13.62.3.33 track_name_size()

Parameters

Returns

Returns the length of the event, which is of the format "0x00 0xFF 0x03 len track-name-bytes".

13.62.3.34 errdump() [1/2]

It adds the file offset to the message.

Parameters

msg The main error message string, without an ending newline character.

Returns

The constructed string is returned as a side-effect, in case we want to pass it along to the externally-accessible error-message buffer.

13.62.3.35 errdump() [2/2]

It adds the file offset to the message.

Parameters

msg	The main error message string, without an ending newline character.
value	The long value to show as part of the message.

Returns

The constructed string is returned as a side-effect, in case we want to pass it along to the externally-accessible error-message buffer.

13.62.3.36 write_track()

13.62.3.37 seq_number_size()

```
long seq64::midifile::seq_number_size ( ) const [inline], [private]
```

13.62.3.38 track_end_size()

```
long seq64::midifile::track_end_size ( ) const [inline], [private]
```

13.62.3.39 is_sysex_special_id()

Parameters

ch Provides the byte to be checked against 0x7D through 0x7F.

Returns

Returns true if the byte is SysEx special ID.

13.62.4 Field Documentation

```
13.62.4.1 m_mutex
```

```
mutex seq64::midifile::m_mutex [mutable], [private]
```

Made mutable for use in certain locked getter functions.

```
13.62.4.2 m_file_size
```

```
int seq64::midifile::m_file_size [private]
```

This variable was added when loading a file that caused an attempt to load data well beyond the file-size of the midicvt test file Dixie04.mid.

```
13.62.4.3 m_error_message
```

```
std::string seq64::midifile::m_error_message [private]
```

If empty, there's no pending error. Currently most useful in the parse() function.

```
13.62.4.4 m_error_is_fatal
```

```
bool seq64::midifile::m_error_is_fatal [private]
```

The caller can query for this value after getting the return value from parse().

```
13.62.4.5 m_disable_reported
```

```
bool seq64::midifile::m_disable_reported [private]
```

Once is enough.

```
13.62.4.6 m_pos
```

```
int seq64::midifile::m_pos [private]
```

This is at least a 31-bit value in the recent architectures running Linux and Windows, so it will handle up to 2 Gb of data. This member is used as the offset into the m_data vector.

```
13.62.4.7 m_name
```

```
const std::string seq64::midifile::m_name [private]
```

```
13.62.4.8 m_data
```

```
std::vector<midibyte> seq64::midifile::m_data [private]
```

We could also use a string of characters, unsigned. This member is resized to the putative size of the MIDI file, in the parse() function. Then the whole file is read into it, as if it were an array. This member is an input buffer.

```
13.62.4.9 m_char_list
```

```
std::list<midibyte> seq64::midifile::m_char_list [private]
```

The class pushes each MIDI byte into this list using the write_byte() function. Also note that the write() function calls sequence::fill_list() to fill a temporary std::list<char> (!) buffer, then writes that data backwards to this member. This member is an output buffer.

```
13.62.4.10 m_new_format
```

```
bool seq64::midifile::m_new_format [private]
```

In the new format, each sequencer-specific value (0x242400xx, as defined in the globals module) is preceded by the sequencer-specific prefix, 0xFF 0x7F len id/date). By default, the new format is used, but the user can specify the —legacy (-I) option, or make a soft link to the sequence24 binary called "seq24", to write the data in the old format. [We will eventually add the —legacy option to the "rc" configuration file.] Note that reading can handle either format transparently.

```
13.62.4.11 m_global_bgsequence
```

```
bool seq64::midifile::m_global_bgsequence [private]
```

```
13.62.4.12 m_ppqn
```

```
int seq64::midifile::m_ppqn [private]
```

```
13.62.4.13 m_use_default_ppqn
```

```
bool seq64::midifile::m_use_default_ppqn [private]
```

13.62.4.14 m_smf0_splitter

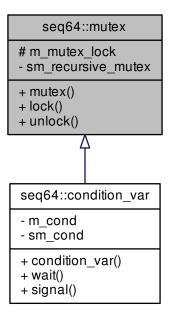
```
midi_splitter seq64::midifile::m_smf0_splitter [private]
```

This object holds all of the information needed to split a multi-channel sequence.

13.63 seq64::mutex Class Reference

The mutex class provides a simple wrapper for the pthread_mutex_t type used as a recursive mutex.

Inheritance diagram for seq64::mutex:



Public Member Functions

• mutex ()

The constructor assigns the recursive mutex to the local locking mutex.

· void lock () const

Lock the mutex.

· void unlock () const

Unlock the mutex.

Protected Attributes

pthread_mutex_t m_mutex_lock

Provides a mutex lock usable by a single module or class.

Static Private Attributes

• static const pthread_mutex_t sm_recursive_mutex

Provides a recursive mutex that can be used by the whole application, and is, apparently.

13.63.1 Constructor & Destructor Documentation

```
13.63.1.1 mutex()
seq64::mutex::mutex ( )
```

13.63.2 Member Function Documentation

```
13.63.2.1 lock()

void seq64::mutex::lock ( ) const

13.63.2.2 unlock()
```

void seq64::mutex::unlock () const

13.63.3 Field Documentation

```
13.63.3.1 sm_recursive_mutex
const pthread_mutex_t seq64::mutex::sm_recursive_mutex [static], [private]
```

Define the static recursive mutex and its condition variable.

```
13.63.3.2 m_mutex_lock

pthread_mutex_t seq64::mutex::m_mutex_lock [mutable], [protected]
```

However, this mutex ends up being a copy of the static sm_recursive_mutex (and, of course, a different "object").

13.64 seq64::editable_event::name_value_t Struct Reference

Provides a type that contains the pair of values needed for the various lookup maps that are needed to manage editable events.

Data Fields

• unsigned short event_value

Holds a midibyte value (0x00 to 0xFF) or SEQ64_END_OF_MIDIBYTE_TABLE to indicate the end of an array of name_value_t items.

· std::string event name

Holds the human-readable name for an event code or other numeric value in an array of name_value_t items.

13.64.1 Field Documentation

13.64.1.1 event_value

```
unsigned short seq64::editable_event::name_value_t::event_value
```

13.64.1.2 event_name

```
std::string seq64::editable_event::name_value_t::event_name
```

13.65 seq64::options Class Reference

This class supports a full tabbed options dialog.

Inherits Dialog.

Public Member Functions

• options (Gtk::Window &parent, perform &p, bool showjack=false)

Private Types

```
    enum button {
        e_jack_transport,
        e_jack_master,
        e_jack_master_cond,
        e_jack_midi,
        e_jack_start_mode_live,
        e_jack_start_mode_song,
        e_jack_connect,
        e_jack_disconnect }
```

Defines buttons indices or IDs for some controls related to JACK.

Private Member Functions

• perform & perf ()

'Getter' function for member m_mainperf

- void clock_callback_off (int bus, Gtk::RadioButton *button)
- void clock callback on (int bus, Gtk::RadioButton *button)
- void clock callback mod (int bus, Gtk::RadioButton *button)
- void clock_mod_callback (Gtk::Adjustment *adj)
- void input_callback (int bus, Gtk::Button *button)
- void filter callback (Gtk::Button *button)
- void transport_callback (button type, Gtk::Button *button)
- void mouse_seq24_callback (Gtk::RadioButton *)
- void mouse_fruity_callback (Gtk::RadioButton *)
- void mouse_mod4_callback (Gtk::CheckButton *)
- void mouse snap split callback (Gtk::CheckButton *)
- void mouse click edit callback (Gtk::CheckButton *)
- void lash_support_callback (Gtk::CheckButton *)
- void add_midi_clock_page ()
- void add_midi_input_page ()
- void add_keyboard_page ()
- · void add extended keys page ()
- void add_mouse_page ()
- void add_jack_sync_page ()

Private Attributes

• Gtk::Tooltips * m_tooltips

A repository for GTK tooltip support.

perform & m_mainperf

The performance object to which some of these options apply.

• Gtk::Button * m button ok

The famous "OK" button's pointer.

• Gtk::CheckButton * m_button_jack_transport

Main JACK transport selection.

• Gtk::CheckButton * m_button_jack_master

Main JACK transport master selection.

Gtk::CheckButton * m_button_jack_master_cond

Main JACK transport master-conditional selection.

• Gtk::Button * m_button_jack_connect

JACK Connect button, which we need to enable/disable for clarity and some additional safety.

• Gtk::Button * m button jack disconnect

JACK Disonnect button, which we need to enable/disable for clarity and some additional safety.

Gtk::Notebook * m notebook

Not sure yet what this notebook is for.

13.65.1 Member Enumeration Documentation

13.65.1.1 button

```
enum seq64::options::button [private]
```

These values are handled in options::transport_callback(). Some of them set JACK-related values in the rc_settings object, while the others set up or tear down the JACK support of sequencer64.

The JACK Transport settings are a little messy. They should be radio buttons, and control each other's settings. Currently, if the user wants to set up for JACK Master, the JACK Transport button must also be checked.

Enumerator

e_jack_transport	Turns on the "with JACK Transport" option, rc_settings :: with_jack_transport().
e_jack_master	Turns on the "with JACK Master" option, rc_settings :: with_jack_master(). If another application is already JACK Master, this will fail.
e_jack_master_cond	Turns on the "with JACK Master" option rc_settings :: with_jack_master_cond(). This option makes sequencer64 the JACK Master conditionally, that is, if no other application has claimed that role.
e_jack_midi	Turns on the "Native JACK MIDI" option rc_settings :: with_jack_midi(). This is a setting independent of the JACK Transport settings. This is use only in the "rtmidi" implementation os Sequencer64, seq64.
e_jack_start_mode_live	Doesn't directly do anything; the live mode versus song mode is set by the e_jack_start_mode_song value.
e_jack_start_mode_song	Sets the "JACK start mode" value to true, which means that sequencer64 is in song mode. This value is obtained via rc_settings :: song_start_mode(). It will eventually be the start mode that applies to either ALSA or JACK playback.
e_jack_connect	Causes the perform object's JACK initialization function, perform::init_jack(), to be called.
e_jack_disconnect	Causes the perform object's JACK deinitialization function, perform::deinit_jack(), to be called.

13.65.2 Constructor & Destructor Documentation

13.65.2.1 options()

```
seq64::options::options (
          Gtk::Window & parent,
          perform & p,
          bool showjack = false )
```

13.65.3 Member Function Documentation

```
13.65.3.1 perf()
```

```
perform& seq64::options::perf ( ) [inline], [private]
```

13.65.3.2 clock_callback_off() void seq64::options::clock_callback_off (int bus, Gtk::RadioButton * button) [private] 13.65.3.3 clock_callback_on() void seq64::options::clock_callback_on (int bus, Gtk::RadioButton * button) [private] 13.65.3.4 clock_callback_mod() void seq64::options::clock_callback_mod (int bus, Gtk::RadioButton * button) [private] 13.65.3.5 clock_mod_callback() void seq64::options::clock_mod_callback (Gtk::Adjustment * adj) [private] 13.65.3.6 input_callback() void $seq64::options::input_callback$ (int bus, Gtk::Button * button) [private] 13.65.3.7 filter_callback()

Generated by Doxygen

void seq64::options::filter_callback (

Gtk::Button * button) [private]

```
13.65.3.8 transport_callback()
```

```
void seq64::options::transport_callback (
            button type,
             Gtk::Button * button ) [private]
13.65.3.9 mouse_seq24_callback()
void seq64::options::mouse_seq24_callback (
            Gtk::RadioButton * ) [private]
13.65.3.10 mouse_fruity_callback()
void seq64::options::mouse_fruity_callback (
             Gtk::RadioButton * ) [private]
13.65.3.11 mouse_mod4_callback()
void seq64::options::mouse_mod4_callback (
            Gtk::CheckButton * ) [private]
13.65.3.12 mouse_snap_split_callback()
void seq64::options::mouse_snap_split_callback (
            Gtk::CheckButton * ) [private]
13.65.3.13 mouse_click_edit_callback()
void seq64::options::mouse_click_edit_callback (
             Gtk::CheckButton * ) [private]
13.65.3.14 lash_support_callback()
void seq64::options::lash\_support\_callback (
             Gtk::CheckButton * ) [private]
```

```
13.65.3.15 add_midi_clock_page()
void seq64::options::add_midi_clock_page ( ) [private]
13.65.3.16 add_midi_input_page()
void seq64::options::add_midi_input_page ( ) [private]
13.65.3.17 add_keyboard_page()
void seq64::options::add_keyboard_page ( ) [private]
13.65.3.18 add_extended_keys_page()
void seq64::options::add_extended_keys_page ( ) [private]
13.65.3.19 add_mouse_page()
void seq64::options::add_mouse_page ( ) [private]
13.65.3.20 add_jack_sync_page()
void seq64::options::add_jack_sync_page ( ) [private]
13.65.4 Field Documentation
13.65.4.1 m_tooltips
Gtk::Tooltips* seq64::options::m_tooltips [private]
```

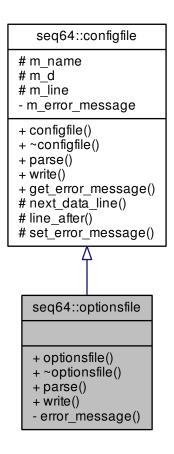
Must be a GTK thang.

```
13.65.4.2 m_mainperf
perform& seq64::options::m_mainperf [private]
13.65.4.3 m_button_ok
Gtk::Button* seq64::options::m_button_ok [private]
13.65.4.4 m_button_jack_transport
Gtk::CheckButton* seq64::options::m_button_jack_transport [private]
13.65.4.5 m_button_jack_master
Gtk::CheckButton* seq64::options::m_button_jack_master [private]
13.65.4.6 m_button_jack_master_cond
Gtk::CheckButton* seq64::options::m_button_jack_master_cond [private]
13.65.4.7 m_button_jack_connect
Gtk::Button* seq64::options::m_button_jack_connect [private]
13.65.4.8 m_button_jack_disconnect
Gtk::Button* seq64::options::m_button_jack_disconnect [private]
13.65.4.9 m_notebook
Gtk::Notebook* seq64::options::m_notebook [private]
```

13.66 seq64::optionsfile Class Reference

Provides a file for reading and writing the application' main configuration file.

Inheritance diagram for seq64::optionsfile:



Public Member Functions

• optionsfile (const std::string &name)

Principal constructor.

∼optionsfile ()

A rote destructor.

• bool parse (perform &perf)

Parse the \sim /.seq24rc or \sim /.config/sequencer64/sequencer64.rc file.

• bool write (const perform &perf)

This options-writing function is just about as complex as the options-reading function.

Private Member Functions

• bool error_message (const std::string §ionname, const std::string &additional="") Helper function for error-handling.

Additional Inherited Members

13.66.1 Detailed Description

The settings that are passed around are provided or used by the perform class.

13.66.2 Constructor & Destructor Documentation

13.66.2.1 optionsfile()

Parameters

name | Provides the name of the options file; this is usually a full path file-specification.

13.66.2.2 \sim optionsfile()

```
seq64::optionsfile::~optionsfile ( )
```

13.66.3 Member Function Documentation

13.66.3.1 parse()

[midi-control]

Get the number of sequence definitions provided in the [midi-control] section. Ranges from 32 on up. Then read in all of the sequence lines. The first 32 apply to the first screen set. There can also be a comment line "# mute in group" followed by 32 more lines. Then there are addditional comments and single lines for BPM up, BPM down, Screen Set Up, Screen Set Down, Mod Replace, Mod Snapshot, Mod Queue, Mod Gmute, Mod Glearn, and Screen Set Play. These are all forms of MIDI automation useful to control the playback while not sitting near the computer.

[mute-group]

The mute-group starts with a line that indicates up to 32 mute-groups are defined. A common value is 1024, which means there are 32 groups times 32 keys. But this value is currently thrown away. This value is followed by 32

lines of data, each contained 4 sets of 8 settings. See the seq24-doc project on GitHub for a much more detailed description of this section.

[midi-clock]

The MIDI-clock section defines the clocking value for up to 16 output busses. The first number, 16, indicates how many busses are specified. Generally, these busses are shown to the user with names such as "[1] seq24 1".

[keyboard-control]

The keyboard control defines the keys that will toggle the stage of each of up to 32 patterns in a pattern/sequence box. These keys are displayed in each box as a reminder. The first number specifies the Key number, and the second number specifies the Sequence number.

[keyboard-group]

The keyboard group specifies more automation for the application. The first number specifies the Key number, and the second number specifies the Group number. This section should be better described in the seq24-doc project on GitHub.

[extended-keys]

Additional keys (not yet represented in the Options dialog) to support additional keys for tempo-tapping, Seq32's new transport and connection functionality, and maybe a little more.

[New-keys]

Conditional support for reading Seq32 "rc" files.

[jack-transport]

This section covers various JACK settings, one setting per line. In order, the following numbers are specfied:

```
- jack_transport - Enable sync with JACK Transport.
- jack_master - Seq24 will attempt to serve as JACK Master.
- jack_master_cond - Seq24 will fail to be Master if there is already a Master set.
- song_start_mode:
- 0 = Playback will be in Live mode. Use this to allow muting and unmuting of loops.
- 1 = Playback will use the Song Editor's data.
```

[midi-input]

This section covers the MIDI input busses, and has a format similar to "[midi-clock]". Generally, these busses are shown to the user with names such as "[1] seq24 1", and currently there is only one input buss. The first field is the port number, and the second number indicates whether it is disabled (0), or enabled (1).

[midi-clock-mod-ticks]

This section covers.... One common value is 64.

[manual-alsa-ports]

Set to 1 if you want seq24 to create its own ALSA ports and not connect to other clients.

[last-used-dir]

This section simply holds the last path-name that was used to read or write a MIDI file. We still need to add a check for a valid path, and currently the path must start with a "/", so it is not suitable for Windows.

[interaction-method]

This section specified the kind of mouse interaction.

- 0 = 'seq24' (original Seq24 method).
- 1 = 'fruity' (similar to a certain fruity sequencer we like).

The second data line is set to "1" if Mod4 can be used to keep seq24 in note-adding mode even after the right-click is released, and "0" otherwise.

p Provides the performance object to which all of these options apply.

Returns

Returns true if the file was able to be opened for reading. Currently, there is no indication if the parsing actually succeeded.

Implements seq64::configfile.

13.66.3.2 write()

```
bool seq64::optionsfile::write ( {\tt const\ perform\ \&\ p\ )} \quad [{\tt virtual}]
```

Parameters

p Provides a const reference to the main perform object. However, we have to cast away the constness, because too many of the perform getter functions are used in non-const contexts.

Returns

Returns true if the write operations all succeeded.

New boolean to show sequence numbers; ignored in legacy mode.

Implements seq64::configfile.

13.66.3.3 error_message()

Parameters

sectionname	Provides the name of the section for reporting the error.
additional	Additional context information to help in finding the error.

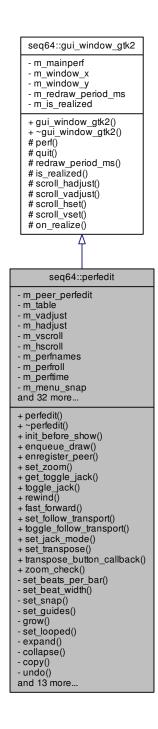
Returns

Always returns false.

13.67 seq64::perfedit Class Reference

This class supports a Performance Editor that is used to arrange the patterns/sequences defined in the patterns panel.

Inheritance diagram for seg64::perfedit:



Public Member Functions

• perfedit (perform &p, bool second_perfedit=false, int ppqn=SEQ64_USE_DEFAULT_PPQN)

Principal constructor, has a reference to a perform object.

virtual ~perfedit ()

This rote destructor does nothing.

void init_before_show ()

This function forwards its call to the perfroll function of the same name.

void enqueue draw (bool forward=true)

Helper wrapper for calling perfroll::queue_draw() for one or both perfedits.

void enregister peer (perfedit *peer)

Register the peer perfedit object.

void set_zoom (int z)

Implements the horizontal zoom feature.

bool get_toggle_jack ()

Gets the state fo the JACK toggle button in the Song editor, when compiled with seq32 JACK support.

void toggle_jack ()

Sets the state fo the JACK toggle button in the Song editor, when compiled with seq32 JACK support.

• void rewind (bool press)

Implements the seq32/stazed rewind operation.

void fast_forward (bool press)

Implements the seq32/stazed fast-forward operation.

void set_follow_transport ()

Sets the transport status when compiled for seq32 JACK support.

void toggle follow transport ()

Toggles the transport status when compiled for seq32 JACK support.

void set_jack_mode ()

Sets the JACK transport status, based on the status of the JACK button in the Song editor, when compiled for seq32 JACK support.

• void set_transpose (int transpose)

Sets the value of transposition for this window.

void transpose_button_callback (int transpose)

The button callback for transposition for this window.

Static Public Member Functions

static bool zoom_check (int z)

Checks zoom values for the z/Z keystrokes used in perfroll and perftime.

Private Member Functions

void set beats per bar (int bpm)

Sets the beats-per-measure text and value to the given value, and then calls set_guides().

void set_beat_width (int bw)

Sets the BW (beat width, or the denominator in the time signature) text and values to the given value, and then calls set_guides().

void set_snap (int snap)

Sets the snap text and values to the given value, and then calls set_guides().

void set_guides ()

Sets the guides, which are the L and R user-interface elements.

• void grow ()

Increments the size of the perfroll and perftime objects.

void set_looped ()

Set the looping in the perform object.

• void expand ()

Implement the expand action.

void collapse ()

Implement the collapse action.

• void copy ()

Implement the copy (actually, expand-and-copy) action.

• void undo ()

Implement the undo feature (Ctrl-Z).

• void redo ()

Implement the redo feature (Ctrl-R).

void popup_menu (Gtk::Menu *menu)

Opens the given popup menu.

void draw_sequences ()

Forces a redraw of the sequences, though currently just the perfnames part of each sequence in the performance editor.

· bool timeout ()

Handles a drawing timeout.

void set_image (bool isrunning)

Changes the image used for the pause/play button.

void start_playing ()

Implement the playing.

void pause_playing ()

Pauses the playing of the song, leaving the progress bar where it stopped.

void stop_playing ()

Stop the playing.

void toggle_playing ()

Reverses the state of playback.

• void on_realize ()

This callback function calls the base-class on_realize() function, and then connects the perfedit::timeout() function to the Glib signal-timeout, with a redraw timeout of redraw period ms().

bool on_key_press_event (GdkEventKey *ev)

This function is the callback for a key-press event.

• bool on_key_release_event (GdkEventKey *ev)

This function is the callback for a key-release event.

bool on_delete_event (GdkEventAny *)

All this callback function does is return false.

Private Attributes

• perfedit * m_peer_perfedit

The partner instance of perfedit.

• Gtk::Table * m table

A whole horde of GUI elements.

• Gtk::Adjustment * m_vadjust

Vertical adjust for piano roll.

Gtk::Adjustment * m_hadjust

Horizontal adjust for piano roll.

• Gtk::VScrollbar * m_vscroll

Vertical scroll for piano roll.

• Gtk::HScrollbar * m_hscroll Horizonatl scroll for piano roll. • perfnames * m_perfnames Pattern names in leftmost column. • perfroll * m_perfroll The piano roll in the song editor. • perftime * m perftime The time/measures bar above roll. • Gtk::Menu * m menu snap The menu for grid-snap selection. • Gtk::Menu * m menu xpose The menu for transpose selection. Gtk::Button * m_button_xpose Button to bring up transpose menu. • Gtk::Entry * m_entry_xpose Text edit for the transpose value. Gtk::Image * m_image_play The image for the play button. • Gtk::Button * m_button_snap Button to bring up the snap menu. • Gtk::Entry * m_entry_snap Text edit for the grid-snap value. • Gtk::Button * m_button_stop The Stop Play button object. • Gtk::Button * m_button_play Implements the yellow two-bar pause button. Gtk::ToggleButton * m_button_loop Button for Left-to-Right looping. • Gtk::Button * m_button_expand Button for Left/Right expansion. • Gtk::Button * m_button_collapse Button for Left/Right collapse. • Gtk::Button * m_button_copy Expand and copy between L/R. • Gtk::Button * m button grow Expand grid (bottom-right button). • Gtk::Button * m button undo Button to undo previous action. Gtk::Button * m_button_redo Button to redo previous action. Gtk::ToggleButton * m button jack Button to toggle JACK connection. • Gtk::ToggleButton * m_button_follow Button to toggle JACK following. • Gtk::Button * m button bpm Beats-per-measure menu button. • Gtk::Entry * m_entry_bpm

Text-edit for beat-width.

Gtk::HBox * m hbox

Horizontal box (which?) in table.

Gtk::HBox * m hlbox

Horizontal box for buttons at top.

Gtk::Tooltips * m tooltips

Container for tool-tips.

• Gtk::Menu * m_menu_bpm

Menus for time signature, beats per measure, beat width.

• Gtk::Menu * m menu bw

Drop-down menu for beat-width.

• int m_snap

Sets the horizontal grid snap-to in units of "pulses" or "ticks".

• int m bpm

The current "beats per measure" value.

• int m bw

The current "beat width" value.

• int m_ppqn

The current "parts per quarter note" value.

• bool m_is_running

Holds the current status of running, for use in display the play versus pause icon.

• int m standard bpm

The standard "beats per measure" of Sequencer64, which here matches the beats-per-measure displayed in the perfroll (piano roll).

Friends

· void update perfedit sequences ()

This global function in the seq64 namespace calls perfedit :: draw_sequences(), if the global perfedit objects exist.

Additional Inherited Members

13.67.1 Detailed Description

It has a segroll and piano roll? No, it has a perform, a perfnames, a perfroll, and a perftime.

13.67.2 Constructor & Destructor Documentation

13.67.2.1 perfedit()

We've reordered the pointer members and put them in the initializer list to make the constructor a bit cleaner.

Todo Offload most of the work into an initialization function like options does.

р	Refers to the main performance object.
second_perfedit	If true, this object is the second perfedit object.
ppqn	The optionally-changed PPQN value to use for the performance editor.

```
13.67.2.2 ~perfedit()
virtual seq64::perfedit::~perfedit ( ) [inline], [virtual]
```

We're going to have to run the application through valgrind to make sure that nothing is left behind.

13.67.3 Member Function Documentation

```
13.67.3.1 init_before_show()
void seq64::perfedit::init_before_show ( )
```

It does not seem to need to also forward to the perftime function of the same name.

Note that we call the children's queue_draw() functions, not enqueue_draw(), otherwise we'll get stack overflow.

Parameters

forward If true (the default), pass the call to the peer. When passing this call to the peer, this parameter is set to false to prevent an infinite loop and the resultant stack overflow.

13.67.3.3 zoom_check()

It has to range from greater than 1 (the highest zoom-in causes an unexplained drawing artifact at this time), and not greater than four times the c_perf_scale_x value, at which point we have zoomed out so far that the measure numbers are almost completely obscured.

z The desired zoom value to validate.

13.67.3.4 enregister_peer()

This function is meant to be called by mainwnd, which creates the perfedits and then makes sure they get along. Only the first call to this function will work; only one peer can be registered.

Parameters

peer The peer perfedit object to register, if not null.

13.67.3.5 set_zoom()

Parameters

z The zoom value to be set. The child zoom functions called each check that this value is valid.

13.67.3.6 get_toggle_jack()

```
bool seq64::perfedit::get_toggle_jack ( )
```

Returns

Returns the JACK button's get_active() status.

13.67.3.7 toggle_jack()

```
void seq64::perfedit::toggle_jack ( )
```

Note that this will trigger the button signal callback.

13.67.3.8 rewind()

The timeout is in milliseconds, and is currently hard-wired to 120.

Note the use of "&perf()" to get the address of the perform object.

Parameters

press True if the operation is a key press, false if the operation is a key release.

13.67.3.9 fast_forward()

Parameters

press True if the operation is a key press, false if the operation is a key release.

13.67.3.10 set_follow_transport()

```
void seq64::perfedit::set_follow_transport ( )
```

Note that this will trigger the button signal callback.

13.67.3.11 toggle_follow_transport()

```
void seq64::perfedit::toggle_follow_transport ( )
```

Note that this will trigger the button signal callback.

13.67.3.12 set_jack_mode()

```
void seq64::perfedit::set_jack_mode ( )
```

To avoid a lot of pointer dereferencing, much of the code is offloaded to perform::set_jack_mode(), which now returns a boolean.

13.67.3.13 set_transpose()

transpose	The amount to transpose the transposable sequences. We need to add validation at some point, if
	the widget does not enforce that.

13.67.3.14 transpose_button_callback()

Parameters

transpose The amount to transpose the transposable sequences.

13.67.3.15 set_beats_per_bar()

The usage of is modified was faulty. Offloaded it to the perform object to make it more foolproof. See the perform ::modify() function.

Parameters

bpm Provides the beats/measure or beats/bar value to be set. This value is basically the numerator of the time signature.

13.67.3.16 set_beat_width()

The usage of is modified was faulty. Offloaded it to the perform object to make it more foolproof. See the perform ::modify() function.

Parameters

bw Provides the beat width to be set. The beat width is basically the denominator of the time signature.

13.67.3.17 set_snap()

Parameters

snap Provide the snap value to be set. This value is basically the numerator of the expression "1 / snap".

13.67.3.18 set_guides()

```
void seq64::perfedit::set_guides ( ) [private]
```

See the set_snap() function.

It's a little confusing; I assigned the label "m_standard_bpm" to the value 4 in "measure_pulse = $192 * 4 * m_bpm / m_bw$ ", but I am not sure I understand this equation... why the extra factor of 4? That 4 appears in "c_ppqn * 4" a lot in the original code.

13.67.3.19 grow()

```
void seq64::perfedit::grow ( ) [private]
```

Make sure that setting the modified flag makes sense for this operation. It doesn't seem to modify members.

13.67.3.20 set_looped()

```
void seq64::perfedit::set_looped ( ) [private]
```

13.67.3.21 expand()

```
void seq64::perfedit::expand ( ) [private]
```

This action opens up a space of events between the L and R (left and right) markers. This action is preceded by pushing an Undo operation in the perform object, moving its triggers, and telling the perfroll to redraw.

13.67.3.22 collapse()

```
void seq64::perfedit::collapse ( ) [private]
```

This action removes all events between the L and R (left and right) markers. This action is preceded by pushing an Undo operation in the perform object, not moving its triggers (they go away), and telling the perfroll to redraw.

```
13.67.3.23 copy()
void seq64::perfedit::copy ( ) [private]
```

This action opens up a space of events between the L and R (left and right) markers, and copies the information from the same amount of events that follow the R marker. This action is preceded by pushing an Undo operation in the perform object, copying its triggers, and telling the perfoll to redraw.

```
13.67.3.24 undo()
void seq64::perfedit::undo ( ) [private]
```

We pop an Undo trigger, and then ask the perfroll to queue up a (re)drawing action.

```
13.67.3.25 redo()
void seq64::perfedit::redo ( ) [private]
```

We pop an Redo trigger, and then ask the perfroll to queue up a (re)drawing action.

```
13.67.3.26 popup_menu()
```

```
13.67.3.27 draw_sequences()
```

```
void seq64::perfedit::draw_sequences ( ) [private]
```

This is meant to be called when the focus of an open seqedit or eventedit window changes.

```
13.67.3.28 timeout()
```

```
bool seq64::perfedit::timeout ( ) [private]
```

It redraws "dirty" sequences in the perfroll and the perfnames objects, and shows draw progress on the perfroll. It also changes the pause/play image if the status of running has changed. This function is called frequently and continuously. It will work for both perfedit windows, if both are up.

```
13.67.3.29 set_image()
```

isrunning If true, the image should be the pause image. Otherwise, it should be the play image.

13.67.3.30 start_playing()

```
void seq64::perfedit::start_playing ( ) [private]
```

JACK will be used if it is present and, in the application, enabled and working. Note the new flag to let perform know that it is a pause/play request from the perfedit window. In other words, a forced Song mode.

13.67.3.31 pause_playing()

```
void seq64::perfedit::pause_playing ( ) [private]
```

Keeps the stop button enabled as a kind of rewind for ALSA. Stop in place!

13.67.3.32 stop_playing()

```
void seq64::perfedit::stop_playing ( ) [private]
```

We need to make the progress line move back to the beginning right away here.

13.67.3.33 toggle_playing()

```
void seq64::perfedit::toggle_playing ( ) [inline], [private]
```

Meant only to be called when the "Play" button is pressed. Currently, the GUI does not change. This function will ultimately act like a Pause/Play button, but currently the pause functionality on works (partially) for JACK transport. Currently not used.

13.67.3.34 on_realize()

```
void seq64::perfedit::on_realize ( ) [private]
```

13.67.3.35 on_key_press_event()

By default, the space-bar starts the playing, and the Escape key stops the playing. The start/end key may be the same key (i.e. space-bar), allow toggling when the same key is mapped to both triggers. Note that we now pass false in the call to perform::playback_key_event(), if SEQ64_PAUSE_SUPPORT is compiled in. Song mode doesn't yield the pause effect we want.

ev Provides the key event to implement.

13.67.3.36 on_key_release_event()

It is needed to turn off the fast-forward and rewind keys functionality when released.

Parameters

ev Provides the key event to implement.

13.67.3.37 on_delete_event()

13.67.4 Friends And Related Function Documentation

13.67.4.1 update_perfedit_sequences

```
void update_perfedit_sequences ( ) [friend]
```

It is used by other objects (seqedit and eventedit) that can modify the currently-edited sequence shown in the perfedit (song window).

13.67.5 Field Documentation

13.67.5.1 m_peer_perfedit

```
perfedit* seq64::perfedit::m_peer_perfedit [private]
```

```
13.67.5.2 m_table
Gtk::Table* seq64::perfedit::m_table [private]
Layout table for song editor.
13.67.5.3 m_vadjust
Gtk::Adjustment* seq64::perfedit::m_vadjust [private]
13.67.5.4 m_hadjust
Gtk::Adjustment* seq64::perfedit::m_hadjust [private]
13.67.5.5 m_vscroll
Gtk::VScrollbar* seq64::perfedit::m_vscroll [private]
13.67.5.6 m_hscroll
Gtk::HScrollbar* seq64::perfedit::m_hscroll [private]
13.67.5.7 m_perfnames
perfnames* seq64::perfedit::m_perfnames [private]
13.67.5.8 m_perfroll
perfroll* seq64::perfedit::m_perfroll [private]
13.67.5.9 m_perftime
perftime* seq64::perfedit::m_perftime [private]
```

```
13.67.5.10 m_menu_snap
Gtk::Menu* seq64::perfedit::m_menu_snap [private]
13.67.5.11 m_menu_xpose
Gtk::Menu* seq64::perfedit::m_menu_xpose [private]
13.67.5.12 m_button_xpose
Gtk::Button* seq64::perfedit::m_button_xpose [private]
13.67.5.13 m_entry_xpose
Gtk::Entry* seq64::perfedit::m_entry_xpose [private]
13.67.5.14 m_image_play
Gtk::Image* seq64::perfedit::m_image_play [private]
13.67.5.15 m button snap
Gtk::Button* seq64::perfedit::m_button_snap [private]
13.67.5.16 m_entry_snap
Gtk::Entry* seq64::perfedit::m_entry_snap [private]
13.67.5.17 m_button_stop
Gtk::Button* seq64::perfedit::m_button_stop [private]
```

```
13.67.5.18 m_button_play
Gtk::Button* seq64::perfedit::m_button_play [private]
The Play button object.
13.67.5.19 m_button_loop
Gtk::ToggleButton* seq64::perfedit::m_button_loop [private]
13.67.5.20 m_button_expand
Gtk::Button* seq64::perfedit::m_button_expand [private]
13.67.5.21 m_button_collapse
Gtk::Button* seq64::perfedit::m_button_collapse [private]
13.67.5.22 m_button_copy
Gtk::Button* seq64::perfedit::m_button_copy [private]
13.67.5.23 m_button_grow
Gtk::Button* seq64::perfedit::m_button_grow [private]
13.67.5.24 m_button_undo
Gtk::Button* seq64::perfedit::m_button_undo [private]
13.67.5.25 m_button_redo
Gtk::Button* seq64::perfedit::m_button_redo [private]
```

```
13.67.5.26 m_button_jack
Gtk::ToggleButton* seq64::perfedit::m_button_jack [private]
13.67.5.27 m_button_follow
Gtk::ToggleButton* seq64::perfedit::m_button_follow [private]
13.67.5.28 m_button_bpm
Gtk::Button* seq64::perfedit::m_button_bpm [private]
13.67.5.29 m_entry_bpm
Gtk::Entry* seq64::perfedit::m_entry_bpm [private]
13.67.5.30 m_button_bw
Gtk::Button* seq64::perfedit::m_button_bw [private]
13.67.5.31 m_entry_bw
Gtk::Entry* seq64::perfedit::m_entry_bw [private]
13.67.5.32 m_hbox
Gtk::HBox* seq64::perfedit::m_hbox [private]
13.67.5.33 m_hlbox
Gtk::HBox* seq64::perfedit::m_hlbox [private]
```

```
13.67.5.34 m_tooltips
Gtk::Tooltips* seq64::perfedit::m_tooltips [private]
13.67.5.35 m_menu_bpm
Gtk::Menu* seq64::perfedit::m_menu_bpm [private]
Drop-down menu for beats/minute.
13.67.5.36 m_menu_bw
Gtk::Menu* seq64::perfedit::m_menu_bw [private]
13.67.5.37 m_snap
int seq64::perfedit::m_snap [private]
13.67.5.38 m_bpm
int seq64::perfedit::m_bpm [private]
Do not confuse it with BPM (beats per minute). The numerator of the time signature.
13.67.5.39 m_bw
int seq64::perfedit::m_bw [private]
The denominator of the time signature.
13.67.5.40 m_ppqn
int seq64::perfedit::m_ppqn [private]
13.67.5.41 m_is_running
```

bool seq64::perfedit::m_is_running [private]

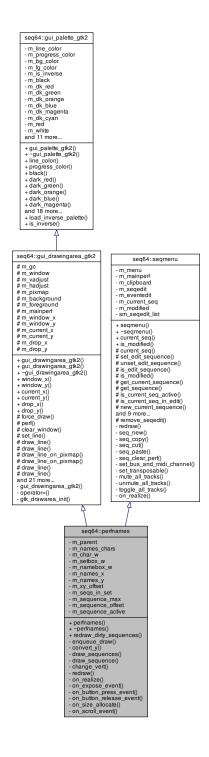
13.67.5.42 m_standard_bpm

```
int seq64::perfedit::m_standard_bpm [private]
```

13.68 seq64::perfnames Class Reference

This class implements the left-side keyboard in the patterns window.

Inheritance diagram for seq64::perfnames:



Public Member Functions

• perfnames (perform &p, perfedit &parent, Gtk::Adjustment &vadjust)

Principal constructor for this user-interface object.

virtual ∼perfnames ()

Let's provide a do-nothing virtual destructor.

• void redraw_dirty_sequences ()

Redraws sequences that have been modified.

Private Member Functions

void enqueue_draw ()

Wraps queue_draw() and forwards the call to the parent perfedit, so that it can forward it to any other perfedit that exists, and to the other sub-elements of the song editor.

• int convert_y (int y)

Converts a y-value into a sequence number and returns it.

void draw sequences ()

New function to encapsulate forced redrawing of all sequence names in the current viewport.

void draw_sequence (int sequence)

Draw the given sequence.

void change_vert ()

Change the vertial offset of a sequence/pattern.

· void redraw (int sequence)

Redraw the given sequence.

void on_realize ()

Handles the callback when the window is realized.

• bool on_expose_event (GdkEventExpose *ev)

Handles an on-expose event.

• bool on button press event (GdkEventButton *ev)

Provides the callback for a button press, and it handles only a left mouse button [the right mouse button is handled in on_button_release_event()].

bool on_button_release_event (GdkEventButton *ev)

Handles a button-release for the right button, bringing up a popup menu that is identical to the right-click popup menu for a slot in the patterns panel (mainwid), and context sensitive.

void on_size_allocate (Gtk::Allocation &)

Handles a size-allocation event.

• bool on_scroll_event (GdkEventScroll *ev)

Handle the vertical scrolling of the window.

Private Attributes

· perfedit & m_parent

Provides a link to the perfedit that created this object.

· int m names chars

Provides the number of the characters in the name box.

int m_char_w

Provides the "real" width of a character.

· int m setbox w

Provides the width of the "set number" box.

int m_namebox_w

Provides the width of the "name" box.

• int m_names_x

Provides the width of the names box, which is the width of a character for 24 characters.

· int m_names_y

Provides the height of the names box, which is hardwired to 24 pixels.

• int m_xy_offset

Provides the horizontal and vertical offsets of the text relative to the names box.

const int m_seqs_in_set

The number of sequences in a set, currently still hardwired to 32.

• const int m_sequence_max

The maximum number of sequences, current $32 \times 32 = 1024$.

• int m_sequence_offset

The offset from the 0th sequence, which is determined by the vertical view of the piano roll, controlled by the vertical scroll-bar.

• bool m sequence active [c max sequence]

Indicates if the given sequence is active or not.

Friends

· class perfedit

Additional Inherited Members

13.68.1 Detailed Description

It inherits from gui_drawingarea_gtk2 to support the font, color, and other GUI functionality, and from seqmenu to support the right-click Edit/New/Cut right-click menu.

Obsolete Note the usage of virtual base classes. Since these can add some extra overhead, we should determine if we can do without the virtuality (and indeed it doesn't seem to be needed).

13.68.2 Constructor & Destructor Documentation

13.68.2.1 perfnames()

Weird is that the window (x,y) are set to (c_names_x, 100), when c_names_y is 22 (now 24) in globals.h.

Parameters

р	Provides a reference to the main performance object of the application.
parent	Provides a reference to the object that contains this object, so that this object can tell the parent to
	queue up a drawing operation.
Gevaration Perfective the vertical scrollbar object needed so that perfnames can respond to scrollbar	
	cursor/thumb movement.

```
13.68.2.2 \sim perfnames()
```

virtual seq64::perfnames::~perfnames () [inline], [virtual]

13.68.3 Member Function Documentation

```
13.68.3.1 redraw_dirty_sequences()
void seq64::perfnames::redraw_dirty_sequences ( )
13.68.3.2 enqueue_draw()
```

void seq64::perfnames::enqueue_draw () [private]

The parent perfedit will call perfnames::queue_draw() on behalf of this object, and it will pass a perfnames...:enqueue draw() to the peer perfedit's perfnames, if the peer exists.

```
13.68.3.3 convert_y()
```

Used in figuring out which sequence to mute/unmute in the performance editor.

Parameters

y The y value (within the vertical limits of the perfnames column to the left of the performance editor's piano roll.

Returns

Returns the sequence number corresponding to the y value.

13.68.3.4 draw_sequences()

```
void seq64::perfnames::draw_sequences ( ) [private]
```

13.68.3.5 draw_sequence()

This function has to be prepared to handle an almost endless list of sequences, including unused ones, to draw them all with compatible styles. The sequences are grouped by set-number. The set-number occurs every 32 sequences in the leftmost column of the window.

- 1. Render the set number, or a blank box, in leftmost column. If the y height of the first draw_rectangle is m_names_y + 1, then we get a black line for the blank tracks, looks ugly.
- 2. Make sure that the rectangle drawn with the proper background colors for various combinations of muting and highlighting, otherwise just the name is properly colored.
- 3. Render the column with the name of the sequence. The channel number ranges from 1 to 16, but SMF 0 is indicated on-screen by a channel number of 0. We get the label format from the perform object, for consistency across windows.

Parameters

seqnum Index to the sequence information to be drawn.

13.68.3.6 change_vert()

```
void seq64::perfnames::change_vert ( ) [private]
```

13.68.3.7 redraw()

This function is a virtual function of seqmenu that must be overridden in this class.

Parameters

sequence Provides the number of the sequence to be redrawn.

Implements seq64::segmenu.

13.68.3.8 on_realize()

```
void seq64::perfnames::on_realize ( ) [private]
```

It first calls the base-class version of on_realize(). Then it allocates any additional resources needed.

13.68.3.9 on_expose_event()

```
bool seq64::perfnames::on\_expose\_event (
            GdkEventExpose * ev ) [private]
```

It draws all of the sequences that will be visible.

We could actually optimize this a tiny bit, to save some additions in the for loop.

Parameters

```
ev
     The expose event, not used.
```

Returns

Always returns true.

13.68.3.10 on_button_press_event()

```
bool seq64::perfnames::on_button_press_event (
            GdkEventButton * ev ) [private]
```

Two operations are supported by left-clicking on the sequence/track name:

- Normal. Toggles the mute status of the sequence that is clicked. Shift. Toggles the mutes status of all other sequences, making this operation an easy way to preview a single sequence in the performance editor, then bring back the rest of the tracks.

Parameters

```
The mouse button event.
```

Returns

Always returns true.

13.68.3.11 on_button_release_event()

```
bool seq64::perfnames::on\_button\_release\_event (
             GdkEventButton * p0 ) [private]
```

Parameters

The button event.

Returns

Always returns false.

13.68.3.12 on_size_allocate()

It first calls the base-class version of this function.

Parameters

The allocation event. It is passed to the base-class on_size_allocate() function, and then m_window_x and m_window_y are set to the width and height, respectively, of the allocation.

13.68.3.13 on_scroll_event()

The vertical value is incremented or decremented by the amount of the step increment, and the page is clamped to the new value.

Parameters

ev The scrolling event.

Returns

Always returns true.

13.68.4 Friends And Related Function Documentation

13.68.4.1 perfedit

friend class perfedit [friend]

13.68.5 Field Documentation

```
13.68.5.1 m_parent
```

```
perfedit& seq64::perfnames::m_parent [private]
```

We want to support two perfedit windows, but the children of perfedit will have to communicate changes requiring a redraw through the parent.

```
13.68.5.2 m_names_chars
```

```
int seq64::perfnames::m_names_chars [private]
```

Pretty much hardwired to 24 at present.

```
13.68.5.3 m_char_w
```

```
int seq64::perfnames::m_char_w [private]
```

This value is obtained from a font-renderer accessor function.

```
13.68.5.4 m_setbox_w
```

```
int seq64::perfnames::m_setbox_w [private]
```

This used to be hardwired to 6 * 2 (character-width times two).

```
13.68.5.5 m_namebox_w
```

```
int seq64::perfnames::m_namebox_w [private]
```

This used to be a weird calculation based on character width.

```
13.68.5.6 m_names_x
```

```
int seq64::perfnames::m_names_x [private]
```

```
13.68.5.7 m_names_y
```

```
int seq64::perfnames::m_names_y [private]
```

This value was once 22 pixels, but we need a little extra room for our new font. This extra room is compatible enough with the old font, as well.

```
13.68.5.8 m_xy_offset
int seq64::perfnames::m_xy_offset [private]

Currently hardwired.

13.68.5.9 m_seqs_in_set

const int seq64::perfnames::m_seqs_in_set [private]

13.68.5.10 m_sequence_max

const int seq64::perfnames::m_sequence_max [private]

13.68.5.11 m_sequence_offset
int seq64::perfnames::m_sequence_offset [private]

13.68.5.12 m_sequence_active

bool seq64::perfnames::m_sequence_active[c_max_sequence] [private]
```

If this really is the true meaning of this value, we ought to get it directly from the sequence if we can.

13.69 seq64::perform Class Reference

This class supports the performance mode.

Public Types

```
    enum mute_op_t {
        MUTE_TOGGLE,
        MUTE_OFF,
        MUTE_ON }
        Provides settings for muting.
    enum ff_rw_button_t {
        FF_RW_REWIND,
        FF_RW_NONE,
        FF_RW_FORWARD }
```

Provides setting for the fast-forward and rewind functionality.

Public Member Functions

perform (gui_assistant &mygui, int ppqn=SEQ64_USE_DEFAULT_PPQN)

This construction initializes a vast number of member variables, some of them public (but we're working on that)!

∼perform ()

The destructor sets some running flags to false, signals this condition, then joins the input and output threads if the were launched.

• bool is_modified () const

'Getter' function for member m_is_modfied

· void modify ()

'Setter' function for member m_is_modified This setter only sets the modified-flag to true.

• int ppqn () const

'Getter' function for member m_ppqn

• int sequence_count () const

'Getter' function for member m_sequence_count It is better to call this getter before bothering to even try to use a sequence.

• int sequence_max () const

'Getter' function for member m_sequence_max

bool is_control_status () const

'Getter' function for member m_control_status

void set_edit_sequence (int seqnum)

'Setter' function for member m_edit_sequence

· void unset_edit_sequence (int seqnum)

'Setter' function for member m_edit_sequence

bool is edit sequence (int seqnum) const

'Getter' function for member m_edit_sequence

• int get_beats_per_bar () const

'Getter' function for member m_beats_per_bar

void set_beats_per_bar (midibpm bpm)

'Setter' function for member m_beats_per_bar

• int get_beat_width () const

'Getter' function for member m_beat_width

void set_beat_width (int bw)

 ${\it 'Setter' function for member m_beat_width}$

void clocks_per_metronome (int cpm)

'Setter' function for member m_clocks_per_metronome

• int clocks_per_metronome () const

'Getter' function for member m_clocks_per_metronome

void set_32nds_per_quarter (int tpq)

'Setter' function for member m_32nds_per_quarter

• int get_32nds_per_quarter () const

'Getter' function for member m_32nds_per_quarter

void us_per_quarter_note (long upqn)

'Setter' function for member m_us_per_quarter_note

· long us per quarter note () const

'Getter' function for member m_us_per_quarter_note

const gui_assistant & gui () const

'Getter' function for member m_gui_support The const getter.

• gui_assistant & gui ()

'Getter' function for member $m_gui_support$ The un-const getter.

• const keys_perform & keys () const

'Getter' function for member m_gui_support.keys() The const getter.

keys_perform & keys ()

'Getter' function for member m_gui_support.keys() The un-const getter.

· mastermidibus & master bus ()

'Getter' function for member m_master_bus Obviously, this is a dangerous function, but we've got ya covered.

void filter_by_channel (bool flag)

'Setter' function for member m_master_bus.filter_by_channel()

• bool is_running () const

'Getter' function for member m_running Could also be called "is_playing()".

• bool is_pattern_playing () const

'Setter' function for member m_is_pattern_playing

bool toggle_song_start_mode ()

'Setter' function for member m_song_start_mode

void song start mode (bool flag)

'Setter' function for member m_song_start_mode

bool song_start_mode () const

'Getter' function for member m_song_start_mode

· bool is jack running () const

'Getter' function for member m_jack_asst.is_running() This function is useful for announcing the status of JACK in user-interface items that only have access to the perform object.

· bool is_jack_master () const

'Getter' function for member m_jack_asst.is_master() Also now includes is_jack_running(), since one cannot be JACK Master if JACK is not running.

void enregister (performcallback *pfcb)

Adds a pointer to an object to be notified by this perform object.

- void toggle_jack_mode ()
- bool set_jack_mode (bool mode)

Encapsulates behavior needed by perfedit.

bool get_toggle_jack () const

'Getter' function for member m_jack_asst.get_jack_mode()

void set_jack_stop_tick (midipulse tick)

'Setter' function for member m_jack_asst.set_jack_stop_tick()

unsigned short combine_bytes (midibyte b0, midibyte b1)

Combines bytes into an unsigned-short value.

• void FF_rewind ()

Implements the fast-forward or rewind functionality imported from seq32.

bool FF RW timeout ()

Convenience function.

void start_from_perfedit (bool flag)

'Setter' function for member m_start_from_perfedit

· bool start_from_perfedit () const

'Getter' function for member m_start_from_perfedit

void set_follow_transport (bool flag)

'Getter' function for member m_jack_asst.set_follow_transport()

bool get follow transport () const

'Getter' function for member m_jack_asst.get_follow_transport()

void toggle_follow_transport ()

'Setter' function for member m_jack_asst.toggle_follow_transport()

void set_reposition (bool postype=true)

'Getter' function for member m_reposition

• ff_rw_button_t ff_rw_type ()

'Getter' function for member m_FF_RW_button_type

void ff_rw_type (ff_rw_button_t button_type)

'Getter' function for member m_FF_RW_button_type

· void rewind (bool press)

Sets the rewind status.

· void fast forward (bool press)

Sets the fast-forward status.

void reposition (midipulse tick)

Encapsulates some repositioning code needed to move the position to the mouse pointer location in perfroll.

· bool clear_all ()

Clears all of the patterns/sequences.

void launch (int ppqn)

Calls the MIDI buss and JACK initialization functions and the input/output thread-launching functions.

void new_sequence (int seq)

Creates a new pattern/sequence for the given slot, and sets the new pattern's master MIDI bus address.

void add sequence (sequence *seq, int perf)

Adds a pattern/sequence pointer to the list of patterns.

• void delete_sequence (int seq)

Deletes a pattern/sequence by number.

bool is sequence in edit (int seq)

Check if the pattern/sequence, given by number, has an edit in progress.

void clear_sequence_triggers (int seq)

Clears the patterns/sequence for the given sequence, if it is active.

void print_triggers () const

Shows all the triggers of all the sequences.

• void finish ()

The rough opposite of launch(); it doesn't stop the threads.

• midipulse get_tick () const

'Getter' function for member m_tick

· void set tick (midipulse tick)

'Setter' function for member m_tick

midipulse get_jack_tick () const

'Getter' function for member m_jack_tick

void set_jack_tick (midipulse tick)

'Setter' function for member m jack tick

void set_left_tick (midipulse tick, bool setstart=true)

Set the left marker at the given tick.

• midipulse get_left_tick () const

'Getter' function for member m_left_tick

void set_start_tick (midipulse tick)

'Setter' function for member m_starting_tick

• midipulse get_start_tick () const

'Setter' function for member m_starting_tick

void set_right_tick (midipulse tick, bool setstart=true)

Set the right marker at the given tick.

· midipulse get_right_tick () const

'Getter' function for member m_right_tick

• double left_right_size () const

Convenience function for JACK support when loop in song mode.

bool is_active (int seq) const

Checks the pattern/sequence for activity.

void apply_song_transpose ()

Calls the apply_song_transpose() function for all active sequences.

void set transpose (int t)

'Setter' function for member m_transpose For sanity's sake, the values are restricted to +-64.

int get_transpose () const

'Getter' function for member m_transpose

midibpm get_beats_per_minute ()

'Getter' function for member m_master_bus.get_beats_per_minute Retrieves the BPM setting of the master MIDI buss.

void set_sequence_control_status (int status)

If the given status is present in the c_status_snapshot, the playing state is saved.

void unset sequence control status (int status)

If the given status is present in the c_status_snapshot, the playing state is restored.

void sequence_playing_toggle (int seq)

If the given sequence is active, then it is toggled as per the current value of m_control_status.

void sequence playing change (int seq, bool on)

Turn the playing of a sequence on or off, if it is active.

void sequence_playing_on (int seq)

Calls sequence_playing_change() with a value of true.

void sequence_playing_off (int seq)

Calls sequence_playing_change() with a value of false.

void mute_all_tracks (bool flag=true)

Mutes/unmutes all tracks in the current set of active patterns/sequences.

void toggle_all_tracks ()

Toggles the mutes status of all tracks in the current set of active patterns/sequences.

• bool armed_saved () const

'Getter' function for member m_armed_saved

- void toggle_playing_tracks ()
- void mute_screenset (int ss, bool flag=true)

Mutes/unmutes all tracks in the desired screen-set.

void output_func ()

Performance output function.

void input_func ()

This function is called by input_thread_func().

void set_group_mute_state (int gtrack, bool muted)

This function sets the mute state of an element in the m_mute_group array.

bool get_group_mute_state (int gtrack)

The opposite of set_group_mute_state(), it gets the value of the desired track.

void set_offset (int offset)

Calculates the offset into the screen sets.

• int get offset () const

'Getter' function for member m_offset

void save_playing_state ()

For all active patterns/sequences, this function gets the playing status and saves it in m_sequence_state[i].

void restore playing state ()

For all active patterns/sequences, this function gets the playing status from m_sequence_state[i] and sets it for the sequence.

std::string key_name (unsigned int k) const

Here follows a few forwarding functions for the keys_perform-derived classes.

keys_perform::SlotMap & get_key_events ()

Forwarding function for key events.

• keys_perform::SlotMap & get_key_groups ()

Forwarding function for key groups.

keys_perform::RevSlotMap & get_key_events_rev ()

Forwarding function for reverse key events.

keys_perform::RevSlotMap & get_key_groups_rev ()

Forwarding function for reverse key groups.

bool show_ui_sequence_key () const

'Getter' function for member m_show_ui_sequency_key Provides access to keys().show_ui_sequence_key().

void show_ui_sequence_key (bool flag)

'Setter' function for member m_show_ui_sequency_key

bool show_ui_sequence_number () const

'Getter' function for member m_show_ui_sequency_number Provides access to keys().show_ui_sequence_number().

void show_ui_sequence_number (bool flag)

'Getter' function for member m_show_ui_sequency_number

unsigned int lookup_keyevent_key (int seqnum)

Gets the event key for the given sequence.

long lookup_keyevent_seq (unsigned int keycode)

Gets the sequence number for the given event key.

unsigned int lookup_keygroup_key (long groupnum)

Gets the group key for the given sequence.

long lookup keygroup group (unsigned int keycode)

Gets the group number for the given group key.

void start_playing (bool songmode=false)

Encapsulates a series of calls used in mainwnd.

void pause_playing (bool songmode=false)

Pause playback, so that progress bars stay where they are, and playback always resumes where it left off, at least in ALSA mode, which doesn't have to worry about being a "slave".

void stop_playing ()

Encapsulates a series of calls used in mainwnd.

void start_key (bool songmode=false)

Invoke the start key functionality.

void pause_key (bool songmode=false)

Invoke the pause key functionality.

void stop_key ()

Invoke the stop key functionality.

• void learn_toggle ()

Encapsulates some calls used in mainwnd.

• midibpm decrement_beats_per_minute ()

Encapsulates some calls used in mainwnd.

• midibpm increment_beats_per_minute ()

Encapsulates some calls used in mainwnd.

midibpm page_decrement_beats_per_minute ()

Provides additional coarse control over the BPM value, which comes into force when the Page-Up/Page-Down keys are pressed.

· midibpm page increment beats per minute ()

Provides additional coarse control over the BPM value, which comes into force when the Page-Up/Page-Down keys are pressed.

int decrement_screenset ()

Encapsulates some calls used in mainwnd.

• int increment screenset ()

Encapsulates some calls used in mainwnd.

bool highlight (const sequence &seq) const

True if a sequence is empty and should be highlighted.

bool is_smf_0 (const sequence &seq) const

True if the sequence is an SMF 0 sequence.

const sequence * get_sequence (int seq) const

Retrieves the actual sequence, based on the pattern/sequence number.

sequence * get sequence (int seq)

Retrieves the actual sequence, based on the pattern/sequence number.

void sequence_key (int seq)

Handle a sequence key to toggle the playing of an active pattern in the selected screen-set.

std::string sequence label (const sequence &seq)

Provides a way to format the sequence parameters string for display in the mainwid or perfnames modules.

void set_input_bus (int bus, bool input_active)

Sets the input bus, and handles the special "key labels on sequence" and "sequence numbers on sequence" functionality.

void set_clock_bus (int bus, clock_e clocktype)

Sets the clock value, as specified in the Options / MIDI Clocks tab.

bool mainwnd key event (const keystroke &k)

Provided for mainwnd :: on_key_press_event() and mainwnd :: on_key_release_event() to call.

bool perfroll_key_event (const keystroke &k, int drop_sequence)

Provided for perfroll :: on key press event() and perfroll :: on key release event() to call.

bool playback key event (const keystroke &k, bool songmode=false)

New function provided to unify the stop/start (space/escape) behavior of the various windows where playback can be started, paused, or stopped.

void move_triggers (bool direction)

If the left tick is less than the right tick, then, for each sequence that is active, its triggers are moved by the difference between the right and left in the specified direction.

• void copy triggers ()

If the left tick is less than the right tick, then, for each sequence that is active, its triggers are copied, offset by the difference between the right and left.

void push_trigger_undo (int track=SEQ64_ALL_TRACKS)

For every active sequence, call that sequence's push_trigger_undo() function.

void pop_trigger_undo ()

For every active sequence, call that sequence's pop_trigger_undo() function.

• void pop trigger redo ()

For every active sequence, call that sequence's pop_trigger_redo() function.

bool is_dirty_main (int seq)

Checks the pattern/sequence for main-dirtiness.

bool is_dirty_edit (int seq)

Checks the pattern/sequence for edit-dirtiness.

• bool is_dirty_perf (int seq)

Checks the pattern/sequence for perf-dirtiness.

bool is_dirty_names (int seq)

Checks the pattern/sequence for names-dirtiness.

• bool is_exportable (int seq) const

Indicates that the desired sequence is active, unmuted, and has a non-zero trigger count.

· void set screenset (int ss)

Sets the m_screenset value (the index or ID of the current screen set).

• int get_screenset () const

'Getter' function for member m_screenset

int get_playing_screenset () const

'Getter' function for member m_playing_screen

• bool toggle_other_seqs (int seqnum, bool isshiftkey)

This code handles the use of the Shift key to toggle the mute state of all other sequences.

bool toggle other names (int segnum, bool isshiftkey)

This code handles the use of the Shift key to toggle the mute state of all other sequences.

Private Member Functions

· bool have_undo () const

'Getter' function for member m_have_undo

void set_have_undo (bool undo)

'Setter' function for member m_have_undo Note that, if the undo parameter is true, then we mark the performance as modified.

· bool have redo () const

'Getter' function for member m_have_redo

• void set_have_redo (bool redo)

'Setter' function for member m_have_redo

void split_trigger (int seqnum, midipulse tick)

Convenience function for perfroll's split-trigger functionality.

midipulse get_max_trigger ()

Locates the largest trigger value among the active sequences.

· void collapse ()

Convenience function for perfedit's collapse functionality.

void copy ()

Convenience function for perfedit's copy functionality.

void expand ()

Convenience function for perfedit's expand functionality.

midi_control & midi_control_toggle (int ctl)

Retrieves a reference to a value from m_midi_cc_toggle[].

midi_control & midi_control_on (int ctl)

Retrieves a reference to a value from m_midi_cc_on[].

• midi_control & midi_control_off (int ctl)

Retrieves a reference to a value from m_midi_cc_off[].

void midi_control_event (const event &ev)

This function encapsulates code in input_func() to make it easier to read and understand.

• void handle_midi_control (int control, bool state)

Handle the MIDI Control values that provide some automation for the application.

bool handle_midi_control_ex (int control, midi_control::action a)

Provides operation of the new MIDI controls.

• const std::string & get_screen_set_notepad (int screen_set) const

Retrieves the given string from m_screen_set_notepad[].

• const std::string & current_screen_set_notepad () const

Returns the notepad text for the current screen-set.

void set screen set notepad (int screenset, const std::string ¬e)

Copies the given string into m_screen_set_notepad[].

void set_screen_set_notepad (const std::string ¬e)

Sets the notepad text for the current screen-set.

· void set playing screenset ()

Sets the screen set that is active, based on the value of m_screenset.

bool any_group_unmutes () const

'Getter' function for member m_mute_group[]

void mute_group_tracks ()

If m_mode_group is true, then this function operates.

void select_and_mute_group (int g_group)

Select a mute group and then mutes the track in the group.

void set_song_mute (mute_op_t op)

Provides for various settings of the song-mute status of all sequences in the song.

void set_mode_group_mute ()

'Setter' function for member m_mode_group

void unset_mode_group_mute ()

'Setter' function for member m_mode_group Unsets this member.

void select_group_mute (int g_mute)

If we're in group-learn mode, then this function gets the playing statuses of all of the sequences in the current playscreen, and copies them into the desired mute-group.

void set_mode_group_learn ()

Sets the group-mute mode, then the group-learn mode, then notifies all of the notification subscribers.

• void unset_mode_group_learn ()

Notifies all of the notification subscribers that group-learn is being turned off.

bool is group learning ()

'Getter' function for member m_mode_group_learn

void set_and_copy_mute_group (int group)

When in group-learn mode, for active sequences, the mute-group settings are set based on the playing status of each sequence.

· bool activate ()

Performs a controlled activation of the jack_assistant and other JACK modules.

void start (bool state)

If JACK is not running, call inner_start() with the given state.

• void stop ()

If JACK is not running, call inner_stop().

· void start_jack ()

If JACK is supported, starts the JACK transport.

void stop_jack ()

If JACK is supported, stops the JACK transport.

void position jack (bool state, midipulse tick=0)

If JACK is supported and running, sets the position of the transport.

void off_sequences ()

For all active patterns/sequences, set the playing state to false.

void all notes off ()

For all active patterns/sequences, turn off its playing notes.

void set_active (int seq, bool active)

Sets or unsets the active state of the given pattern/sequence number.

void set_was_active (int seq)

Sets was-active flags: main, edit, perf, and names.

• void reset_sequences (bool pause=false)

For all active patterns/sequences, get its playing state, turn off the playing notes, set playing to false, zero the markers, and, if not in playback mode, restore the playing state.

• void play (midipulse tick)

Plays all notes to the current tick.

void set_orig_ticks (midipulse tick)

For every pattern/sequence that is active, sets the "original tick" value for the pattern.

void set_beats_per_minute (midibpm bpm)

Sets the value of the BPM into the master MIDI buss, after making sure it is squelched to be between 20 and 500.

void set_looping (bool looping)

'Setter' function for member m looping

· int max active set () const

Checks the whole universe of sequences to determine the current last-active set, that is, the highest set that has any active sequences in it.

void launch_input_thread ()

Creates the input thread using input_thread_func().

void launch output thread ()

Creates the output thread using output_thread_func().

bool init_jack_transport ()

Initializes JACK support, if SEQ64_JACK_SUPPORT is defined.

bool deinit_jack_transport ()

Tears down the JACK infrastructure.

bool seq_in_playing_screen (int seq)

A helper function for determining if the mode group is in force, the playing screenset is the same as the current screenset, and the sequence is in the range of the playing screenset.

· void is modified (bool flag)

'Setter' function for member m_is_modified

· bool valid_midi_control_seq (int seq) const

Checks the parameter against c_midi_controls_extended.

bool is screenset valid (int screenset) const

Checks the screenset against m_max_sets.

void set running (bool running)

'Setter' function for member m_running

void is_pattern_playing (bool flag)

'Setter' function for member m_is_pattern_playing

void set_playback_mode (bool playbackmode)

'Setter' function for member m_playback_mode

int mute_group_offset (int track)

A helper function to calculate the index into the mute-group array, based on the desired track.

bool is_seq_valid (int seq) const

Provides common code to check for the bounds of a sequence number.

• bool is_mseq_valid (int seq) const

Validates the sequence number, which is important since they're currently used as array indices.

• bool install_sequence (sequence *seq, int seqnum)

A private helper function for add_sequence() and new_sequence().

void inner_start (bool state)

Locks on m_condition_var.

• void inner_stop (bool midiclock=false)

Unconditionally, and without locking, clears the running status, resets the sequences, and sets m_usemidiclock false.

• int clamp_track (int track) const

Provides common code to keep the track value valid.

void set_key_event (unsigned int keycode, long sequence_slot)

At construction time, this function sets up one keycode and one event slot.

void set_key_group (unsigned int keycode, long group_slot)

At construction time, this function sets up one keycode and one group slot.

• bool create_master_bus ()

Creates the mastermidibus.

void add_clock (clock_e clocktype)

Saves the clock settings read from the "rc" file so that they can be passed to the mastermidibus after it is created.

void set_clock (int bus, clock_e clocktype)

Sets a single clock item, if in the currently existing range.

void add_input (bool flag)

Saves the input settings read from the "rc" file so that they can be passed to the mastermidibus after it is created.

void set_input (int bus, bool inputing)

Sets a single input item, if in the currently existing range.

- bool get_input (int bus)
- bool is input system port (int bus)

Private Attributes

· bool m song start mode

If true, playback is done in Song mode, not Live mode.

bool m_start_from_perfedit

Indicates that, no matter what the current Song/Live setting, the playback was started from the perfedit window.

· bool m_reposition

It seems that this member, if true, forces a repositioning to the left (L) tick marker.

float m_excell_FF_RW

Provides an "acceleration" factor for the fast-forward and rewind functionality.

• ff rw button t m FF RW button type

Indicates whether the fast-forward or rewind key is in effect in the perfedit window.

bool m_mute_group [c_max_sequence]

Mute group support.

bool m_armed_saved

Indicates if the m_saved_armed_statuses[] values are the saved state of the sequences, and can be restored.

• bool m_armed_statuses [c_max_sequence]

Holds the "global" saved status of the playing tracks, for restoration after saving.

bool m_tracks_mute_state [c_seqs_in_set]

Holds the current mute states of each track.

• bool m_mode_group

If true, indicates that a mode group is selected, and playing statuses will be "memorized".

• bool m mode group learn

If true, indicates that a group learn is selected, which also "memorizes" a mode group, and notifies subscribers of a group-learn change.

· int m mute group selected

Selects a group to mute.

int m_playing_screen

Playing screen support.

int m_playscreen_offset

Playing screen sequence number offset.

• sequence * m_seqs [c_max_sequence]

Provides a "vector" of patterns/sequences.

bool m_seqs_active [c_max_sequence]

Each boolean value in this array is set to true if a sequence is active, meaning that it will be used to hold some kind of MIDI data, even if only Meta events.

• bool m_was_active_main [c_max_sequence]

Each boolean value in this array is set to true if a sequence was active, meaning that it was found to be active at the time we were setting it to inactive.

• bool m was active edit [c max sequence]

Each boolean value in this array is set to true if a sequence was active, meaning that it was found to be active at the time we were setting it to inactive.

bool m_was_active_perf [c_max_sequence]

Each boolean value in this array is set to true if a sequence was active, meaning that it was found to be active at the time we were setting it to inactive.

bool m was active names [c max sequence]

Each boolean value in this array is set to true if a sequence was active, meaning that it was found to be active at the time we were setting it to inactive.

bool m_sequence_state [c_max_sequence]

Saves the current playing state of each pattern.

· int m transpose

Holds the global MIDI transposition value.

· pthread t m out thread

Provides information for managing pthreads.

pthread_t m_in_thread

Provides a "handle" to the input thread.

· bool m out thread launched

Indicates that the output thread has been started.

· bool m_in_thread_launched

Indicates that the input thread has been started.

bool m running

Indicates that playback is running.

· bool m_is_pattern_playing

Indicates that a pattern is playing.

· bool m_inputing

Indicates that events are being written to the MIDI input busses in the input thread.

bool m_outputing

Indicates that events are being written to the MIDI output busses in the output thread.

bool m_looping

Indicates that status of the "loop" button in the performance editor.

· bool m playback mode

Specifies the playback mode.

• int m_ppqn

Holds the current PPQN for usage in various actions.

midibpm m_bpm

Holds the current BPM (beats per minute) for later usage.

• int m_beats_per_bar

Holds the beats/bar value as obtained from the MIDI file.

· int m_beat_width

Holds the beat width value as obtained from the MIDI file.

int m_clocks_per_metronome

Augments the beats/bar and beat-width with the additional values included in a Time Signature meta event.

int m_32nds_per_quarter

Augments the beats/bar and beat-width with the additional values included in a Time Signature meta event.

• long m_us_per_quarter_note

Augments the beats/bar and beat-width with the additional values included in a Tempo meta event.

mastermidibus * m_master_bus

Provides our MIDI buss.

• std::vector< clock e > m master clocks

Saves the clock settings obtained from the "rc" (options) file so that they can be loaded into the mastermidibus once it is created.

• std::vector< bool > m_master_inputs

Saves the input settings obtained from the "rc" (options) file so that they can be loaded into the mastermidibus once it is created.

• midipulse m_one_measure

Holds the "one measure's worth" of pulses (ticks), which is normally m_ppqn * 4.

midipulse m_left_tick

Holds the position of the left (L) marker, and it is first defined as 0.

midipulse m_right_tick

Holds the position of the right (R) marker, and it is first defined as the end of the fourth measure.

· midipulse m starting tick

Holds the starting tick for playing.

· midipulse m_tick

MIDI Clock support.

midipulse m_jack_tick

Let's try to save the last JACK pad structure tick for re-use with resume after pausing.

· bool m_usemidiclock

More MIDI clock support.

bool m_midiclockrunning

More MIDI clock support.

· int m midiclocktick

More MIDI clock support.

• int m_midiclockpos

More MIDI clock support.

· bool m dont reset ticks

Support for pause, which does not reset the "last tick" when playback stops/starts.

• std::string m_screen_set_notepad [c_max_sets]

Used in the mainwnd class to set the notepad text for the given set.

midi_control m_midi_cc_toggle [c_midi_controls_extended]

Provides the settings of MIDI Toggle, as read from the "rc" file.

midi_control m_midi_cc_on [c_midi_controls_extended]

Provides the settings of MIDI On, as read from the "rc" file.

midi_control m_midi_cc_off [c_midi_controls_extended]

Provides the settings of MIDI Off, as read from the "rc" file.

int m_offset

Holds the current offset into the screen-sets.

int m_control_status

Holds the OR'ed control status values.

• int m_screenset

Indicates the number of the currently-selected screen-set.

int m_seqs_in_set

 $We \textit{ will eventually replace } c_seqs_in_set \textit{ with this member, which defaults to the value of } c_seqs_in_set.$

int m_max_sets

A replacement for the c_max_sets constant.

int m_sequence_count

Keeps track of created sequences, whether or not they are active.

int m_sequence_max

A replacement for the c_max_sequence constant.

int m_sequence_high

Indicates the highest-number sequence.

· int m edit sequence

Hold the number of the currently-in-edit sequence.

bool m_is_modified

It may be a good idea to eventually centralize all of the dirtiness of a performance here.

condition_var m_condition_var

A condition variable to protect playback.

· jack_assistant m_jack_asst

A wrapper object for the JACK support of this application.

- · bool m_have_undo
- std::vector< int > m_undo_vect

Holds the "track" numbers or the "all tracks" values for undo operations.

- bool m_have_redo
- std::vector< int > m redo vect

Holds the "track" numbers or the "all tracks" values for redo operations.

- std::vector< performcallback * > m_notify
- gui_assistant & m_gui_support

Support for a wide range of GUI-related operations.

Static Private Attributes

· static midi_control sm_mc_dummy

Provides a dummy, inactive midi_control object to handle out-of-range midi_control indicies.

Friends

- · class jack assistant
- · class keybindentry
- · class mainwnd
- · class midifile
- · class optionsfile
- · class options
- · class perfedit
- class perfroll
- void * input_thread_func (void *myperf)

Set up the performance, and set the process to realtime privileges.

void * output_thread_func (void *myperf)

Global functions defined in perform.cpp.

int jack_sync_callback (jack_transport_state_t state, jack_position_t *pos, void *arg)

Global functions for JACK support and JACK sessions.

- int jack_transport_callback (jack_nframes_t nframes, void *arg)
- void jack_shutdown (void *arg)
- void jack_timebase_callback (jack_transport_state_t state, jack_nframes_t nframes, jack_position_t *pos, int new_pos, void *arg)

The JACK timebase function defined here sets the JACK position structure.

• long get_current_jack_position (void *arg)

13.69.1 Detailed Description

It has way too many data members, one of them public. Might be ripe for refactoring. That has its own dangers, of course.

13.69.2 Member Enumeration Documentation

13.69.2.1 mute_op_t

enum seq64::perform::mute_op_t

Enumerator

MUTE_TOGGLE	
MUTE_OFF	
MUTE_ON	

```
13.69.2.2 ff_rw_button_t
enum seq64::perform::ff_rw_button_t
```

Enumerator

FF_RW_REWIND	
FF_RW_NONE	
FF_RW_FORWARD	

13.69.3 Constructor & Destructor Documentation

Also note that we have a little issue with the fact that various sequences (patterns) can potentially have different beats/measure and beat-width values.

Currently, when reading the MIDI file, the beats/minute value is obtained from the MIDI file, if present, and this value is passed to perform::set_beats_per_minute(), which forwards it to the master MIDI buss and JACK assistant objects. This Tempo setting comes from both the Tempo meta event in track 0, and from the Seq24's c_bpm SeqSpec section! This setting is now also made for the two Time Signature values.

But note that Sequencer64 now scales the c_bpm value so that two extra digits of precision can be saved with the MIDI file. We went throughout the code, changing BPM from an integer to a double.

Parameters

mygui	Provides access to the GUI assistant that holds many things, including the containers of keys and the "events" they provide. This is a base-class reference; for a real class, see the gui_assistant_gtk2 class in the seq_gtkmm2 GUI-specific library. Note that we access the m_gui_support member using the gui() accessor function.
ppqn	The default, choosable, or actual PPQN value.

```
13.69.3.2 ~perform() seq64::perform::~perform ( )
```

Finally, any active or inactive (but allocated) patterns/sequences are deleted, and their pointers nullified.

Note that we could use m_sequence_high to replace m_sequence_max in the for-loop, but who cares, we are exiting!

13.69.4 Member Function Documentation

```
13.69.4.1 is_modified() [1/2]
bool seq64::perform::is_modified ( ) const [inline]

13.69.4.2 modify()
void seq64::perform::modify ( ) [inline]
```

The setter that will, is_modified(), is private. No one but perform and its friends should falsify this flag.

```
13.69.4.3 ppqn()
```

```
int seq64::perform::ppqn ( ) const [inline]
```

13.69.4.4 sequence_count()

```
int seq64::perform::sequence_count ( ) const [inline]
```

In many cases at startup, or when loading a file, there are no sequences yet, and still the code calls functions that try to access them.

```
13.69.4.5 sequence_max()
```

```
int seq64::perform::sequence_max ( ) const [inline]
```

13.69.4.6 is_control_status()

```
bool seq64::perform::is_control_status ( ) const [inline]
```

Returns

Returns true if the m_control_status value is non-zero, which means that there is a queue, replace, or snapshot functionality in progress.

13.69.4.7 set_edit_sequence()

Parameters

seqnum

Pass in -1 to disable the edit-sequence number unconditionally. Use unset_edit_sequence() to disable it if it matches the current edit-sequence number.

13.69.4.8 unset_edit_sequence()

Disables the edit-sequence number if it matches the parameter.

Parameters

seqnum	The sequence number of the sequence to unset.
--------	---

13.69.4.9 is_edit_sequence()

Parameters

seqnum

Tests the parameter against m_edit_sequence. Returns true if that member is not -1, and the parameter matches it.

13.69.4.10 get_beats_per_bar()

```
int seq64::perform::get_beats_per_bar ( ) const [inline]
```

13.69.4.11 set_beats_per_bar()

Parameters

bpm

Provides the value for beats/measure. Also used to set the beats/measure in the JACK assistant object.

```
13.69.4.12 get_beat_width()
int seq64::perform::get_beat_width ( ) const [inline]
13.69.4.13 set_beat_width()
void seq64::perform::set_beat_width (
             int bw ) [inline]
Parameters
      Provides the value for beat-width. Also used to set the beat-width in the JACK assistant object.
13.69.4.14 clocks_per_metronome() [1/2]
void seq64::perform::clocks_per_metronome (
             int cpm ) [inline]
13.69.4.15 clocks_per_metronome() [2/2]
int seq64::perform::clocks_per_metronome ( ) const [inline]
13.69.4.16 set_32nds_per_quarter()
void seq64::perform::set_32nds_per_quarter (
             int tpq ) [inline]
13.69.4.17 get_32nds_per_quarter()
int seq64::perform::get_32nds_per_quarter ( ) const [inline]
```

```
13.69.4.18 us_per_quarter_note() [1/2]
void seq64::perform::us_per_quarter_note (
            long upqn ) [inline]
13.69.4.19 us_per_quarter_note() [2/2]
long seq64::perform::us_per_quarter_note ( ) const [inline]
13.69.4.20 gui() [1/2]
const gui_assistant& seq64::perform::gui ( ) const [inline]
13.69.4.21 gui() [2/2]
gui_assistant& seq64::perform::gui ( ) [inline]
13.69.4.22 keys() [1/2]
const keys_perform& seq64::perform::keys ( ) const [inline]
13.69.4.23 keys() [2/2]
keys_perform& seq64::perform::keys ( ) [inline]
13.69.4.24 master_bus()
mastermidibus& seq64::perform::master_bus ( ) [inline]
```

```
13.69.4.25 filter_by_channel()
void seq64::perform::filter_by_channel (
             bool flag ) [inline]
13.69.4.26 is_running()
bool seq64::perform::is_running ( ) const [inline]
13.69.4.27 is_pattern_playing() [1/2]
bool seq64::perform::is_pattern_playing ( ) const [inline]
13.69.4.28 toggle_song_start_mode()
bool seq64::perform::toggle_song_start_mode ( ) [inline]
13.69.4.29 song_start_mode() [1/2]
void seq64::perform::song_start_mode (
             bool flag ) [inline]
13.69.4.30 song_start_mode() [2/2]
bool seq64::perform::song_start_mode ( ) const [inline]
13.69.4.31 is_jack_running()
bool seq64::perform::is_jack_running ( ) const [inline]
13.69.4.32 is_jack_master()
bool seq64::perform::is_jack_master ( ) const [inline]
13.69.4.33 enregister()
void seq64::perform::enregister (
             performcallback * pfcb ) [inline]
```

Parameters

pfcb Provides the pointer to the performance callback.

13.69.4.34 toggle_jack_mode()

```
void seq64::perform::toggle_jack_mode ( ) [inline]
```

13.69.4.35 set_jack_mode()

Note that we moved some of the code from perfedit::set_jack_mode() [the seq32 version] to this function.

Parameters

Returns

Returns true if JACK is running currently, and false otherwise.

13.69.4.36 get_toggle_jack()

```
bool seq64::perform::get_toggle_jack ( ) const [inline]
```

13.69.4.37 set_jack_stop_tick()

13.69.4.38 combine_bytes()

```
unsigned short seq64::perform::combine\_bytes ( midibyte\ b0, midibyte\ b1 )
```

http://www.blitter.com/~russtopia/MIDI/~jglatt/tech/midispec/wheel.htm

Two data bytes follow the status. The two bytes should be combined together to form a 14-bit value. The first data byte's bits 0 to 6 are bits 0 to 6 of the 14-bit value. The second data byte's bits 0 to 6 are really bits 7 to 13 of the 14-bit value. In other words, assuming that a C program has the first byte in the variable First and the second data byte in the variable Second, here's how to combine them into a 14-bit value (actually 16-bit since most computer CPUs deal with 16-bit, not 14-bit, integers).

Parameters

b0	The first byte to be combined.
b1	The second byte to be combined.

Returns

Returns the bytes basically OR'd together.

13.69.4.39 FF_rewind()

```
void seq64::perform::FF_rewind ( )
```

It changes m_tick by a quarter of the number of ticks in a standard measure, with m_excell_FF_RW (defaults to one) to factor the difference.

13.69.4.40 FF_RW_timeout()

```
bool seq64::perform::FF_RW_timeout ( )
```

This function is used in the free function version of FF_RW_timeout() as a callback to the gtk_timeout() function. It multiplies m_excell_FF_RW by 1.1 as long as one of the fast-forward or rewind keys is held, and is less than 60.

Returns

Returns true if one of the fast-forward or rewind keys was held, leaving m_excell_FF_RW at the last value it had. Otherwise, it resets the value to 1, and returns false.

```
13.69.4.41 start_from_perfedit() [1/2]
```

```
13.69.4.42 start_from_perfedit() [2/2]
bool seq64::perform::start_from_perfedit ( ) const [inline]
13.69.4.43 set_follow_transport()
void seq64::perform::set_follow_transport (
            bool flag ) [inline]
13.69.4.44 get_follow_transport()
bool seq64::perform::get_follow_transport ( ) const [inline]
13.69.4.45 toggle_follow_transport()
void seq64::perform::toggle_follow_transport ( ) [inline]
13.69.4.46 set_reposition()
void seq64::perform::set_reposition (
            bool postype = true ) [inline]
13.69.4.47 ff_rw_type() [1/2]
ff_rw_button_t seq64::perform::ff_rw_type ( ) [inline]
13.69.4.48 ff_rw_type() [2/2]
void seq64::perform::ff_rw_type (
             ff_rw_button_t button_type ) [inline]
13.69.4.49 rewind()
void seq64::perform::rewind (
             bool press ) [inline]
```

Parameters

press If true, the status is set to FF_RW_REWIND, otherwise it is set to FF_RW_NONE.

13.69.4.50 fast_forward()

Parameters

press If true, the status is set to FF_RW_FORWARD, otherwise it is set to FF_RW_NONE.

13.69.4.51 reposition()

Used only in perfroll :: on_key_press_event() to implement the Seq32 pointer-position feature.

Parameters

tick Provides the position value to be set.

13.69.4.52 clear_all()

```
bool seq64::perform::clear_all ( )
```

The mainwnd module calls this function. Note that perform now handles the "is modified" flag on behalf of all external objects, to centralize and simplify the dirtying of a MIDI tune.

Anything else to clear? What about all the other sequence flags? We can beef up delete_sequence() for them, at some point.

Added stazed code from 1.0.5 to abort clearing if any of the sequences are in editing.

Returns

Returns true if the clear-all operation could be performed. If false, then at least one active sequence was in editing mode.

13.69.4.53 launch()

This function is called in main(). We collected all the calls here as a simplification, and renamed it because it is more than just initialization. This function must be called after the perform constructor and after the configuration file and command-line configuration overrides. The original implementation, where the master buss was an object, was too inflexible to handle a JACK implementation.

Parameters

ppqn Provides the PPQN value, which is either the default value (192) or is read from the "user" configuration file.

Todo We probably need a bpm parameter for consistency at some point.

```
13.69.4.54 new_sequence()
```

Then it activates the pattern [this is done in the install_sequence() function]. It doesn't deal with thrown exceptions.

This function is called by the seqmenu and mainwid objects to create a new sequence. We now pass this sequence to install_sequence() to better handle potential memory leakage, and to make sure the sequence gets counted. Also, adding a new sequence from the user-interface is a significant modification, so the "is modified" flag gets set.

Change Note ca 2016-05-15 If enabled, wire in the MIDI buss override.

Parameters

seq The prospective sequence number of the new sequence.

13.69.4.55 add_sequence()

No check is made for a null pointer, but the install_sequence() call will make sure such a pointer is officially logged.

This function checks for the preferred sequence number. This is the number that was specified by the Sequence Number meta-event for the current track. If the preferred sequence number is in the valid range (0 to m_sequence — max) and it is not active, add it and activate it. Otherwise, iterate through all patterns from prefnum to m_ sequence _max and add and activate the first one that is not active, and then finish.

Finally, note that this function is used only by midifile, when reading in a MIDI song. Therefore, the "is modified" flag is *not* set by this function; loading a sequence from a file is not a modification that should lead to a prompt for saving the file later.

Todo Shouldn't we wrap around the sequence list if we can't find an empty sequence slot after prefnum?

Todo This function needs some deeper analysis against the original, in my opinion.

Warning

The logic of the if-statement in this function was such that *prefnum* could be out-of-bounds in the else-clause. We reworked the logic to be airtight. This bug was caught by gcc 4.8.3 on CentOS, but not on gcc 4.9.3 on Debian Sid!

Parameters

seq	The pointer to the pattern/sequence to add.
prefnum	The preferred sequence number of the pattern, as explained above. If this value is out-of-range, then it is basically ignored.

13.69.4.56 delete_sequence()

We now also solidify the deletion by setting the pointer to null after deletion, so it will blow up if accidentally accessed. The final act is to raise the "is modified" flag, since deleting an existing sequence is always a significant modification.

Now, this function obviously sets the "active" flag for the sequence to false. But there are a few other flags that are not modified; shouldn't we also falsify them here?

Parameters

The sequence number of the sequence to be deleted. It is validated	ed.
--	-----

13.69.4.57 is_sequence_in_edit()

Parameters

seg Provides the sequence number to be checked.

Returns

Returns truen if the sequence's get_editing() call returns true. Otherwise, false is returned, which can also indicate an illegal sequence number.

Parameters

13.69.4.64 set_jack_tick()

void seq64::perform::set_jack_tick (

midipulse tick) [inline]

seq Provides the desired sequence. The is_active() function validates this value.

Parameters

tick Provides the current JACK tick (pulse) value to set.

13.69.4.65 set_left_tick()

We let the caller determine if this setting is a modification. If the left tick is later than the right tick, the right tick is move to one measure past the left tick.

Todo The perform::m_one_measure member is currently hardwired to PPQN * 4.

Parameters

tick	The tick (MIDI pulse) at which to place the left tick. If the left tick is greater than or equal to the right
	tick, then the right ticked is moved forward by one "measure's length" (m_ppqn $*$ 4) past the left tick.
setstart	If true (the default, and long-standing implicit setting), then the starting tick is also set to the left tick.

13.69.4.66 get_left_tick()

```
midipulse seq64::perform::get_left_tick ( ) const [inline]
```

13.69.4.67 set_start_tick()

Parameters

tick Provides the starting JACK tick (pulse) value to set.

13.69.4.68 get_start_tick()

```
midipulse seq64::perform::get_start_tick ( ) const [inline]
```

13.69.4.69 set_right_tick()

This setting is made only if the tick parameter is at or beyond the first measure. We let the caller determine is this setting is a modification.

Parameters

tick	The tick (MIDI pulse) at which to place the right tick. If less than or equal to the left tick setting, then the left tick is backed up by one "measure's worth" ($m_ppqn * 4$) worth of ticks from the new right tick.
setstart	If true (the default, and long-standing implicit setting), then the starting tick is also set to the left tick, if that got changed.

13.69.4.70 get_right_tick()

```
midipulse seq64::perform::get_right_tick ( ) const [inline]
```

13.69.4.71 left_right_size()

```
double seq64::perform::left_right_size ( ) const [inline]
```

Returns

Returns the difference between the right and left tick, cast to double.

13.69.4.72 is_active()

Parameters

seq The pattern number. It is checked for invalidity. This can lead to "too many" (i.e. redundant) checks, but we're trying to centralize such checks in this function.

Returns

Returns the value of the active-flag, or false if the sequence was invalid or null.

```
13.69.4.73 apply_song_transpose()
void seq64::perform::apply_song_transpose ( )
13.69.4.74 set_transpose()
void seq64::perform::set_transpose (
             int t ) [inline]
13.69.4.75 get_transpose()
int seq64::perform::get_transpose ( ) const [inline]
13.69.4.76 get_beats_per_minute()
midibpm seq64::perform::get_beats_per_minute ( ) [inline]
Returns
     Returns the value of beats/minute from the master buss.
13.69.4.77 set_sequence_control_status()
void seq64::perform::set_sequence_control_status (
              int status )
Then the given status is OR'd into the m_control_status.
Parameters
 status
         The status to be used.
13.69.4.78 unset_sequence_control_status()
void seq64::perform::unset_sequence_control_status (
```

int status)

Then the given status is reversed in m_control_status.

Parameters

status The status to be used.	
-------------------------------	--

13.69.4.79 sequence_playing_toggle()

If m_control_status is c_status_queue, then the sequence's toggle_queued() function is called. This is the "mod queue" implementation.

Otherwise, if it is c_status_replace, then the status is unset, and all sequences are turned off. Then the sequence's toggle-playing() function is called, which should turn it back on. This is the "mod replace" implementation; it is like a Solo. But can it be undone?

This function is called in sequence_key() to implement a toggling of the sequence of the pattern slot in the current screen-set that is represented by the keystroke.

This function is also called in midi_control_event() if the control number represents a sequence number in a screen-set, that is, it ranges from 0 to

1. This value is offset by the current screen-set number, m_offset before passing it to this function.

Parameters

seq The sequence number of the sequence to be potentially toggled. This value must be a valid and active sequence number.

13.69.4.80 sequence_playing_change()

Used for the implementation of sequence_playing_on() and sequence_playing_off().

Parameters

seq	The number of the sequence to be turned off.
on	True if the sequence is to be turned on, false if it is to be turned off.

13.69.4.81 sequence_playing_on()

Parameters

seq The sequence number of the sequence to turn on.

13.69.4.82 sequence_playing_off()

Parameters

seq The sequence number of the sequence to turn off.

13.69.4.83 mute_all_tracks()

```
void seq64::perform::mute_all_tracks (
          bool flag = true )
```

Covers tracks from 0 to m_sequence_max.

We have to also set the sequence's playing status, in opposition to the mute status, in order to see the sequence status change on the user-interface. HMMMMMM.

Parameters

flag If true (the default), the song-mute of the sequence is turned on. Otherwise, it is turned off.

13.69.4.84 toggle_all_tracks()

```
void seq64::perform::toggle_all_tracks ( )
```

Covers tracks from 0 to m_sequence_max.

13.69.4.85 armed_saved()

```
bool seq64::perform::armed_saved ( ) const [inline]
```

13.69.4.86 toggle_playing_tracks()

```
void seq64::perform::toggle_playing_tracks ( )
```

13.69.4.87 mute_screenset()

Parameters

ss	The screen-set to be operated upon.]
flag	If true (the default), the song-mute of the sequence is turned on. Otherwise, it is turned off.]

13.69.4.88 output_func()

```
void seq64::perform::output_func ( )
```

This function is called by the free function output_thread_func(). Here's how it works:

```
It runs while m_outputing is true.MORE TO COME. Yeah, a lot more to come. It is a complex function.
```

Change Note ca 2016-01-26 Hurray, seq24 is coming back to life! We see that there is a fix for clock tick drift here, which relies on using long and long long values. See the Changelog for seq24 0.9.3.

Warning

Valgrind shows that output_func() is being called before the JACK client pointer is being initialized!!!

- 1. Get delta time (current last).
- 2. Get delta ticks from time.
- 3. Add to current_ticks.
- 4. Compute prebuffer ticks.
- 5. Play from current tick to prebuffer.

Figure out how much time we need to sleep, and do it.

Now we want to trigger every c_thread_trigger_width_us, and it took us delta_us to play(). Also known as the "sleeping_us".

Check MIDI clock adjustment. Note that we replaced " $60000000.0f / m_ppqn / bpm$ " with a call to a function. We also removed the "f" specification from the constants.

13.69.4.89 input_func()

```
void seq64::perform::input_func ( )
```

Stazed:

```
http://www.blitter.com/~russtopia/MIDI/~jglatt/tech/midispec/ssp.htm
```

Example: If a Song Position value of 8 is received, then a sequencer (or drum box) should cue playback to the third quarter note of the song. (8 MIDI beats \star 6 MIDI clocks per MIDI beat = 48 MIDI Clocks. Since there are 24 MIDI Clocks in a quarter note, the first quarter occurs on a time of 0 MIDI Clocks, the second quarter note occurs upon the 24th MIDI Clock, and the third quarter note occurs on the 48th MIDI Clock).

8 MIDI beats * 6 MIDI clocks per MIDI beat = 48 MIDI Clocks.

13.69.4.90 set_group_mute_state()

The index value is the track number offset by the number of the selected mute group (which is equivalent to a set number) times the number of sequences in a set. This function is used in midifile and optionsfile when parsing the file to get the initial mute-groups.

Parameters

ç	gtrack	The number of the track to be muted/unmuted.
r	muted	This boolean indicates the state to which the track should be set.

13.69.4.91 get_group_mute_state()

Uses the mute_group_offset() function. This function is used in midifile and optionsfile when writing the file to get the initial mute-groups.

Parameters

gtrack	The number of the track for which the state is to be obtained. Like set_group_mute_state(), this value
	is offset by adding m_mute_group_selected * m_seqs_in_set.

Returns

Returns the desired m_mute_group[] value.

13.69.4.92 set_offset()

Sets m_offset = offset * c_mainwnd_rows * c_mainwnd_cols.

Parameters

```
offset The desired offset.
```

13.69.4.93 get_offset()

```
int seq64::perform::get_offset ( ) const [inline]
```

13.69.4.94 save_playing_state()

```
void seq64::perform::save_playing_state ( )
```

Inactive patterns get the value set to false. Used in unsetting the snapshot status (c_status_snapshot).

13.69.4.95 restore_playing_state()

```
void seq64::perform::restore_playing_state ( )
```

Used in unsetting the snapshot status (c_status_snapshot).

13.69.4.96 key_name()

Parameters

k The key number for which to return the string name of the key.

```
13.69.4.97 get_key_events()
keys_perform::SlotMap& seq64::perform::get_key_events ( ) [inline]
13.69.4.98 get_key_groups()
keys_perform::SlotMap& seq64::perform::get_key_groups ( ) [inline]
13.69.4.99 get_key_events_rev()
keys_perform::RevSlotMap& seq64::perform::get_key_events_rev ( ) [inline]
13.69.4.100 get_key_groups_rev()
keys_perform::RevSlotMap& seq64::perform::get_key_groups_rev () [inline]
13.69.4.101 show_ui_sequence_key() [1/2]
bool seq64::perform::show_ui_sequence_key ( ) const [inline]
Used in mainwid, options, optionsfile, userfile, and perform.
13.69.4.102 show_ui_sequence_key() [2/2]
void seq64::perform::show_ui_sequence_key (
             bool flag ) [inline]
Parameters
       Provides the flag to set into keys().show_ui_sequence_key().
 flag
```

Provides the value to set into keys().show_ui_sequence_number().

13.69.4.105 lookup_keyevent_key()

If we're not in legacy mode, then we adjust for the screenset, so that screensets greater than 0 can also show the correct key name, instead of a question mark.

Legacy seq24 already responds to the toggling of the mute state via the shortcut keys even if screenset > 0, but it shows the question mark.

Parameters

flag

seqnum The number of the sequence for which to return the event key.

Returns

Returns the desired key. If there is no such value, then the period ('?') character is returned.

13.69.4.106 lookup_keyevent_seq()

The inverse of lookup_keyevent_key().

Parameters

keycode The number of the event key for which to return the configured sequence number.

Returns

Returns the desired sequence. If there is no such value, then a sequence number of 0 is returned.

13.69.4.107 lookup_keygroup_key()

Parameters

groupnum The number of the sequence for which to return the group key.

Returns

Returns the desired key. If there is no such value, then the period ('.') character is returned.

13.69.4.108 lookup_keygroup_group()

The inverse of lookup_keygroup_key().

Parameters

keycode The number of the group key for which to return the configured sequence number.

Returns

Returns the desired group number. If there is no such value, then a group number of 0 is returned.

13.69.4.109 start_playing()

We've reversed the start() and start_jack() calls so that JACK is started first, to match all of the other use-cases for playing that we've found in the code. Note that the complementary function, stop_playing(), is an inline function defined in the header file.

The perform::start() function passes its boolean flag to perform::inner_start(), which sets the playback mode to that flag; if that flag is false, that turns off "song" mode. So that explains why mute/unmute is disabled.

Playback use cases:

```
These use cases are meant to apply to either a Seq32 or a regular build
of Sequencer64, eventually. Currently, the regular build does not have
a concept of a "global" perform song-mode flag.
-# mainwnd.
    -# Play. If the perform song-mode is "Song", then use that mode.
       Otherwise, use "Live" mode.
    -# Stop. This action is modeless here. In ALSA, it will cause
       a rewind (but currently seqroll doesn't rewind until Play is
       clicked, a minor bug).
    -# Pause. Same processing as Play or Stop, depending on current
       status. When stopping, the progress bars in seqroll and
       perfroll remain at their current point.
-# perfedit.
    -# Play. Override the current perform song-mode to use "Song".
    -# Stop. Revert the perfedit setting, in case play is restarted
       or resumed via mainwnd.
    -# Pause. Same processing as Play or Stop, depending on current
       status.
 -# ALSA versus JACK. One issue here is that, if JACK isn't "running"
   at all (i.e. we are in ALSA mode), then we cannot be JACK Master.
```

Helgrind shows a read/write race condition in m_start_from_perfedit bewteen jack_transport_callback() and start← _playing() here. Is inline function access of a boolean atomic?

Parameters

songmode	Indicates if the caller wants to start the playback
	to as "JACK mode"). In the seq32 code at GitHul
	"global_jack_start_mode" flag, which is true for S

in Song mode (sometimes erroneously referred b, this flag was identical to the Song mode, and false for Live mode. False disables Song mode, and is the default, which matches seq24. Generally, we pass true in this parameter if we're starting playback from the perfedit window. It alters the m start from perfedit member, not the m song start mode member (which replaces the global flag now).

13.69.4.110 pause_playing()

```
void seq64::perform::pause_playing (
            bool sonamode = false)
```

Currently almost the same as stop playing(), but expanded as noted in the comments so that we ultimately have more granular control over what happens. We're researching the whole sequence of stopping and starting, and it can be tricky to make correct changes.

We still need to make restarting pick up at the same place in ALSA mode; in JACK mode, JACK transport takes care of that feature.

Change Note ca 2016-10-11 User layk noted this call, and it makes sense to not do this here, since it is unknown at this point what the actual status is. Note that we STILL need to FOLLOW UP on calls to pause playing() and stop playing() in perfedit, mainwnd, etc.

is_pattern_playing(false);

Parameters

songmode	Indicates that, if resuming play, it should play in Song mode (true) or Live mode (false). See the
	comments for the start_playing() function.

13.69.4.111 stop_playing()

```
void seq64::perform::stop_playing ( )
```

Stops playback, turns off the (new) m_dont_reset_ticks flag, and set the "is-pattern-playing" flag to false. With stop, reset the start-tick to either the left-tick or the 0th tick (to be determined, currently resets to 0).

13.69.4.112 start_key()

```
void seq64::perform::start_key (
          bool songmode = false )
```

Meant to be used by GUIs to unify the treatment of keys versus buttons. Also handy in the extended MIDI controls that people have requested.

Parameters

songmode | The live/play mode parameter to be passed along to the key processor. Defaults to false (live mode).

13.69.4.113 pause_key()

Meant to be used by GUIs to unify the treatment of keys versus buttons. Also handy in the extended MIDI controls that people have requested.

Parameters

songmode The live/play mode parameter to be passed along to the key processor, when starting playback.

Defaults to false (live mode).

13.69.4.114 stop_key()

```
void seq64::perform::stop_key ( )
```

Meant to be used by GUIs to unify the treatment of keys versus buttons. Also handy in the extended MIDI controls that people have requested.

13.69.4.115 learn_toggle()

```
void seq64::perform::learn_toggle ( ) [inline]
```

```
13.69.4.116 decrement_beats_per_minute()
```

```
midibpm seq64::perform::decrement_beats_per_minute ( )
```

Actually does a lot of work in those function calls.

Returns

Returns the resultant BPM, as a convenience.

13.69.4.117 increment_beats_per_minute()

```
midibpm seq64::perform::increment_beats_per_minute ( )
```

Actually does a lot of work in those function calls.

Returns

Returns the resultant BPM, as a convenience.

13.69.4.118 page_decrement_beats_per_minute()

```
midibpm seq64::perform::page_decrement_beats_per_minute ( )
```

Encapsulates some calls used in mainwnd. Actually does a lot of work in those function calls.

Returns

Returns the resultant BPM, as a convenience.

13.69.4.119 page_increment_beats_per_minute()

```
midibpm seq64::perform::page_increment_beats_per_minute ( )
```

Encapsulates some calls used in mainwnd. Actually does a lot of work in those function calls.

Returns

Returns the resultant BPM, as a convenience.

13.69.4.120 decrement_screenset()

```
int seq64::perform::decrement_screenset ( ) [inline]
```

13.69.4.121 increment_screenset()

```
int seq64::perform::increment_screenset ( ) [inline]
```

13.69.4.122 highlight()

This setting is currently a build-time option, but could be made a run-time option later.

Parameters

seq Provides a reference to the desired sequence.

13.69.4.123 is_smf_0()

Parameters

seg | Provides a reference to the desired sequence.

```
13.69.4.124 get_sequence() [1/2]
const sequence* seq64::perform::get_sequence (
```

int seq) const [inline]

This is the const version.

Parameters

seq The prospective sequence number.

Returns

Returns the value of m_seqs[seq] if seq is valid. Otherwise, a null pointer is returned.

Parameters

Returns

Returns the value of m_seqs[seq] if seq is valid. Otherwise, a null pointer is returned.

```
13.69.4.126 sequence_key()
void seq64::perform::sequence_key (
    int seq )
```

This function is use in mainwnd when toggling the mute/unmute setting using keyboard keys.

Parameters

seq The sequence's control-key number, which is relative to the current screen-set.

This string goes on the bottom-left of those user-interface elements.

The format of this string is something like the following example, depending on the "show sequence numbers" option. The values shown are, in this order, sequence number (if allowed), buss number, channel number, beats per bar, and beat width.

```
No sequence number: 31-16 4/4
Sequence number: 9 31-16 4/4
```

The sequence number and buss number are re 0, while the channel number is displayed re 1, unless it is an SMF 0 null channel (0xFF), in which case it is 0.

Note

Later, we could add the sequence hot-key to this string, though showing that is not much use in perfnames. Also, this function is a stilted mix of direct access and access through sequence number.

Parameters

seq

Provides the reference to the sequence, use for getting the sequence parameters to be written to the label string.

Returns

Returns the filled in label if the sequence is active. Otherwise, an empty string is returned.

13.69.4.128 set_input_bus()

This function is called by options::input_callback().

Note that the mastermidibus::set_input() function passes the setting along to the input busarray.

Tricky Code See the bus parameter. We should provide two separate functions for this feature, but it is already combined into one input-callback function with a lot of other functionality in the options module.

Parameters

bus	If this value is greater than SEQ64_DEFAULT_BUSS_MAX (32), then it is treated as a user-interface flag (PERFORM_KEY_LABELS_ON_SEQUENCE) that causes all the sequences to be dirtied, and thus get redrawn with the new user-interface setting.
active	Indicates whether the buss or the user-interface feature is active or inactive.

13.69.4.129 set_clock_bus()

Note that the call to mastermidibus::set_clock() also sets the clock in the output busarray.

Parameters

bus	The bus index to be set.
clocktype	Indicates whether the buss or the user-interface feature is e_clock_off, e_clock_pos, and
	e_clock_mod.

13.69.4.130 mainwnd_key_event()

This function handles the keys for the functions of replace, queue, keep-queue, snapshots, toggling mute groups, group learn, and playing screenset. For further keystroke processing, see mainwind :: on_key_press_event().

Keys not handled here are handled in mainwnd: bpm up & down; screenset up & down.

Parameters

```
k The keystroke object to be handled.
```

Returns

Returns true if the key was handled.

13.69.4.131 perfroll_key_event()

It handles the Ctrl keys for cut, copy, paste, and undo.

The "is modified" flag is raised if something is deleted, but we cannot yet handle the case where we undo all the changes. So, for now, we play it safe with the user, even if the user gets annoyed because he knows that he undid all the changes.

Parameters

k	The keystroke object to be handled.
drop_sequence	Provides the index of the sequence whose selected trigger is to be cut, copied, or pasted.
	Undo and redo are now supported.

Returns

Returns true if the key was handled.

13.69.4.132 playback_key_event()

To be used in mainwnd, perfedit, and seqroll.

The start/end key may be the same key (e.g. Space) to allow toggling when the same key is mapped to both triggers.

Checking is_running() may not work completely in JACK.

Change Note layk 2016-10-11 Issue #42 to prevent inadvertent step-edit in sequence :: stream_event(). We did it slightly different to save a little code; also found a spot that was missed.

Parameters

k	Provides the encapsulated keystroke to check.
songmode	Provides the "jack flag" needed by the mainwnd, seqroll, and perfedit windows. Defaults to false,
	which disables Song mode, and enables Live mode. But using Song mode seems to make the
	pause key not work in the performance editor.

Returns

Returns true if the keystroke matched the start, stop, or (new) pause keystrokes. Generally, no further keystroke processing is needed in this case.

13.69.4.133 move_triggers()

Parameters

	direction	Specifies the desired direction; false = left, true = right.
--	-----------	--

13.69.4.134 copy_triggers()

```
void seq64::perform::copy_triggers ( )
```

This copies the triggers between the L marker and R marker to the R marker.

13.69.4.135 push_trigger_undo()

Too bad we cannot yet keep track of all the undoes for the sake of properly handling the "is modified" flag.

This function now has a new parameter. Not added to this function is the seemingly redundant undo-push the seq32 code does; is this actually a seq42 thing?

Also, there is still an issue with our undo-handling for a single track. See pop_trigger_undo().

track

A new parameter (found in the stazed seq32 code) that allows this function to operate on a single track. A parameter value of SEQ64_ALL_TRACKS (-1, the default) implements the original behavior.

```
13.69.4.136 pop_trigger_undo()
void seq64::perform::pop_trigger_undo ( )
```

Todo Look at seq32/src/perform.cpp and the perform :: push_trigger_undo(track) function, which has a track parameter that has a -1 values the supports all tracks. It requires two new vectors (one for undo, one for redo), two new flags (likewise). We've put this code in place, no longer macroed out, now permanent.

See the sequence::is_dirty_main() function.

Parameters

seq The pattern number. It is checked for validity.

Returns

Returns the was-active-main flag value, before setting it to false. Returns false if the pattern was invalid.

```
13.69.4.139 is_dirty_edit()
```

seq The pattern number. It is checked for validity.

Returns

Returns the was-active-edit flag value, before setting it to false. Returns false if the pattern was invalid.

13.69.4.140 is_dirty_perf()

Parameters

seq The pattern number. It is checked for validity.

Returns

Returns the was-active-perf flag value, before setting it to false. Returns false if the pattern/sequence number was invalid.

13.69.4.141 is_dirty_names()

Parameters

seq The pattern number. It is checked for validity.

Returns

Returns the was-active-names flag value, before setting it to false. Returns false if the pattern/sequence number was invalid.

13.69.4.142 is_exportable()

```
seq The index of the desired sequence.
```

Returns

Returns true if the sequence has the three properties noted above.

13.69.4.143 set_screenset()

It's not clear that we need to set the "is modified" flag just because we changed the screen set, so we don't.

As a new feature, we would like to queue-mute the previous screenset, and queue-unmute the newly-selected screenset. Still working on getting it right.

Parameters

The index of the desired new screen set. It is forced to range from 0 to m_max_sets - 1. The clamping seems weird, but hews to seq24. What it does is let the user wrap around the screen-sets in the user interface.

13.69.4.144 get_screenset()

```
int seq64::perform::get_screenset ( ) const [inline]
```

13.69.4.145 get_playing_screenset()

```
int seq64::perform::get_playing_screenset ( ) const [inline]
```

13.69.4.146 toggle_other_seqs()

See mainwid::on_button_release_event(). If the Shift key is pressed, toggle the mute state of all other sequences. Inactive sequences are skipped.

seqnum	The sequence that is being clicked on. It must be active in order to allow toggling.
isshiftkey	Indicates if the shift-key functionality for toggling all of the other sequences is active.

Returns

Returns true if the full toggling was able to be performed.

13.69.4.147 toggle_other_names()

See perfnames::on_button_press_event(). If the Shift key is pressed, toggle the mute state of all other sequences. Inactive sequences are skipped.

Parameters

seqnum	The sequence that is being clicked on. It must be active in order to allow toggling.
isshiftkey	Indicates if the shift-key functionality for toggling all of the other sequences is active.

Returns

Returns true if the toggling was able to be performed.

13.69.4.148 have_undo()

```
bool seq64::perform::have_undo ( ) const [inline], [private]
```

13.69.4.149 set_have_undo()

Once it is set, it remains set, unless cleared by saving the file.

13.69.4.150 have_redo()

```
bool seq64::perform::have_redo ( ) const [inline], [private]
```

13.69.4.151 set_have_redo()

```
void seq64::perform::set_have_redo (
                bool redo ) [inline], [private]
```

13.69.4.152 split_trigger()

Parameters

seqnum	Indicates the sequence that needs to have its trigger split.
tick	The MIDI pulse number at which the trigger should be split.

13.69.4.153 get_max_trigger()

```
midipulse seq64::perform::get_max_trigger ( ) [private]
```

Returns

Returns the highest trigger value, or zero. It is not clear why this function doesn't return a "no trigger found" value. Is there always at least one trigger, at 0?

13.69.4.154 collapse()

```
void seq64::perform::collapse ( ) [inline], [private]
```

13.69.4.155 copy()

```
void seq64::perform::copy ( ) [inline], [private]
```

13.69.4.156 expand()

```
void seq64::perform::expand ( ) [inline], [private]
```

13.69.4.157 midi_control_toggle()

Recall that the midi_control object specifies if a control is active, inversely-active, what status byte initiates it, what data byte initiates it, and the min/max values. Note that the status byte determines what category of event it is (e.g. note on/off versus a continuous controller), and the data byte indicates the note value or the type of continuous controller.

Parameters

ctl

Provides the index to pass to valid_midi_control_seq() to obtain a control value (such as c_midi_control_bpm_up) to use to retrieve the desired midi_control object.

Returns

Returns the "toggle" value if the control value is valid, or a reference to sm_mc_dummy otherwise.

13.69.4.158 midi_control_on()

Parameters

ctl

Provides the index to pass to valid_midi_control_seq() to obtain a control value (such as c_midi_control_bpm_up) to use to retrieve the desired midi_control object.

Returns

Returns the "on" value if the control value is valid, and a reference to sm_mc_dummy otherwise.

13.69.4.159 midi_control_off()

Parameters

ctl Provides a control value (such as c_midi_control_bpm_up) to use to retrieve the desired midi_control object.

Returns

Returns the "off" value if the control value is valid, and a reference to sm_mc_dummy otherwise.

```
13.69.4.160 midi_control_event()
```

Here is the processing involved in this function

Incorporates pull request #24, arnaud-jacquemin, issue #23 "MIDI controller toggles wrong pattern".

Change Note ca 2016-10-05 Issue #35. Changed "on" to "off".

QUESTIONS/TODO:

```
1. Why go above the sequence numbers, why not
just go up to c_midi_track_ctrl?
```

2. What about our new extended controls?

13.69.4.161 handle_midi_control()

```
void seq64::perform::handle_midi_control (
            int ctl,
            bool state ) [private]
       c_midi_control_mod_replace
       c_midi_control_mod_snapshot
       {\tt c\_midi\_control\_mod\_queue}
       c_midi_control_mod_gmute
       c_midi_control_mod_glearn
Other values supported:
       c_midi_control_bpm_up
       c_midi_control_bpm_dn
       c_midi_control_ss_up
       c_midi_control_ss_dn
       c_midi_control_play_ss
We have added the following extended values:
       c_midi_control_playback (for pause/toggle, start, and stop)
       c_midi_control_record
       c_midi_control_solo
                                  (for toggle, on, or off)
       c_midi_control_thru
       c_midi_control_bpm_page_up
       c_midi_control_bpm_page_dn
        The extended values will actually be handled by a new function,
handle_midi_control_ex().
c_midi_control_solo probably will need a parameter.
Values from 32 through 2 \times 32 are normalized by subtracting 32 and passed to
the select_and_mute_group() function. Otherwise, the following apply:
We also reserve a few control values above that for expansion.
```

ctl	The MIDI control value to use to perform an operation.
state	The state of the control, used with the following
	values:

13.69.4.162 handle_midi_control_ex()

Parameters

ctl	The MIDI control value to use to perform an operation.
а	The action of the control.

Returns

Returns true if the control was an extended control and was acted on.

13.69.4.163 get_screen_set_notepad()

Parameters

screenset	The ID number of the screen set, an index into the m_screen_set_notepad[] array. This value is
	validated.

Returns

Returns a reference to the desired string, or to an empty string if the screen-set number is invalid.

13.69.4.164 current_screen_set_notepad()

```
const std::string& seq64::perform::current_screen_set_notepad ( ) const [inline], [private]
```

stead of nice
stea

Parameters

note The string value to set into the notepad text.

```
13.69.4.167 set_playing_screenset()
```

```
void seq64::perform::set_playing_screenset ( ) [private]
```

This function is called when one of the snapshot keys is pressed.

For each value up to m_seqs_in_set (32), the index of the current sequence in the current screen set (m_playing _ _screen) is obtained. If the sequence is active and the sequence actually exists, it is processed; null sequences are no longer flagged as an error, they are just ignored.

Modifies m_playing_screen, m_playscreen_offset, stores the current playing-status of each sequence in m_tracks ← _mute_state[], and then calls mute_group_tracks(), turns on unmuted tracks in the current screen-set.

Basically, this function retrieves and saves the playing status of the sequences in the current play-screen, sets the play-screen to the current screen-set, and then mutes the previous play-screen. It is called via the c_midi_control _ play_ss value or via the set-playing-screen-set keystroke.

```
13.69.4.168 any_group_unmutes()
```

```
bool seg64::perform::any_group_unmutes ( ) const [private]
```

Returns

Returns true if there are any unmute statuses in the mute-group array. If they're all zero, we don't need to save them.

13.69.4.169 mute_group_tracks()

```
void seq64::perform::mute_group_tracks ( ) [private]
```

It loops through every screen-set. In each screen-set, it acts on each active sequence. If the active sequence is in the current "in-view" screen-set (m_screenset as opposed to m_playing_screen), and its m_track_mute_state[] is true, then the sequence is turned on, otherwise it is turned off.

Change Note tdeagan 2015-12-22 via git pull. Replaced m_playing_screen with m_screenset.

It seems to us that the for (g) clause should have g range from 0 to m max sets, not m seqs in set.

13.69.4.170 select_and_mute_group()

Called in perform and in mainwnd.

Parameters

group Provides the group number for the group to be muted.

13.69.4.171 set song mute()

The sequence::set_song_mute() and toggle_song_mute() functions do all the work, including mp-dirtying the sequence.

We've modified this function to call multe_all_tracks() and toggle_all_tracks() in order to consolidate the code and (cough cough) fix a bug in this functionality from the mainwand menu.

Parameters

op Provides the "flag" that indicates if this function is to set mute on, off, or to toggle the mute status.

13.69.4.172 set_mode_group_mute()

```
void seq64::perform::set_mode_group_mute ( ) [inline], [private]
```

```
13.69.4.173 unset_mode_group_mute()
```

```
void seq64::perform::unset_mode_group_mute ( ) [inline], [private]
```

13.69.4.174 select_group_mute()

```
void seq64::perform::select_group_mute (
    int mutegroup ) [private]
```

Then, no matter what, it makes the desired mute-group the selected mute-group. Compare to set_and_copy_
mute_group().

One thing to note is that, once saved, then, if used, it is applied to the current screen-set, even if it is not the screen-set whose playing status were saved.

Parameters

mutegroup	The number of the desired mute group, clamped to be between 0 and m_seqs_in_set-1.	
	Obviously, it is the set whose state is to be stored, if in group-learn mode.	

13.69.4.175 set_mode_group_learn()

```
void seq64::perform::set_mode_group_learn ( ) [private]
```

This function is called via a MIDI control c_midi_control_mod_glearn and via the group-learn keystroke.

```
13.69.4.176 unset_mode_group_learn()
```

```
void seq64::perform::unset_mode_group_learn ( ) [private]
```

Then unsets the group-learn mode flag. This function is called via a MIDI control c_midi_control_mod_glearn, via the group-learn keystroke, and in mainwnd::on_key_press_event(), to end the group-learn mode.

Shouldn't this function also call this one, to perfectly complement set_mode_group_learn: unset_mode_group_

mute(). Too tricky.

13.69.4.177 is_group_learning()

```
bool seq64::perform::is_group_learning ( ) [inline], [private]
```

13.69.4.178 set_and_copy_mute_group()

Then the mute-group is stored in m_tracks_mute_state[], which holds states for only the number of sequences in a set.

Compare to select_group_mute(); its main difference is that it will at least copy the states even if not in group-learn mode. And, if in group-learn mode, it will grab the playing states of the sequences before copying them.

This function is used only once, in select_and_mute_group(). It used to be called just select_mute_group(), but that's too easy to confuse with select_group_mute().

Change Note tdeagan 2015-12-22 via git pull: git pull https://github.com/TDeagan/sequencer64. ← git mute_groups m_screenset replaces m_playscreen_offset.

Parameters

mutegroup	Provides the mute-group to select.
-----------	------------------------------------

13.69.4.179 activate()

```
bool seq64::perform::activate ( ) [private]
```

Currently does work only for JACK; the activate() calls for other APIs just return true without doing anything.

13.69.4.180 start()

Parameters

songmode | If true, playback is to be in Song mode. Otherwise, it is to be in Live mode.

13.69.4.181 stop()

```
void seq64::perform::stop ( ) [private]
```

The logic seems backward here, in that we call inner_stop() if JACK is not running. Or perhaps we misunderstand the meaning of m_jack_running?

Stazed:

This function's sole purpose was to prevent inner_stop() from being called internally when JACK was running... potentially twice. inner_stop() was called by output_func() when JACK sent a JackTransportStopped message. If seq42 initiated the stop, then stop_jack() was called which then triggered the JackTransportStopped message to output_func() which then triggered the bool stop_jack to call inner_stop(). The output_func() call to inner_stop() is only necessary when some other JACK client sends a jack_transport_stop message to JACK, not when it is initiated by seq42. The method of relying on JACK to call $inner_stop()$ when internally initiated caused a (very) obscure apparent freeze if you press and hold the start/stop key if set to toggle. This occurs because of the delay between JackTransportStarting and JackTransportStopped if both triggered in rapid succession by holding the toggle key down. The variable global_is_running gets set false by a delayed inner_stop() from JACK after the start (true) is already sent. This means the global is set to true when JACK is actually off (false). Any subsequent presses to the toggle key send a stop message because the global is set to true. Because JACK is not running, output_func() is not running to send the inner_stop() call which resets the global to false. Thus an apparent freeze as the toggle key endlessly sends a stop, but $inner_stop()$ never gets called to reset. Whoo! So, to fix this we just need to call inner_stop() directly rather than wait for JACK to send a delayed stop, only when running. This makes the whole purpose of this stop() function unneeded. The check for m_jack_running is commented out and this function could be removed. It is being left for future generations to ponder!!!

13.69.4.182 start_jack()

```
void seq64::perform::start_jack ( ) [inline], [private]
```

13.69.4.183 stop_jack()

```
void seq64::perform::stop_jack ( ) [inline], [private]
```

13.69.4.184 position_jack()

Parameters

songmode	If true, playback is to be in Song mode. Otherwise, it is to be in Live mode.
tick	Provides the pulse position to be set. The default value is 0.

```
13.69.4.185 off_sequences()
void seq64::perform::off_sequences ( ) [private]
Replaces "for (int s = 0; s < m_sequence_max; ++s)"

13.69.4.186 all_notes_off()</pre>
```

void seq64::perform::all_notes_off () [private]

Then flush the master MIDI buss.

```
13.69.4.187 set_active()
```

If setting it active, the sequence::number() setter is called. It won't modify the sequence's internal copy of the sequence number if it has already been set.

Parameters

seq	Provides the prospective sequence number.
active	True if the sequence is to be set to the active state.

13.69.4.188 set_was_active()

Why do we need this routine?

Parameters

seq The pattern number. It is checked for validity.

13.69.4.189 reset_sequences()

Note that these calls are folded into one member function of the sequence class. Finally, flush the master MIDI buss.

pause Try to prevent notes from lingering on pause if true. By default, it is false.

13.69.4.190 play()

Starts the playing of all the patterns/sequences.

This function just runs down the list of sequences and has them dump their events. It skips sequences that have no playable MIDI events.

Note how often the "s" (sequence) pointer was used. It was worth offloading all these calls to a new sequence function. Hence the new sequence::play_queue() function.

Finally, we stop the looping at m_sequence_high rather than m_sequence_max, to save a little time.

Parameters

tick Provides the tick at which to start playing. This value is also copied to m_tick.

13.69.4.191 set_orig_ticks()

This is really the "last tick" value, so we renamed sequence::set_orig_tick() to sequence::set_last_tick().

Parameters

tick Provides the last-tick value to be set for each sequence that is active.

13.69.4.192 set_beats_per_minute()

Replaces perform::set_bpm() from seq24.

The value is set only if neither JACK nor this performance object are running.

It's not clear that we need to set the "is modified" flag just because we changed the beats per minute. This setting does get saved to the MIDI file, with the c_bpmtag.

bpm

Provides the beats/minute value to be set. It is clamped, if necessary, between the values SEQ64_MINIMUM_BPM to SEQ64_MAXIMUM_BPM. They provide a wide range of speeds, well beyond what normal music needs.

13.69.4.193 set_looping()

```
void seq64::perform::set_looping (
                bool looping ) [inline], [private]
```

Parameters

looping

The boolean value to set for looping, used in the performance editor.

13.69.4.194 max_active_set()

```
int seq64::perform::max_active_set ( ) const [private]
```

Returns

Returns the value of the highest active set. A value of 0 represents the first set. If no sequences are active, then -1 is returned.

13.69.4.195 launch_input_thread()

```
void seq64::perform::launch_input_thread ( ) [private]
```

This might be a good candidate for a small thread class derived from a small base class.

13.69.4.196 launch_output_thread()

```
void seq64::perform::launch_output_thread ( ) [private]
```

This might be a good candidate for a small thread class derived from a small base class.

13.69.4.197 init_jack_transport()

```
bool seq64::perform::init_jack_transport ( ) [inline], [private]
```

Who calls this routine? The main() routine of the application [via launch()], and the options module, when the Connect button is pressed.

Returns

Returns the result of the init() call; true if JACK sync is now running. If JACK support is not built into the application, then this function returns false, to indicate that JACK is (definitely) not running.

13.69.4.198 deinit_jack_transport()

```
bool seq64::perform::deinit_jack_transport ( ) [inline], [private]
```

Called by launch() and in the options module, when the Disconnect button is pressed.

Returns

Returns the result of the init() call; false if JACK sync is now no longer running. If JACK support is not built into the application, then this function returns true, to indicate that JACK is (definitely) not running.

13.69.4.199 seq_in_playing_screen()

Parameters

seq | Provides the index of the desired sequence.

Returns

Returns true if the sequence adheres to the conditions noted above.

13.69.4.200 is_modified() [2/2]

flag The value of the modified flag to be set.

13.69.4.201 valid_midi_control_seq()

```
bool seq64::perform::valid_midi_control_seq (
    int seq ) const [inline], [private]
```

We were checking against c_midi_track_ctrl as well, but that was a bug. This function is meant to check that the supplied sequence number does not exceed the value of c_midi_controls_extended (32 * 2 + 10 + 10 = 84). The track (sequence or pattern) controls rangoe from 0 to 64. Next come the "c_midi_control" values: bpm_up, bpm_dn, ..., play_ss, plus some extended controls that are relatively new, and, lastly, c_midi_controls_extended itself.

Parameters

seq

The sequence number value that should be inside the c_midi_controls_extended range. This value can specify not only a sequence number, but larger control values as well, so the function and parameter are mildly mis-named.

Returns

Returns true if the sequence number is valid for accessing the MIDI control values. For this function, no error print-out is generated.

13.69.4.202 is screenset valid()

```
bool seq64::perform::is_screenset_valid (
    int screenset ) const [inline], [private]
```

Parameters

screenset	The prospective screenset value.
-----------	----------------------------------

Returns

Returns true if the parameter is valid. For this function, no error print-out is generated.

13.69.4.203 set_running()

```
void seq64::perform::set_running (
                bool running ) [inline], [private]
```

running The value of the running flag to be set.

```
13.69.4.204 is_pattern_playing() [2/2]
```

13.69.4.205 set_playback_mode()

```
void seq64::perform::set_playback_mode (
    bool playbackmode ) [inline], [private]
```

Parameters

playbackmode The value of the playback mode flag to be set.

13.69.4.206 mute_group_offset()

Parameters

track The number of the desired track.

13.69.4.207 is_seq_valid()

Also see the function is_mseq_valid(), which also checks the pointer stored in the m_seq[] array.

We considered checking the *seq* param against sequence_count(), but this function is called while creating sequences that add to that count, so we continue checking against the "container" size. Also, it is possible to have holes in the array representing inactive sequences, so that sequencer_count() would be too limiting.

seq	The sequencer number, in interval [0, m_sequence_max).
-----	--

Returns

Returns true if the sequence number is valid.

13.69.4.208 is_mseq_valid()

It also evaluates the m_seq[seq] pointer value.

Note

Since we can have holes in the sequence array, where there are inactive sequences, we check if the sequence is even active before emitting a message about a null pointer for the sequence. We only want to see messages that indicate actual problems.

Parameters

seq

Provides the sequence number to be checked. It is checked for validity. We cannot compare the sequence number versus the sequence_count(), because the current implementation can have inactive holes (with null pointers) interspersed with active pointers.

Returns

Returns true if the sequence number is valid as per is_seq_valid(), and the sequence pointer is not null.

13.69.4.209 install_sequence()

It is common code and using it prevents inconsistences. It assumes values have already been checked. It does not set the "is modified" flag, since adding a sequence by loading a MIDI file should not set it. Compare new_\circ sequence(), used by mainwid and seqmenu, with add sequence(), used by midifile.

Parameters

seq The pointer to the pattern/sequence to add.	
seqnum	The sequence number of the pattern to be added. Not validated, to save some time.

Returns

Returns true if a sequence was removed, or the sequence was successfully added. In other words, if a real change in sequence pointers occurred. It is up to the caller to decide if the change warrants setting the "is modified" flag.

13.69.4.210 inner_start()

Then, if not is_running(), the playback mode is set to the given state. If that state is true, call off_sequences(). Set the running status, and signal the condition. Then unlock.

Minor issue:

```
In ALSA mode, restarting the sequence moves the progress bar to the beginning of the sequence, even if just pausing. This is fixed by compiling with SEQ64_PAUSE_SUPPORT, which disables calling off_sequences() when starting playback from the song editor / performance window.
```

Parameters

songmode	Sets the playback mode, and, if true, turns off all of the sequences before setting the is-running	
	condition.	

13.69.4.211 inner_stop()

```
void seq64::perform::inner_stop (
          bool midiclock = false ) [private]
```

Note that we do need to set the running flag to false here, even when JACK is running. Otherwise, JACK starts ping-ponging back and forth between positions under some circumstances.

However, if JACK is running, we do not want to reset the sequences... this causes the progress bar for each sequence to move to near the end of the sequence.

Parameters

```
midiclock If true, indicates that the MIDI clock should be used.
```

13.69.4.212 clamp_track()

Fixed the bug we found, where we checked for track > m_seqs_in_set, instead of using the >= operator.

Parameters

```
track The track value to be checked and rectified as necessary.
```

Returns

Returns the track parameter, clamped between 0 and m_seqs_in_set-1, inclusive.

13.69.4.213 set_key_event()

It is called 32 times, corresponding to the pattern/sequence slots in the Patterns window. It first removes the given key-code from the regular and reverse slot-maps. Then it removes the sequence-slot from the regular and reverse slot-maps. Finally, it adds the sequence-slot with a key value of key-code, and adds the key-code with a value of sequence-slot.

Parameters

keycode	The keycode for which to set the sequence slot.
sequence_slot	The sequence slot to be set.

13.69.4.214 set_key_group()

It is called 32 times, corresponding the pattern/sequence slots in the Patterns window. Compare it to the set_key ← _events() function.

Parameters

keycode	The keycode for which to set the group slot.
group_slot	The group slot to be set.

13.69.4.215 create_master_bus()

```
bool seq64::perform::create_master_bus ( ) [private]
```

We need to delay creation until launch time, so that settings can be obtained before determining just how to set up the application.

Once the master buss is created, we then copy the clocks and input setting that were read from the "rc" file, via the mastermidibus::port_settings() function, to use in determining whether to initialize and connect the input ports at start-up. Seq24 wouldn't connect unconditionally, and Sequencer64 shouldn't, either.

However, the devices actually on the system at start time might be different from what was saved in the "rc" file after the last run of Sequencer64.

Returns

Returns true if the creation succeeded.

13.69.4.216 add_clock()

Parameters

```
clocktype The clock value read from the "rc" file.
```

13.69.4.217 set_clock()

Mostly meant for use by the Options / MIDI Input tab.

13.69.4.218 add_input()

```
void seq64::perform::add_input (
                bool flag ) [inline], [private]
```

Parameters

```
flag The input flag read from the "rc" file.
```

13.69.4.219 set_input()

```
void seq64::perform::set_input (
```

```
int bus,
bool inputing ) [inline], [private]
```

Mostly meant for use by the Options / MIDI Input tab.

```
13.69.4.220 get_input()
```

```
bool seq64::perform::get_input (
                int bus ) [inline], [private]
```

13.69.4.221 is_input_system_port()

13.69.5 Friends And Related Function Documentation

13.69.5.1 jack_assistant

```
friend class jack_assistant [friend]
```

13.69.5.2 keybindentry

```
friend class keybindentry [friend]
```

13.69.5.3 mainwnd

friend class mainwnd [friend]

13.69.5.4 midifile

friend class midifile [friend]

13.69.5.5 optionsfile

```
friend class optionsfile [friend]
```

13.69.5.6 options

```
friend class options [friend]
```

13.69.5.7 perfedit

```
friend class perfedit [friend]
```

13.69.5.8 perfroll

```
friend class perfroll [friend]
```

13.69.5.9 input_thread_func

Parameters

myperf

Provides the perform object instance that is to be used. Its output_func() is called. Currently, this parameter is not validated, for speed.

Returns

Always returns nullptr.

13.69.5.10 output_thread_func

Set up the performance, set the process to realtime privileges, and then start the output function.

myperf	Provides the perform object instance that is to be used. Its output_func() is called. Currently, this
	parameter is not validated, for speed.

Returns

Always returns nullptr.

13.69.5.11 jack_sync_callback

```
int jack_sync_callback (
          jack_transport_state_t state,
          jack_position_t * pos,
          void * arg ) [friend]
```

This JACK synchronization callback informs the specified perform object of the current state and parameters of JACK.

The transport state will be:

JackTransportStopped when a new position is requested.
 JackTransportStarting when the transport is waiting to start.
 JackTransportRolling when the timeout has expired, and the position is now a moving target.

This is the slow-sync callback, which the stazed code replaces with jack_transport_callback().

Parameters

state	The JACK Transport state.
pos	The JACK position value.
arg	The pointer to the jack_assistant object. Currently not checked for nullity, nor dynamic-casted.

Returns

Returns 1 if the function works, and 0 if something was wrong.

13.69.5.12 jack_transport_callback

13.69.5.13 jack_shutdown

13.69.5.14 jack_timebase_callback

```
void jack_timebase_callback (
          jack_transport_state_t state,
          jack_nframes_t nframes,
          jack_position_t * pos,
          int new_pos,
          void * arg ) [friend]
```

The original version of the function worked properly with Hydrogen, but not with Klick. The new code seems to work with both. More testing and clarification is needed. This new code was "discovered" in the source-code for the "SooperLooper" project:

```
http://essej.net/sooperlooper/
```

The first difference with the new code is that it handles the case where the JACK position is moved (new_pos == true). If this is true, and the JackPositionBBT bit is off in pos->valid, then the new BBT value is set.

The seconds set of differences are in the "else" clause. In the new code, it is very simple: calculate the new tick value, back it off by the number of ticks in a beat, and perhaps go to the first beat of the next bar.

In the old code (complex!), the simple BBT adjustment is always made. This changes (perhaps) the beats_per_bar, beat_type, etc. We need to make these settings use the actual global values for beats set for Sequencer64. Then, if transitioning from JackTransportStarting to JackTransportRolling (instead of checking new_pos!), the BBT values (bar, beat, and tick) are finally adjusted. Here are the steps, with old and new steps noted:

- -# Calculate the "delta" ticks based on the current frame, the ticks_per_beat, the beats_per_minute, and the frame_rate. The old code saves this in a local, the new code assigns it to pos->tick.
- -# Old code: save this delta as a positive value.
- -# Figure out the settings and modify bar, beat, tick, and bar_start_tick. The old and new code seem to have the same intent, but it seems like the new code is faster and also correct.
 - Old code: Calculations are made by division and mod operations.
 - New code: Calculations are made by increments and decrements in a while loop.

Stazed:

The call to jack_timebase_callback() to supply JACK with BBT, etc. would occasionally fail when the pos information had zero or some garbage in the pos.frame_rate variable. This would occur when there was a rapid change of frame position by another client... i.e. qjackctl. From the JACK API:

pos address of the position structure for the next cycle; pos->frame will be its frame number. If new_pos is FALSE, this structure contains extended position information from the current cycle. If TRUE, it contains whatever was set by the requester. The timebase_callback's task is to update the extended information here."

The "If TRUE" line seems to be the issue. It seems that qjackctl does not always set pos.frame_rate so we get garbage and some strange BBT calculations that display in qjackctl. So we need to set it here and just use m_jack_frame_rate for calculations instead of pos.frame_rate.

Parameters

state	Indicates the current state of JACK transport.
nframes	The number of JACK frames in the current time period.
pos	Provides the position structure to be filled in, the address of the position structure for the next cycle; pos->frame will be its frame number. If new_pos is FALSE, this structure contains extended position information from the current cycle. If TRUE, it contains whatever was set by the requester. The timebase_callback's task is to update the extended information here.
new_pos	TRUE (non-zero) for a newly requested pos, or for the first cycle after the timebase_callback is defined. This is usually 0 in Sequencer64 at present, and 1 if one, say, presses "rewind" in qjackctl.
arg	Provides the jack_assistant pointer, currently unchecked for nullity.

13.69.5.15 get_current_jack_position

Warning

Currently valgrind flags j->client() as uninitialized.

13.69.6 Field Documentation

```
13.69.6.1 sm_mc_dummy
```

```
midi_control seq64::perform::sm_mc_dummy [static], [private]
```

Instantiate the dummy midi_control object, which is used in lieu of a null pointer.

We're taking code that basically works already, in the sense that it never seems to access a null pointer. So we're not even risking data transfers between this dummy object and the ones we really want to use.

However, it would be nice to be able to detect any errors that occur. How?

```
13.69.6.2 m_song_start_mode
```

```
bool seq64::perform::m_song_start_mode [private]
```

This is a replacement for the global setting, but is essentially a global setting itself, and is saved to and restored from the "rc" configuration file. Sometimes called "JACK start mode", it used to be a JACK setting, but now applies to any playback. Do not confuse this setting with m_playback_mode, which has a similar meaning but is more transitory. Probably, the concept needs some clean-up.

```
13.69.6.3 m_start_from_perfedit
```

```
bool seq64::perform::m_start_from_perfedit [private]
```

13.69.6.4 m_reposition

```
bool seq64::perform::m_reposition [private]
```

13.69.6.5 m_excell_FF_RW

```
float seq64::perform::m_excell_FF_RW [private]
```

It starts out at 1.0, and can range up to 60.0, being multiplied by 1.1 by the FF/RW timeout function.

13.69.6.6 m_FF_RW_button_type

```
ff_rw_button_t seq64::perform::m_FF_RW_button_type [private]
```

It has values of FF_RW_REWIND, FF_RW_NONE, or FF_RW_FORWARD. This was a free (global in a namespace) int in perfedit.

13.69.6.7 m_mute_group

```
bool seq64::perform::m_mute_group[c_max_sequence] [private]
```

This value determines whether a particular track will be muted or unmuted, and it can handle all tracks available in the application (currently c_max_sets * c_seqs_in_set, i.e. 1024). Note that the current state of playing can be "learned", and stored herein as the desired state for the track.

13.69.6.8 m_armed_saved

```
bool seq64::perform::m_armed_saved [private]
```

13.69.6.9 m_armed_statuses

```
bool seq64::perform::m_armed_statuses[c_max_sequence] [private]
```

13.69.6.10 m_tracks_mute_state

```
bool seq64::perform::m_tracks_mute_state[c_seqs_in_set] [private]
```

Unlike the m_mute_group[] array, this holds the current state, rather than the state desired by activating a mute group, and it applies to only one screen-set.

13.69.6.11 m_mode_group

```
bool seq64::perform::m_mode_group [private]
```

This value starts out true. It is altered by the c_midi_control_mod_gmute handler or when the keys().group_off() or the keys().group_on() keys are struck.

13.69.6.12 m_mode_group_learn

```
bool seq64::perform::m_mode_group_learn [private]
```

13.69.6.13 m_mute_group_selected

```
int seq64::perform::m_mute_group_selected [private]
```

It seems like a "group" is essentially a "set" that is selected for the saving and restoring of the status of all patterns in that set.

13.69.6.14 m_playing_screen

```
int seq64::perform::m_playing_screen [private]
```

In seq24, this value is altered by set_playing_screenset(), which is called by handle_midi_control(c_midi_control ← _play_ss, state).

13.69.6.15 m_playscreen_offset

```
int seq64::perform::m_playscreen_offset [private]
```

Saves some multiplications, should make the code easier to grok, and centralizes the use of c_seqs_in_set, which we want to be able to change at run-time, as a future enhancement.

13.69.6.16 m_seqs

```
sequence* seq64::perform::m_seqs[c_max_sequence] [private]
```

Todo First, make the sequence array a vector, and second, put allof these flags into a structure and access those members indirectly.

```
13.69.6.17 m_seqs_active
```

```
bool seq64::perform::m_seqs_active[c_max_sequence] [private]
```

This array can have "holes" with inactive sequences, so every sequence needs to be checked before using it.

13.69.6.18 m_was_active_main

```
bool seq64::perform::m_was_active_main[c_max_sequence] [private]
```

This value seems to be used only in maintaining dirtiness-status; did some process modify the sequence? Was it's mute/unmute status changed?

13.69.6.19 m_was_active_edit

```
bool seq64::perform::m_was_active_edit[c_max_sequence] [private]
```

This value seems to be used only in maintaining dirtiness-status for editing the mute/unmute status during pattern editing.

13.69.6.20 m_was_active_perf

```
bool seq64::perform::m_was_active_perf[c_max_sequence] [private]
```

This value seems to be used only in maintaining dirtiness-status for editing the mute/unmute status during performance/song editing.

13.69.6.21 m_was_active_names

```
bool seq64::perform::m_was_active_names[c_max_sequence] [private]
```

This value seems to be used only in maintaining dirtiness-status for editing the mute/unmute status during performance names editing. Not sure that it serves a real purpose; perhaps created with an eye to editing the pattern name in the song editor?

13.69.6.22 m_sequence_state

```
bool seq64::perform::m_sequence_state[c_max_sequence] [private]
```

13.69.6.23 m_transpose

```
int seq64::perform::m_transpose [private]
```

```
13.69.6.24 m_out_thread
```

```
pthread_t seq64::perform::m_out_thread [private]
```

Provides a "handle" to the output thread.

13.69.6.25 m_in_thread

```
pthread_t seq64::perform::m_in_thread [private]
```

13.69.6.26 m_out_thread_launched

```
bool seq64::perform::m_out_thread_launched [private]
```

13.69.6.27 m_in_thread_launched

```
bool seq64::perform::m_in_thread_launched [private]
```

13.69.6.28 m_running

```
bool seq64::perform::m_running [private]
```

However, this flag is conflated with some JACK support, and we have to supplement it with another flag, m_\leftarrow pattern_playing.

13.69.6.29 m_is_pattern_playing

```
bool seq64::perform::m_is_pattern_playing [private]
```

It replaces rc_settings :: is_pattern_playing(), which is gone, since the perform object is now visible to all classes that care about it.

13.69.6.30 m_inputing

```
bool seq64::perform::m_inputing [private]
```

```
13.69.6.31 m_outputing
bool seq64::perform::m_outputing [private]
13.69.6.32 m_looping
bool seq64::perform::m_looping [private]
If true, the performance will loop between the L and R markers in the performance editor.
13.69.6.33 m_playback_mode
bool seq64::perform::m_playback_mode [private]
There are two, "live" and "song", indicated by the following values:
        m_playback_mode == false: live mode
m_playback_mode == true: playback/song mode
13.69.6.34 m_ppqn
int seq64::perform::m_ppqn [private]
13.69.6.35 m_bpm
midibpm seq64::perform::m_bpm [private]
13.69.6.36 m_beats_per_bar
int seq64::perform::m_beats_per_bar [private]
The default value is SEQ64_DEFAULT_BEATS_PER_MEASURE (4).
13.69.6.37 m_beat_width
int seq64::perform::m_beat_width [private]
```

The default value is SEQ64_DEFAULT_BEAT_WIDTH (4).

```
13.69.6.38 m_clocks_per_metronome
```

```
int seq64::perform::m_clocks_per_metronome [private]
```

This value provides the number of MIDI clocks between metronome clicks. The default value of this item is 24. It can also be read from some SMF 1 files, such as our hymne.mid example.

```
13.69.6.39 m_32nds_per_quarter
```

```
int seq64::perform::m_32nds_per_quarter [private]
```

Useful in export. A duplicate of the same member in the sequence class.

```
13.69.6.40 m_us_per_quarter_note
```

```
long seq64::perform::m_us_per_quarter_note [private]
```

Useful in export. A duplicate of the same member in the sequence class.

```
13.69.6.41 m_master_bus
```

```
mastermidibus* seq64::perform::m_master_bus [private]
```

We changed this item to a pointer so that we can delay the creation of this object until after all settings have been read.

```
13.69.6.42 m_master_clocks
```

```
std::vector<clock_e> seq64::perform::m_master_clocks [private]
```

```
13.69.6.43 m_master_inputs
```

```
std::vector<bool> seq64::perform::m_master_inputs [private]
```

13.69.6.44 m_one_measure

```
midipulse seq64::perform::m_one_measure [private]
```

We can save some multiplications, and, more importantly, later define a more flexible definition of "one measure's worth" than simply four quarter notes.

```
13.69.6.45 m_left_tick
midipulse seq64::perform::m_left_tick [private]
Note that "tick" is actually "pulses".
13.69.6.46 m_right_tick
midipulse seq64::perform::m_right_tick [private]
Note that "tick" is actually "pulses".
13.69.6.47 m_starting_tick
midipulse seq64::perform::m_starting_tick [private]
By default, this value is always reset to the value of the "left tick". We want to eventually be able to leave it at the
last playing tick, to support a "pause" functionality. Note that "tick" is actually "pulses".
13.69.6.48 m tick
midipulse seq64::perform::m_tick [mutable], [private]
The m_tick member holds the tick to be used in displaying the progress bars and the maintime pill. It is mutable
because sometimes we want to adjust it in a const function for pause functionality.
13.69.6.49 m_jack_tick
midipulse seq64::perform::m_jack_tick [private]
13.69.6.50 m_usemidiclock
bool seq64::perform::m_usemidiclock [private]
13.69.6.51 m_midiclockrunning
bool seq64::perform::m_midiclockrunning [private]
```

```
13.69.6.52 m_midiclocktick
int seq64::perform::m_midiclocktick [private]
13.69.6.53 m_midiclockpos
int seq64::perform::m_midiclockpos [private]
13.69.6.54 m_dont_reset_ticks
bool seq64::perform::m_dont_reset_ticks [private]
All this member is used for is keeping the last tick from being reset.
13.69.6.55 m_screen_set_notepad
std::string seq64::perform::m_screen_set_notepad[c_max_sets] [private]
13.69.6.56 m_midi_cc_toggle
midi_control seq64::perform::m_midi_cc_toggle[c_midi_controls_extended] [private]
13.69.6.57 m_midi_cc_on
midi_control seq64::perform::m_midi_cc_on[c_midi_controls_extended] [private]
13.69.6.58 m_midi_cc_off
midi_control seq64::perform::m_midi_cc_off[c_midi_controls_extended] [private]
13.69.6.59 m_offset
int seq64::perform::m_offset [private]
```

It is used in the MIDI control of the playback status of the sequences in the current screen-set. It is also used to offset the sequence numbers so that the control (mute/unmute) keys can be shown on any screen-set.

13.69.6.60 m_control_status

```
int seq64::perform::m_control_status [private]
```

Need to learn more about this one. It is used in the replace, snapshot, and queue functionality.

13.69.6.61 m_screenset

```
int seq64::perform::m_screenset [private]
```

This is merely the screen-set that is in view. The fix of tdeagan substitutes the "in-view" screen-set for the "playing" screen-set.

13.69.6.62 m_seqs_in_set

```
int seq64::perform::m_seqs_in_set [private]
```

This change will require some arrays to be dynamically allocated (vectors).

13.69.6.63 m_max_sets

```
int seq64::perform::m_max_sets [private]
```

Again, currently set to the old value, which is used in hard-wired array sizes. To make it variable will require a move from arrays to vectors.

13.69.6.64 m_sequence_count

```
int seq64::perform::m_sequence_count [private]
```

Used by the install_sequence() function. Note that this value is not a suitable replacement for c_max_sequence/m

_sequence_max, because there can be inactive sequences amidst the active sequences. See the m_sequence

_limit member.

13.69.6.65 m_sequence_max

```
int seq64::perform::m_sequence_max [private]
```

However, this value is already 32 * 32 = 1024, and is probably enough for any usage. Famous last words?

13.69.6.66 m_sequence_high

```
int seq64::perform::m_sequence_high [private]
```

This value starts as 0, to indicate no sequences loaded, and then contains the highest sequence number hitherto loaded, plus 1 so that it can be used as a for-loop limit similar to m_sequence_max. It's maximum value should be m_sequence_max (c_max_sequence).

Currently meant only for limited context to try to squeeze a little extra speed out of playback. There's no easy way to lower this value when the highest sequence is deleted, though.

```
13.69.6.67 m_edit_sequence
```

```
int seq64::perform::m_edit_sequence [private]
```

Moving this status from segmenu into perform for better centralized management.

```
13.69.6.68 m_is_modified
```

```
bool seq64::perform::m_is_modified [private]
```

All the GUIs seem to use a perform object.

```
13.69.6.69 m_condition_var
```

```
condition_var seq64::perform::m_condition_var [private]
```

It is signalled if playback has been started. The output thread function waits on this variable until m_running and m_outputing are false. This variable is also signalled in the perform destructor.

```
13.69.6.70 m_jack_asst
```

```
jack_assistant seq64::perform::m_jack_asst [private]
```

It implements most of the JACK stuff.

```
13.69.6.71 m_have_undo
```

```
bool seq64::perform::m_have_undo [private]
```

```
13.69.6.72 m_undo_vect
```

```
std::vector<int> seq64::perform::m_undo_vect [private]
```

See the push_trigger_undo() function.

```
13.69.6.73 m_have_redo
```

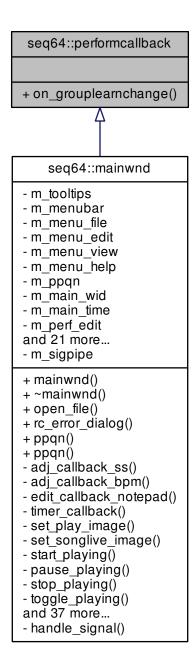
```
bool seq64::perform::m_have_redo [private]
```

```
13.69.6.74 m_redo_vect
std::vector<int> seq64::perform::m_redo_vect [private]
See the pop_trigger_undo() function.
13.69.6.75 m_notify
std::vector<performcallback *> seq64::perform::m_notify [private]
13.69.6.76 m_gui_support
gui_assistant& seq64::perform::m_gui_support [private]
```

13.70 seq64::performcallback Struct Reference

Provides for notification of events.

Inheritance diagram for seq64::performcallback:



Public Member Functions

virtual void on_grouplearnchange (bool)
 A do-nothing callback.

13.70.1 Detailed Description

Provide a response to a group-learn change event.

13.70.2 Member Function Documents	atio	<u>neni</u>	ocume	D	Function	Member	3.70.2	13
-----------------------------------	------	-------------	-------	---	----------	--------	--------	----

13.70.2.1 on_grouplearnchange()

"state" is an Unused parameter.

Reimplemented in seq64::mainwnd.

13.71 seq64::perfroll Class Reference

This class implements the performance roll user interface.

Inheritance diagram for seq64::perfroll:



Public Member Functions

perfroll (perform &perf, perfedit &parent, Gtk::Adjustment &hadjust, Gtk::Adjustment &vadjust, int ppqn=S←
 EQ64_USE_DEFAULT_PPQN)

Principal constructor.

• virtual \sim perfroll ()

This destructor deletes the interaction object.

· void set_guides (int snap, int measure, int beat)

This function sets the m_snap, m_measure_length, and m_beat_length members directly from the function parameters, which are in units of pulses (sometimes misleadingly called "ticks".)

· void update sizes ()

Updates the sizes of various items.

void init_before_show ()

Sets the roll-lengths ticks member.

void fill_background_pixmap ()

This function updates the background of the piano roll.

void increment_size ()

Increments the value of m_roll_length_ticks by the PPQN * 512, then calls update_sizes().

• void draw all ()

Provides a very common sequence of calls used in perfroll_input.

- void follow_progress ()
- void redraw_progress ()

Helper function to simplify the client call.

Private Member Functions

• void draw progress ()

Draws the progress line that shows where we are in the performance.

void redraw_dirty_sequences ()

Redraws patterns/sequences that have been modified.

void set_ppqn (int ppqn)

Handles changes to the PPQN value in one place.

void convert_xy (int x, int y, midipulse &ticks, int &seq)

Converts (x, y) coordinates on the piano roll to tick (pulse) and sequence numbers.

void convert_x (int x, midipulse &ticks)

Converts an x-coordinate to a tick-offset on the x axis.

void snap x (int &x)

This function performs a 'snap' action on x.

· void draw_sequence_on (int seqnum)

Draws the given pattern/sequence on the given drawable area.

void draw_background_on (int seqnum)

Draws the given pattern/sequence background on the given drawable area.

void draw_drawable_row (long y)

Not quite sure what this draws yet.

• void change horz ()

Changes the 4-bar horizontal offset member and queues up a draw operation.

void change_vert ()

Changes the vertical offset member and queues up a draw operation.

void split_trigger (int sequence, midipulse tick)

Splits a trigger, whatever that means.

void enqueue_draw ()

Wraps queue_draw() and forwards the call to the parent perfedit, so that it can forward it to any other perfedit that exists.

void set_zoom (int z)

Implements the horizontal zoom feature.

void convert_drop_xy ()

A convenience function.

void horizontal_adjust (double step)

This function provides optimization for the on_scroll_event() function.

void vertical adjust (double step)

This function provides optimization for the on_scroll_event() function.

void horizontal_set (double value)

Sets the exact position of a horizontal scroll-bar.

· void vertical set (double value)

Sets the exact position of a vertical scroll-bar.

• void on_realize ()

Provides the on-realization callback.

bool on expose event (GdkEventExpose *ev)

Handles the on-expose event.

bool on_button_press_event (GdkEventButton *ev)

This callback function handles a button press by forwarding it to the interaction object's button-press function.

bool on button release event (GdkEventButton *ev)

This callback function handles a button release by forwarding it to the interaction object's button-release function.

bool on_motion_notify_event (GdkEventMotion *ev)

Handles motion notification by forwarding it to the interaction object's motion-notification callback function.

• bool on scroll event (GdkEventScroll *ev)

Handles horizontal and vertical scrolling.

bool on_focus_in_event (GdkEventFocus *ev)

This callback handles an in-focus event by setting the flag to HAS_FOCUS.

bool on_focus_out_event (GdkEventFocus *ev)

This callback handles an out-of-focus event by resetting the flag HAS_FOCUS.

• void on_size_allocate (Gtk::Allocation &al)

Upon a size allocation event, this callback calls the base-class version of this function, then sets m_window_x and m_window_y, and calls update_sizes().

bool on_key_press_event (GdkEventKey *ev)

This callback function handles a key-press event.

void on size request (GtkRequisition *)

This do-nothing callback effectively throws away a size request.

Private Attributes

· perfedit & m parent

Provides a link to the perfedit that created this object.

int m_h_page_increment

Provides the horizontal page increment for the horizontal scrollbar.

• int m_v_page_increment

Provides the vertical page increment for the vertical scrollbar.

• int m_snap

The amount of horizontal snap.

• int m_ppqn

Parts-per-quarter-note value.

int m_page_factor

4096, horizonal page sizing.

· int m divs per beat

Holds current tick scaling value.

· midipulse m ticks per bar

Holds current bar scaling value.

int m_perf_scale_x

Scaling based on zoom and PPQN.

• int m zoom

New value to attempt a rudimentary time-zoom feature.

int m_names_y

The maximum height of the perfroll names box, in pixes.

int m background x

The width of the perfroll background.

· int m_size_box_w

This is a basically constant value set to s_perfroll_size_box_w = 3.

· int m measure length

The legnth of a measure, in beat units.

· int m_beat_length

The length of a beat, in parts-per-quarter note.

· midipulse m old progress ticks

Saves the position of the progress bar, for erasing it in preparation for drawing it at the next tick value.

· bool m have button press

Used in the fruity and seq24 perfroll input classes to help with trigger push/pop management.

bool m_transport_follow

Indicates that the application should follow JACK transport.

· bool m trans button press

Indicates if the follow-transport button is pressed.

· midipulse m 4bar offset

Holds the horizontal offset related to the horizontal scroll-bar position.

int m_sequence_offset

This value is the vertical version of m_4bar_offset.

int m_roll_length_ticks

Provides the width of the piano roll in ticks.

• midipulse m_drop_tick

The horizontal location for section movement.

midipulse m_drop_tick_trigger_offset

The horizontal trigger location for section movement.

int m_drop_sequence

Holds the currently-selected sequence being moved.

· int m sequence max

Currently, just a class-specific version of c_max_sequence, meant for the future.

• bool m_sequence_active [c_max_sequence]

Used when drawing an active sequence.

• FruityPerfInput m_fruity_interaction

We need both styles of interaction object present.

• Seq24PerfInput m_seq24_interaction

Provides support for standard Seq24 mouse handling, plus the keystroke handlers.

• AbstractPerfInput & m_interaction

Provides a reference to the selected (at startup time) method of mouse interaction.

bool m_moving

Used in the Seq24 or Fruity processing when moving a section of triggers.

• bool m_growing

Used in the Seq24 or Fruity processing when growing a section of triggers.

bool m_grow_direction

Used in the Seq24 or Fruity processing when growing a section of triggers.

Friends

class FruityPerfInput

These friend implement interaction-specific behavior, although only the Seq24 interactions support full keyboard processing, except for some common functionality provided by perform::perfroll_key_event().

- class Seq24PerfInput
- · class perfedit

Additional Inherited Members

13.71.1 Constructor & Destructor Documentation

```
13.71.1.1 perfroll()
```

```
13.71.1.2 \simperfroll()
```

```
seq64::perfroll::~perfroll ( ) [virtual]
```

Well, now there are two objects, so no explicit deletion necessary.

13.71.2 Member Function Documentation

13.71.2.1 set_guides()

This function then fills in the background, and queues up a draw operation.

Parameters

snap	Provides the number of snap-pulses (pulses per snap interval) as calculated in perfedit::set_guides (). This is actually equal to the measure-pulses divided by the snap value in perfedit; the snap value defaults to 8.
measure	Provides the number of measure-pulses (pulses per measure) as calculated in perfedit::set_guides().
beat	Provides the number of beat-pulses (pulses per beat) as calculated in perfedit::set_guides().

13.71.2.2 update_sizes()

```
void seq64::perfroll::update_sizes ( )
```

Note

Trying to figure out what the 16 is. So take the "bars-visible" calculation, the c_perf_scale_x value, assume that "ticks" is another name for "pulses", and assume that "beats" is a quarter note. Ignoring the numbers, the units come out to:

```
pixels * ticks / pixel
bars = ------
ticks / beat * beats / bar
```

Thus, the 16 is a "beats per bar" or "beats per measure" value. This doesn't quite make sense, but there are 16 divisions per beat on the perfroll user-interface. So for now we'll call it the latter, and make a variable called "m_divs_per_beat", see its definition in the class initializer list.

13.71.2.3 init_before_show()

```
void seq64::perfroll::init_before_show ( )
```

First, it gets the largest trigger value among the active sequences. Then it truncates this value to the nearest PPQN * 16 ticks. Then it adds PPQN * 4096 ticks.

13.71.2.4 fill_background_pixmap()

```
void seq64::perfroll::fill_background_pixmap ( )
```

The first thing done is to clear the background by painting it with a filled white rectangle.

This function is called whenever something occurs (e.g. zoom) that can affect how the piano roll is drawn.

void seq64::perfroll::increment_size ()

13.71.2.6 draw_all()

13.71.2.5 increment_size()

```
void seq64::perfroll::draw_all ( )
```

m_drop_y is adjusted by perfroll::change_vert() for any scroll after it was originally selected. The call below to draw_drawable_row() will have the wrong y location and un-select will not occur if the user scrolls the track up or down to a new y location, if not adjusted.

13.71.2.7 follow_progress()

```
void seq64::perfroll::follow_progress ( )
```

13.71.2.8 redraw_progress()

```
void seq64::perfroll::redraw_progress ( ) [inline]
```

13.71.2.9 draw_progress()

```
void seq64::perfroll::draw_progress ( ) [private]
```

We would like to be able to leave the line there when the progress is paused while running off of JACK transport. How? The perf().get_tick() call always returns 0 when stop is in force.

If we comment out the erasure of the old line, we see that the progress bar is also erased when a pattern boundary is hit (triggers), and when the sequence is stopped by the user.

In order to support true pause in the song editor, we tried to replace perform::get_tick() () [a new experimental function]. But those replacements here always return 0, even as perform::get_tick() () increases. Now were are trying a newer function, perform::get_max_tick(), which seems to do the trick for resuming (instead of rewinding) the progress bar. It's still a tiny bit laggy, so we have to find a faster way to get the maximum. (Note that the draw_progress function is called at every timeout, that is, constantly.)

The perform::get_max_tick() call doesn't work with JACK: the progress bar rewinds to the beginning when playback is paused, though it does resume where it left off. It also may cause the progress bar to backtrack through any gap. Let's restore the get_tick() call.

```
13.71.2.10 redraw_dirty_sequences()
```

```
void seq64::perfroll::redraw_dirty_sequences ( ) [private]
```

Change Note ca 2016-05-30 Lets try not drawing sequences greater than the maximum, at all.

13.71.2.11 set_ppqn()

The m_ticks_per_bar member replaces the global ppqn times 16. This construct is parts-per-quarter-note times 4 quarter notes times 4 sixteenth notes in a bar. (We think...)

The m_perf_scale_x member starts out at c_perf_scale_x, which is 32 ticks per pixel at the default tick rate of 192 PPQN. We adjust this now. But note that this calculation still involves the c_perf_scale_x constant.

Todo Resolve the issue of c_perf_scale_x versus m_perf_scale_x in perfroll.

13.71.2.12 convert_xy()

```
void seq64::perfroll::convert_xy (
          int x,
          int y,
          midipulse & d_tick,
          int & d_seq ) [private]
```

The results are returned via the d_tick and d_seq parameters. The sequence number is clipped to a legal value (0 to m_sequence_max).

Parameters

	X	The x coordinate of the mouse pointer.
	У	The y coordinate of the mouse pointer.
out	d_tick	Holds the calculated tick value.
out	d_seq	Holds the calculated sequence-number value.

13.71.2.13 convert_x()

The result is returned via the tick parameter. Note that m_4bar_offset already includes the m_ticks_per_bar = ppqn * 16 factor, for speed.

Parameters

	Х	The input x (pixel) value.
out	tick	Holds the result of the calculation.

```
13.71.2.14 snap_x()
```

- m_snap = number pulses to snap to
- m_perf_scale_x = number of pulses per pixel

Therefore mod = m_snap/m_perf_scale_x equals the number pixels to snap to.

```
13.71.2.15 draw_sequence_on()
```

Statement nesting from hell!

13.71.2.16 draw_background_on()

```
void seq64::perfroll::draw_background_on (
    int seqnum ) [private]
```

13.71.2.17 draw_drawable_row()

It is involved in the drawing of a greyed (selected) row.

What's weird is that we divide y by m_names_y, then multiply it by m_names_y, before passing the result to draw—drawable(). However, if we just use y casted to an int, then the drawing of the row is only partial, vertically.

13.71.2.18 change_horz()

```
void seq64::perfroll::change_horz ( ) [private]
```

Since the m_4bar_offset value is always multiplied by m_ticks_per_bar before usage, let's just do it here and not have to multiply it later.

```
13.71.2.19 change_vert()
```

```
void seq64::perfroll::change_vert ( ) [private]
```

Stazed:

Must adjust m_drop_y or perfroll_input's unselect_triggers() will not work if scrolled up or down to a new location. See the note in on_button_press_event() in the perfroll_input module. Also see the note in the draw_all() function.

13.71.2.20 split_trigger()

13.71.2.21 enqueue_draw()

```
void seq64::perfroll::enqueue_draw ( ) [private]
```

The parent perfedit will call perfroll::queue_draw() on behalf of this object, and it will pass a perfroll::enqueue_draw() to the peer perfedit's perfroll, if the peer exists.

13.71.2.22 set_zoom()

Change Note ca 2016-04-05 The initial zoom value is c_perf_scale_x (32). We allow it to range from 1 to 128, for now. Smaller values zoom in.

13.71.2.23 convert_drop_xy()

```
void seq64::perfroll::convert_drop_xy ( ) [inline], [private]
```

13.71.2.24 horizontal_adjust()

A duplicate of the one in segroll.

Parameters

step

Provides the step value to use for adjusting the horizontal scrollbar. See gui_drawingarea_gtk2::scroll_hadjust() for more information.

13.71.2.25 vertical_adjust()

A near-duplicate of the one in seqroll.

Parameters

step

Provides the step value to use for adjusting the vertical scrollbar. See gui_drawingarea_gtk2::scroll_vadjust() for more information.

13.71.2.26 horizontal_set()

Parameters

value

The desired position. Mostly this is either 0.0 or 9999999.0 (an "infinite" value to select the start or end position.

13.71.2.27 vertical_set()

Parameters

value

The desired position. Mostly this is either 0.0 or 9999999.0 (an "infinite" value to select the start or end position.

```
13.71.2.28 on_realize()
```

```
void seq64::perfroll::on_realize ( ) [private]
```

Calls the base-class version first.

Then it allocates the additional resources need, that couldn't be initialized in the constructor, and makes some connections.

Stazed:

```
This creation of m_background needs to be set to the max width for proper drawing of zoomed measures or they will get truncated with high beats per measure and low beat width. Since this is a constant size, it cannot be adjusted later for zoom. The constant c_perfroll_background_x is set to the max amount by default for use here. The drawing functions fill_background_pixmap() and draw_background_on() which use c_perfroll_background_x also, could be adjusted by zoom with a substituted variable. Not sure if there is any benefit to doing the adjustment... Perhaps a small benefit in speed? Maybe FIXME if really, really bored...
```

13.71.2.29 on expose event()

Draws a vertical page of the performance editor. The part drawn starts at m_sequence_offset and continues until the last sequence that can be at least partially seen given the height of the window.

If we're at the bottom of the sequences (1024, a non-existent sequence) would be the last sequence shown, we don't bother drawing it. This prevents debug messages about an illegal sequence, and can show a black bottom row that is a clear sign we're at the end of the legal sequences.

Parameters

ev Provides the expose event.

Returns

Always returns true.

13.71.2.30 on_button_press_event()

This gives us Seq24 versus Fruity behavior.

One minor issue: Fruity behavior doesn't yet provide the keystroke behavior we now handle for the Seq24 mode of operation.

13.71.2.31 on_button_release_event()

```
bool seq64::perfroll::on_button_release_event ( {\tt GdkEventButton} \ *\ ev\ ) \quad [{\tt private}]
```

This gives us Seq24 versus Fruity behavior.

13.71.2.32 on_motion_notify_event()

```
bool seq64::perfroll::on_motion_notify_event (
    GdkEventMotion * ev ) [private]
```

13.71.2.33 on_scroll_event()

If the Shift key is held while scrolling, then the scrolling is horizontal, otherwise it is vertical. This matches the convention of the seqedit class.

Note that, unlike the sequedit class, Ctrl-Scroll is not used to modify the zoom value. Rather than mess up legacy behavior, we will rely on keystrokes (z, 0, Z, and Ctrl-Page-Up and Ctrl-Page-Down) to implement this zoom.

Parameters

```
ev Provides the scroll event.
```

Returns

Currently always returns true.

13.71.2.34 on_focus_in_event()

13.71.2.35 on_focus_out_event()

13.71.2.36 on_size_allocate()

13.71.2.37 on_key_press_event()

```
bool seq64::perfroll::on_key_press_event (
        GdkEventKey * ev ) [private]
```

If we don't check the event type first, then the ev->keyval value is something weird like 65507. Note that we pass the functionality on to the perform::perfroll_key_event() function for the handling of delete, cut, copy, paste, and undo operations. If the keystroke is not handled by that function, then we handle it here.

Note that only the Seq24 input interaction object handles additional keystrokes not handled by the perfroll_key_ event() function.

The perfroll_key_event() call handles Del, Ctrl-X, Ctrl-V, and Ctrl-Z (which does nothing at present).

We've also added support for moving up and down in the piano roll (Up and Down arrows), paging up and down (Page-Up and Page-Down keys), paging left and right (Shift Page-Up and Page-Down), paging to top and bottom (Home and End), and paging to start and end (Shift Home and End).

The Keypad-End key is an issue on our ASUS "gaming" laptop. Whether it is seen as a "1" or an "End" key depends on an interaction between the Shift and the Num Lock key. Annoying, takes some time to get used to.

Stazed: there are many changes from seq32 that need to be studied before including them here. Note that, even though we filter out the Ctrl key here, it still works for Ctrl-X (cut) and Ctrl-V (paste). For undo, the Undo button can be used, Ctrl-Z never worked in this view anyway.

Warning

We see that 'x' and 'z' are already handled in perfroll_key_event() if the Ctrl key was pressed. Be careful.

13.71.2.38 on_size_request()

13.71.3 Friends And Related Function Documentation

13.71.3.1 FruityPerfInput

```
friend class FruityPerfInput [friend]
```

The perfedit class needs access to the private enqueue_draw() function.

13.71.3.2 Seq24PerfInput

```
friend class Seq24PerfInput [friend]
```

13.71.3.3 perfedit

```
friend class perfedit [friend]
```

13.71.4 Field Documentation

13.71.4.1 m_parent

```
perfedit& seq64::perfroll::m_parent [private]
```

We want to support two perfedit windows, but the children of perfedit will have to communicate changes requiring a redraw through the parent.

13.71.4.2 m_h_page_increment

```
int seq64::perfroll::m_h_page_increment [private]
```

It was set to 1, the same as the step increment. That is too little. This value will be set to 4, for now. Might be a useful "user" configuration option.

13.71.4.3 m_v_page_increment

```
int seq64::perfroll::m_v_page_increment [private]
```

It was set to 1, the same as the step increment. That is too little. This value will be set to 8, for now. Might be a useful "user" configuration option.

13.71.4.4 m_snap

```
int seq64::perfroll::m_snap [private]
```

13.71.4.5 m_ppqn

```
int seq64::perfroll::m_ppqn [private]
```

```
13.71.4.6 m_page_factor
int seq64::perfroll::m_page_factor [private]
13.71.4.7 m_divs_per_beat
int seq64::perfroll::m_divs_per_beat [private]
13.71.4.8 m_ticks_per_bar
midipulse seq64::perfroll::m_ticks_per_bar [private]
13.71.4.9 m_perf_scale_x
int seq64::perfroll::m_perf_scale_x [private]
13.71.4.10 m_zoom
int seq64::perfroll::m_zoom [private]
It seems to work pretty well now.
13.71.4.11 m_names_y
int seq64::perfroll::m_names_y [private]
This is currently semantically a constant set to c_names_y = 24.
13.71.4.12 m_background_x
int seq64::perfroll::m_background_x [private]
```

This is based on the m_ppqn value and the value of c_perf_scale_x (or is m_perf_scale_x preferable?)

```
13.71.4.13 m_size_box_w
```

```
int seq64::perfroll::m_size_box_w [private]
```

It is used in drawing the short lines of the small box that sits at the top-left and bottom-right corners of each segment in the pattern editor. These can be used to lengthen and shorten a section in the song editor. We will increase this size, perhaps double it, to make it easier to grab.

13.71.4.14 m_measure_length

```
int seq64::perfroll::m_measure_length [private]
```

13.71.4.15 m_beat_length

```
int seq64::perfroll::m_beat_length [private]
```

13.71.4.16 m_old_progress_ticks

```
midipulse seq64::perfroll::m_old_progress_ticks [private]
```

See the draw_progress() function. This could almost be static inside that function.

13.71.4.17 m_have_button_press

```
bool seq64::perfroll::m_have_button_press [private]
```

13.71.4.18 m_transport_follow

```
bool seq64::perfroll::m_transport_follow [private]
```

The alternative is ...?

13.71.4.19 m_trans_button_press

```
bool seq64::perfroll::m_trans_button_press [private]
```

```
13.71.4.20 m_4bar_offset
```

```
midipulse seq64::perfroll::m_4bar_offset [private]
```

Used in drawing the progress bar and the sequence events. Also used in convert_x() and convert_xy(). This used to be the offset in units of bar ticks, but now we use it as a full-fledged ticks value. See the change_horz() function.

```
13.71.4.21 m_sequence_offset
```

```
int seq64::perfroll::m_sequence_offset [private]
```

It is obtained or changed when the vertical scroll-bar moves. It is used for drawing the correct vertical window in the piano roll.

```
13.71.4.22 m_roll_length_ticks
```

```
int seq64::perfroll::m_roll_length_ticks [private]
```

Calculated in init_before_show() based on the maximum trigger found in the perform object, the ticks/bar, the $P \leftarrow PQN$, and the page factor. Also can be increased in size in the increment_size() function [tied to the Grow button]. Used in update_sizes().

```
13.71.4.23 m_drop_tick
```

```
midipulse seq64::perfroll::m_drop_tick [private]
```

Used only by the friend modules perfroll_input and fruityperfroll_input.

```
13.71.4.24 m_drop_tick_trigger_offset
```

```
midipulse seq64::perfroll::m_drop_tick_trigger_offset [private]
```

Used only by the friend modules perfroll_input and fruityperfroll_input.

```
13.71.4.25 m_drop_sequence
```

```
int seq64::perfroll::m_drop_sequence [private]
```

Used for redrawing the sequence.

```
13.71.4.26 m_sequence_max
```

```
int seq64::perfroll::m_sequence_max [private]
```

```
13.71.4.27 m_sequence_active
```

```
bool seq64::perfroll::m_sequence_active[c_max_sequence] [private]
```

Not sure yet why we can't just use the sequence's member function to access this status boolean.

13.71.4.28 m_fruity_interaction

```
FruityPerfInput seq64::perfroll::m_fruity_interaction [private]
```

Even if the user specifies the fruity interaction, the Seq24 interaction is still needed to handle our new keystroke support for the perfroll. We need both objects to exist all the time, similar to the Fruity/Seq24 roles in the seqroll object.

Obsolete AbstractPerfInput * m_interaction

```
13.71.4.29 m_seq24_interaction
```

```
Seq24PerfInput seq64::perfroll::m_seq24_interaction [private]
```

13.71.4.30 m_interaction

```
AbstractPerfInput& seq64::perfroll::m_interaction [private]
```

13.71.4.31 m_moving

```
bool seq64::perfroll::m_moving [private]
```

13.71.4.32 m_growing

```
bool seq64::perfroll::m_growing [private]
```

13.71.4.33 m_grow_direction

```
bool seq64::perfroll::m_grow_direction [private]
```

Determines whether the section is growing to the left or to the right.

13.72 seq64::perftime Class Reference

This class implements drawing the piano time at the top of the "performance window" (the "song editor"). Inheritance diagram for seq64::perftime:



Public Member Functions

perftime (perform &perf, perfedit &parent, Gtk::Adjustment &hadjust, int ppqn=SEQ64_USE_DEFAULT_P
 — PQN)

Principal constructor.

virtual ~perftime ()

Let's provide a do-nothing virtual destructor.

- · void reset ()
- void set scale (int scale)
- void set guides (int snap, int measure)

Sets the m_snap value and the m_measure_length members directly from the function parameters, which are in units of pulses (sometimes misleadingly called "ticks".)

• void increment_size ()

This function does nothing.

Private Member Functions

• void enqueue draw ()

Wraps queue_draw() and forwards the call to the parent perfedit, so that it can forward it to any other perfedit that exists.

void set_zoom (int z)

Implements the horizontal zoom feature.

void draw_background ()

Separated out the drawing done in on_expose_event(), so that it can be redone when the zoom changes.

- void draw_progress_on_window ()
- void change horz ()

Changes the m_4bar_offset and queues a draw operation.

void set_ppqn (int ppqn)

Handles changes to the PPQN value in one place.

long tick_to_pixel (midipulse tick)

Common calculation to convert a pulse/tick value to a perftime x value.

midipulse pixel_to_tick (long pixel)

The inverse of tick_to_pixel().

· int tick offset ()

Centralizes calculation of the tick offset of the time bar.

• void update_sizes ()

This function does nothing.

• int idle_progress ()

This function just returns true.

• void update_pixmap ()

This function does nothing.

void draw_pixmap_on_window ()

This function does nothing.

• void on_realize ()

Implements the on-realization event, then allocates some resources the could not be allocated in the constructor.

bool on expose event (GdkEventExpose *ev)

Implements the on-expose event.

bool on_button_press_event (GdkEventButton *ev)

Implement the button-press event to set the L and R ticks.

• void on_size_allocate (Gtk::Allocation &r)

Implements a size-allocation event.

bool on_button_release_event (GdkEventButton *)

This button-release handler does nothing.

bool key_press_event (GdkEventKey *ev)

This callback function handles a key-press event.

Private Attributes

· perfedit & m_parent

Provides a link to the perfedit that created this object.

· int m 4bar offset

Not yet sure exactly what this member represents.

int m_tick_offset

This member is m_4bar_offset times 16 times the current PPQN, to save some calculations and centralize this value.

• int m_ppqn

The current value of PPQN, which we are trying to get to work everywhere, when PPQN is changed from the global ppqn = 192.

• int m_snap

Snap value, starts out very small, equal to m_ppqn.

• int m_measure_length

Provides the length of a measure in pulses or ticks.

int m_left_marker_tick

Holds the current location of the left (L) marker when arrow movement is in force.

int m_right_marker_tick

Holds the current location of the right (R) marker when arrow movement is in force.

• int m_perf_scale_x

A class version of the global c_perf_scale_x factor.

· int m_timearea_y

A class version of the global c_timerarea_y factor.

Friends

class perfedit

Additional Inherited Members

13.72.1 Constructor & Destructor Documentation

13.72.1.1 perftime()

In the constructor you can only allocate colors; get_window() returns 0 because we have not been realized.

Note

Note that we still have to use a global constant in the base-class constructor; we cannot assign it to the corresponding member beforehand.

p	Provides a reference to the main performance object of the application.	
parent	Provides a reference to the object that contains this object, so that this object can tell the parent to	
	queue up a drawing operation.	
hadjust	Provides the horizontal scrollbar object needed so that perftime can respond to scrollbar	
	cursor/thumb movement.	
ppqn	An optional override of the default PPQN value for the application.	

13.72.1.2 ∼perftime()

```
virtual seq64::perftime::~perftime ( ) [inline], [virtual]
```

13.72.2 Member Function Documentation

13.72.2.1 reset()

```
void seq64::perftime::reset ( )
```

13.72.2.2 set_scale()

13.72.2.3 set_guides()

This function then fills in the background, and queues up a draw operation.

Parameters

snap	Provides the number of snap-pulses (pulses per snap interval) as calculated in perfedit::set_guides(). This is actually equal to the measure-pulses divided by the snap value in perfedit; the snap value defaults to 8.	
measure	Provides the number of measure-pulses (pulses per measure) as calculated in perfedit::set_guides().	

```
13.72.2.4 increment_size()

void seq64::perftime::increment_size ( ) [inline]

Compare it to perfroll::increment_size().

13.72.2.5 enqueue_draw()

void seq64::perftime::enqueue_draw ( ) [private]
```

The parent perfedit will call perftime::queue_draw() on behalf of this object, and it will pass a perftime::enqueue_ draw() to the peer perfedit's perftime, if the peer exists.

Redraws the background if the new zoom checked out.

Parameters

z Provides the zoom value, which is checked, and then copied into m_perf_scale_x.

```
13.72.2.7 draw_background()
```

```
void seq64::perftime::draw_background ( ) [private]
```

Note that m_measure_length == 0 will cause integer overflow.

```
13.72.2.8 draw_progress_on_window()
```

```
void seq64::perftime::draw_progress_on_window ( ) [private]
```

13.72.2.9 change_horz()

```
void seq64::perftime::change_horz ( ) [private]
```

Again, uses the constant, 16 [now offloaded to the new tick_offset() function.].

13.72.2.10 set_ppqn()

It also modifies m_snap, m_measure_length (but always for four measures!), and m_tick_offset.

Todo We need make the 4 constant variable per the number of beats (quarter-notes) per bar, and also at least make 16 (4x4) a meaningful manifest constant.

Parameters

13.72.2.11 tick_to_pixel()

Parameters

tick The horizontal tick value to convert to an x pixel value, based on tick-offset and the x-scale.

Returns

Returns the x-pixel representing the time location parameter.

13.72.2.12 pixel_to_tick()

Parameters

pixel The pixel value.

Returns

Returns the time value represented b the pixel.

```
13.72.2.13 tick_offset()
int seq64::perftime::tick_offset ( ) [inline], [private]
Returns
     Returns m_4bar_offset * 16 * m_ppqn.
13.72.2.14 update_sizes()
void seq64::perftime::update_sizes ( ) [inline], [private]
13.72.2.15 idle_progress()
int seq64::perftime::idle_progress ( ) [inline], [private]
13.72.2.16 update_pixmap()
void seq64::perftime::update_pixmap ( ) [inline], [private]
13.72.2.17 draw_pixmap_on_window()
void seq64::perftime::draw_pixmap_on_window ( ) [inline], [private]
13.72.2.18 on_realize()
void seq64::perftime::on_realize ( ) [private]
It is important to call the base-class version of this function.
The former work of this function is now done in base-class's on_realize() and in its constructor now.
        m_window = get_window();
        m_gc = Gdk::GC::create(m_window);
        m_window->clear();
```

```
set_size_request(10, m_timearea_y);
```

13.72.2.19 on_expose_event()

```
bool seq64::perftime::on_expose_event (
            GdkEventExpose * ev ) [private]
```

Redraws the background.

Note

The perfedit object is created early on. When brought on-screen from mainwnd (the main window), first, perftime::on realize() is called, then this event is called.

ev The expose event, not used.

Returns

Always returns true.

13.72.2.20 on_button_press_event()

```
bool seq64::perftime::on_button_press_event ( {\tt GdkEventButton} \ * \ p0 \ ) \quad [{\tt private}]
```

Added functionality to try to set the start-tick if ctrl-left-click is pressed.

Parameters

p0 The button event.

Returns

Always returns true.

Why is setting the start-tick disabled? We re-enable it and see if it works. To our surprise, it works, but it sticks between stop/pause and the next playback in the performance editor. We added a feature where stop sets the start-tick to the left tick (or the beginning tick).

13.72.2.21 on_size_allocate()

13.72.2.22 on_button_release_event()

"ev", The button event parameter, is not used.

Returns

Always returns false

13.72.2.23 key_press_event()

Can't get the keystroke events to be seen by perfroll or perftime here using the normal callback function for keystrokes, and not sure why. The perfedit object can call this function, and that call works, so the perfedit class, which does get keystrokes, calls this function to do the work.

This function uses the "I" key to activate the movement of the "L" marker with the arrow keys, by the interval of on snap value for each press. It also uses the "r" key to activate the movement of the "R" marker, and the "x" to deactivate either movement move.

Be aware that there is no visual feedback, as yet, that one is in the movement mode.

Also be aware the changing the name of this function from "key_press_event()" to "on_key_press_event()" will disrupt the process, causing keystrokes to not get here. Too tricky.

13.72.3 Friends And Related Function Documentation

13.72.3.1 perfedit

```
friend class perfedit [friend]
```

13.72.4 Field Documentation

13.72.4.1 m_parent

```
perfedit& seq64::perftime::m_parent [private]
```

We want to support two perfedit windows, but the children of perfedit will have to communicate changes requiring a redraw through the parent.

13.72.4.2 m_4bar_offset

```
int seq64::perftime::m_4bar_offset [private]
```

Also, why always 4/16 in the calculations of this value? Might be able to get rid of this member, though it's a bit tricky.

13.72.4.3 m_tick_offset

```
int seq64::perftime::m_tick_offset [private]
```

Why 16?

13.72.4.4 m_ppqn int seq64::perftime::m_ppqn [private] 13.72.4.5 m_snap int seq64::perftime::m_snap [private]

13.72.4.6 m_measure_length

```
int seq64::perftime::m_measure_length [private]
```

This value is $m_ppqn * 4$, though eventually we want to employ a more flexible representation of measure length. Supports perftime's keystroke processing.

```
13.72.4.7 m_left_marker_tick
```

```
int seq64::perftime::m_left_marker_tick [private]
```

Otherwise it is -1. Supports perftime's keystroke processing.

```
13.72.4.8 m_right_marker_tick
```

```
int seq64::perftime::m_right_marker_tick [private]
```

Otherwise it is -1. Supports perftime's keystroke processing.

13.72.4.9 m_perf_scale_x

```
int seq64::perftime::m_perf_scale_x [private]
```

13.72.4.10 m_timearea_y

```
int seq64::perftime::m_timearea_y [private]
```

13.73 seq64::midi_port_info::port_info_t Struct Reference

Hold the information for a single port.

Data Fields

· int m_client_number

The major buss number of the port.

std::string m_client_name

The system's name for the client.

· int m_port_number

The minor port number of the port.

• std::string m_port_name

The system's name for the port.

int m_queue_number

A number used in some APIs.

• bool m_is_input

Indicates an input port.

bool m_is_virtual

Indicates an manual/virtual port.

• bool m_is_system

Built-in port, almost always false.

13.73.1 Detailed Description

Except for the virtual-vs-normal status, this information is obtained by scanning the system at the startup time of the application.

13.73.2 Field Documentation

13.73.2.1 m_client_number

```
int seq64::midi_port_info::port_info_t::m_client_number
```

13.73.2.2 m_client_name

```
std::string seq64::midi_port_info::port_info_t::m_client_name
```

13.73.2.3 m_port_number

```
int seq64::midi_port_info::port_info_t::m_port_number
```

13.73.2.4 m_port_name

std::string seq64::midi_port_info::port_info_t::m_port_name

13.73.2.5 m_queue_number

int seq64::midi_port_info::port_info_t::m_queue_number

13.73.2.6 m_is_input

bool seq64::midi_port_info::port_info_t::m_is_input

13.73.2.7 m_is_virtual

bool seq64::midi_port_info::port_info_t::m_is_virtual

13.73.2.8 m_is_system

bool seq64::midi_port_info::port_info_t::m_is_system

13.74 seq64::rc_settings Class Reference

This class contains the options formerly named "global_xxxxxx".

Public Member Functions

rc_settings ()

Default constructor.

• rc_settings (const rc_settings &rhs)

Copy constructor.

rc_settings & operator= (const rc_settings &rhs)

Principal assignment operator.

std::string config filespec () const

Constructs the full path and file specification for the "rc" file based on whether or not the legacy Seq24 filenames are being used.

• std::string user_filespec () const

Constructs the full path and file specification for the "user" file based on whether or not the legacy Seq24 filenames are being used.

· void set defaults ()

Sets the default values.

• bool auto_option_save () const

'Getter' function for member m_auto_option_save

bool legacy_format () const

'Getter' function for member m_legacy_format

bool lash_support () const

'Getter' function for member m_lash_support

• bool allow_mod4_mode () const

'Getter' function for member m_allow_mod4_mode

• bool allow_snap_split () const

'Getter' function for member m_allow_snap_split

bool allow_click_edit () const

'Getter' function for member m_allow_click_edit

bool show_midi () const

 ${\it 'Getter' function for member m_show_midi}$

bool priority () const

'Getter' function for member m_priority

· bool stats () const

'Getter' function for member m_stats

• bool pass_sysex () const

'Getter' function for member m_pass_sysex

bool with_jack_transport () const

'Getter' function for member m_with_jack_transport

bool with_jack_master () const

'Getter' function for member m_with_jack_master

bool with_jack_master_cond () const

'Getter' function for member m_with_jack_master_cond

bool with_jack_midi () const

'Getter' function for member m_with_jack_midi

• bool with jack () const

'Getter' function for member m_with_jack_transport m_with_jack_master, and m_with_jack_master_cond, to save client code some trouble.

bool filter_by_channel () const

'Getter' function for member m_filter_by_channel

· bool manual alsa ports () const

'Getter' function for member m_manual_alsa_ports

bool reveal_alsa_ports () const

'Getter' function for member m_reveal_alsa_ports

• bool print_keys () const

'Getter' function for member m_print_keys

• bool device_ignore () const

'Getter' function for member m_device_ignore

• int device_ignore_num () const

'Getter' function for member m_device_ignore_num

· interaction method t interaction method () const

'Getter' function for member m_interaction_method

const std::string & filename () const

'Getter' function for member m filename

const std::string & jack_session_uuid () const

'Getter' function for member m_jack_session_uuid

const std::string & last_used_dir () const

'Getter' function for member m_last_used_dir

const std::string & config_directory () const

'Getter' function for member m_config_directory

• const std::string & config_filename () const

'Getter' function for member m_config_filename

· const std::string & user filename () const

'Getter' function for member m_user_filename

· const std::string & config filename alt () const

'Getter' function for member m config filename alt;

· const std::string & user filename alt () const

'Getter' function for member m_user_filename_alt

const std::string application_name () const

'Getter' function for member m application name

const std::string & app_client_name () const

'Getter' function for member m_app_client_name

Protected Member Functions

void auto_option_save (bool flag)

'Setter' function for member m_auto_option_save

void legacy_format (bool flag)

'Setter' function for member m_legacy_format

void lash support (bool flag)

'Setter' function for member m_lash_support

void allow_mod4_mode (bool flag)

'Setter' function for member m_allow_mod4_mode

void allow snap split (bool flag)

'Setter' function for member m_allow_snap_split

void allow_click_edit (bool flag)

'Setter' function for member m_allow_click_edit

void show_midi (bool flag)

'Setter' function for member m_show_midi

• void priority (bool flag)

'Setter' function for member m_priority

void stats (bool flag)

'Setter' function for member m stats

void pass_sysex (bool flag)

'Setter' function for member m_pass_sysex

void with_jack_transport (bool flag)

'Setter' function for member m_with_jack_transport

void with_jack_master (bool flag)

'Setter' function for member m_with_jack_master

void with jack master cond (bool flag)

'Setter' function for member m_with_jack_master_cond

void with_jack_midi (bool flag)

'Setter' function for member m_with_jack_midi

• void filter_by_channel (bool flag)

'Setter' function for member m_filter_by_channel

void manual_alsa_ports (bool flag)

'Setter' function for member m_manual_alsa_ports

void reveal_alsa_ports (bool flag)

'Setter' function for member m_reveal_alsa_ports

void print_keys (bool flag)

'Setter' function for member m_print_keys

void device_ignore (bool flag)

'Setter' function for member m_device_ignore

void device_ignore_num (int value)

'Setter' function for member m_device_ignore_num However, please note that this value, while set in the options processing of the main module, does not appear to be used anywhere in the code in seq24, Sequencer24, and this application.

void interaction_method (interaction_method_t value)

'Setter' function for member m_interaction_method

void filename (const std::string &value)

'Setter' function for member m filename

void jack session uuid (const std::string &value)

'Setter' function for member m_jack_session_uuid

void last_used_dir (const std::string &value)

'Setter' function for member m_last_used_dir

void config_directory (const std::string &value)

'Setter' function for member m_config_directory

void set_config_files (const std::string &value)

'Setter' function for member m_config_filename and m_user_filename

· void config_filename (const std::string &value)

'Setter' function for member m_config_filename ("rc")

void user_filename (const std::string &value)

'Setter' function for member m_user_filename ("usr")

void config_filename_alt (const std::string &value)

'Setter' function for member m_config_filename_alt

void user_filename_alt (const std::string &value)

'Setter' function for member m_user_filename_alt

Private Member Functions

• std::string home_config_directory () const

Provides the directory for the configuration file, and also creates the directory if necessary.

Private Attributes

• bool m_auto_option_save

[auto-option-save] setting.

· bool m_legacy_format

Write files in legacy format.

• bool m_lash_support

Enable LASH, if compiled in.

· bool m allow mod4 mode

Allow Mod4 to hold drawing mode.

• bool m_allow_snap_split

Allow snap-split of a trigger.

bool m_allow_click_edit

Allow double-click edit pattern.

• bool m_show_midi

Show MIDI events to console.

bool m_priority

Run at high priority (Linux only).

• bool m_stats

Show some output statistics.

bool m_pass_sysex

Pass SysEx to outputs, not ready.

• bool m_with_jack_transport

Enable synchrony with JACK.

bool m_with_jack_master

Serve as a JACK transport Master.

bool m_with_jack_master_cond

Serve as JACK Master if possible.

bool m_with_jack_midi

Use JACK MIDI.

· bool m filter by channel

Record only sequence channel data.

bool m_manual_alsa_ports

[manual-alsa-ports] setting.

bool m_reveal_alsa_ports

[reveal-alsa-ports] setting.

bool m_print_keys

Show hot-key in main window slot.

bool m_device_ignore

From seq24 module, unused!

• int m_device_ignore_num

From seq24 module, unused!

interaction_method_t m_interaction_method

[interaction-method]

• std::string m filename

Provides the name of current MIDI file.

std::string m_jack_session_uuid

Holds the JACK UUID value that makes this JACK connection unique.

· std::string m last used dir

Holds the directory from which the last MIDI file was opened (or saved).

• std::string m_config_directory

Holds the current "rc" and "user" configuration directory.

• std::string m_config_filename

Holds the current "rc" configuration filename.

• std::string m_user_filename

Holds the current "user" configuration filename.

• std::string m_config_filename_alt

Holds the legacy "rc" filename, ".seq24rc".

• std::string m_user_filename_alt

Holds the legacy "user" filename, ".seq24usr".

· const std::string m_application_name

Holds the application name, e.g.

std::string m_app_client_name

Holds the client name for the application.

Friends

- · class optionsfile
- · class options
- · class mainwnd
- · class rtmidi info
- int parse_command_line_options (perform &p, int argc, char *argv [])

Parses the command-line options on behalf of the application.

• bool help_check (int argc, char *argv [])

Checks to see if the first option is a help or version argument, just so we can skip the "Reading configuration ..." messages.

13.74.1 Detailed Description

It gives us a whole lot more encapsulation and control over how the options of the "rc" file (optionsfile) are set and used. Note that this class does not support the hot-keys options; those are handled in the keys_perform class.

13.74.2 Constructor & Destructor Documentation

rhs The source of the data for the copy.

13.74.3 Member Function Documentation

13.74.3.1 operator=()

Parameters

rhs The source of the data for the assignment.

Returns

Returns a reference to the destination for use in serial assignments.

13.74.3.2 config_filespec()

```
\verb|std::string| seq64::rc\_settings::config_filespec ( ) const
```

Returns

If home_config_directory() returns a non-empty string, then the legacy or normal "rc" configuration file-name is appended to that result, and returned. Otherwise, an empty string is returned.

13.74.3.3 user_filespec()

```
std::string seq64::rc_settings::user_filespec ( ) const
```

Returns

If home_config_directory() returns a non-empty string, then the legacy or normal "user" configuration file-name is appended to that result, and returned. Otherwise, an empty string is returned.

```
13.74.3.4 set_defaults()
void seq64::rc_settings::set_defaults ( )
13.74.3.5 auto_option_save() [1/2]
bool seq64::rc_settings::auto_option_save ( ) const [inline]
13.74.3.6 legacy_format() [1/2]
bool seq64::rc_settings::legacy_format ( ) const [inline]
13.74.3.7 lash_support() [1/2]
bool seq64::rc_settings::lash_support ( ) const [inline]
13.74.3.8 allow_mod4_mode() [1/2]
\verb|bool seq64::rc_settings::allow_mod4_mode ( ) const [inline]|\\
13.74.3.9 allow_snap_split() [1/2]
bool seq64::rc_settings::allow_snap_split ( ) const [inline]
13.74.3.10 allow_click_edit() [1/2]
bool seq64::rc_settings::allow_click_edit ( ) const [inline]
13.74.3.11 show_midi() [1/2]
bool seq64::rc_settings::show_midi ( ) const [inline]
```

```
13.74.3.12 priority() [1/2]
bool seq64::rc_settings::priority ( ) const [inline]
13.74.3.13 stats() [1/2]
bool seq64::rc_settings::stats ( ) const [inline]
13.74.3.14 pass_sysex() [1/2]
bool seq64::rc_settings::pass_sysex ( ) const [inline]
13.74.3.15 with_jack_transport() [1/2]
bool seq64::rc_settings::with_jack_transport ( ) const [inline]
13.74.3.16 with_jack_master() [1/2]
bool seq64::rc_settings::with_jack_master ( ) const [inline]
13.74.3.17 with_jack_master_cond() [1/2]
bool seq64::rc_settings::with_jack_master_cond ( ) const [inline]
13.74.3.18 with_jack_midi() [1/2]
bool seq64::rc_settings::with_jack_midi ( ) const [inline]
13.74.3.19 with_jack()
bool seq64::rc_settings::with_jack ( ) const [inline]
```

Do not confuse these original options with the new "no JACK MIDI" option.

```
13.74.3.20 filter_by_channel() [1/2]
bool seq64::rc_settings::filter_by_channel ( ) const [inline]
13.74.3.21 manual_alsa_ports() [1/2]
bool seq64::rc_settings::manual_alsa_ports ( ) const [inline]
13.74.3.22 reveal_alsa_ports() [1/2]
bool seq64::rc_settings::reveal_alsa_ports ( ) const [inline]
13.74.3.23 print_keys() [1/2]
bool seq64::rc_settings::print_keys ( ) const [inline]
13.74.3.24 device_ignore() [1/2]
bool seq64::rc_settings::device_ignore ( ) const [inline]
13.74.3.25 device_ignore_num() [1/2]
int seq64::rc_settings::device_ignore_num ( ) const [inline]
13.74.3.26 interaction_method() [1/2]
interaction_method_t seq64::rc_settings::interaction_method ( ) const [inline]
13.74.3.27 filename() [1/2]
const std::string& seq64::rc_settings::filename ( ) const [inline]
```

```
13.74.3.28 jack_session_uuid() [1/2]
const std::string& seq64::rc_settings::jack_session_uuid ( ) const [inline]
13.74.3.29 last_used_dir() [1/2]
const std::string& seq64::rc_settings::last_used_dir ( ) const [inline]
13.74.3.30 config_directory() [1/2]
const std::string& seq64::rc_settings::config_directory ( ) const [inline]
13.74.3.31 config_filename() [1/2]
const std::string& seq64::rc_settings::config_filename ( ) const [inline]
13.74.3.32 user_filename() [1/2]
const std::string& seq64::rc_settings::user_filename ( ) const [inline]
13.74.3.33 config_filename_alt() [1/2]
const std::string& seq64::rc_settings::config_filename_alt ( ) const [inline]
13.74.3.34 user_filename_alt() [1/2]
const std::string& seq64::rc_settings::user_filename_alt ( ) const [inline]
13.74.3.35 application_name()
const std::string seq64::rc_settings::application_name ( ) const [inline]
```

```
13.74.3.36 app_client_name()
const std::string& seq64::rc_settings::app_client_name ( ) const [inline]
13.74.3.37 auto_option_save() [2/2]
void seq64::rc\_settings::auto\_option\_save (
            bool flag ) [inline], [protected]
13.74.3.38 legacy_format() [2/2]
void seq64::rc_settings::legacy_format (
            bool flag ) [inline], [protected]
13.74.3.39 lash_support() [2/2]
void seq64::rc_settings::lash_support (
             bool flag ) [inline], [protected]
13.74.3.40 allow_mod4_mode() [2/2]
void seq64::rc_settings::allow_mod4_mode (
             bool flag ) [inline], [protected]
13.74.3.41 allow_snap_split() [2/2]
void seq64::rc_settings::allow_snap_split (
            bool flag ) [inline], [protected]
13.74.3.42 allow_click_edit() [2/2]
void seq64::rc_settings::allow_click_edit (
             bool flag ) [inline], [protected]
```

```
13.74.3.43 show_midi() [2/2]
void seq64::rc_settings::show_midi (
            bool flag ) [inline], [protected]
13.74.3.44 priority() [2/2]
void seq64::rc_settings::priority (
             bool flag ) [inline], [protected]
13.74.3.45 stats() [2/2]
void seq64::rc_settings::stats (
            bool flag ) [inline], [protected]
13.74.3.46 pass_sysex() [2/2]
void seq64::rc_settings::pass_sysex (
             bool flag ) [inline], [protected]
13.74.3.47 with_jack_transport() [2/2]
void seq64::rc_settings::with_jack_transport (
             bool flag ) [protected]
13.74.3.48 with_jack_master() [2/2]
void seq64::rc_settings::with_jack_master (
            bool flag ) [protected]
13.74.3.49 with_jack_master_cond() [2/2]
void seq64::rc\_settings::with\_jack\_master\_cond (
             bool flag ) [protected]
```

```
13.74.3.50 with_jack_midi() [2/2]
void seq64::rc_settings::with_jack_midi (
            bool flag ) [inline], [protected]
13.74.3.51 filter_by_channel() [2/2]
void seq64::rc_settings::filter_by_channel (
            bool flag ) [inline], [protected]
13.74.3.52 manual_alsa_ports() [2/2]
void seq64::rc\_settings::manual\_alsa\_ports (
             bool flag ) [inline], [protected]
13.74.3.53 reveal_alsa_ports() [2/2]
void seq64::rc_settings::reveal_alsa_ports (
             bool flag ) [inline], [protected]
13.74.3.54 print_keys() [2/2]
void seq64::rc_settings::print_keys (
             bool flag ) [inline], [protected]
13.74.3.55 device_ignore() [2/2]
void seq64::rc_settings::device_ignore (
             bool flag ) [inline], [protected]
13.74.3.56 device_ignore_num() [2/2]
void seq64::rc_settings::device_ignore_num (
             int value ) [protected]
```

value The value to use to make the setting.

```
13.74.3.57 interaction_method() [2/2]
```

Parameters

value The value to use to make the setting.

```
13.74.3.58 filename() [2/2]
```

Parameters

value The value to use to make the setting.

```
13.74.3.59 jack_session_uuid() [2/2]
```

Parameters

value The value to use to make the setting.

```
13.74.3.60 last_used_dir() [2/2]
```

value The value to use to make the setting.

13.74.3.61 config_directory() [2/2]

Parameters

value The value to use to make the setting.

13.74.3.62 set_config_files()

Implements the -config option to change both configuration files ("rc" and "usr") with one option.

Parameters

value

The value to use to make the setting, if the string is not empty. If the value has an extension, it is stripped first.

13.74.3.63 config_filename() [2/2]

Parameters

value

The value to use to make the setting, if the string is not empty. If there is no period in the string, then ".rc" is appended to the end of the filename.

13.74.3.64 user_filename() [2/2]

value

The value to use to make the setting, if the string is not empty. If there is no period in the string, then ".usr" is appended to the end of the filename.

```
13.74.3.65 config_filename_alt() [2/2]
```

Parameters

value The value to use to make the setting, if the string is not empty.

13.74.3.66 user_filename_alt() [2/2]

Parameters

value The value to use to make the setting.

13.74.3.67 home_config_directory()

```
std::string seq64::rc_settings::home_config_directory ( ) const [private]
```

If the legacy format is in force, then the home directory for the configuration is (in Linux) "/home/username", and the configuration file is ".seq24rc".

If the new format is in force, then the home directory is (in Linux) "/home/username/.config/sequencer64", and the configuration file is "sequencer64.rc".

Returns

Returns the selected home configuration directory. If it does not exist, or could not be created, then an empty string is returned.

13.74.4 Friends And Related Function Documentation

13.74.4.1 optionsfile

```
friend class optionsfile [friend]
```

13.74.4.2 options

```
friend class options [friend]
```

13.74.4.3 mainwnd

```
friend class mainwnd [friend]
```

13.74.4.4 rtmidi_info

```
friend class rtmidi_info [friend]
```

13.74.4.5 parse_command_line_options

Note that, since we call this function twice (once before the configuration files are parsed, and once after), we have to make sure that the global value optind is reset to 0 before calling this function. Note that the traditional reset value for optind is 1, but 0 is used in GNU code to trigger the internal initialization routine of get_opt().

Parameters

р	The performance object that implements some of the command-line options.
argc	The number of command-line arguments.
argv	The array of command-line argument pointers.

Returns

Returns the value of optind if no help-related options were provided.

13.74.4.6 help_check

Also check for the –legacy option. Finally, it also checks for the "?" option that people sometimes use as a guess to get help.

Parameters

argo	The number of command-line arguments.
argv	The array of command-line argument pointers.

Returns

Returns true only if -v, -V, -version, -h, -help, or "?" were encountered. If the legacy options occurred, then rc().legacy_format(true) is called, as a side effect, because it will be needed before we parse the options.

13.74.5 Field Documentation

13.74.5.1 m_auto_option_save

```
bool seq64::rc_settings::m_auto_option_save [private]
```

13.74.5.2 m_legacy_format

```
bool seq64::rc_settings::m_legacy_format [private]
```

13.74.5.3 m_lash_support

```
bool seq64::rc_settings::m_lash_support [private]
```

13.74.5.4 m_allow_mod4_mode

```
bool seq64::rc_settings::m_allow_mod4_mode [private]
```

```
13.74.5.5 m_allow_snap_split
bool seq64::rc_settings::m_allow_snap_split [private]
13.74.5.6 m_allow_click_edit
bool seq64::rc_settings::m_allow_click_edit [private]
13.74.5.7 m_show_midi
bool seq64::rc_settings::m_show_midi [private]
13.74.5.8 m_priority
bool seq64::rc_settings::m_priority [private]
13.74.5.9 m_stats
bool seq64::rc_settings::m_stats [private]
13.74.5.10 m_pass_sysex
bool seq64::rc_settings::m_pass_sysex [private]
13.74.5.11 m_with_jack_transport
bool seq64::rc_settings::m_with_jack_transport [private]
13.74.5.12 m_with_jack_master
```

bool seq64::rc_settings::m_with_jack_master [private]

13.74.5.13 m_with_jack_master_cond

bool seq64::rc_settings::m_with_jack_master_cond [private]

13.74.5.14 m_with_jack_midi

bool seq64::rc_settings::m_with_jack_midi [private]

13.74.5.15 m_filter_by_channel

bool seq64::rc_settings::m_filter_by_channel [private]

13.74.5.16 m_manual_alsa_ports

bool seq64::rc_settings::m_manual_alsa_ports [private]

13.74.5.17 m_reveal_alsa_ports

bool seq64::rc_settings::m_reveal_alsa_ports [private]

13.74.5.18 m_print_keys

bool seq64::rc_settings::m_print_keys [private]

13.74.5.19 m_device_ignore

bool seq64::rc_settings::m_device_ignore [private]

13.74.5.20 m_device_ignore_num

int seq64::rc_settings::m_device_ignore_num [private]

```
13.74.5.21 m_interaction_method
interaction_method_t seq64::rc_settings::m_interaction_method [private]
13.74.5.22 m_filename
std::string seq64::rc_settings::m_filename [private]
13.74.5.23 m_jack_session_uuid
std::string seq64::rc_settings::m_jack_session_uuid [private]
13.74.5.24 m_last_used_dir
std::string seq64::rc_settings::m_last_used_dir [private]
13.74.5.25 m_config_directory
std::string seq64::rc_settings::m_config_directory [private]
This value is "\sim/.config/sequencer64" by default.
13.74.5.26 m_config_filename
std::string seq64::rc_settings::m_config_filename [private]
This value is "sequencer64.rc" by default.
13.74.5.27 m_user_filename
std::string seq64::rc_settings::m_user_filename [private]
This value is "sequencer64.usr" by default.
```

13.74.5.28 m_config_filename_alt

std::string seq64::rc_settings::m_config_filename_alt [private]

13.74.5.29 m_user_filename_alt

```
std::string seq64::rc_settings::m_user_filename_alt [private]
```

13.74.5.30 m_application_name

```
const std::string seq64::rc_settings::m_application_name [private]
```

"sequencer64", "seq64portmidi", or "seq64". This is a constant, set to SEQ64_APP_NAME. Also see the seq_← app_name() function.

13.74.5.31 m_app_client_name

```
std::string seq64::rc_settings::m_app_client_name [private]
```

This is much like the application name, but in the future will be a configuration option. For now it is just the value of the SEQ64_CLIENT_NAME macro. Also see the seq_client_name() function.

13.75 seq64::rect Class Reference

A small helper class representing a rectangle.

Data Fields

• int x

The x-coordinate of the origin of the rectangle.

• int y

The y-coordinate of the origin of the rectangle.

· int height

The height of the rectangle, in units of pixels.

· int width

The width of the rectangle, in units of pixels.

13.75.1 Field Documentation

13.75.1.1 x

int seq64::rect::x

13.75.1.2 y int seq64::rect::y 13.75.1.3 height int seq64::rect::height 13.75.1.4 width int seq64::rect::width

13.76 seq64::gui_drawingarea_gtk2::rect Struct Reference

A small helper structure representing a rectangle.

Data Fields

- int x
- int y
- · int height
- int width

13.76.1 Field Documentation

```
13.76.1.1 x
int seq64::gui_drawingarea_gtk2::rect::x

13.76.1.2 y
int seq64::gui_drawingarea_gtk2::rect::y
```

13.76.1.3 height

```
int seq64::gui_drawingarea_gtk2::rect::height
```

13.76.1.4 width

int seq64::gui_drawingarea_gtk2::rect::width

13.77 seq64::rterror Class Reference

Exception handling class for rtexmidi.

Inherits exception.

Public Types

enum Type {
 WARNING,
 DEBUG_WARNING,
 UNSPECIFIED,
 NO_DEVICES_FOUND,
 INVALID_DEVICE,
 MEMORY_ERROR,
 INVALID_PARAMETER,
 INVALID_USE,
 DRIVER_ERROR,
 SYSTEM_ERROR,
 THREAD_ERROR }

Public Member Functions

- rterror (const std::string &message, Type type=rterror::UNSPECIFIED)
- virtual ~rterror ()
- virtual void print_message () const

Prints thrown error message to stderr.

virtual const Type & getType () const

Returns the thrown error message type.

virtual const std::string & get_message () const

Returns the thrown error message string.

• virtual const char * what () const noexcept

Returns the thrown error message as a c-style string.

Private Attributes

• std::string m_message

Holds the latest message information for the exception.

Type m_type

Holds the type or severity of the exception.

13.77.1 Detailed Description

The rterror class is quite simple but it does allow errors to be "caught" by rterror::Type. See the rtexmidi documentation to know which methods can throw an rterror.

Please note that, in this refactoring of rtmidi, we've done away with all the exception specifications, on the advice of Herb Sutter. They may be more relevent to C++11 and beyond, but this library is too small to worry about them, for now

13.77.2 Member Enumeration Documentation

13.77.2.1 Type

```
enum seq64::rterror::Type
```

Enumerator

WARNING	A non-critical error.
DEBUG_WARNING	Non-critical error useful for debugging.
UNSPECIFIED	The default, unspecified error type.
NO_DEVICES_FOUND	No devices found on system.
INVALID_DEVICE	An invalid device ID was specified.
MEMORY_ERROR	An error occured during memory allocation.
INVALID_PARAMETER	Invalid parameter specified to a function.
INVALID_USE	The function was called incorrectly.
DRIVER_ERROR	A system driver error occured.
SYSTEM_ERROR	A system error occured.
THREAD_ERROR	A thread error occured.

13.77.3 Constructor & Destructor Documentation

```
13.77.3.1 rterror()
```

virtual seq64::rterror::~rterror () [inline], [virtual]

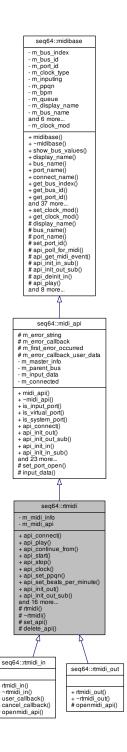
13.77.4 Member Function Documentation

```
13.77.4.1 print_message()
virtual void seq64::rterror::print_message ( ) const [inline], [virtual]
13.77.4.2 getType()
virtual const Type& seq64::rterror::getType ( ) const [inline], [virtual]
13.77.4.3 get_message()
virtual const std::string& seq64::rterror::get_message ( ) const [inline], [virtual]
13.77.4.4 what()
virtual const char* seq64::rterror::what ( ) const [inline], [virtual], [noexcept]
13.77.5 Field Documentation
13.77.5.1 m_message
std::string seq64::rterror::m_message [private]
13.77.5.2 m_type
Type seq64::rterror::m_type [private]
```

13.78 seq64::rtmidi Class Reference

The main class of the rtmidi API.

Inheritance diagram for seq64::rtmidi:



Public Member Functions

• virtual bool api_connect ()

```
    virtual void api_play (event *e24, midibyte channel)

    virtual void api_continue_from (midipulse tick, midipulse beats)

• virtual void api_start ()
• virtual void api stop ()
· virtual void api clock (midipulse tick)
virtual void api_set_ppqn (int ppqn)

    virtual void api set beats per minute (midibpm bpm)

virtual bool api_init_out ()
• virtual bool api init out sub ()
• virtual bool api init in ()
• virtual bool api init in sub ()
• virtual bool api deinit in ()

    virtual bool api_get_midi_event (event *inev)

• virtual int api_poll_for_midi ()
virtual void api_sysex (event *e24)
virtual void api_flush ()
• virtual bool is_port_open () const
      Returns true if a port is open and false if not.

    virtual int get bus id ()

      Gets the buss/client ID for a MIDI interfaces.
virtual std::string get_bus_name ()
virtual int get_port_id ()

    virtual std::string get_port_name ()

int get_port_count ()
• int full_port_count ()
• const midi_api * get_api () const
```

'Getter' function for member m_midi_api const version

'Getter' function for member m_midi_api non-const version

Protected Member Functions

midi_api * get_api ()

```
    rtmidi (midibus &parentbus, rtmidi_info &info)
    Default constructor.
```

virtual ∼rtmidi ()

Destructor.

void set_api (midi_api *ma)

'Setter' function for member m_midi_api

· void delete api ()

'Setter' function for member m_midi_api

Private Attributes

```
• rtmidi_info & m_midi_info
```

Holds a reference to the "global" midi_info wrapper object.

midi_api * m_midi_api

Points to the API I/O object (e.g.

Friends

· class midibus

Additional Inherited Members

13.78.1 Detailed Description

We moved the enum Api definition into the new rtmidi_types.hpp module to make refactoring the code easier.

13.78.2 Constructor & Destructor Documentation

13.78.2.1 rtmidi()

Parameters

parentbus	This is the midibus that the rtmidi object is going to implement, by forwarding calls to the selected MIDI subsystem (e.g. ALSA or JACK).
info	This object provides the system's enumerated busses/ports as found by the selected MIDI subsystem.

13.78.2.2 ∼rtmidi()

```
seq64::rtmidi::~rtmidi ( ) [protected], [virtual]
```

13.78.3 Member Function Documentation

13.78.3.1 api_connect()

```
virtual bool seq64::rtmidi::api_connect ( ) [inline], [virtual]
```

Reimplemented from seq64::midi_api.

```
13.78.3.2 api_play()
virtual void seq64::rtmidi::api_play (
             event * e24,
             midibyte channel ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.3 api_continue_from()
virtual void seq64::rtmidi::api_continue_from (
             midipulse tick,
             midipulse beats ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.4 api_start()
virtual void seq64::rtmidi::api_start ( ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.5 api_stop()
virtual void seq64::rtmidi::api_stop ( ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.6 api_clock()
virtual void seq64::rtmidi::api_clock (
             midipulse tick ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.7 api_set_ppqn()
virtual void seq64::rtmidi::api_set_ppqn (
             int ppqn ) [inline], [virtual]
Implements seq64::midi_api.
```

```
13.78.3.8 api_set_beats_per_minute()
virtual void seq64::rtmidi::api_set_beats_per_minute (
             midibpm bpm ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.9 api_init_out()
virtual bool seq64::rtmidi::api_init_out ( ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.10 api_init_out_sub()
virtual bool seq64::rtmidi::api_init_out_sub ( ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.11 api_init_in()
virtual bool seq64::rtmidi::api_init_in ( ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.12 api_init_in_sub()
virtual bool seq64::rtmidi::api_init_in_sub ( ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.13 api_deinit_in()
virtual bool seq64::rtmidi::api_deinit_in ( ) [inline], [virtual]
Implements seq64::midi_api.
```

```
13.78.3.14 api_get_midi_event()
virtual bool seq64::rtmidi::api_get_midi_event (
              event * inev ) [inline], [virtual]
Implements seq64::midi api.
13.78.3.15 api_poll_for_midi()
virtual int seq64::rtmidi::api_poll_for_midi ( ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.16 api_sysex()
virtual void seq64::rtmidi::api_sysex (
              event * e24 ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.17 api_flush()
virtual void seq64::rtmidi::api_flush ( ) [inline], [virtual]
Implements seq64::midi_api.
13.78.3.18 is_port_open()
virtual bool seq64::rtmidi::is_port_open ( ) const [inline], [virtual]
13.78.3.19 get_bus_id()
virtual int seq64::rtmidi::get_bus_id ( ) [inline], [virtual]
This is the left-hand side of a X:Y pair (such as 128:0).
This function is a new part of the RtMidi interface.
```

Returns

Returns the buss/client value as provided by the selected API.

```
13.78.3.20 get_bus_name()
```

```
virtual std::string seq64::rtmidi::get_bus_name ( ) [inline], [virtual]
```

Returns

Returns the buss name from the selected API subsystem.

13.78.3.21 get_port_id()

```
virtual int seq64::rtmidi::get_port_id ( ) [inline], [virtual]
```

Returns

Returns the port ID number from the selected API subsystem.

13.78.3.22 get_port_name()

```
virtual std::string seq64::rtmidi::get_port_name ( ) [inline], [virtual]
```

Returns

Returns the port name from the selected API subsystem.

13.78.3.23 get_port_count()

```
int seq64::rtmidi::get_port_count ( ) [inline]
```

Returns

This value depends on the MIDI mode setting (input versus output).

13.78.3.24 full_port_count()

```
int seq64::rtmidi::full_port_count ( ) [inline]
```

Returns

This value is the sum of the number of input and output ports.

```
13.78.3.25 get_api() [1/2]
const midi_api* seq64::rtmidi::get_api ( ) const [inline]
13.78.3.26 get_api() [2/2]
midi_api* seq64::rtmidi::get_api ( ) [inline]
13.78.3.27 set_api()
void seq64::rtmidi::set_api (
            midi_api * ma ) [inline], [protected]
13.78.3.28 delete_api()
void seq64::rtmidi::delete_api ( ) [inline], [protected]
13.78.4 Friends And Related Function Documentation
13.78.4.1 midibus
friend class midibus [friend]
13.78.5 Field Documentation
```

```
13.78.5.1 m_midi_info
rtmidi_info& seq64::rtmidi::m_midi_info [private]
```

Unlike the original RtMidi library, this library separates the port-enumeration code ("info") from the port-usage code ("api").

We might make it a static object at some point.



midi_api* seq64::rtmidi::m_midi_api [private]

midi_alsa or midi_jack) for which this class is a wrapper.

13.79 seq64::rtmidi_in Class Reference

A realtime MIDI input class.

Inheritance diagram for seq64::rtmidi_in:



Public Member Functions

rtmidi_in (midibus &parentbus, rtmidi_info &info)

Constructs the desired MIDI API.

• virtual ~rtmidi_in ()

A do-nothing virtual destructor.

• void user_callback (rtmidi_callback_t callback, void *userdata=nullptr)

Set a callback function to be invoked for incoming MIDI messages.

void cancel_callback ()

Cancel use of the current callback function (if one exists).

Protected Member Functions

void openmidi_api (rtmidi_api api, rtmidi_info &info)
 Opens the desired MIDI API.

Additional Inherited Members

13.79.1 Detailed Description

This class provides a common, platform-independent API for realtime MIDI input. It allows access to a single MIDI input port. Incoming MIDI messages are either saved to a queue for retrieval using the get_message() function or immediately passed to a user-specified callback function. Create multiple instances of this class to connect to more than one MIDI device at the same time. With the OS-X, Linux ALSA, and JACK MIDI APIs, it is also possible to open a virtual input port to which other MIDI software clients can connect.

13.79.2 Constructor & Destructor Documentation

13.79.2.1 rtmidi_in()

If no compiled support for specified API value is found, we issue a warning and continue as if no API was specified. In this case, we iterate through the compiled APIs and return as soon as we find one with at least one port or we reach the end of the list.

Parameters

parentbus	This is the midibus that the rtmidi object is going to implement, by forwarding calls to the selected MIDI subsystem (e.g. ALSA or JACK).
info	Contains information about the existing ports and the selected MIDI API. Actually, the selected
	MIDI API is a static value of the rtmidi_info class.

Exceptions

This function will throw an rterror object if it cannot find a MIDI API to use.

13.79.2.2 ∼rtmidi_in()

```
seq64::rtmidi_in::~rtmidi_in ( ) [virtual]
```

13.79.3 Member Function Documentation

13.79.3.1 user_callback()

The callback function will be called whenever an incoming MIDI message is received. While not absolutely necessary, it is best to set the callback function before opening a MIDI port to avoid leaving some messages in the queue.

Parameters

callback	A callback function must be given.	
userdata	Optionally, a pointer to additional data can be passed to the callback function whenever it is called.	Ī

13.79.3.2 cancel_callback()

```
void seq64::rtmidi_in::cancel_callback ( ) [inline]
```

Subsequent incoming MIDI messages will be written to the queue and can be retrieved with the *get_message* function.

13.79.3.3 openmidi_api()

Parameters

api	The enum value for the desired MIDI API. If not specified, first JACK is tried, then ALSA.
info	Holds information about the API's setup (e.g. a list of the ALSA ports found on the system, the main
	ALSA "handle" pointer, plus the PPQN, BPM, and other information).

13.80 seq64::rtmidi_in_data Class Reference

The rtmidi_in_data structure is used to pass private class data to the MIDI input handling function or thread.

Public Member Functions

- rtmidi in data ()
- const midi_queue & queue () const

'Getter' function for member m_queue const

• midi_queue & queue ()

'Getter' function for member m_queue non-const

- · const midi_message & message () const
- midi message & message ()
- · midibyte ignore flags () const
- bool test ignore flags (midibyte testbits)
- void ignore_flags (midibyte setbits)
- · bool do input () const
- void do_input (bool flag)
- bool first_message () const
- void first_message (bool flag)
- bool continue_sysex () const
- void continue_sysex (bool flag)
- bool using_callback () const
- void using_callback (bool flag)
- const void * api_data () const

'Getter' function for member m_api_data const

void * api_data ()

'Getter' function for member m_api_data non-const

- void api_data (void *dataptr)
- const void * user_data () const

'Getter' function for member m_user_data const

void * user data ()

'Getter' function for member m_user_data const

void user_data (void *dataptr)

'Setter' function for member m_user_data

rtmidi_callback_t user_callback () const

'Getter' function for member m_user_callback

void user_callback (rtmidi_callback_t cbptr)

'Setter' function for member m_user_callback This should be done immediately after opening the port to avoid having incoming messages written to the queue instead of sent to the callback function.

Private Attributes

- midi queue m queue
- midi_message m_message
- midibyte m_ignore_flags
- bool m do input
- bool m_first_message
- void * m api data
- · bool m using callback
- rtmidi_callback_t m_user_callback
- void * m_user_data
- bool m_continue_sysex

13.80.1 Detailed Description

Used to be nested in the rtmidi_in class.

13.80.2 Constructor & Destructor Documentation

```
13.80.2.1 rtmidi_in_data()
seq64::rtmidi_in_data::rtmidi_in_data ( )
13.80.3 Member Function Documentation
13.80.3.1 queue() [1/2]
const midi_queue& seq64::rtmidi_in_data::queue ( ) const [inline]
13.80.3.2 queue() [2/2]
midi_queue& seq64::rtmidi_in_data::queue ( ) [inline]
13.80.3.3 message() [1/2]
const midi_message& seq64::rtmidi_in_data::message ( ) const [inline]
13.80.3.4 message() [2/2]
midi_message& seq64::rtmidi_in_data::message ( ) [inline]
13.80.3.5 ignore_flags() [1/2]
midibyte seq64::rtmidi_in_data::ignore_flags ( ) const [inline]
```

```
13.80.3.6 test_ignore_flags()
bool seq64::rtmidi_in_data::test_ignore_flags (
            midibyte testbits ) [inline]
13.80.3.7 ignore_flags() [2/2]
void seq64::rtmidi_in_data::ignore_flags (
           midibyte setbits ) [inline]
13.80.3.8 do_input() [1/2]
bool seq64::rtmidi_in_data::do_input ( ) const [inline]
13.80.3.9 do_input() [2/2]
void seq64::rtmidi_in_data::do_input (
            bool flag ) [inline]
13.80.3.10 first_message() [1/2]
bool seq64::rtmidi_in_data::first_message ( ) const [inline]
13.80.3.11 first_message() [2/2]
void seq64::rtmidi_in_data::first_message (
             bool flag ) [inline]
13.80.3.12 continue_sysex() [1/2]
bool seq64::rtmidi_in_data::continue_sysex ( ) const [inline]
```

```
13.80.3.13 continue_sysex() [2/2]
void seq64::rtmidi_in_data::continue_sysex (
            bool flag ) [inline]
13.80.3.14 using_callback() [1/2]
bool seq64::rtmidi_in_data::using_callback ( ) const [inline]
13.80.3.15 using_callback() [2/2]
void seq64::rtmidi_in_data::using_callback (
            bool flag ) [inline]
13.80.3.16 api_data() [1/3]
const void* seq64::rtmidi_in_data::api_data ( ) const [inline]
13.80.3.17 api_data() [2/3]
void* seq64::rtmidi_in_data::api_data ( ) [inline]
13.80.3.18 api_data() [3/3]
void seq64::rtmidi_in_data::api_data (
             void * dataptr ) [inline]
13.80.3.19 user_data() [1/3]
const void* seq64::rtmidi_in_data::user_data ( ) const [inline]
```

```
13.80.3.20 user_data() [2/3]
void* seq64::rtmidi_in_data::user_data ( ) [inline]
13.80.3.21 user_data() [3/3]
void seq64::rtmidi_in_data::user_data (
           void * dataptr ) [inline]
13.80.3.22 user_callback() [1/2]
rtmidi_callback_t seq64::rtmidi_in_data::user_callback ( ) const [inline]
13.80.3.23 user_callback() [2/2]
void seq64::rtmidi_in_data::user_callback (
             rtmidi_callback_t cbptr ) [inline]
13.80.4 Field Documentation
13.80.4.1 m_queue
midi_queue seq64::rtmidi_in_data::m_queue [private]
13.80.4.2 m_message
midi_message seq64::rtmidi_in_data::m_message [private]
13.80.4.3 m_ignore_flags
midibyte seq64::rtmidi_in_data::m_ignore_flags [private]
```

```
13.80.4.4 m_do_input
bool seq64::rtmidi_in_data::m_do_input [private]
13.80.4.5 m_first_message
bool seq64::rtmidi_in_data::m_first_message [private]
13.80.4.6 m_api_data
void* seq64::rtmidi_in_data::m_api_data [private]
13.80.4.7 m_using_callback
bool seq64::rtmidi_in_data::m_using_callback [private]
13.80.4.8 m_user_callback
rtmidi_callback_t seq64::rtmidi_in_data::m_user_callback [private]
13.80.4.9 m_user_data
void* seq64::rtmidi_in_data::m_user_data [private]
13.80.4.10 m_continue_sysex
```

13.81 seq64::rtmidi_info Class Reference

bool seq64::rtmidi_in_data::m_continue_sysex [private]

A class for enumerating MIDI clients and ports.

Public Member Functions

 rtmidi_info (rtmidi_api api=RTMIDI_API_UNSPECIFIED, const std::string &appname="rtmidiapp", int ppqn=SEQ64_DEFAULT_PPQN, midibpm bpm=SEQ64_DEFAULT_BPM)

Default constructor.

virtual ∼rtmidi info ()

Destructor.

• bool midi_mode () const

Sets the input or output mode for getting data.

• void midi mode (bool flag)

Sets the input or output mode for getting data.

• void clear ()

Clear the MIDI port container.

void add_input (const midibus *m)

Add midibus information to the input ports.

void add_output (const midibus *m)

Add midibus information to the output ports.

void add_bus (const midibus *m)

Adds the bus to a list of busses to be connected by the API at the right time (currently applies only to JACK).

• int get_bus_id (int index) const

Gets the buss/client ID for a MIDI interfaces.

- · std::string get bus name (int index) const
- int get_port_count () const
- int full_port_count () const
- int get_port_id (int index) const
- std::string get_port_name (int index) const
- bool get_input (int index) const
- · bool get_virtual (int index) const
- bool get_system (int index) const
- int get_all_port_info ()
- int queue_number (int index) const
- const std::string & app name () const
- int global_queue () const
- int ppqn () const
- void api_set_ppqn (int p)
- · midibpm bpm () const
- void api_set_beats_per_minute (midibpm b)
- · void api port start (mastermidibus &masterbus, int bus, int port)
- bool api_get_midi_event (event *inev)
- void api_flush ()
- int api poll for midi ()
- std::string port_list () const

Returns a list of all the ports as an ASCII string.

const midi_info * get_api_info () const

'Getter' function for member m_info_api const version

• midi info * get api info ()

'Getter' function for member m_info_api non-const version

Static Public Member Functions

• static std::string get_version ()

'Getter' function for member SEQ64_RTMIDI_VERSION This is a static function to replace the midi_api version.

static void get_compiled_api (std::vector< rtmidi_api > &apis)

Gets the list of APIs compiled into the application.

• static rtmidi_api & selected_api ()

'Getter' function for member sm_selected_api

Protected Member Functions

- bool api_connect ()
- bool set_api_info (midi_info *ma)

'Setter' function for member m_info_api This function also checks the pointer and returns false if it is not valid.

void delete_api ()

'Setter' function for member m_info_api

• bool openmidi_api (rtmidi_api api, const std::string &appname, int ppqn, midibpm bpm)

Opens the desired MIDI API.

Static Protected Member Functions

• static void selected_api (const rtmidi_api &api)

'Setter' function for member sm_selected_api

Private Attributes

midi_info * m_info_api

Provides access to the selected API (currently only JACK or ALSA).

Static Private Attributes

• static rtmidi_api sm_selected_api

To save repeated queries, we save this value.

Friends

- · class mastermidibus
- · class midibus
- · class rtmidi_in
- · class rtmidi_out

13.81.1 Detailed Description

New, but ripe for refactoring nonetheless.

13.81.2 Constructor & Destructor Documentation

13.81.2.1 rtmidi_info()

```
seq64::rtmidi_info::rtmidi_info (
    rtmidi_api api = RTMIDI_API_UNSPECIFIED,
    const std::string & appname = "rtmidiapp",
    int ppqn = SEQ64_DEFAULT_PPQN,
    midibpm bpm = SEQ64_DEFAULT_BPM )
```

Code basically cut-and-paste from rtmidi_in or rtmidi_out. Common code!

```
13.81.2.2 \simrtmidi_info()
```

```
seq64::rtmidi_info::~rtmidi_info ( ) [virtual]
```

13.81.3 Member Function Documentation

13.81.3.1 get_version()

```
std::string seq64::rtmidi_info::get_version ( ) [static]
```

13.81.3.2 get_compiled_api()

Note that we make ALSA versus JACK a runtime option as it is in the legacy Sequencer64 application.

This is a static function to replace the midi_api version.

Parameters

```
apis The API structure.
```

```
13.81.3.3 midi_mode() [1/2]
```

```
bool seq64::rtmidi_info::midi_mode ( ) const [inline]
```

Also adds the midibus to a lit of busses to connect in mastermidibus. This function is meant for virtual ports.

```
13.81.3.7 add_output()
```

Also adds the midibus to a lit of busses to connect in mastermidibus. This function is meant for virtual ports.

13.81.3.8 add_bus()

See the calls to this function in mastermidibus.

13.81.3.9 get_bus_id()

This is the left-hand side of a X:Y pair (such as 128:0).

This function is a new part of the RtMidi interface.

Parameters

index	The ordinal index of the desired interface to look up.
-------	--

Returns

Returns the buss/client value as provided by the selected API.

```
13.81.3.10 get_bus_name()
std::string seq64::rtmidi_info::get_bus_name (
            int index ) const [inline]
13.81.3.11 get_port_count()
int seq64::rtmidi_info::get_port_count ( ) const [inline]
13.81.3.12 full_port_count()
int seq64::rtmidi_info::full_port_count ( ) const [inline]
13.81.3.13 get_port_id()
int seq64::rtmidi_info::get_port_id (
            int index ) const [inline]
13.81.3.14 get_port_name()
std::string seq64::rtmidi_info::get_port_name (
```

int index) const [inline]

```
13.81.3.15 get_input()
bool seq64::rtmidi_info::get_input (
            int index ) const [inline]
13.81.3.16 get_virtual()
bool seq64::rtmidi_info::get_virtual (
           int index ) const [inline]
13.81.3.17 get_system()
bool seq64::rtmidi_info::get_system (
             int index ) const [inline]
13.81.3.18 get_all_port_info()
int seq64::rtmidi_info::get_all_port_info ( ) [inline]
13.81.3.19 queue_number()
int seq64::rtmidi_info::queue_number (
            int index ) const [inline]
13.81.3.20 app_name()
const std::string& seq64::rtmidi_info::app_name ( ) const [inline]
13.81.3.21 global_queue()
int seq64::rtmidi_info::global_queue ( ) const [inline]
```

```
13.81.3.22 ppqn()
int seq64::rtmidi_info::ppqn ( ) const [inline]
13.81.3.23 api_set_ppqn()
void seq64::rtmidi_info::api_set_ppqn (
            int p) [inline]
13.81.3.24 bpm()
midibpm seq64::rtmidi_info::bpm ( ) const [inline]
13.81.3.25 api_set_beats_per_minute()
void seq64::rtmidi_info::api_set_beats_per_minute (
             midibpm b ) [inline]
13.81.3.26 api_port_start()
void seq64::rtmidi_info::api_port_start (
             mastermidibus & masterbus,
             int bus,
             int port ) [inline]
13.81.3.27 api_get_midi_event()
bool seq64::rtmidi_info::api_get_midi_event (
             event * inev ) [inline]
13.81.3.28 api_flush()
void seq64::rtmidi_info::api_flush ( ) [inline]
```

```
13.81.3.29 api_poll_for_midi()
int seq64::rtmidi_info::api_poll_for_midi ( ) [inline]
13.81.3.30 port_list()
std::string seq64::rtmidi_info::port_list ( ) const [inline]
13.81.3.31 selected_api() [1/2]
static rtmidi_api& seq64::rtmidi_info::selected_api ( ) [inline], [static]
13.81.3.32 get_api_info() [1/2]
const midi_info* seq64::rtmidi_info::get_api_info ( ) const [inline]
13.81.3.33 get_api_info() [2/2]
midi_info* seq64::rtmidi_info::get_api_info ( ) [inline]
13.81.3.34 api_connect()
bool seq64::rtmidi_info::api_connect ( ) [inline], [protected]
13.81.3.35 selected_api() [2/2]
static void seq64::rtmidi_info::selected_api (
             const rtmidi_api & api ) [inline], [static], [protected]
```

13.81.3.36 set_api_info()

This feature is important to allow a missing API (e.g. the JACK server is not running) to be detected.

13.81.3.37 delete_api()

```
void seq64::rtmidi_info::delete_api ( ) [inline], [protected]
```

13.81.3.38 openmidi_api()

If the JACK API is tried, and found missing, we turn off all of the other JACK flags found in the "rc" configuration file. Also, the loop in the constructor will come back here to try the other compiled-in APIs (currently just ALSA).

Parameters

api	The desired MIDI API.
appname	The name of the application, to be passed to the midi_info-derived constructor.
ppqn	The PPQN value to pass along to the midi_info_derived constructor.
bpm	The BPM (beats per minute) value to pass along to the midi_info_derived constructor.

Returns

Returns true if a valid API is found. A valid API is on that is both compiled into the application and is found existing on the host computer (system).

13.81.4 Friends And Related Function Documentation

13.81.4.1 mastermidibus

```
friend class mastermidibus [friend]
```

friend class midibus [friend] 13.81.4.3 rtmidi_in friend class rtmidi_in [friend] 13.81.4.4 rtmidi_out friend class rtmidi_out [friend]

```
13.81.5.1 m_info_api

midi_info* seq64::rtmidi_info::m_info_api [private]
```

```
13.81.5.2 sm_selected_api

rtmidi_api seq64::rtmidi_info::sm_selected_api [static], [private]
```

Holds the selected API code.

Its default value is RTMIDI_API_UNSPECIFIED.

13.82 seq64::rtmidi_out Class Reference

A realtime MIDI output class.

Inheritance diagram for seq64::rtmidi_out:



Public Member Functions

• rtmidi_out (midibus &parentbus, rtmidi_info &info)

Principal constructor.

virtual ∼rtmidi_out ()

The destructor closes any open MIDI connections.

Protected Member Functions

• void openmidi_api (rtmidi_api api, rtmidi_info &info)

Additional Inherited Members

13.82.1 Detailed Description

This class provides a common, platform-independent API for MIDI output. It allows one to probe available MIDI output ports, to connect to one such port, and to send MIDI bytes immediately over the connection. Create multiple instances of this class to connect to more than one MIDI device at the same time. With the OS-X, Linux ALSA and JACK MIDI APIs, it is also possible to open a virtual port to which other MIDI software clients can connect.

13.82.2 Constructor & Destructor Documentation

```
13.82.2.1 rtmidi_out()
```

Attempt to open the specified API. If there is no compiled support for specified API value, then issue a warning and continue as if no API was specified. In that case, we Iterate through the compiled APIs and return as soon as we find one with at least one port or we reach the end of the list.

Parameters

parentbus	This is the midibus that the rtmidi object is going to implement, by forwarding calls to the selected MIDI subsystem (e.g. ALSA or JACK).
info	Contains information about the existing ports and the selected MIDI API. Actually, the selected MIDI API is a static value of the rtmidi_info class.

Exceptions

This	function will throw an rterror object if it cannot find a MIDI API to use.
------	--

```
13.82.2.2 ~rtmidi_out()
```

```
seq64::rtmidi\_out::\sim rtmidi\_out ( ) [virtual]
```

A do-nothing virtual destructor.

13.82.3 Member Function Documentation

13.82.3.1 openmidi_api()

13.83 seq64::Seq24PerfInput Class Reference

Implements the default (Seq24) performance input characteristics of this application.

Inheritance diagram for seq64::Seq24PerfInput:

seq64::AbstractPerfInput - m_adding - m_adding_pressed + AbstractPerfInput() # ~AbstractPerfInput() # on_button_press_event() # on_button_release_event() # on_motion_notify_event() # activate_adding() # handle_motion_key() # is_adding() # set_adding() # is_adding_pressed() # set_adding_pressed() seq64::Seq24PerfInput - m effective tick + Seq24PerfInput() + on_button_press_event() + on_button_release_event() + on_motion_notify_event() - activate_adding() - handle_motion_key()

Public Member Functions

- Seq24PerfInput ()
- bool on_button_press_event (GdkEventButton *a_ev, perfroll &roll)

Handles the normal variety of button-press event.

• bool on_button_release_event (GdkEventButton *a_ev, perfroll &roll)

Handles various button-release events.

bool on_motion_notify_event (GdkEventMotion *a_ev, perfroll &roll)

Handles the normal motion-notify event.

Private Member Functions

• virtual void activate_adding (bool a_adding, perfroll &roll)

A popup menu (which one?) calls this.

• bool handle_motion_key (bool is_left, perfroll &roll)

Handles the keystroke motion-notify event for moving a pattern back and forth in the performance.

Private Attributes

• midipulse m_effective_tick

The current tick for the current segment?

Friends

class perfroll

Additional Inherited Members

13.83.1 Constructor & Destructor Documentation

13.83.1.1 Seq24PerfInput()

```
seq64::Seq24PerfInput::Seq24PerfInput ( ) [inline]
```

13.83.2 Member Function Documentation

13.83.2.1 on_button_press_event()

Is there any easy way to use ctrl-left-click as the middle button here?

Stazed:

```
roll.m_drop_y will be adjusted by perfroll::change_vert() for any scroll after it was originally selected. The call here to draw_drawable_row() [now folded into draw_all()] will have the wrong y location and un-select will not occur (or the wrong sequence will be unselected) if the user scrolls the track up or down to a new y location, if not adjusted.
```

Returns

Returns true if a modification occurred.

Implements seq64::AbstractPerfInput.

13.83.2.2 on_button_release_event()

Any use for the middle-button or ctrl-left-click we can add?

Returns

Returns true if any modification occurred.

Implements seq64::AbstractPerfInput.

13.83.2.3 on_motion_notify_event()

Returns

Returns true if a modification occurs. This function used to always return true.

Implements seq64::AbstractPerfInput.

13.83.2.4 activate_adding()

What does it mean?

Implements seq64::AbstractPerfInput.

13.83.2.5 handle_motion_key()

What happens when the mouse is used to drag the pattern is that, first, roll.m_drop_tick is set by left-clicking into the pattern to select it. As the pattern is dragged, the drop-tick value does not change, but the tick (converted from the moving x value) does.

Then the button-handler sets roll.m_moving = true, and calculates roll.m_drop_tick_trigger_offset = roll.m_drop_tick - p.get_sequence(dropseq)->selected_trigger_start();

The motion handler sees that roll.m_moving is true, gets the new tick value from the new x value, offsets it, and calls p.get sequence(dropseq)->move selected triggers to(tick, true).

When the user releases the left button, then roll.m_growing is turned of and the roll draw_all()'s.

Parameters

is_left	False denotes the right arrow key, and true denotes the left arrow key.
roll	Provides a reference to the parent roll, which keeps track of most of the information about the status of
	the window.

Returns

Returns true if there was some action able to happen that would necessitate a window update. We've updated triggers::move_selected() [called indirectly near the end of this routine] to return false if no more movement could be made. This prevents this routine from moving way ahead after movement of the selected (in the user-interface) trigger stops.

Implements seq64::AbstractPerfInput.

13.83.3 Friends And Related Function Documentation

13.83.3.1 perfroll

```
friend class perfroll [friend]
```

13.83.4 Field Documentation

13.83.4.1 m_effective_tick

```
midipulse seg64::Seg24PerfInput::m_effective_tick [private]
```

13.84 seq64::Seq24SeqEventInput Struct Reference

This structure implement the normal interaction methods for Seq24.

Public Member Functions

Seq24SeqEventInput ()

Default constructor.

void set_adding (bool adding, seqevent &ths)

Changes the mouse cursor to a pencil or a left pointer in the given seqevent object, depending on the first parameter.

• bool on_button_press_event (GdkEventButton *ev, seqevent &ths)

Implements the on-button-press event callback.

• bool on_button_release_event (GdkEventButton *ev, seqevent &ths)

Implements the on-button-release callback.

bool on_motion_notify_event (GdkEventMotion *ev, seqevent &ths)

Implements the on-motion-notify event.

Data Fields

· bool m_adding

True if we're adding events via the mouse.

13.84.1 Constructor & Destructor Documentation

13.84.1.1 Seq24SeqEventInput()

```
seq64::Seq24SeqEventInput::Seq24SeqEventInput ( ) [inline]
```

13.84.2 Member Function Documentation

13.84.2.1 set_adding()

Modifies m_adding as well.

Parameters

adding	The value to set m_adding to, and if true, sets the mouse cursor to a pencil icon, otherwise sets it to a standard mouse-pointer icon.
seqev	The seqevent whose window will be set to "adding" mode.

13.84.2.2 on_button_press_event()

Set values for dragging, then reset the box that holds dirty redraw spot. Then do the rest.

Parameters

ev	The button event for the press of a mouse button.
seqev	Provides the sequeent strip to be affected by this button event.

Returns

Returns true if a likely modification was made. This function used to return true all the time.

Needs update. seqev.m_seq.unselect(); ???????

13.84.2.3 on_button_release_event()

Parameters

ev		The button event for the release of a mouse button.
seq	ev	Provides the seqevent strip to be affected by this button event.

Returns

Returns true if a likely modification was made. This function used to return true all the time.

13.84.2.4 on_motion_notify_event()

ev	The button event for the motion of the mouse cursor.
seqev	Provides the sequeent strip to be affected by this button event.

Returns

Returns true if a likely modification was made. This function used to return true all the time.

13.84.3 Field Documentation

13.84.3.1 m_adding

bool seq64::Seq24SeqEventInput::m_adding

13.85 seq64::seqdata Class Reference

This class supports drawing piano-roll eventis on a window.

Inheritance diagram for seq64::seqdata:

```
seq64::gui_palette_gtk2
                                                                       and 11 more...
+ gui_palette_gtk2()
+ gui_palette_gtk2()
+ gui_palette_gtk2()
+ line_color()
+ progress_color()
+ black()
+ dark_green()
+ dark_green()
+ dark_blue()
+ dark_magenta()
and 18 more...
+ load_inverse_palette()
+ is_inverse()
                       seq64::gui_drawingarea_gtk2
#m_drop_y

+gui_drawingarea_gk2()
+gui_drawingarea_gk2()
+-gui_drawingarea_gk2()
+-gui_drawingarea_gk2()
--gui_drawingarea_gk2()
+-gui_drawingarea_gk2()
+-gui_drawingarea_gk2()
+-gui_drawingarea_gk2()
# draw_line()
# draw_line
```

Public Member Functions

- seqdata (sequence &seq, perform &p, int zoom, Gtk::Adjustment &hadjust)

 Principal constructor.
- virtual ∼seqdata ()

Let's provide a do-nothing virtual destructor.

• void reset ()

This function calls update_size().

void redraw ()

Calls change_horz() to update the pixmap and queue up a redraw operation.

void set zoom (int a zoom)

Sets the zoom to the given value and resets the view via the reset function.

void set_data_type (midibyte status, midibyte control)

Sets the status to the given value, and the control to the optional given value, which defaults to 0, then calls redraw().

Private Member Functions

• int idle_redraw ()

Draws events on this object's built-in window and pixmap.

void update_sizes ()

Updates the sizes in the pixmap if the view is realized, and queues up a draw operation.

void update_pixmap ()

Simply calls draw_events_on_pixmap().

• void draw_line_on_window ()

Draws on vertical line on the data window.

void xy_to_rect (int x1, int y1, int x2, int y2, int &rx, int &ry, int &rw, int &rh)

This function takes two points, and returns an XWin rectangle, returned via the last four parameters.

void draw_events_on (Glib::RefPtr< Gdk::Drawable > drawable)

Draws events on the given drawable object.

• void change horz ()

Change the scrolling offset on the x-axis, and redraw.

void convert_x (int x, midipulse &tick)

This function takes screen coordinates, and gives the horizontaol tick value based on the current zoom, returned via the second parameter.

• void render_number (Glib::RefPtr< Gdk::Pixmap > &pixmap, int x, int y, const char *const num)

Convenience function for rendering numbers.

void draw_events_on_pixmap ()

Simply calls draw_events_on() for this object's built-in pixmap.

• void draw_pixmap_on_window ()

Simply queues up a draw operation.

void on_realize ()

Implements the on-realization event, by calling the base-class version and then allocating the resources that could not be allocated in the constructor.

bool on_expose_event (GdkEventExpose *ev)

Implements the on-expose event by calling draw_drawable() on the event.

bool on_button_press_event (GdkEventButton *ev)

Implements a mouse button-press event.

bool on_button_release_event (GdkEventButton *ev)

Implement a button-release event.

bool on_motion_notify_event (GdkEventMotion *ev)

Handles a motion-notify event.

bool on_leave_notify_event (GdkEventCrossing *ev)

Handles an on-leave notification event.

• bool on_scroll_event (GdkEventScroll *ev)

Implements the on-scroll event.

• void on_size_allocate (Gtk::Allocation &)

Handles a size-allocation event by updating m_window_x and m_window_y, and then updating all of the sizes of the data pane in update_sizes().

Private Attributes

· sequence & m seq

Points to the sequence whose data is being affected by this class.

• int m_zoom

Sets the zoom value for this part of the sequence editor, one pixel $== m_z$ oom ticks, i.e.

· int m scroll offset ticks

The value of the leftmost tick in the data pane.

int m_scroll_offset_x

The value of the leftmost pixel in the data pane.

• int m number w

The adjusted width of a digit in a data number.

· int m_number_h

The adjusted height of all digits in a data number.

• int m number offset y

A new value to make it easier to adapt the vertical number drawing of a data item's numeric value to a different font.

· midibyte m_status

Holds the status byte of the next event in the sequence, and indicates What the data window is currently editing or drawing.

• midibyte m_cc

Holds the MIDI CC byte of the next event in the sequence, and indicates What the data window is currently editing or drawing.

Glib::RefPtr< Gdk::Pixmap > m_numbers [c_dataarea_y]

Holds the pixmaps for each number (0 to 127) that can be drawn for a data value in the data pane.

• GdkRectangle m_old

This rectangle is used in blanking out a data line in draw_line_on_window().

- bool m_drag_handle
- bool m_dragging

This value is true if the mouse is being dragged in the data pane, which is done in order to change the height and value of each data line.

Friends

- class Ifownd
- · class segevent
- · class segroll

Additional Inherited Members

13.85.1 Constructor & Destructor Documentation

13.85.1.1 seqdata()

In the constructor one can only allocate colors, get_window() returns 0 because this pane has not yet been realized.

seq	The sequence that is being displayed and edited by this data pane.
р	The performance object that oversees all of the sequences. This object is needed here only to access the perform::modify() function.
zoom	The starting zoom of this pane.
hadjust	The horizontal adjustment object provided by the parent class, seqedit, that created this pane.

```
13.85.1.2 ∼seqdata()
```

```
virtual seq64::seqdata::\simseqdata ( ) [inline], [virtual]
```

13.85.2 Member Function Documentation

13.85.2.1 reset()

```
void seq64::seqdata::reset ( )
```

Then, regardless of whether the view is realized, updates the pixmap and queues up a draw operation.

Note

If it weren't for the is_realized() condition, we could just call update_sizes(), which does all this anyway.

13.85.2.2 redraw()

```
void seq64::seqdata::redraw ( ) [inline]
```

13.85.2.3 set_zoom()

```
void seq64::seqdata::set_zoom (
    int z)
```

Called by seqedit::set_zoom(), which validates the zoom value.

Parameters

z The desired zoom value, assumed to be validated already. See the seqedit::set_zoom() function.

13.85.2.4 set_data_type()

Perhaps we should check that at least one of the parameters causes a change.

Parameters

status	The MIDI event byte (status byte) to set.
control	The MIDI CC value to set.

13.85.2.5 idle_redraw()

```
int seq64::seqdata::idle_redraw ( ) [private]
```

This drawing is done only if there is no dragging in progress, to guarantee no flicker.

13.85.2.6 update_sizes()

```
void seq64::seqdata::update_sizes ( ) [private]
```

It creates a pixmap with window dimensions given by m_window_x and m_window_y.

We thought there was a potential memory leak, since m_pixmap is created every time the window is resized, but valgrind says otherwise... maybe. An awful lot of Gtk leaks!

13.85.2.7 update_pixmap()

```
void seq64::seqdata::update_pixmap ( ) [private]
```

13.85.2.8 draw_line_on_window()

```
void seq64::seqdata::draw_line_on_window ( ) [private]
```

13.85.2.9 xy_to_rect()

It checks the mins/maxes, then fills in x, y, and width, height.

	x1	The input x value for the first data point.
	y1	The input y value for the first data point.
	x2	The input x value for the second data point.
	y2	The input y value for the second data point.
out	rx	The output for the x value of the XWin rectangle.
out	ry	The output for the y value of the XWin rectangle.
out	rw	The output for the width value of the XWin rectangle.
out	rh	The output for the height of the XWin rectangle.

13.85.2.10 draw_events_on()

Very similar to seqevent :: draw_events_on(). And yet it doesn't handle zooming as well, must fix!

Stazed:

For Note On there can be multiple events on the same vertical in which the selected item can be covered. For Note On the selected item needs to be drawn last so it can be seen. So, for other events the variable num_selected_events will be -1 for ALL_EVENTS. For Note On only, the variable will be the number of selected events. If 0 then only one pass is needed. If > 0 then two passes are needed, one for unselected (first), and one for selected (last). For the first pass, if any events are selected, the selection type is EVENTS_UNSELECTED. For the second pass, it will be set to num_selected_events.

We now draw the data line for selected event in dark orange, instead of black. We're not likely to adopt the Stazed convention of drawing in blue. Also, there seem to be some bugs in how the data selection works. Needs more evaluation.

Also, if we decide to draw handle on each vertical data line, it would look nicer if a circle.

Parameters

drawable	The given drawable object.

13.85.2.11 change_horz()

```
void seq64::seqdata::change_horz ( ) [private]
```

Basically identical to seqevent::change_horz().

13.85.2.12 convert_x()

13.85.2.13 render_number()

```
void seq64::seqdata::render_number (
        Glib::RefPtr< Gdk::Pixmap > & pixmap,
        int x,
        int y,
        const char *const num ) [inline], [private]
```

Parameters

pixmap	The reference pointer to the GDK pixmap onto which this number will be drawing.
Х	The x-coordinate of the position of the text.
У	The y-coordinate of the position of the text.
num	The number to be rendered. This should be a string reference, but oh well.

13.85.2.14 draw_events_on_pixmap()

```
void seq64::seqdata::draw_events_on_pixmap ( ) [inline], [private]
```

13.85.2.15 draw_pixmap_on_window()

```
void seq64::seqdata::draw_pixmap_on_window ( ) [inline], [private]
```

13.85.2.16 on_realize()

```
void seq64::seqdata::on_realize ( ) [private]
```

It also connects up the change_horz() function.

Note that this function creates a small pixmap for every possible y-value, where y ranges from 0 to MIDI_COUNT \leftarrow _MAX-1 = 127. It then fills each pixmap with a numeric representation of that y value, up to three digits (left-padded with spaces).

13.85.2.17 on_expose_event()

ev Provides the expose-event.

Returns

Always returns true.

13.85.2.18 on_button_press_event()

This function pushes the undo information for the sequence, sets the drop-point, resets the box that holds dirty redraw spot, and sets m_dragging to true.

Parameters

ev Provides the button-press event.

Returns

Always returns true.

13.85.2.19 on_button_release_event()

```
bool seq64::seqdata::on_button_release_event (
    GdkEventButton * ev ) [private]
```

Sets the current point. If m_dragging is true, then the sequence data is changed, the performance modification flag is set, and m_dragging is reset.

Parameters

ev Provides the button-release event.

Returns

Returns true if a modification occurred, and in that case also sets the perform modification flag.

13.85.2.20 on_motion_notify_event()

It converts the x,y of the mouse to ticks, then sets the events in the event-data-range, updates the pixmap, draws events in the window, and draws a line on the window.

Parameters

ev The motion event.

Returns

Returns true if a change in event data occurred. If true, then the perform modification flag is set.

13.85.2.21 on_leave_notify_event()

Parameter "p0", the crossing point for the event, is unused.

13.85.2.22 on_scroll_event()

This scroll event only handles basic scrolling, without any modifier keys such as the Ctrl of Shift masks. If there is a note (seqroll pane) or event (seqevent pane) selected, and mouse hovers over the data area (seqdata pane), then this scrolling action will increase or decrease the value of the data item, which lengthens of shortens the line drawn.

Todo DOCUMENT the sequata scrolling behavior in the documentation projects.

Parameters

ev Provides the scroll-event.

Returns

Always returns true.

```
13.85.2.23 on_size_allocate()
```

```
r Provides the allocation event.
```

13.85.3 Friends And Related Function Documentation

```
13.85.3.1 Ifownd
```

```
friend class lfownd [friend]
```

13.85.3.2 seqevent

```
friend class sequent [friend]
```

13.85.3.3 seqroll

```
friend class seqroll [friend]
```

13.85.4 Field Documentation

```
13.85.4.1 m_seq
```

```
sequence& seq64::seqdata::m_seq [private]
```

13.85.4.2 m_zoom

```
int seq64::seqdata::m_zoom [private]
```

the unit is ticks/pixel.

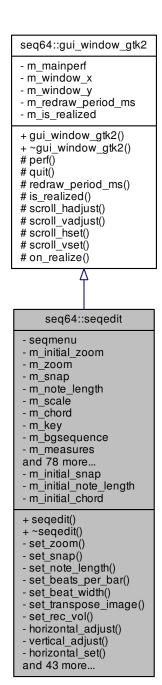
```
13.85.4.3 m_scroll_offset_ticks
int seq64::seqdata::m_scroll_offset_ticks [private]
Adjusted in the <a href="mailto:change_horz">change_horz</a>() function.
13.85.4.4 m_scroll_offset_x
int seq64::seqdata::m_scroll_offset_x [private]
Adjusted in the change_horz() function. It is the offset ticks divided by the zoom value, i.e. the unit is pixels..
13.85.4.5 m_number_w
int seq64::seqdata::m_number_w [private]
By "adjusted", well this is just a minor tweak for appearances.
13.85.4.6 m_number_h
int seq64::seqdata::m_number_h [private]
Basically, the character height times 3. By "adjusted", well this is just a minor tweak for appearances.
13.85.4.7 m_number_offset_y
int seq64::seqdata::m_number_offset_y [private]
This value was hardwired as 8, for a character height of 10.
13.85.4.8 m status
midibyte seq64::seqdata::m_status [private]
13.85.4.9 m_cc
midibyte seq64::seqdata::m_cc [private]
13.85.4.10 m_numbers
Glib::RefPtr<Gdk::Pixmap> seq64::seqdata::m_numbers[c_dataarea_y] [private]
This array is filled only once, in the on_realize() function.
```



Implements the Pattern Editor, which has references to:

Generated by Doxygen

Inheritance diagram for seq64::seqedit:



Public Member Functions

- seqedit (perform &perf, sequence &seq, int pos, int ppqn=SEQ64_USE_DEFAULT_PPQN)
 Principal constructor.
- virtual \sim seqedit ()

A rote destructor.

Private Member Functions

void set zoom (int zoom)

Selects the given zoom value.

void set_snap (int snap)

Selects the given snap value, which is the number of ticks in a snap-sized interval.

void set_note_length (int note_length)

Selects the given note-length value.

void set beats per bar (int bpm)

Set the bpm (beats per measure) value, using the given parameter, and some internal values passed to apply_\leftarrow length().

void set beat width (int bw)

Set the bw (beat width) value, using the given parameter, and some internal values passed to apply_length().

void set transpose image (bool istransposable)

Changes the image used for the transpose button.

void set_rec_vol (int recvol)

Passes the given parameter to sequence::set_rec_vol().

void horizontal_adjust (double step)

This function provides optimization for the on_scroll_event() function.

void vertical_adjust (double step)

This function provides optimization for the on_scroll_event() function.

void horizontal_set (double value)

Sets the exact position of a horizontal scroll-bar.

• void vertical_set (double value)

Sets the exact position of a vertical scroll-bar.

void set_measures (int lim)

Set the measures value, using the given parameter, and some internal values passed to apply_length().

void apply_length (midibpm bpm, int bw, int measures)

Sets the sequence length based on the three given parameters.

• long get measures ()

Calculates the measures value based on the bpm (beats per measure), ppqn (parts per quarter note), and bw (beat width) values, and returns the resultant measures value.

• void set_midi_channel (int midichannel, bool user_change=false)

Selects the given MIDI channel parameter in the main sequence object, so that it will use that channel.

void set midi bus (int midibus, bool user change=false)

Selects the given MIDI buss parameter in the main sequence object, so that it will use that buss.

• void set scale (int scale)

Selects the given scale value.

- void set_chord (int chord)
- void set_key (int note)

Selects the given key (signature) value.

void set background sequence (int seq)

Draws the given background sequence on the Pattern editor so that the musician has something to see that can be played against.

void transpose_change_callback ()

Passes the transpose status to the sequence object.

• void name_change_callback ()

Set the name for the main sequence to this object's entry name.

• void play_change_callback ()

Passes the play status to the sequence object.

void record_change_callback ()

Passes the recording status to the sequence object.

• void q_rec_change_callback ()

Passes the quantized-recording status to the sequence object.

· void thru change callback ()

Passes the MIDI Thru status to the sequence object.

void undo_callback ()

Pops an undo operation from the sequence object, and then tells the segroll, seqtime, seqdata, and seqevent objects to redraw.

void redo callback ()

Pops a redo operation from the sequence object, and then tell the segroll, seqtime, seqdata, and seqevent objects to redraw.

void set_data_type (midibyte status, midibyte control=0)

Sets the data type based on the given parameters.

- void update all windows ()
- · void fill top bar ()

This function inserts the user-interface items into the top bar or panel of the pattern editor; this bar has two rows of user interface elements.

void create_menus ()

Creates the various menus by pushing menu elements into the menus.

void popup_menu (Gtk::Menu *menu)

Pops up the given pop-up menu.

• void popup event menu ()

Populates the event-selection menu that drops from the "Event" button in the bottom row of the Pattern editor.

void popup_midibus_menu ()

Populates the MIDI Output buss pop-up menu.

void popup sequence menu ()

Populates the "set background sequence" menu (drops from the button that has some note-bars on it at the right of the second row of the top bar).

void popup_tool_menu ()

Sets up the pop-up menus that are brought up by pressing the Tools button, which shows a hammer image.

· void popup midich menu ()

Populates the MIDI Channel pop-up menu.

Gtk::Image * create_menu_image (bool state=false)

Sets the menu pixmap depending on the given state, where true is a full menu (black backgroun), and empty menu (gray background).

• bool timeout ()

Update the window after a time out, based on dirtiness and on playback progress.

void do_action (int action, int var)

Implements the actions brought forth from the Tools (hammer) button.

- void mouse_action (mouse_action_e action)
- void start playing ()
- void stop_playing ()
- · void change_focus (bool set_it=true)

Changes what perform and mainwid see as the "current sequence".

• void handle_close ()

Handles closing the sequence editor.

· void on realize ()

On realization, calls the base-class version, and connects the redraw timeout signal, timed at redraw_period_ms().

void on_set_focus (Widget *focus)

On receiving focus, attempt to tell mainwid that this sequence is now the current sequence.

bool on focus in event (GdkEventFocus *)

Implements the on-focus event handling.

bool on_focus_out_event (GdkEventFocus *)

Implements the on-unfocus event handling.

bool on_delete_event (GdkEventAny *event)

Handles an on-delete event.

bool on scroll event (GdkEventScroll *ev)

Handles an on-scroll event.

bool on_key_press_event (GdkEventKey *ev)

Handles a key-press event.

Private Attributes

- · friend segmenu
- · const int m initial zoom

Provides the initial zoom, used for restoring the original zoom using the 0 key.

• int m_zoom

Provides the zoom values: 1 2 3 4, and 1, 2, 4, 8, 16.

• int m_snap

Used in setting the snap-to value in pulses, off = 1.

· int m_note_length

The default length of a note to be inserted by a right-left-click operation.

• int m scale

Setting for the music scale, can now be saved with the sequence.

· int m chord

Setting for the current chord generation; not now saved with the sequence.

int m key

Setting for the music key, can now be saved with the sequence.

· int m bgsequence

Setting for the background sequence, can now be saved with the sequence.

long m_measures

Provides the length of the sequence in measures.

• int m_ppqn

Holds a copy of the current PPQN for the sequence (and the entire MIDI file).

- int m_pp_whole
- int m_pp_eighth
- int m pp sixteenth
- sequence & m_seq

Holds a reference to the sequence that this window represents.

• Gtk::MenuBar * m_menubar

A number of user-interface objects for common.

• Gtk::Menu * m_menu_tools

The "hammer" tool button menu.

• Gtk::Menu * m_menu_zoom

Magnifying glass zoom menu.

• Gtk::Menu * m_menu_snap

Two-arrows grid-snap menu.

• Gtk::Menu * m menu note length

Notes menu for note length.

• Gtk::Menu * m_menu_length

Pattern-length "bars" menu.

• Gtk::ToggleButton * m_toggle_transpose

Transpose toggle button.

```
    Gtk::Image * m_image_transpose

     Image for transpose button.
• Gtk::Menu * m menu midich
     MIDI channel DIN menu button.
• Gtk::Menu * m_menu_midibus
     MIDI output buss menu button.
• Gtk::Menu * m_menu_data
      "Event" button to select data.
• Gtk::Menu * m menu key
      "Music key" menu button.
• Gtk::Menu * m menu scale
      "Music scale" menu button.

    Gtk::Menu * m_menu_chords

      "Chords" menu button.

    Gtk::Menu * m_menu_sequences

      "Background sequence" button.
• Gtk::Menu * m_menu_bpm
     Beats/measure numerator menu.
• Gtk::Menu * m_menu_bw
     Beat-width denominator menu.
• Gtk::Menu * m_menu_rec_vol
     Recording level "Vol" button.

    Gtk::Adjustment * m_vadjust

     Scrollbar and adjustment objects for horizontal and vertical panning.

    Gtk::Adjustment * m_hadjust

     Horizontal motion scratchpad.

    Gtk::VScrollbar * m vscroll new

     Main vertical scroll-bar.
• Gtk::HScrollbar * m_hscroll_new
     Main horizontal scroll-bar.
• seqkeys * m_seqkeys_wid
     Handles the piano-keys part of the pattern-editor user-interface.
• seqtime * m_seqtime_wid
     Handles the time-line (bar or measures) part of the pattern-editor user-interface.

    seqdata * m_seqdata_wid

     Handles the event-data part of the pattern-editor user-interface.
· seqevent * m seqevent wid
     Handles the small event part of the pattern-editor user-interface, where events can be moved and added.

    segroll * m segroll wid

     Handles the piano-roll part of the pattern-editor user-interface.

    Gtk::Button * m button Ifo

      The LFO button in the pattern editor.
• Ifownd * m_lfo_wnd
      The LFO window object used by the pattern editor.
• Gtk::Table * m table
     More user-interface elements.

    Gtk::VBox * m vbox

     Layout box for 3 h-boxes.

    Gtk::HBox * m hbox
```

Topmost menu/text dialog row.

Gtk::HBox * m hbox2

Second row of buttons.

• Gtk::Button * m_button_undo

Undo-edit button.

• Gtk::Button * m_button_redo

Redo-edit button.

Gtk::Button * m_button_quantize

Quantize-pattern button.

• Gtk::Button * m button tools

Button for the Tools menu.

• Gtk::Button * m_button_sequence

Button for Background pattern.

• Gtk::Entry * m_entry_sequence

Text for background pattern.

• Gtk::Button * m_button_bus

Button for MIDI Buss menu.

• Gtk::Entry * m_entry_bus

Text showing MIDI Buss name.

• Gtk::Button * m_button_channel

Button for the MIDI Channel.

• Gtk::Entry * m_entry_channel

Text for the MIDI Channel.

• Gtk::Button * m_button_snap

Button for the Grid-snap menu.

• Gtk::Entry * m_entry_snap

Text for selected Grid-snap.

• Gtk::Button * m_button_note_length

Button for Note-length menu.

• Gtk::Entry * m_entry_note_length

Text showing the Note-length.

• Gtk::Button * m button zoom

Button for the Zoom menu.

• Gtk::Entry * m_entry_zoom

Text for the selected Zoom.

• Gtk::Button * m_button_length

Button for pattern-length.

• Gtk::Entry * m_entry_length

Text for the pattern-length.

• Gtk::Button * m_button_key

Button for the Music Key.

• Gtk::Entry * m_entry_key

Text for selected Music Key.

• Gtk::Button * m button scale

Button for the Music Scale.

• Gtk::Entry * m_entry_scale

Text for the Music Scale.

• Gtk::Button * m button chord

Button for the current Chord.

• Gtk::Entry * m_entry_chord

Text for the current Chord.

• Gtk::Tooltips * m_tooltips

Tooltip collector for dialog.

• Gtk::Button * m_button_data

• Gtk::Entry * m_entry_data

Text for the selected Event.

Button for Event (data) menu.

• Gtk::Button * m_button_bpm

Button for Beats/Measure menu.

• Gtk::Entry * m_entry_bpm

Text for chosen Beats/Measure.

• Gtk::Button * m button bw

Button for Beat-Width menu.

Gtk::Entry * m_entry_bw

Text for chosen Beat-Width.

• Gtk::Button * m button rec vol

Button for recording volume.

Gtk::ToggleButton * m_toggle_play

Pattern-to-MIDI record button.

• Gtk::ToggleButton * m_toggle_record

MIDI-port-to-pattern button.

• Gtk::ToggleButton * m_toggle_q_rec

Quantized-record MIDI button.

• Gtk::ToggleButton * m_toggle_thru

MIDI-to-pattern-MIDI button.

• Gtk::Entry * m_entry_name

Name of the sequence.

• midibyte m_editing_status

Indicates what MIDI event/status the data window currently editing.

• midibyte m_editing_cc

Indicates what MIDI CC value the data window currently editing.

bool m_have_focus

Indicates that the focus has already been changed to this sequence.

Static Private Attributes

• static int m_initial_snap

Static data members.

- static int m_initial_note_length
- · static int m initial chord

Additional Inherited Members

13.86.1 Detailed Description

- perform
- · segroll
- seqkeys
- seqdata
- seqtime
- seqevent
- · sequence

This class has a metric ton of user-interface objects and other members.

13.86.2 Constructor & Destructor Documentation

13.86.2.1 seqedit()

If provided, override the scale, key, and background-sequence with the values stored in the file with the sequence, if they are set to non-default values. This is a new feature.

Todo Offload most of the work into an initialization function like options does.

Horizontal Gtk::Adjustment constructor: The initial value was 0 on a range from 0 to 1, with step and page increments of 1, and a page_size of 1. We can fix these values here, or create an h_adjustment() function similar to eventedit
∴v_adjustment(), which first gets called in on_realize().

Parameters

р	The performance object of which the sequence is a part.	
seq	The sequence object this window object represents.	
pos	The sequence number (pattern slot number) for this sequence and window.	
ppqn	The optional PPQN parameter for this sequence. Warning: not really used by the caller, need to square	
	that!	

13.86.2.2 \sim seqedit()

```
seq64::seqedit::~seqedit ( ) [virtual]
```

13.86.3 Member Function Documentation

13.86.3.1 set_zoom()

It is passed to the seqroll, seqtime, seqdata, and seqevent objects, as well. This function doesn't check if the zoom will change, because this function might be used to initialize the zoom of the children.

The notation for zoom in the user-interface is in pixels:ticks, but I would prefer to use pulses/pixel (pulses per pixel). Oh well. Note that this value of zoom is saved to the "user" configuration file when Sequencer64 exit.

z The prospective zoom value to set. It is applied only if between the minimum and maximum allowed zoom values, inclusive. See the usr().min zoom() and usr().max zoom() function.

13.86.3.2 set_snap()

It is passed to the segroll, sequence objects, as well.

The default initial snap is the default PPQN divided by 4, or the equivalent of a 16th note (48 ticks). The snap divisor is 192 * 4 / 48 or 16.

Parameters

The prospective snap value to set. It is checked only to make sure it is greater than 0, to avoid a numeric exception.

13.86.3.3 set_note_length()

It is passed to the seqroll object, as well.

Warning

Currently, we don't handle changes in the global PPQN after the creation of the menu. The creation of the menu hard-wires the values of note-length. To adjust for a new global PQN, we will need to store the original PPQN (m_original_ppqn = m_ppqn), and then adjust the notelength based on the new PPQN. For example if the new PPQN is twice as high as 192, then the notelength should double, though the text displayed in the "Note length" field should remain the same. However, we do adjust for a non-default PPQN at startup time.

Parameters

notelength Provides the note length in units of MIDI pulses.

13.86.3.4 set_beats_per_bar()

```
void seq64::seqedit::set_beats_per_bar (
    int bpm ) [private]
```

Todo Check if verification is needed at this point.

Parameters

bpm Provides the BPM (beats per measure) value to set.

13.86.3.5 set_beat_width()

Todo Check if verification is needed at this point.

Parameters

bw Provides the beat-width value to set.

13.86.3.6 set_transpose_image()

Parameters

istransposable	If true, set the image to the "Transpose" icon. Otherwise, set it to the "Drum" (not
	transposable) icon.

13.86.3.7 set_rec_vol()

This function also changes the button's text to match the selection, and also changes the global velocity-override setting in user_settings. Note that the setting will not be saved to the "usr" configuration file unless Sequencer64 was run with the "--user-save" option.

Parameters

recvol The setting to be made, obtained from the recording-volume ("Vol") menu.

13.86.3.8 horizontal_adjust()

A duplicate of the one in seqroll.

Parameters

step

Provides the step value to use for adjusting the horizontal scrollbar. See gui_drawingarea_gtk2::scroll_hadjust() for more information.

13.86.3.9 vertical_adjust()

A near-duplicate of the one in seqroll.

Parameters

step

Provides the step value to use for adjusting the vertical scrollbar. See gui_drawingarea_gtk2::scroll_vadjust() for more information.

13.86.3.10 horizontal_set()

Parameters

value

The desired position. Mostly this is either 0.0 or 9999999.0 (an "infinite" value to select the start or end position.

13.86.3.11 vertical_set()

value

The desired position. Mostly this is either 0.0 or 9999999.0 (an "infinite" value to select the start or end position.

13.86.3.12 set_measures()

Todo Check if verification is needed at this point.

Parameters

lim

Provides the sequence length, in measures.

13.86.3.13 apply_length()

There's an implicit "adjust-triggers = true" parameter used in sequence::set_length().

Then the seqroll, seqtime, seqdata, and seqevent objects are reset().

13.86.3.14 get_measures()

```
long seq64::seqedit::get_measures ( ) [private]
```

Todo Create a sequence::set_units() function or a sequence::get_measures() function to forward to.

13.86.3.15 set_midi_channel()

Should this change set the is-modified flag? Where should validation occur?

midichannel	The MIDI channel value to set.
user_change	True if the user made this change, and thus has potentially modified the song.

13.86.3.16 set_midi_bus()

Should this change set the is-modified flag? Where should validation against the ALSA or JACK buss limits occur?

Also, it would be nice to be able to update this display of the MIDI bus in the field if we set it from the segmenu.

Parameters

bus	The buss value to set.
user_change	True if the user made this change, and thus has potentially modified the song.

13.86.3.17 set_scale()

It is passed to the seqroll and seqkeys objects, as well. As a new feature, it is also passed to the sequence, so that it can be saved as part of the sequence data.

Note that the "initial value" for this parameter is a static variable that gets set to the new value, so that opening up another sequence causes the sequence to take on the new "initial value" as well. A feature, but should it be optional? Now it is, based on the setting of usr().global_seq_feature().

13.86.3.18 set_chord()

It is passed to the sequence, so that it can be saved as part of the sequence data.

Note that the "initial value" for this parameter is a static variable that gets set to the new value, so that opening up another sequence causes the sequence to take on the new "initial value" as well. A feature, but should it be optional? Now it is, based on the setting of usr().global_seq_feature().

13.86.3.20 set_background_sequence()

```
void seq64::seqedit::set_background_sequence (
    int seqnum ) [private]
```

As a new feature, it is also passed to the sequence, so that it can be saved as part of the sequence data, but only if less or equal to the maximum single-byte MIDI value, 127.

Note that the "initial value" for this parameter is a static variable that gets set to the new value, so that opening up another sequence causes the sequence to take on the new "initial value" as well. A feature, but should it be optional? Now it is, based on the setting of usr().global seq feature().

13.86.3.21 transpose_change_callback()

```
void seq64::seqedit::transpose_change_callback ( ) [private]
```

13.86.3.22 name_change_callback()

```
void seq64::seqedit::name_change_callback ( ) [private]
```

That name is the name the user has given to the sequence being edited.

13.86.3.23 play_change_callback()

```
void seq64::seqedit::play_change_callback ( ) [private]
```

13.86.3.24 record_change_callback()

```
void seq64::seqedit::record_change_callback ( ) [private]
```

Stazed:

```
Both record_change_callback() and thru_change_callback() will call set_sequence_input() for the same sequence. We only need to call it if it is not already set, if setting. And, we should not unset it if the m_toggle_thru->get_active() is true.
```

13.86.3.25 q_rec_change_callback()

```
void seq64::seqedit::q_rec_change_callback ( ) [private]
```

13.86.3.26 thru_change_callback()

```
void seq64::seqedit::thru_change_callback ( ) [private]
```

Stazed:

Both record_change_callback() and thru_change_callback() will call set_sequence_input() for the same sequence. We only need to call it if it is not already set, if setting. And, we should not unset it if the m_toggle_thru->get_active() is true.

13.86.3.27 undo_callback()

```
void seq64::seqedit::undo_callback ( ) [private]
```

13.86.3.28 redo_callback()

```
void seq64::seqedit::redo_callback ( ) [private]
```

13.86.3.29 set_data_type()

This function uses the hardwired array c_controller_names.

Parameters

status	The current editing status.
control	The control value. However, we really need to validate it!

13.86.3.30 update_all_windows()

```
void seq64::seqedit::update_all_windows ( ) [private]
```

```
13.86.3.31 fill_top_bar()
void seq64::seqedit::fill_top_bar ( ) [private]
```

Note that, if a non-default title for the sequence is in force, then we immediately force the focus to the sequol "widget", so that the space bar can be used to control playback, instead of immediately erasing the name of the sequence. The following commented radio-buttons were a visual way to select the modes of note editing (select, draw, and grow). These can easily be done with the left mouse button, keystrokes, or some other tricks, though.

```
13.86.3.32 create_menus()
void seq64::seqedit::create_menus ( ) [private]
```

The first menu is the Zoom menu, represented in the pattern/sequence editor by a button with a magnifying glass. The values are "pixels to ticks", where "ticks" are actually the "pulses" of "pulses per quarter note". We would prefer the notation "n" instead of "1:n", as in "n pulses per pixel".

Note that many of the setups here could be loops through data structures. The Snap menu is actually the Grid Snap button, which shows two arrows pointing to a central bar. This menu somewhat duplicates the same menu in perfedit.

To reduce the amount of written code, we now use a static array to initialize some of the sequedit menu entries. 0 denotes the separator. This same setup is used to set up both the snap and note menu, since they are exactly the same. Saves a *lot* of code.

This menu lets one set the key of the sequence, and is brought up by the button with the "golden key" image on it.

This button shows a down around for the bottom half of the time signature. It's tooltip is "Time signature. Length of beat." But it is called bw, or beat width, in the code.

This menu is shown when pressing the button at the bottom of the window that has "Vol" as its label. Let's show the numbers as well to help the user. And we'll have to document this change.

This menu sets the scale to show on the panel, and the button shows a "staircase" image. See the c_music_scales enumeration defined in the globals module.

This section sets up two different menus. The first is m_menu_length. This menu lets one set the sequence length in bars. The second menu is the m_menu_bpm, or BPM, which here means "beats per measure" (not "beats per minute").

This menu has a large number of items. I think they are filled in in code, but can also be loaded from \sim /.seq24usr. To be determined. Create the 8 sub-menus for the various ranges of controller changes, shown 16 per sub-menu.

```
13.86.3.35 popup_midibus_menu()
```

```
void seq64::seqedit::popup_midibus_menu ( ) [private]
```

The MIDI busses are obtained by getting the mastermidibus object, and iterating through the busses that it contains.

However, JACK counts the playback ports, such as "yoshimi:midi in", as "input" ports... the application outputs to the input ports. So we have to deal with that somehow.

```
13.86.3.36 popup_sequence_menu()
```

```
void seq64::seqedit::popup_sequence_menu ( ) [private]
```

It is populated with an "Off" menu entry, and a second "[0]" menu entry that pulls up a drop-down menu of all of the patterns/sequences that are present in the MIDI file for screen-set 0. If more screensets have active sequences, then their screen-set number appears in the screen-set section of the menu.

Now, at present, we can only save background sequence numbers that are less than 128, which means the sequences from 0 to 127, or the first four screen sets. Higher sequences can be selected, but, right now, they cannot be saved. We'll probably fix that at some point, low priority.

```
13.86.3.37 popup tool menu()
```

```
void seq64::seqedit::popup_tool_menu ( ) [private]
```

This button shows three sub-menus that need to be filled in by this function. All the functions accessed here seem to be implemented by the do_action() function.

```
13.86.3.38 popup_midich_menu()
```

```
void seq64::seqedit::popup_midich_menu ( ) [private]
```

13.86.3.39 create_menu_image()

13.86.3.40 timeout()

```
bool seq64::seqedit::timeout ( ) [private]
```

Note the new call to seqroll::follow_progress(). This allows the seqroll to pop to the next frame of events to continue to show the moving progress bar. Does this need to be an option? It only affects patterns longer than a measure or two, whatever the width of the seqroll window is. This is a new feature that is not in seq24.

What about seqtime? That doesn't change.

13.86.3.41 do_action()

Note that the push_undo() calls push all of the current events (in sequence::m_events) onto the stack (as a single entry).

13.86.3.42 mouse_action()

13.86.3.43 start_playing()

```
void seq64::seqedit::start_playing ( ) [private]
```

13.86.3.44 stop_playing()

```
void seq64::seqedit::stop_playing ( ) [private]
```

13.86.3.45 change_focus()

```
void seq64::seqedit::change_focus (
          bool set_it = true ) [private]
```

Similar to the same function in eventedit.

Parameters

set⊷	If true (the default value), indicates we want focus, otherwise we want to give up focus.
it	

13.86.3.46 handle_close()

```
void seq64::seqedit::handle_close ( ) [private]
```

```
13.86.3.47 on_realize()
void seq64::seqedit::on_realize ( ) [private]
13.86.3.48 on_set_focus()
void seq64::seqedit::on_set_focus (
             Widget * focus ) [private]
Only works in certain circumstances.
13.86.3.49 on_focus_in_event()
bool seq64::seqedit::on_focus_in_event (
            GdkEventFocus * ) [private]
13.86.3.50 on_focus_out_event()
bool seq64::seqedit::on_focus_out_event (
             GdkEventFocus * ) [private]
13.86.3.51 on_delete_event()
bool seq64::seqedit::on_delete_event (
```

It tells the sequence to stop recording, tells the perform object's mastermidibus to stop processing input, and sets the sequence object's editing flag to false.

Warning

This function also calls "delete this"!

GdkEventAny * event) [private]

Returns

Always returns false.

13.86.3.52 on_scroll_event()

This handles moving the scroll wheel on a mouse or do a two-fingered scrolling action on a touchpad. If no modifier key is pressed, this moves the view up or down on the "notes" coordinate, showing different piano keys. This behavior is implemented in seqkeys::on scroll event(), and is called into play by returning false here.

If the Ctrl key is pressed, then the scrolling action causes the view to zoom in or out. This behavior is implemented here.

If the Shift key is pressed, then the scrolling action moves the view horizontally on the time-line (measures-line) of the piano roll. This behavior is implemented here.

```
13.86.3.53 on_key_press_event()
```

A number of new keystrokes are processed, so that we can lessen the reliance on the mouse and work a little faster.

```
Ctrl-W keypress. This keypress closes the sequence/pattern editor window by way of calling on_delete_event(). We could apply this convention to all the other windows.
z 0 Z zoom keys. "z" zooms out, "Z" (Shift-z) zooms in, and "0" resets the zoom to the default.
Page-Up and Page-Down. Moves up and down in the piano roll.
Home and End. Page to the top or the bottom of the piano roll.
Shift-Page-Up and Shift-Page-Down. Move left and right in the piano roll.
Shift-Home and Shift-End. Page to the start or the end of the piano roll.
Ctrl-Page-Up and Ctrl-Page-Down. Mirrors the zoom-in and zoom-out capabilities of scrolling up and down with the mouse while the Ctrl key is pressed.
```

The Keypad-End key is an issue on our ASUS "gaming" laptop. Whether it is seen as a "1" or an "End" key depends on an interaction between the Shift and the Num Lock key. Annoying, takes some time to get used to.

Change Note layk 2016-10-17 Issue #46. Undoing (ctrl-z) removes two instances of history. To reproduce this bug, if one makes three notes one at a time and presses ctrl-z once only the first one remains. Same goes for moving notes. This is due to this else-if statement where we call seqroll::on_key_press_event() making first removal. This if statement is never true and seqroll::on_key_press_event() is called again as Gtk::Window::on_key_press_event(), making another m_seq.pop_undo() in seqroll. Note that the code here was an (ill-advised) attempt to avoid the pattern title field from grabbing the initial keystrokes; better to just get used to clicking the piano roll first. Finally, fixing the undo bug also let's ctrl-page-up/page-down change the zoom. Lastly, we've removed the undo here... seqroll already handles both undo and redo keystrokes.

Change Note ca 2016-10-18 Issue #46. In addition to layk's fixes, we have to properly determine if we're inside the "Sequence Name" ("GtkEntry") field, as opposed to the "GtkDrawingArea" field, to avoid grabbing and using keystrokes intended for the text-entry field. We may have to rethink the whole seqroll vs. seqedit key-press process at some point, as this is a bit too tricky. Please note that the name "gtkmm__GtkEntry" likely applies only to GNU's C++ compiler, g++. This will be an issue in any port to Microsoft's C++ compiler.

Parameters

ev Provides the keystroke event to be handled.

Returns

Returns true if we handled the keystroke here. Otherwise, returns the value of Gtk::Window::on_key_press-event(ev).

13.86.4 Field Documentation

13.86.4.1 seqmenu

```
friend seq64::seqedit::seqmenu [private]
```

13.86.4.2 m_initial_snap

```
int seq64::seqedit::m_initial_snap [static], [private]
```

These items apply to all of the instances of seqedit, and are passed on to the following constructors:

- · seqdata
- · seqevent
- · segroll
- · seqtime

The snap and note-length defaults would be good to write to the "user" configuration file. The scale and key would be nice to write to the proprietary section of the MIDI song. Or, even more flexibly, to each sequence, if that makes sense to do, since all tracks would generally be in the same key. Right, Charles Ives?

Note that, currently, that some of these "initial values" are modified, so that they are "contagious". That is, the next sequence to be opened in the sequence editor will adopt these values. This is a long-standing feature of Seq24, but strikes us as a bit surprising.

Change Note ca 2016-04-10 If we just double the PPQN, then the snap divisor becomes 32, and the snap interval is a 32nd note. We would like to keep it at a 16th note. We correct the snap ticks to the actual PPQN ratio.

13.86.4.3 m_initial_note_length

```
int seq64::seqedit::m_initial_note_length [static], [private]
```

13.86.4.4 m_initial_chord

```
int seq64::seqedit::m_initial_chord [static], [private]
```

```
13.86.4.5 m_initial_zoom
const int seq64::seqedit::m_initial_zoom [private]
13.86.4.6 m_zoom
int seq64::seqedit::m_zoom [private]
The value of zoom is the same as the number of pixels per tick on the piano roll.
13.86.4.7 m_snap
int seq64::seqedit::m_snap [private]
13.86.4.8 m_note_length
int seq64::seqedit::m_note_length [private]
13.86.4.9 m_scale
int seq64::seqedit::m_scale [private]
13.86.4.10 m_chord
int seq64::seqedit::m_chord [private]
13.86.4.11 m_key
int seq64::seqedit::m_key [private]
13.86.4.12 m_bgsequence
```

int seq64::seqedit::m_bgsequence [private]

```
13.86.4.13 m_measures
long seq64::seqedit::m_measures [private]
13.86.4.14 m_ppqn
int seq64::seqedit::m_ppqn [private]
13.86.4.15 m_pp_whole
int seq64::seqedit::m_pp_whole [private]
13.86.4.16 m_pp_eighth
int seq64::seqedit::m_pp_eighth [private]
13.86.4.17 m_pp_sixteenth
int seq64::seqedit::m_pp_sixteenth [private]
13.86.4.18 m_seq
sequence& seq64::seqedit::m_seq [private]
13.86.4.19 m_menubar
```

Many of these are menu items, and are associated with buttons that, when pressed, bring up the menu for display and selection of its entries. The top bar with menu buttons.

Gtk::MenuBar* seq64::seqedit::m_menubar [private]

```
13.86.4.20 m_menu_tools
Gtk::Menu* seq64::seqedit::m_menu_tools [private]
13.86.4.21 m_menu_zoom
Gtk::Menu* seq64::seqedit::m_menu_zoom [private]
13.86.4.22 m_menu_snap
Gtk::Menu* seq64::seqedit::m_menu_snap [private]
13.86.4.23 m_menu_note_length
Gtk::Menu* seq64::seqedit::m_menu_note_length [private]
13.86.4.24 m_menu_length
Gtk::Menu* seq64::seqedit::m_menu_length [private]
13.86.4.25 m_toggle_transpose
Gtk::ToggleButton* seq64::seqedit::m_toggle_transpose [private]
13.86.4.26 m_image_transpose
Gtk::Image* seq64::seqedit::m_image_transpose [private]
13.86.4.27 m_menu_midich
Gtk::Menu* seq64::seqedit::m_menu_midich [private]
```

```
13.86.4.28 m_menu_midibus
Gtk::Menu* seq64::seqedit::m_menu_midibus [private]
13.86.4.29 m_menu_data
Gtk::Menu* seq64::seqedit::m_menu_data [private]
13.86.4.30 m_menu_key
Gtk::Menu* seq64::seqedit::m_menu_key [private]
13.86.4.31 m_menu_scale
Gtk::Menu* seq64::seqedit::m_menu_scale [private]
13.86.4.32 m_menu_chords
Gtk::Menu* seq64::seqedit::m_menu_chords [private]
13.86.4.33 m_menu_sequences
Gtk::Menu* seq64::seqedit::m_menu_sequences [private]
13.86.4.34 m_menu_bpm
Gtk::Menu* seq64::seqedit::m_menu_bpm [private]
13.86.4.35 m_menu_bw
```

Gtk::Menu* seq64::seqedit::m_menu_bw [private]

```
13.86.4.36 m_menu_rec_vol
Gtk::Menu* seq64::seqedit::m_menu_rec_vol [private]
13.86.4.37 m_vadjust
Gtk::Adjustment* seq64::seqedit::m_vadjust [private]
Vertical position descriptor.
13.86.4.38 m_hadjust
Gtk::Adjustment* seq64::seqedit::m_hadjust [private]
13.86.4.39 m_vscroll_new
Gtk::VScrollbar* seq64::seqedit::m_vscroll_new [private]
13.86.4.40 m_hscroll_new
Gtk::HScrollbar* seq64::seqedit::m_hscroll_new [private]
13.86.4.41 m_segkeys_wid
seqkeys* seq64::seqedit::m_seqkeys_wid [private]
This item draws the piano-keys at the left of the segedit window.
13.86.4.42 m_seqtime_wid
seqtime* seq64::seqedit::m_seqtime_wid [private]
This is the location where the measure numbers and the END marker are shown.
13.86.4.43 m_seqdata_wid
seqdata* seq64::seqedit::m_seqdata_wid [private]
```

This is the area at the bottom of the window that shows value lines for the selected kinds of events.

```
13.86.4.44 m_seqevent_wid
seqevent* seq64::seqedit::m_seqevent_wid [private]
13.86.4.45 m_seqroll_wid
seqroll* seq64::seqedit::m_seqroll_wid [private]
13.86.4.46 m_button_lfo
Gtk::Button* seq64::seqedit::m_button_lfo [private]
This item will always be an optional part of the build, enabled by defining SEQ64_STAZED_LFO_SUPPORT.
13.86.4.47 m_lfo_wnd
lfownd* seq64::seqedit::m_lfo_wnd [private]
This item get the seqdata window hooked into it, and so must follow that item in the C++ initializer list.
13.86.4.48 m_table
Gtk::Table* seq64::seqedit::m_table [private]
These items provide a number of buttons and text-entry fields, as well as their layout. The layout table for editor.
13.86.4.49 m_vbox
Gtk::VBox* seq64::seqedit::m_vbox [private]
13.86.4.50 m_hbox
Gtk::HBox* seq64::seqedit::m_hbox [private]
13.86.4.51 m_hbox2
Gtk::HBox* seq64::seqedit::m_hbox2 [private]
```

```
13.86.4.52 m_button_undo
Gtk::Button* seq64::seqedit::m_button_undo [private]
13.86.4.53 m_button_redo
Gtk::Button* seq64::seqedit::m_button_redo [private]
13.86.4.54 m_button_quantize
Gtk::Button* seq64::seqedit::m_button_quantize [private]
13.86.4.55 m_button_tools
Gtk::Button* seq64::seqedit::m_button_tools [private]
13.86.4.56 m_button_sequence
Gtk::Button* seq64::seqedit::m_button_sequence [private]
13.86.4.57 m_entry_sequence
Gtk::Entry* seq64::seqedit::m_entry_sequence [private]
13.86.4.58 m_button_bus
Gtk::Button* seq64::seqedit::m_button_bus [private]
13.86.4.59 m_entry_bus
```

Gtk::Entry* seq64::seqedit::m_entry_bus [private]

```
13.86.4.60 m_button_channel
Gtk::Button* seq64::seqedit::m_button_channel [private]
13.86.4.61 m_entry_channel
Gtk::Entry* seq64::seqedit::m_entry_channel [private]
13.86.4.62 m_button_snap
Gtk::Button* seq64::seqedit::m_button_snap [private]
13.86.4.63 m_entry_snap
Gtk::Entry* seq64::seqedit::m_entry_snap [private]
13.86.4.64 m_button_note_length
Gtk::Button* seq64::seqedit::m_button_note_length [private]
13.86.4.65 m_entry_note_length
Gtk::Entry* seq64::seqedit::m_entry_note_length [private]
13.86.4.66 m_button_zoom
Gtk::Button* seq64::seqedit::m_button_zoom [private]
```

Generated by Doxygen

13.86.4.67 m_entry_zoom

Gtk::Entry* seq64::seqedit::m_entry_zoom [private]

```
13.86.4.68 m_button_length
Gtk::Button* seq64::seqedit::m_button_length [private]
13.86.4.69 m_entry_length
Gtk::Entry* seq64::seqedit::m_entry_length [private]
13.86.4.70 m_button_key
Gtk::Button* seq64::seqedit::m_button_key [private]
13.86.4.71 m_entry_key
Gtk::Entry* seq64::seqedit::m_entry_key [private]
13.86.4.72 m_button_scale
Gtk::Button* seq64::seqedit::m_button_scale [private]
13.86.4.73 m_entry_scale
Gtk::Entry* seq64::seqedit::m_entry_scale [private]
13.86.4.74 m_button_chord
Gtk::Button* seq64::seqedit::m_button_chord [private]
13.86.4.75 m_entry_chord
```

Gtk::Entry* seq64::seqedit::m_entry_chord [private]

```
13.86.4.76 m_tooltips
Gtk::Tooltips* seq64::seqedit::m_tooltips [private]
13.86.4.77 m_button_data
Gtk::Button* seq64::seqedit::m_button_data [private]
13.86.4.78 m_entry_data
Gtk::Entry* seq64::seqedit::m_entry_data [private]
13.86.4.79 m_button_bpm
Gtk::Button* seq64::seqedit::m_button_bpm [private]
13.86.4.80 m_entry_bpm
Gtk::Entry* seq64::seqedit::m_entry_bpm [private]
13.86.4.81 m_button_bw
Gtk::Button* seq64::seqedit::m_button_bw [private]
13.86.4.82 m_entry_bw
Gtk::Entry* seq64::seqedit::m_entry_bw [private]
13.86.4.83 m_button_rec_vol
```

Gtk::Button* seq64::seqedit::m_button_rec_vol [private]

```
13.86.4.84 m_toggle_play
Gtk::ToggleButton* seq64::seqedit::m_toggle_play [private]
13.86.4.85 m_toggle_record
Gtk::ToggleButton* seq64::seqedit::m_toggle_record [private]
13.86.4.86 m_toggle_q_rec
Gtk::ToggleButton* seq64::seqedit::m_toggle_q_rec [private]
13.86.4.87 m_toggle_thru
Gtk::ToggleButton* seq64::seqedit::m_toggle_thru [private]
13.86.4.88 m_entry_name
Gtk::Entry* seq64::seqedit::m_entry_name [private]
13.86.4.89 m_editing_status
midibyte seq64::seqedit::m_editing_status [private]
13.86.4.90 m_editing_cc
midibyte seq64::seqedit::m_editing_cc [private]
13.86.4.91 m_have_focus
bool seq64::seqedit::m_have_focus [private]
```

13.87 seq64::seqevent Class Reference

Implements the piano event drawing area.

Inheritance diagram for seq64::seqevent:



Public Member Functions

seqevent (perform &p, sequence &seq, int zoom, int snap, seqdata &seqdata_wid, Gtk::Adjustment &hadjust, int ppqn=SEQ64_USE_DEFAULT_PPQN)

Principal constructor.

virtual ∼seqevent ()

Let's provide a do-nothing virtual destructor.

· void reset ()

This function basically resets the whole widget as if it was realized again.

• void redraw ()

Adjusts the scrolling offset for ticks, updates the pixmap, and draws it on the window.

void set zoom (int zoom)

Sets zoom to the given value, and resets if the value ended up being changed.

void set_snap (int snap)

'Setter' function for member m_snap Simply sets the snap member.

void set_data_type (midibyte status, midibyte control)

Sets the status to the given parameter, and the CC value to the given optional control parameter, which defaults to 0.

void update_sizes ()

If the window is realized, this function creates a pixmap with window dimensions, the updates the pixmap, and queues up a redraw.

void draw_background ()

This function updates the background.

void draw_events_on_pixmap ()

This function fills the main pixmap with events.

void draw_pixmap_on_window ()

This function currently just queues up a draw operation for the pixmap.

void draw_selection_on_window ()

Draw the selected events on the window.

void update_pixmap ()

Redraws the background pixmap on the main pixmap, then puts the events on.

Private Member Functions

virtual void force_draw ()

Forces a draw on the current drawable area of the window.

• int idle_redraw ()

Implements redraw while idling.

void x_to_w (int x1, int x2, int &x, int &w)

This function checks the mins / maxes.

· void drop_event (midipulse tick)

Drops (adds) an event at the given tick.

void draw_events_on (Glib::RefPtr< Gdk::Drawable > draw)

Draws events on the given drawable object.

• void start_paste ()

Starts a paste operation.

• void change_horz ()

Changes the horizontal scrolling offset for ticks, then updates the pixmap and forces a redraw.

void convert x (int x, midipulse &tick)

Takes the screen x coordinate, multiplies it by the current zoom, and returns the tick value in the given parameter.

void convert_t (midipulse tick, int &x)

Converts the given tick value to an x corrdinate, based on the zoom, and returns it via the second parameter.

void snap y (int &y)

This function performs a 'snap' on y.

void snap_x (int &x)

This function performs a 'snap' on x.

void on_realize ()

Implements the on-realize callback.

bool on_expose_event (GdkEventExpose *ev)

Implements the on-expose event callback.

bool on_button_press_event (GdkEventButton *ev)

Implements the on-button-press event callback.

bool on_button_release_event (GdkEventButton *ev)

Implements the on-button-release event callback.

bool on_motion_notify_event (GdkEventMotion *ev)

Implements the on-motion-notify event callback.

bool on_focus_in_event (GdkEventFocus *)

Responds to a focus event by setting the HAS_FOCUS flag.

bool on_focus_out_event (GdkEventFocus *)

Responds to a unfocus event by resetting the HAS_FOCUS flag.

bool on_key_press_event (GdkEventKey *p0)

Implements the key-press event callback function.

void on_size_allocate (Gtk::Allocation &)

Implements the on-size-allocate event callback.

Private Attributes

FruitySeqEventInput m fruity interaction

Provides the mouse-handling paradigm for the fruity interaction.

Seq24SeqEventInput m_seq24_interaction

Provides the normal mouse-handling for Sequencer64.

• sequence & m_seq

Provides a reference to the sequence whose data is represented in this sequent object.

• int m zoom

Zoom setting, means that one pixel == m_zoom ticks.

int m_snap

The grid-snap setting for the event bar grid.

• int m_ppqn

The value to use for the PPQN for this sequence.

GdkRectangle m_old

Used in drawing the event selection in the thing event row.

• GdkRectangle m_selected

Used in moving and pasting the selected events in the thin event row.

int m_scroll_offset_ticks

Provides the offset of the ticks in the event view based on where the scroll-bar has moved the view "window".

int m_scroll_offset_x

Provides the offset of the pixels in the event view based on where the scroll-bar has moved the view "window".

· seqdata & m_seqdata_wid

The data view that parallels this event view.

bool m selecting

Used when highlighting a bunch of events.

· bool m moving init

Used externally by the fruityseq and seq24seq modules, to initialize the act of moving events.

bool m moving

Indicates that this pane is in the act of moving a selection.

· bool m_growing

Used externally by the fruityseq and seq24seq modules, when growing the event duration.

bool m_painting

Used externally by the fruityseq and seq24seq modules, in painting the selected events.

• bool m_paste

Indicates that we've selected some events and are in paste mode.

• int m_move_snap_offset_x

Used externally by the fruityseq and seq24seq modules, in snapping.

• midibyte m_status

Indicates what is the data window currently editing.

• midibyte m_cc

Indicates what is the data window currently editing.

Friends

- struct FruitySeqEventInput
- struct Seq24SeqEventInput

Additional Inherited Members

13.87.1 Constructor & Destructor Documentation

13.87.1.1 seqevent()

```
seq64::seqevent::seqevent (
    perform & p,
    sequence & seq,
    int zoom,
    int snap,
    seqdata & seqdata_wid,
    Gtk::Adjustment & hadjust,
    int ppqn = SEQ64_USE_DEFAULT_PPQN )
```

Parameters

р	The "parent" perform object controlling all of the sequences.
seq	The current sequence operated on by this object.
zoom	The initial zoom value.
snap	The initial snap value.
seqdata_wid	The data pane that this event pane is associated with.
hadjust	The horizontal scroll-bar.
ppqn	The initial PPQN value.

```
13.87.1.2 ∼ seqevent()
virtual seq64::seqevent::\sim seqevent ( ) [inline], [virtual]
13.87.2 Member Function Documentation
13.87.2.1 reset()
void seq64::seqevent::reset ( )
Basically identical to seqtime::reset().
13.87.2.2 redraw()
void seq64::seqevent::redraw ( )
Somewhat similar to seqroll::redraw().
13.87.2.3 set_zoom()
void seq64::seqevent::set_zoom (
              int z )
Parameters
     The desired zoom value, assumed to be validated already. See the seqedit::set_zoom() function.
13.87.2.4 set_snap()
void seq64::seqevent::set_snap (
              int snap ) [inline]
The parameter is not validated.
```

Then redraws.

13.87.2.5 set_data_type()

status	The status/event byte to set. For example, EVENT_NOTE_ON and EVENT_NOTE off. This byte should have the channel nybble cleared.
control	The MIDI CC byte to set.

```
13.87.2.6 update_sizes()
```

```
void seq64::seqevent::update_sizes ( )
```

This ends up filling the background with dotted lines, etc.

13.87.2.7 draw_background()

```
void seq64::seqevent::draw_background ( )
```

It sets the foreground to white, draws the rectangle, in order to clear the pixmap. The build-time option SEQ64← _SOLID_PIANOROLL_GRID causes solid lines to be drawn, in gray, instead of dotted black lines, for a smoother look

Also, as a trial option, if the current data type is EVENT_NOTE_ON, EVENT_NOTE_OFF, and EVENT_AFTER

TOUCH, we draw the background in light grey to remind the user that there are issues in copying or moving these events around (unlinked) by themselves.

13.87.2.8 draw_events_on_pixmap()

```
void seq64::seqevent::draw_events_on_pixmap ( )
```

13.87.2.9 draw_pixmap_on_window()

```
void seq64::seqevent::draw_pixmap_on_window ( )
```

Old comments:

It then tells event to do the same. We changed something on this window, and chances are we need to update the event widget as well and update our velocity window.

13.87.2.10 draw_selection_on_window()

```
void seq64::seqevent::draw_selection_on_window ( )
```

13.87.2.11 update_pixmap()

```
void seq64::seqevent::update_pixmap ( )
```

13.87.2.12 force_draw()

```
void seq64::seqevent::force_draw ( ) [private], [virtual]
```

Reimplemented from seq64::gui_drawingarea_gtk2.

13.87.2.13 idle_redraw()

```
int seq64::seqevent::idle_redraw ( ) [private]
```

Who calls this routine? Probably the default timer routine, but not sure.

Returns

Always returns true.

13.87.2.14 x_to_w()

Then it fills in x and the width.

Parameters

	x1	The "left" x value.
	x2	The "right" x value.
out	Х	The destination for the converted x value.
out	W	The destination for the converted width value.

13.87.2.15 drop_event()

```
void seq64::seqevent::drop_event (
          midipulse tick ) [private]
```

It sets the first byte properly for after-touch, program-change, channel-pressure, and pitch-wheel. The type of event is determined by m_status.

Parameters

tick The destination time (division, pulse, tick) for the event to be dropped at.

13.87.2.16 draw_events_on()

Very similar to seqdata::draw_events_on().

Parameters

drawable	The given drawable object.
----------	----------------------------

13.87.2.17 start_paste()

```
void seq64::seqevent::start_paste ( ) [private]
```

It gets the clipboard box that selected elements are in, makes a coordinate conversion, and then, sets the m_{\leftarrow} selected rectangle to hold the (x,y,w,h) of the selected events.

13.87.2.18 change_horz()

```
void seq64::seqevent::change_horz ( ) [private]
```

Very similar to seqroll::change_horz(). Basically identical to seqdata::change_horz().

13.87.2.19 convert_x()

Why not just return it normally?

Parameters

		Х	The x (pixel) value to convert.
ou	t	tick	The destination for the converted x value.

13.87.2.20 convert_t()

Why not just return it normally?

Parameters

	tick	The tick (pulse) value to convert.
out	X	The destination for the converted tick value.

13.87.2.21 snap_y()

Parameters

out	У	The return parameter for the conversion. Why not just return the value?	
-----	---	---	--

13.87.2.22 snap_x()

- snap = number pulses to snap to
- m_zoom = number of pulses per pixel
- Therefore snap / m_zoom = number of pixels to snap to.

Parameters

01	ut	X	The output destination for the snapped x value.

13.87.2.23 on_realize()

```
void seq64::seqevent::on_realize ( ) [private]
```

It calls the base-class version, and then allocates additional resource not allocated in the constructor. Finally, it connects up the change horz function.

13.87.2.24 on_expose_event()

Parameters

ev The expose event.

13.87.2.25 on_button_press_event()

It distinguishes between the Seq24 and Fruity varieties of mouse interaction.

Odd. In the legacy code, each case fell through to the next case to the "default" case! We will assume for now that this is incorrect.

Note that returning "true" from a Gtkmm event-handler stops the propagation of the event to higher-level widgets. The Fruity and Seq24 event handlers return true, always. In the legacy code, though, the fall-through code caused false to be returned, always. Not sure what effect this had. Added some fixes, but then commented them out until better testing can be done.

Parameters

```
ev The button event.
```

Returns

Returns true if the button-press was handled.

13.87.2.26 on_button_release_event()

It distinguishes between the Seq24 and Fruity varieties of mouse interaction.

Odd. The fruity case fell through to the Seq24 case. We will assume for now that this is correct. Added some fixes, but then commented them out until better testing can be done.

```
ev The button event.
```

Returns

Returns true if the button-press was handled.

13.87.2.27 on_motion_notify_event()

It distinguishes between the Seq24 and Fruity varieties of mouse interaction.

Odd. The fruity case fell through to the Seq24 case. We will assume for now that this is correct. Added some fixes, but then commented them out until better testing can be done.

Parameters

```
ev The motion event.
```

Returns

Returns true if the motion-event was handled.

13.87.2.28 on_focus_in_event()

Parameter "ev" is the focus event, unused.

Returns

Always returns false.

13.87.2.29 on_focus_out_event()

Parameter "ev" is the focus event, unused.

Returns

Always returns false.

13.87.2.30 on_key_press_event()

It handles deleted a selection via the Backspace or Delete keys, cut via Ctrl-X, copy via Ctrl-C, paste via Ctrl-V, and undo via Ctrl-Z.

Would be nice to provide redo functionality via Ctrl-Y. :-)

Parameters

ev	The key-press event.
----	----------------------

Returns

Returns true if an event was handled. Only some of the handled events also cause the perform modification flag to be set as a side-effect.

13.87.2.31 on_size_allocate()

The m_window_x and m_window_y values are set to the allocation width and height, respectively.

Parameters

a The allocation to be processed.

13.87.3 Friends And Related Function Documentation

13.87.3.1 FruitySeqEventInput

```
friend struct FruitySeqEventInput [friend]
```

13.87.3.2 Seq24SeqEventInput

friend struct Seq24SeqEventInput [friend]

13.87.4 Field Documentation

```
13.87.4.1 m_fruity_interaction
FruitySeqEventInput seq64::seqevent::m_fruity_interaction [private]
Why should we need both at the same time? Just load the one that is specified in the configuration.
13.87.4.2 m_seq24_interaction
Seq24SeqEventInput seq64::seqevent::m_seq24_interaction [private]
13.87.4.3 m_seq
sequence& seq64::seqevent::m_seq [private]
13.87.4.4 m_zoom
int seq64::seqevent::m_zoom [private]
13.87.4.5 m_snap
int seq64::seqevent::m_snap [private]
Same meaning as for the piano roll. This value is the denominator of the note size used for the snap.
13.87.4.6 m_ppqn
int seq64::seqevent::m_ppqn [private]
Used in snap and zoom scaling.
13.87.4.7 m_old
GdkRectangle seq64::seqevent::m_old [private]
```

```
13.87.4.8 m_selected
GdkRectangle seq64::seqevent::m_selected [private]
13.87.4.9 m_scroll_offset_ticks
int seq64::seqevent::m_scroll_offset_ticks [private]
13.87.4.10 m_scroll_offset_x
int seq64::seqevent::m_scroll_offset_x [private]
Set to m_scroll_offset_ticks divided by m_zoom.
13.87.4.11 m_seqdata_wid
seqdata& seq64::seqevent::m_seqdata_wid [private]
13.87.4.12 m_selecting
bool seq64::seqevent::m_selecting [private]
13.87.4.13 m_moving_init
bool seq64::seqevent::m_moving_init [private]
13.87.4.14 m_moving
bool seq64::seqevent::m_moving [private]
WARNING: This operation seems to have a bug. It makes the events very very long. This bug exists in Seq24.
13.87.4.15 m_growing
bool seq64::seqevent::m_growing [private]
```

Does growing work in this view? Need to do some better testing.

13.87.4.16 m_painting

```
bool seq64::seqevent::m_painting [private]
```

13.87.4.17 m_paste

```
bool seq64::seqevent::m_paste [private]
```

13.87.4.18 m_move_snap_offset_x

```
int seq64::seqevent::m_move_snap_offset_x [private]
```

13.87.4.19 m_status

```
midibyte seq64::seqevent::m_status [private]
```

The current status/event byte.

13.87.4.20 m_cc

```
midibyte seq64::seqevent::m_cc [private]
```

The current MIDI CC value.

13.88 seq64::seqkeys Class Reference

This class implements the left side piano of the pattern/sequence editor.

Inheritance diagram for seq64::seqkeys:



Public Member Functions

- seqkeys (sequence &seq, perform &p, Gtk::Adjustment &vadjust)
 - Principal constructor.
- virtual ∼seqkeys ()
 - Let's provide a do-nothing virtual destructor.
- void set_scale (int scale)

Sets the musical scale, then resets.

void set_key (int key)

Sets the musical key, then resets.

void set_hint_key (int key)

Sets a key to grey so that it can serve as a scale hint.

void set_hint_state (bool state)

Sets the hint state to the given value.

Private Member Functions

virtual void force_draw ()

Forces a draw operation on the whole window.

void set_listen_button_press (GdkEventButton *ev)

Sneaky accessors for the segroll friend.

- void set_listen_button_release (GdkEventButton *ev)
- void set_listen_motion_notify (GdkEventMotion *ev)
- · void draw_area ()

Draws the updated pixmap on the drawable area of the window where the keys' location is hardwired.

void update_pixmap ()

Updates the pixmaps to prepare it for the next draw operation.

void convert_y (int y, int ¬e)

Takes the screen y coordinate, and returns the note value in the second parameter.

void draw key (int key, bool state)

Draws the given key according to the given state.

· void change_vert ()

Changes the y offset of the scrolling, and the forces a draw.

- void update_sizes ()
- · void reset ()

Resetting the keys view updates the pixmap and queues up a draw operation.

· bool is black key (int key) const

Detects a black key.

void on_realize ()

Implements the on-realize event.

bool on_expose_event (GdkEventExpose *ev)

Implements the on-expose event, by drawing on the window.

• bool on_button_press_event (GdkEventButton *ev)

Implements the on-button-press event callback.

bool on_button_release_event (GdkEventButton *ev)

Implements the on-button-release event callback.

bool on_motion_notify_event (GdkEventMotion *p0)

Implements the on-motion-notify event handler.

bool on_enter_notify_event (GdkEventCrossing *p0)

Implements the on-enter notification event handler.

bool on_leave_notify_event (GdkEventCrossing *p0)

Implements the on-leave notification event handler.

• bool on_scroll_event (GdkEventScroll *ev)

Implements the on-scroll-event notification event handler.

void on_size_allocate (Gtk::Allocation &)

Implements the on-size-allocation notification event handler.

Private Attributes

• sequence & m_seq

The sequence object that the keys pane will be using.

int m_scroll_offset_key

Provides the value of the current top key in the keys pane.

· int m_scroll_offset_y

Provides the value of the current top key in the keys pane in units of relative pixels.

• bool m_hint_state

Indicates if a piano key is set to indicate where on the pitch scale the mouse cursor is sitting.

int m_hint_key

Indicates the current y-value of the mouse pointer in units of key value.

· bool m_keying

Set to true while the left mouse button is being pressed.

• int m_keying_note

The note to be played when selected in the seqkeys pane.

• int m scale

This member holds the scale value for the musical scale for the current edit of the sequence.

int m key

This member holds the key value for the musical key for the current edit of the sequence.

· bool m_show_octave_letters

The default value is to show the octave letters on the vertical virtual keyboard.

Friends

- · class segroll
- class FruitySeqRollInput

Additional Inherited Members

13.88.1 Detailed Description

Note the friends of this class, seqroll and FruitySeqRollInput. Where is Seq24SeqRollInput? Gone. It has been folded back into seqroll.

13.88.2 Constructor & Destructor Documentation

13.88.2.1 seqkeys()

seq	Provides the sequence object to which this seqkeys pane is associated.
р	Provides the performance object to which this seqkeys pane (and all sequences) are associated.
vadjust	The range object for the vertical scrollbar linked to the position in the seqkeys pane.

13.88.2.2 \sim seqkeys()

```
virtual seq64::seqkeys::\simseqkeys ( ) [inline], [virtual]
```

13.88.3 Member Function Documentation

13.88.3.1 set_scale()

This function is called by the seqedit class.

Parameters

ale The musical scale value to be set.	
--	--

13.88.3.2 set_key()

Parameters

key The musical key value to be set.

13.88.3.3 set_hint_key()

If m_hint_state is true, the key is drawn (again).

key The key value to set the hint-key to.

13.88.3.4 set_hint_state()

Parameters

state | Provides the value for hinting, where true == on, false == off.

13.88.3.5 force_draw()

```
void seq64::seqkeys::force_draw ( ) [private], [virtual]
```

Unlike most other overridden versions of force_draw(), this one does not call the base-class version.

Reimplemented from seq64::gui_drawingarea_gtk2.

13.88.3.6 set_listen_button_press()

From the stazed code.

Parameters

ev The event to be forwarded from the segroll.

13.88.3.7 set_listen_button_release()

```
void seq64::seqkeys::set_listen_button_release (
    GdkEventButton * ev ) [inline], [private]
```

13.88.3.8 set_listen_motion_notify()

void seq64::seqkeys::update_pixmap () [private]

This function draws the keys, which range from 0 to 127 (SEQ64_MIDI_COUNT_MAX - 1 = c_num_keys - 1). Every octave, a key letter and number (e.g. "C4") is shown. The letter is adjusted to match the current scale (e.g. "C#4").

We want to support an option to show the key number rather than the note letter/number combination, and perhaps to toggle between them. The current difficulty is that the fonts used are just a little to high to fit within the vertical limits of each key. We really don't want to change the vertical size at this time, so we just print every other note value.

Also note that this algorithm draws from the top down, so we have to account for that.

13.88.3.11 convert_y()

13.88.3.10 update_pixmap()

Parameters

	У	The y (vertical) screen coordinate to convert.
out	note	The destination for the note calculation. This would be better as a return value.

13.88.3.12 draw_key()

It accounts for the black keys and the white keys, and for the highlighting of the active key.

key	The key to be drawn.	
state	How the key is to be drawn, where false == normal, true == grayed. A key is greyed when the mouse	
	cursor is at the same vertical location on the piano as the key.	

13.88.3.13 change_vert()

```
void seq64::seqkeys::change_vert ( ) [private]
```

Weird, in seq24 and here, the following was used, completely by accident! We fixed it, but must beware!

```
\label{eq:m_scroll_offset_key} $$ m_scroll_offset_key * c_key_y, $$ // comma operator!!! force_draw();
```

13.88.3.14 update_sizes()

```
void seq64::seqkeys::update_sizes ( ) [private]
```

13.88.3.15 reset()

```
void seq64::seqkeys::reset ( ) [private]
```

13.88.3.16 is_black_key()

Parameters

key	The key to analyze.
-----	---------------------

Returns

Returns true if the key is black (value 1, 3, 6, 8, or 10).

```
13.88.3.17 on_realize()
void seq64::seqkeys::on_realize ( ) [private]
```

Call the base-class version and then allocates resources that could not be allocated in the constructor. It connects the change_vert() function and then calls it.

13.88.3.18 on_expose_event()

Parameters

ev The expose-event object.

13.88.3.19 on_button_press_event()

It handles the left and right buttons. The left button, pressed on the piano keyboard, causes m_keying to be set to true, and the given note to play. The right button toggles the note display between letter/number and MIDI note number.

Parameters

ev The mouse-button event to use.

Returns

Always returns true.

13.88.3.20 on_button_release_event()

```
bool seq64::seqkeys::on_button_release_event (
    GdkEventButton * ev ) [private]
```

It currently handles only the left button, and only if m_keying is true.

This function is used after pressing on one of the keys on the left-side piano keyboard, to make it play, and turns off the playing of the note.

```
ev The button-event.
```

Returns

Always returns true.

13.88.3.21 on_motion_notify_event()

```
bool seq64::seqkeys::on_motion_notify_event ( {\tt GdkEventMotion} \ * \ p0 \ ) \quad [{\tt private}]
```

This allows rolling down the keyboard, playing the notes one-by-one.

Parameters

```
p0 The motion event.
```

Returns

Always returns false.

13.88.3.22 on_enter_notify_event()

This greys the current key.

13.88.3.23 on_leave_notify_event()

This un-greys the current key and stops playing the note.

13.88.3.24 on_scroll_event()

Note that there is no usage of the modifier keys (e.g. Shift or Ctrl). Compare this function to seqedit::on_scroll_\cdot event().

ev Provides the direction of the scroll event.

Returns

Always returns true.

13.88.3.25 on_size_allocate()

Parameters

all Provies the allocation and its width and height.

13.88.4 Friends And Related Function Documentation

13.88.4.1 seqroll

friend class seqroll [friend]

13.88.4.2 FruitySeqRollInput

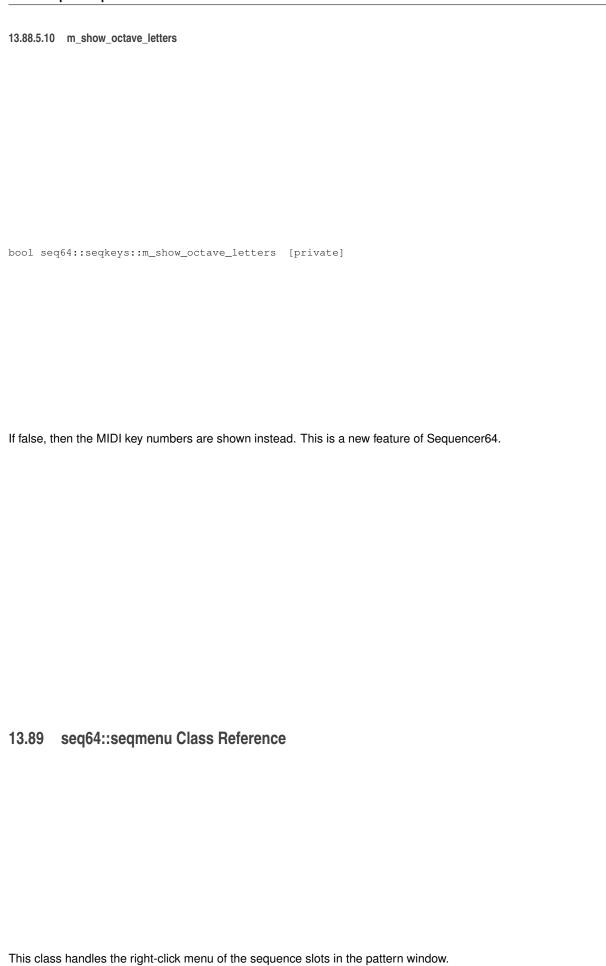
friend class FruitySeqRollInput [friend]

13.88.5 Field Documentation

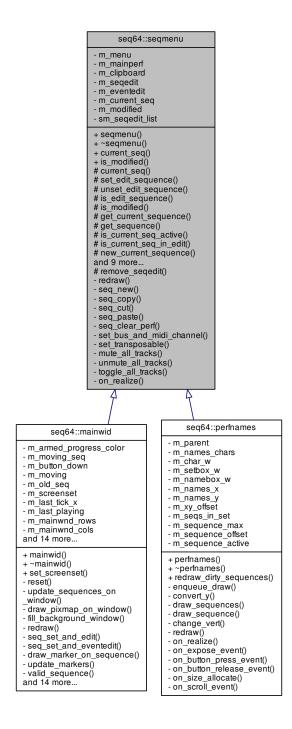
13.88.5.1 m_seq

sequence& seq64::seqkeys::m_seq [private]

```
13.88.5.2 m_scroll_offset_key
int seq64::seqkeys::m_scroll_offset_key [private]
Modified in change_vert().
13.88.5.3 m_scroll_offset_y
int seq64::seqkeys::m_scroll_offset_y [private]
Modified in change_vert().
13.88.5.4 m_hint_state
bool seq64::seqkeys::m_hint_state [private]
13.88.5.5 m_hint_key
int seq64::seqkeys::m_hint_key [private]
13.88.5.6 m_keying
bool seq64::seqkeys::m_keying [private]
Used in playing the sound for each note as it is clicked in the seqkeys pane.
13.88.5.7 m_keying_note
int seq64::seqkeys::m_keying_note [private]
13.88.5.8 m_scale
int seq64::seqkeys::m_scale [private]
13.88.5.9 m_key
int seq64::seqkeys::m_key [private]
```



Inheritance diagram for seq64::seqmenu:



Public Member Functions

seqmenu (perform &a_p)

Principal constructor.

virtual ∼segmenu ()

Provides a rote base-class destructor.

int current_seq () const

'Getter' function for member m_current_seq We're changing the name, so that "seq" indicates an integer by (an imperfect) convention.

bool is_modified () const

'Getter' function for member m modified

Protected Member Functions

void current seg (int seg)

'Setter' function for member m_current_seq

void set_edit_sequence (int seqnum)

'Setter' function for member m_edit_sequence Pass in -1 to disable the edit-sequence number.

void unset_edit_sequence (int seqnum)

 ${\it 'Setter' function for member m_edit_sequence \ Disable \ the \ edit-sequence \ number \ if \ it \ matches \ the \ parameter.}$

• bool is edit sequence (int segnum) const

'Getter' function for member m_edit_sequence Tests the parameter against m_edit_sequence.

· void is_modified (bool flag)

'Setter' function for member m_modified

sequence * get current sequence () const

'Getter' function for member m_mainperf.get_sequence(current_seq()) This call is used many, many times, and well worth wrapping.

sequence * get_sequence (int seqnum) const

Forwards the get-sequence call to the perform object.

· bool is current seg active () const

Forwards the is-sequence-active check to the perform object.

• bool is_current_seq_in_edit () const

Forwards the is-sequence-in-edit check to the perform object.

void new_current_sequence ()

Forwards the new-current-sequence call to the perform object.

void new_sequence (int seqnum)

Forwards the new-sequence call to the perform object.

• void delete_current_sequence ()

Forwards the delete-sequence call to the perform object.

void toggle_current_sequence ()

Forwards the sequence-playing-toggle call to the perform object.

void popup_menu ()

This function sets up the pattern menu entries.

void seq_edit ()

This menu callback launches the sequence-editor (pattern editor) window.

void seq_event_edit ()

This menu callback launches the new event editor window.

seqedit * create_seqedit (sequence &s)

A wrapper function so that we can not only create a new seqedit object, but have some management over it.

virtual void seq_set_and_edit (int seqnum)

Sets the current sequence and then acts as if the user had clicked on its slot.

virtual void seq_set_and_eventedit (int seqnum)

Sets the current sequence and then acts as if the user had right-clicked on its slot and selected "Event Edit".

Static Protected Member Functions

static void remove_seqedit (sequence &s)

A wrapper function to make sure the sequdit object is removed from the list when it goes away.

Private Types

typedef std::map< int, seqedit * > SeqeditMap

An easy type definition for a map of sequent pointers keyed by the sequence number.

typedef std::pair< int, seqedit * > SeqeditPair

A pair to make an entry to add to the seqedit map.

typedef std::map< int, segedit * >::iterator iterator

An iterator for the seqedit map.

typedef std::map< int, seqedit * >::const_iterator const_iterator

A const iterator for the seqedit map.

Private Member Functions

- virtual void redraw (int a_sequence)=0
- void seg new ()

This function sets the new sequence into the perform object, a bit prematurely, though.

void seq_copy ()

Copies the selected (current) sequence to the clipboard sequence.

void seq_cut ()

Deletes the selected (current) sequence and copies it to the clipboard sequence, if it is not in edit mode.

• void seq paste ()

Pastes the sequence clipboard into the current sequence, if the current sequence slot is not active.

void seq_clear_perf ()

If the current sequence is active, this function pushes a trigger undo in the main perform object, clears its sequence triggers for the current sequence, and sets the dirty flag of the sequence.

void set_bus_and_midi_channel (int a_bus, int a_ch)

Sets up the bus, MIDI channel, and dirtiness flag of the current sequence in the main perform object, as per the give parameters.

• void set_transposable (bool flag)

Sets the "is-transposable" flag of the current sequence.

void mute_all_tracks ()

Mutes all tracks in the main perform object.

void unmute_all_tracks ()

Unmutes all tracks in the main perform object.

void toggle_all_tracks ()

Toggles the mute-status of all tracks in the main perform object.

• void on_realize ()

Private Attributes

• Gtk::Menu * m_menu

The menu to pop up when the right-click action is used either on a mainwid pattern slot or on a perfedit pattern name.

· perform & m mainperf

Provides a reference to the central (non-UI) object involved in managing a song and performance.

· sequence m clipboard

Holds a copy of data concerning a sequence, which can then be pasted into another pattern slot.

• seqedit * m_seqedit

Points to the latest sequedit object, if created.

• eventedit * m_eventedit

Points to the latest eventedit object, if created.

• int m_current_seq

References the current sequence by sequence number.

· bool m modified

Indicates if a sequence has been created.

Static Private Attributes

static SeqeditMap sm_seqedit_list
 Holds a list of the currently open seqedit objects, stored as pointers keyed by the sequence number.

Friends

- · class mainwnd
- · class seqedit

13.89.1 Detailed Description

It is an abstract base class.

13.89.2 Member Typedef Documentation

```
13.89.2.1 SeqeditMap

typedef std::map<int, seqedit *> seq64::seqmenu::SeqeditMap [private]

13.89.2.2 SeqeditPair

typedef std::pair<int, seqedit *> seq64::seqmenu::SeqeditPair [private]

13.89.2.3 iterator

typedef std::map<int, seqedit *>::iterator seq64::seqmenu::iterator [private]

13.89.2.4 const_iterator

typedef std::map<int, seqedit *>::const_iterator seq64::seqmenu::const_iterator [private]
```

13.89.3 Constructor & Destructor Documentation

```
13.89.3.1 seqmenu() seq64::seqmenu::seqmenu ( perform & p )
```

Apart from filling in some of the members, this function initializes the clipboard, so that we don't get a crash on a paste with no previous copy.

p | The main performance object representing the whole MIDI song.

```
13.89.3.2 \simseqmenu() seq64::seqmenu::\simseqmenu ( ) [virtual]
```

A rote destructor.

This is necessary in an abstraction base class.

If we determine that we need to delete the m_sequedit pointer, we can do it here. But that is not likely, because we can have many new sequedit objects in play, because we can edit many at once.

13.89.4 Member Function Documentation

Now a pass-along to the perform object.

```
13.89.4.5 unset_edit_sequence()
void seq64::seqmenu::unset_edit_sequence (
             int seqnum ) [inline], [protected]
13.89.4.6 is_edit_sequence()
bool seq64::seqmenu::is_edit_sequence (
             int seqnum ) const [inline], [protected]
Returns true if that member is not -1, and the parameter matches it. Now a pass-along to the perform object.
13.89.4.7 is_modified() [2/2]
void seq64::seqmenu::is_modified (
             bool flag ) [inline], [protected]
13.89.4.8 get_current_sequence()
sequence* seq64::seqmenu::get_current_sequence ( ) const [inline], [protected]
13.89.4.9 get_sequence()
sequence* seq64::seqmenu::get_sequence (
             int seqnum ) const [inline], [protected]
13.89.4.10 is_current_seq_active()
bool seq64::seqmenu::is_current_seq_active ( ) const [inline], [protected]
13.89.4.11 is_current_seq_in_edit()
```

bool seq64::seqmenu::is_current_seq_in_edit () const [inline], [protected]

```
13.89.4.12 new_current_sequence()
void seq64::seqmenu::new_current_sequence ( ) [inline], [protected]
13.89.4.13 new_sequence()
void seq64::seqmenu::new_sequence (
             int seqnum ) [inline], [protected]
13.89.4.14 delete_current_sequence()
void seq64::seqmenu::delete_current_sequence ( ) [inline], [protected]
13.89.4.15 toggle_current_sequence()
void seq64::seqmenu::toggle_current_sequence ( ) [inline], [protected]
13.89.4.16 popup_menu()
```

It also sets up the pattern popup menu entries that are used in mainwid. Note that, for the selected sequence, the "Edit" and "Event Edit" menu entries are not included if a pattern editor or event editor is already running. The new event editor seems to create far-reaching problems that we do not yet understand, so it is now possible to disable it at build time. We have mitigated most of those problems by not allowing both a seq_edit() and a seq_event_edit() at the same time.

```
13.89.4.17 seq_edit()
void seq64::seqmenu::seq_edit ( ) [protected]
```

void seq64::seqmenu::popup_menu () [protected]

If it is already open for that sequence, this function just raises it.

Note that the m_seqedit member to which we save the new pointer is currently there just to avoid a compiler warning.

Also, if a new sequences is created, we set the m_modified flag to true, even though the sequence might later be deleted. Too much modification to keep track of!

An oddity is that calling show_all() here does not work unless the seqedit() constructor makes its show_all() call.

```
13.89.4.18 seq_event_edit()
void seq64::seqmenu::seq_event_edit ( ) [protected]
```

If it is already open for that sequence, this function just raises it.

Note that the m_eventedit member to which we save the new pointer is currently there just to avoid a compiler warning.

This menu entry is available only if the selected sequence is active. That is, if the sequence has already been created.

An oddity is that we need the show_all() call here in order to see the dialog. A situation different from that for sequdit! However, now it doesn't seem to be needed, and we have put it back into the eventedit constructor.

We don't bother checking here if the insert succeeded. If it doesn't, all bets are off.

Parameters

s Provides the sequence for which the sequence will be created. The perform object and the current_seq() value are implicit parameters. This object can obviously be modified by the sequence editor, so cannot be constant.

Returns

Returns the pointer to the new seqedit object.

How do we account for the current screenset? It might not matter if the mute/unmute keystrokes were designed to work only with the current screenset.

Change Note ca 2016-11-01 We would like to be able to right-click on a given pattern slot in mainwid, and figure out if it has a seqedit window open, so that we can update that window. So we need to add that seqedit window to a map of seqedits, keyed by the slot number. Then we can look up that slot and see if it has a seqedit window open. If the seqedit window closes, that window needs to remove itself from the map. This won't be needed for the event editor, which has no functionality from seqmenu.

o edit.
(

Reimplemented in seq64::mainwid.

13.89.4.22 seq_set_and_eventedit()

```
void seq64::seqmenu::seq_set_and_eventedit (
    int seqnum ) [protected], [virtual]
```

Parameters

seqnum	The number of the sequence to event-edit.
--------	---

Reimplemented in seq64::mainwid.

13.89.4.23 redraw()

Implemented in seq64::mainwid, and seq64::perfnames.

13.89.4.24 seq_new()

```
void seq64::seqmenu::seq_new ( ) [private]
```

For one thing, if current_seq() is either a -1 or is greater than the maximum allowed sequence number, perform ::is_active() will return false, and we have no idea whether the sequence is not active or the sequence number is just invalid. So we need to check the pointer we got before trying to use it.

13.89.4.25 seq_copy()

```
void seq64::seqmenu::seq_copy ( ) [private]
```

We use a more appropriate function than operator =() here: sequence::partial_assign().

Todo Can be offloaded to a perform member function that accepts a sequence clipboard non-const reference parameter.

```
13.89.4.26 seq_cut()
```

```
void seq64::seqmenu::seq_cut ( ) [private]
```

Todo A lot of seq_cut() can be offloaded to a (new) perform member function that takes a sequence clipboard non-const reference parameter.

```
13.89.4.27 seq_paste()
```

```
void seq64::seqmenu::seq_paste ( ) [private]
```

Then it sets the dirty flag for the destination sequence.

Todo All of seq_paste() can be offloaded to a (new) perform member function with a const clipboard reference parameter.

13.89.4.28 seq_clear_perf()

```
void seq64::seqmenu::seq_clear_perf ( ) [private]
```

13.89.4.29 set_bus_and_midi_channel()

Parameters

bus	The MIDI buss number to set (bus vs buss? You decide.)
ch	The MIDI channel number to set.

13.89.4.30 set_transposable()

```
void seq64::seqmenu::set_transposable (
          bool flag ) [private]
```

flag The value to use to set the flag.

```
13.89.4.31 mute_all_tracks()
void seq64::seqmenu::mute_all_tracks ( ) [inline], [private]
```

13.89.4.32 unmute_all_tracks()

```
void seq64::seqmenu::unmute_all_tracks ( ) [inline], [private]
```

13.89.4.33 toggle_all_tracks()

```
void seq64::seqmenu::toggle_all_tracks ( ) [inline], [private]
```

13.89.4.34 on_realize()

```
void seq64::seqmenu::on_realize ( ) [private]
```

13.89.5 Friends And Related Function Documentation

13.89.5.1 mainwnd

```
friend class mainwnd [friend]
```

13.89.5.2 seqedit

```
friend class seqedit [friend]
```

13.89.6 Field Documentation

```
13.89.6.1 sm_seqedit_list
seqmenu::SeqeditMap seq64::seqmenu::sm_seqedit_list [static], [private]
```

The single map of sequedit objects, for sequedit updates and management.

We can use this map to look up patterns that we want to change from the right-click seqmenu, and modify the sequedit affected if it is found in the list.

Currently selectable by the USE_SEQEDIT_MACRO until we can make it foolproof.

```
13.89.6.2 m_menu
Gtk::Menu* seq64::seqmenu::m_menu [private]

13.89.6.3 m_mainperf

perform& seq64::seqmenu::m_mainperf [private]
```

```
sequence seq64::seqmenu::m_clipboard [private]
```

```
13.89.6.5 m_seqedit
seqedit* seq64::seqmenu::m_seqedit [private]
```

Change Note Added by Chris on 2015-08-02 based on compiler warnings and a comment warning in the seq_edit() function. We'll save the result of that function here, and will let valgrind tell us later if Gtkmm takes care of it.

```
13.89.6.6 m_eventedit
eventedit* seq64::seqmenu::m_eventedit [private]
```

13.89.6.4 m_clipboard



Inheritance diagram for seq64::seqroll:



Public Member Functions

• seqroll (perform &perf, sequence &seq, int zoom, int snap, seqkeys &seqkeys_wid, int pos, Gtk::Adjustment &hadjust, Gtk::Adjustment &vadjust, int ppqn=SEQ64_USE_DEFAULT_PPQN)

Principal constructor.

virtual ∼seqroll ()

Provides a destructor to delete allocated objects.

void set_snap (int snap)

Sets the snap to the given value, and then resets the view.

void set zoom (int zoom)

Sets the zoom to the given value, and then resets the view.

void set_note_length (int note_length)

'Setter' function for member m_note_length

• int note_off_length () const

'Getter' function for member m_note_length, adjusted for the note_off_margin.

bool add_note (midipulse tick, int note, bool paint=true)

Convenience wrapper for sequence::add_note().

void add_chord (midipulse tick, int note)

Convenience wrapper for sequence::add_chord().

void set_key (int key)

Sets the music key to the given value, and then resets the view.

void set_scale (int scale)

Sets the music scale to the given value, and then resets the view.

void set_chord (int chord)

Sets the current chord to the given value.

void set_data_type (midibyte status, midibyte control)

Sets the status to the given parameter, and the CC value to the given optional control parameter, which defaults to 0.

void set_background_sequence (bool state, int seq)

This function sets the given sequence onto the piano roll of the pattern editor, so that the musician can have another pattern to play against.

void update_pixmap ()

This function draws the background pixmap on the main pixmap, and then draws the events on it.

• void update sizes ()

Update the sizes of items based on zoom, PPQN, BPM, BW (beat width) and more.

void update_background ()

Updates the background of this window.

void draw background on pixmap ()

Draws the main pixmap.

void draw_events_on_pixmap ()

Fills the main pixmap with events.

· void draw selection on window ()

Draws the current selecton on the main window.

void draw_progress_on_window ()

Draw a progress line on the window.

· void reset ()

This function basically resets the whole widget as if it were realized again.

void update_and_draw (int force=false)

Wraps up some common code.

· void redraw ()

Redraws unless m_ignore_redraw is true.

• void redraw_events ()

Redraws events unless m_ignore_redraw is true.

• void start_paste ()

Starts a paste operation.

- void complete_paste ()
- void complete_paste (int x, int y)

Completes a paste operation.

void follow_progress ()

Private Member Functions

virtual void force_draw ()

Set the pixmap into the window and then draws the selection on it.

void horizontal adjust (double step)

This function provides optimization for the on_scroll_event() function.

void vertical_adjust (double step)

This function provides optimization for the on_scroll_event() function.

void snap_y (int &y)

Snaps the y value to the piano-key "height".

void snap_x (int &x)

Performs a 'snap' operation on the x coordinate.

- void convert_xy (int x, int y, midipulse &ticks, int ¬e)
- void convert tn (midipulse ticks, int note, int &x, int &y)

This function takes the given note and tick, and returns the screen coordinates via the pointer parameters.

void xy_to_rect (int x1, int y1, int x2, int y2, int &x, int &y, int &w, int &h)

Converts rectangle corner coordinates to a starting coordinate, plus a width and height.

void convert_tn_box_to_rect (midipulse tick_s, midipulse tick_f, int note_h, int note_l, int &x, int &y, int &w, int &h)

Converts a tick/note box to an x/y rectangle.

• void convert_sel_box_to_rect (midipulse tick_s, midipulse tick_f, int note_h, int note_l)

A convenience function wrapping a common call to convert_tn_box_to_rect().

void get selected box (midipulse &tick s, int ¬e h, midipulse &tick f, int ¬e l)

A convenience function wrapping a common call to m_seq.get_selected_box() and convert_tn_box_to_rect().

void draw_events_on (Glib::RefPtr< Gdk::Drawable > draw)

Draws events on the given drawable area.

• int idle redraw ()

Draw the events on the main window and on the pixmap.

- int idle_progress ()
- void change_horz ()

Change the horizontal scrolling offset and redraw.

· void change_vert ()

Change the vertical scrolling offset and redraw.

void move_selection_box (int dx, int dy)

Function to allow motion of the selection box via the arrow keys.

void move_selected_notes (int dx, int dy)

Proposed new function to encapsulate the movement of selections even more fully.

void grow_selected_notes (int dx)

Proposed new function to encapsulate the movement of selections even more fully.

void set_adding (bool adding)

Changes the mouse cursor pixmap according to whether a note is being added or not.

void update_mouse_pointer (bool adding=false)

Updates the mouse pointer, implementing a context-sensitive mouse.

- bool button_press_initial (GdkEventButton *ev, int &norm_x, int &snapped_x, int &snapped_y)
- void align selection (midipulse &tick s, int ¬e h, midipulse &tick f, int ¬e l, int snapped x)

Get the box that selected elements are in.

bool button_press (GdkEventButton *ev)

Implements the on-button-press event handling for the Seq24 style of mouse interaction.

bool button_release (GdkEventButton *ev)

Implements the on-button-release event handling for the Seq24 style of mouse interaction.

bool motion_notify (GdkEventMotion *ev)

Seq24-style on-motion mouse interaction.

void clear_selected ()

'Setter' function for member m_old

· void clear old ()

'Setter' function for member m_old

· void clear flags ()

Clears all the mouse-action flags.

int scroll offset x (int x) const

Useful x calculation.

int scroll_offset_y (int y) const

Useful y calculation.

void set_current_offset_x_y (int x, int y)

Useful x calculation.

· bool adding () const

'Getter' function for member m_adding

• bool selecting () const

'Getter' function for member m_selecting

· bool growing () const

'Getter' function for member m_growing

· bool normal action () const

Indicates if we're drag-pasting, selecting, moving, growing, or pasting.

· bool select_action () const

Indicates if we're selecting, moving, growing, or pasting.

bool drop_action () const

Indicates if we're moving or pasting.

• bool moving () const

'Getter' function for member m_moving

void on_realize ()

Implements the on-realize event handling.

bool on expose event (GdkEventExpose *ev)

Implements the on-expose event handling.

• bool on button press event (GdkEventButton *ev)

Implements the on-button-press event handling.

bool on_button_release_event (GdkEventButton *ev)

Implements the on-button-release event handling.

• bool on_motion_notify_event (GdkEventMotion *ev)

Implements the on-motion-notify event handling.

bool on_focus_in_event (GdkEventFocus *)

Implements the on-focus event handling.

bool on_focus_out_event (GdkEventFocus *)
 Implements the on-unfocus event handling.

bool on_key_press_event (GdkEventKey *ev)

Implements the on-key-press event handling.

bool on_scroll_event (GdkEventScroll *a_ev)

Implements the on-scroll event handling.

• void on_size_allocate (Gtk::Allocation &)

Implements the on-size-allocate event handling.

bool on_leave_notify_event (GdkEventCrossing *p0)

Implements the on-leave-notify event handling.

bool on_enter_notify_event (GdkEventCrossing *p0)

Implements the on-enter-notify event handling.

Private Attributes

Gtk::Adjustment & m_horizontal_adjust

For accessing on_key_press_event().

Gtk::Adjustment & m_vertical_adjust

We need direct access to the vertical scrollbar if we want to be able to make it follow PageUp and PageDown.

· rect m old

The previous selection rectangle, used for undrawing it.

· rect m selected

Used in moving and pasting notes.

• sequence & m_seq

Provides a reference to the sequence represented by piano roll.

seqkeys & m_seqkeys_wid

Holds a reference to the seqkeys pane that is associated with the seqroll piano roll.

• FruitySeqRollInput m_fruity_interaction

Provides a fruity input object, whether it is needed or not.

int m_pos

A position value.

• int m zoom

Zoom setting, means that one pixel == m_zoom ticks.

int m_snap

The grid-snap setting for the piano roll grid.

• int m ppqn

The value of PPQN for the current MIDI song.

• int m_note_length

Holds the note length in force for this sequence.

• int m scale

Indicates the musical scale in force for this sequence.

• int m_chord

Indicates the current chord in force for this sequence for inserting notes.

· int m key

Indicates the musical key in force for this sequence.

bool m_adding

Set when in note-adding mode.

· bool m selecting

Set when highlighting a bunch of events.

bool m_moving

Set when moving a bunch of events.

bool m_moving_init

Indicates the beginning of moving some events.

bool m_growing

Indicates that the notes are to be extended or reduced in length.

bool m_painting

Indicates the painting of events.

bool m paste

Indicates that we are in the process of painting notes.

bool m_is_drag_pasting

Indicates the drag-pasting of events.

· bool m is drag pasting start

Indicates the drag-pasting of events.

• bool m_justselected_one

Indicates the selection of one event.

• int m move delta x

Tells where the dragging started, the x value.

· int m move delta y

Tells where the dragging started, the y value.

• int m_move_snap_offset_x

This item is used in the fruityseqroll module.

• int m_progress_x

Provides the location of the progress bar.

· int m scroll offset ticks

The horizontal value of the scroll window in units of ticks/pulses/divisions.

· int m_scroll_offset_key

The vertical offset of the scroll window in units of MIDI notes/keys.

int m_scroll_offset_x

The horizontal value of the scroll window in units of pixels.

• int m_scroll_offset_y

The vertical value of the scroll window in units of pixels.

· bool m transport follow

TBD.

• bool m_trans_button_press

TBD.

int m_background_sequence

Holds the value of the musical background sequence that is shown in cyan (formerly grey) on the background of the piano roll.

· bool m_drawing_background_seq

Set to true if the drawing of the background sequence is to be done.

• midibyte m_status

Set to true to avoid the call to update_and_draw().

• midibyte m_cc

The current MIDI control value selected in the sequdit.

Friends

class FruitySeqRollInput

This friend implements fruity interaction-specific behavior.

Additional Inherited Members

13.90.1 Constructor & Destructor Documentation

13.90.1.1 seqroll()

p	The performance object that helps control this piano roll. Note that we can get the perform object from the sequence, and save a parameter. Low priority change.		
seq The sequence object represented by this piano roll.			
zoom The initial zoom of this piano roll.			
snap	The initial grid snap of this piano roll.		
seqkeys_wid	A reference to the piano keys window that is shown to the left of this piano roll.		
pos	A position parameter. See the description of seqroll::m_pos . This is actually the sequence number, and is currently unused. However, we're sure we can find a use for it sometime.		
hadjust	Represents the horizontal scrollbar of this window. It is actually created by the "parent" sequential object.		
vadjust	Represents the vertical scrollbar of this window. It is actually created by the "parent" seqedit object.		
ppqn	The initial value of the PPQN for this sequence. Useful in scale calculations.		

13.90.1.2 \sim seqroll()

```
seq64::seqroll::~seqroll ( ) [virtual]
```

The only thing to delete here is the clipboard. Except it is never used, so is commented out.

13.90.2 Member Function Documentation

13.90.2.1 set_snap()

Parameters

snap	Provides the sname value to set.
------	----------------------------------

13.90.2.2 set_zoom()

zoom The desired zoom value, assumed to be validated already. See the seqedit::set_zoom() function.

13.90.2.3 set_note_length()

```
void seq64::seqroll::set_note_length (
    int note_length ) [inline]
```

13.90.2.4 note_off_length()

```
int seq64::seqroll::note_off_length ( ) const [inline]
```

13.90.2.5 add_note()

The length parameters is obtained from the note_off_length() function. This sets the note length at a little less than the snap value.

Parameters

tick	The time destination of the new note, in pulses.	
note	The pitch destination of the new note.	
paint If true, repaint to be left with just the inserted event. The default is true. The value of false is useful inserting a number of events and saving the repainting until last. It is a bit tricky, as the default paint value for sequence::add_note() is false.		

13.90.2.6 add_chord()

Implicit parameters are the m_chord and note_off_length() members. The latter deducts just a little from the snap value.

tick	The tick at which to add the chord.	
note	The base (bottom) note of the chord.	

13.90.2.7 set_key()

Parameters

key	The desired key value.	
ΛC y	THE desired key value	7.

13.90.2.8 set_scale()

Parameters

scale	The desired scale value.	
Scare	THE DESILED SCALE VALUE.	

13.90.2.9 set_chord()

Parameters

13.90.2.10 set_data_type()

Unlike the same function in seqevent, this version does not redraw. Used by seqedit.

13.90.2.11 set_background_sequence()

```
void seq64::seqroll::set_background_sequence ( bool \ state, int \ seq )
```

The state parameter sets the boolean m_drawing_background_seq.

Parameters

state	If true, the background sequence will be drawn.		
seq	Provides the sequence number, which is checked against the SEQ64_IS_LEGAL_SEQUENCE() macro before being used. This macro allows the value SEQ64_SEQUENCE_LIMIT, which disables the		
	background sequence.		

13.90.2.12 update_pixmap()

```
void seq64::seqroll::update_pixmap ( )
```

13.90.2.13 update_sizes()

```
void seq64::seqroll::update_sizes ( )
```

It brings the scrollbar back to the beginning, resets the upper limit to the number of ticks in the sequence, sets the page-size based on the window size and the zoom factor.

The horizontal step increment is 1 semiquaver (1/16) note per zoom level. The horizontal page increment is currently always one bar. We may want to make that larger for scrolling after the progress bar.

Tha maximum value set for the scrollbar brings it to the last "page" of the piano roll.

The vertical size are also adjusted. More on the story later.

13.90.2.14 update_background()

```
void seq64::seqroll::update_background ( )
```

The first thing done is to clear the background, painting it white.

13.90.2.15 draw_background_on_pixmap()

```
void seq64::seqroll::draw_background_on_pixmap ( )
```

```
13.90.2.16 draw_events_on_pixmap()
void seq64::seqroll::draw_events_on_pixmap ( )

Just calls draw_events_on().

13.90.2.17 draw_selection_on_window()

void seq64::seqroll::draw_selection_on_window ( )
```

Note the parameters of draw_drawable(), which we need to be sure of to draw thicker boxes.

```
    x and y position of rectangle to draw
    x and y position in drawable where rectangle should be drawn
    width and height of rectangle to draw
```

A final parameter of false draws an unfilled rectangle. Orange makes it a little more clear that we're pasting, I think. We also want to try to thicken the lines somehow.

```
13.90.2.18 draw_progress_on_window()
void seq64::seqroll::draw_progress_on_window ( )
```

This is done by first blanking out the line with the background, which contains white space and grey lines, using the the draw_drawable function. Remember that we wrap the draw_drawable() function so it's parameters are xsrc, ysrc, xdest, ydest, width, and height.

Note that the progress-bar position is based on the sequence::get_last_tick() value, the current zoom, and the current scroll-offset x value.

Finally, we had an issue with the selection box flickering, which seems to be solved satisfactorily by not drawing it if a select action is in force. Hopefully no one needs to select notes on the fly and see the progress bar moving at the same time! Another tactic is to draw progress only when the performance is running. This has the benefit/drawback that the progress bar is left where it stops. Consider an enumeration of options: normal, when-not-selecting, and when-running.

```
13.90.2.19 reset()
void seq64::seqroll::reset ( )
```

It's almost identical to the change_horz() function, just calling update_sizes() before update_and_draw().

13.90.2.21 redraw()

force If true, force an immediate draw, otherwise just queue up a draw. This value defaults to false.

```
void seq64::seqroll::redraw ( )
Somewhat similar to seqevent::redraw(). Actually, we don't seem to need to ignore redraw when making settings in
the seqedit constructor, so this member no longer exists.
13.90.2.22 redraw_events()
void seq64::seqroll::redraw_events ( )
Actually, that member is not needed and no longer exists.
13.90.2.23 start_paste()
void seq64::seqroll::start_paste ( )
13.90.2.24 complete_paste() [1/2]
void seq64::seqroll::complete_paste ( ) [inline]
13.90.2.25 complete_paste() [2/2]
void seq64::seqroll::complete_paste (
              int x,
              int y)
13.90.2.26 follow_progress()
void seq64::seqroll::follow_progress ( )
13.90.2.27 force_draw()
void seq64::seqroll::force_draw ( ) [private], [virtual]
Reimplemented from seq64::gui_drawingarea_gtk2.
13.90.2.28 horizontal_adjust()
void seq64::seqroll::horizontal_adjust (
```

double step) [inline], [private]

A duplicate of the one in sequent.

step

Provides the step value to use for adjusting the horizontal scrollbar. See gui_drawingarea_gtk2::scroll_hadjust() for more information.

13.90.2.29 vertical_adjust()

A duplicate of the one in seqedit.

Parameters

step

Provides the step value to use for adjusting the vertical scrollbar. See gui_drawingarea_gtk2::scroll_vadjust() for more information.

13.90.2.30 snap_y()

Parameters

out	У	The y-value to be snapped.
-----	---	----------------------------

13.90.2.31 snap_x()

This function is similar to snap_y(), but it calculates a modulo value from the snap and zoom settings.

```
m_snap = number pulses to snap tom_zoom = number of pulses per pixel
```

Therefore, m_snap / m_zoom = number pixels to snap to.

Parameters

out x Provides the x-value to be snapped and returned. A return value would be better.

13.90.2.32 convert_xy()

13.90.2.33 convert_tn()

This function is the "inverse" of convert xy().

Parameters

	tick	Provides the horizontal value in MIDI pulses.
	note	Provides the vertical value, a note value.
out	Х	Provides the destination x value of the coordinate.
out	У	Provides the destination y value of the coordinate.

13.90.2.34 xy_to_rect()

This function checks the mins / maxes, and then fills in the x, y, width, and height values.

We should refactor this function to use the utility class seqroll::rect as the destination for the conversion.

Parameters

	x1	The x value of the first corner.
	y1	The y value of the first corner.

	x2	The x value of the second corner.
	y2	The y value of the second corner.
out	Х	The destination for the x value in pixels.
out	У	The destination for the y value in pixels.
out	W	The destination for the rectangle width in pixels.
out	h	The destination for the rectangle height value in pixels.

13.90.2.35 convert_tn_box_to_rect()

```
void seq64::seqroll::convert_tn_box_to_rect (
    midipulse tick_s,
    midipulse tick_f,
    int note_h,
    int note_l,
    int & x,
    int & y,
    int & w,
    int & h ) [private]
```

We should refactor this function to use the utility class seqroll::rect as the destination for the conversion.

Parameters

	tick_s	The starting tick of the rectangle.
	tick_f	The finishing tick of the rectangle.
	note⇔	The high note of the rectangle.
	_h	
	note⊷	The low note of the rectangle.
	_/	
out	X	The destination for the x value in pixels.
out	У	The destination for the y value in pixels.
out	W	The destination for the rectangle width in pixels.
out	h	The destination for the rectangle height value in pixels.

13.90.2.36 convert_sel_box_to_rect()

tick_s	The starting tick of the rectangle.
tick_f	The finishing tick of the rectangle.
note← _h	The high note of the rectangle.
note↔ _I	The low note of the rectangle.

13.90.2.37 get_selected_box()

Parameters

out	tick_s	The starting tick of the rectangle.
out	tick_f	The finishing tick of the rectangle.
out	note⊷	The high note of the rectangle.
	_h	
out	note⊷	The low note of the rectangle.
	_/	

13.90.2.38 draw_events_on()

```
void seq64::seqroll::draw_events_on ( \label{lib:cond} {\tt Glib::RefPtr<\ Gdk::Drawable\ >\ draw\ )} \quad [{\tt private}]
```

"Method 0" draws the background sequence, if active. "Method 1" draws the sequence itself.

Parameters

```
draw The "drawable" area to draw on.
```

13.90.2.39 idle_redraw()

```
int seq64::seqroll::idle_redraw ( ) [private]
```

```
13.90.2.40 idle_progress()
int seq64::seqroll::idle_progress ( ) [private]

13.90.2.41 change_horz()

void seq64::seqroll::change_horz ( ) [private]

Roughly similar to seqevent::change_horz().

13.90.2.42 change_vert()

void seq64::seqroll::change_vert ( ) [private]
```

int dx, int dy) [private]

void seq64::seqroll::move_selection_box (

13.90.2.43 move_selection_box()

We now let the Enter key to finish pasting and deselect the moved notes. With the mouse, selecting all notes, copying them, and moving the selection box, pasting can be completed with either a left-click or the Enter key.

We have a weird problem on our main system where the selection box is very flickery. But it works fine on another system. A Gtk-2 issue? Now it seems to work fine, after an update. No, it seems to work well in sequences that have non-note events amongst the note events.

Parameters

dx	The amount to move the selection box. Values are -1, 0, or 11 is left one snap, 0 is no movement horizontally, and 1 is right one snap.
dy	The amount to move the selection box. Values are -1, 0, or 11 is up one snap, 0 is no movement vertically, and 1 is down one snap.

13.90.2.44 move_selected_notes()

Works with the four arrow keys.

Note that the movement vertically is different for the selection box versus the notes. While the movement values are -1, 0, or 1, the differences are as follows:

```
    Selection box vertical movement:

            -1 is up one note snap.
            0 is no vertical movement.
            +1 is down one note snap.

    Note vertical movement:

            -1 is down one note.
            0 is no note vertical movement.
            +1 is up one note.
```

dx	The amount to move the selection box or the selection horizontally. Values are -1 (left one time snap), 0 (no movement), and +1 (right one snap). Obviously values other than +-1 can be used for larger movement, but the GUI doesn't yet support that we could implement movement by "pages" some day.
dy	The amount to move the selection box or the selection vertically. See the notes above.

13.90.2.45 grow_selected_notes()

Parameters

The amount to grow the selection horizontally. Values are -1 (left one time snap), 0 (no stretching), and +1 (right one snap). Obviously values other than +-1 can be used for larger stretching, but the GUI doesn't yet support that.

13.90.2.46 set_adding()

What calls this? It is actually a right click. Not present in the "fruity" implementation. Now moved to the normal seqroll class.

Parameters

```
adding True if adding a note.
```

13.90.2.47 update_mouse_pointer()

Moved here from the "fruity" seqroll class.

13.90.2.48 button_press_initial()

13.90.2.49 align_selection()

Save offset that we get from the snap above. Align selection for drawing. Could be used in XRollInput::on_button
_press_event().

13.90.2.50 button_press()

This function now uses the needs_update flag to determine if the perform object should modify().

Parameters

```
ev Provides the button-press event to process.
```

Returns

Returns the value of needs_update. It used to return only true.

13.90.2.51 button_release()

This function now uses the needs_update flag to determine if the perform object should modify().

Parameters

```
ev | Provides the button-release event to process.
```

Returns

Returns the value of needs_update. It used to return only true.

If in moving mode, adjust for snap and convert deltas into screen coordinates. Since delta_note was from delta_y, it will be flipped (delta_y[0] = note[127], etc.), so we have to adjust.

A left/middle click converts deltas into screen coordinates, then pushs the undo state. Shift causes a "stretch selected" which currently acts like a "move selected" operation. BUG? Otherwise, Ctrl indirectly allows a "grow selected" operation.

Minor new feature. If the Super (Mod4, Windows) key is pressed when release, keep the adding state in force. One can then use the unadorned left-click key to add notes. Right click to reset the adding mode. This feature is enabled only if allowed by the settings (but is true by default). See the same code in perfrollingut.cpp.

13.90.2.52 motion_notify()

We could allow move painting for chords, but that would take some tricky code to move all of the notes of the chord. And allowing painting here currently affects only one note after the chord itself is created.

Parameters

ev Provides the button-release event to process.

Returns

Returns true if the event was processed.

13.90.2.53 clear_selected()

```
void seq64::seqroll::clear_selected ( ) [inline], [private]
```

13.90.2.54 clear_old()

```
void seq64::seqroll::clear_old ( ) [inline], [private]
```

13.90.2.55 clear_flags()

```
void seq64::seqroll::clear_flags ( ) [inline], [private]
```

13.90.2.56 scroll_offset_x()

Offsets the \boldsymbol{x} value by the \boldsymbol{x} origin of the current page.

```
x The x value to offset.
```

13.90.2.57 scroll_offset_y()

Offsets the y value by the y origin of the current page.

Parameters

```
y The y value to offset.
```

13.90.2.58 set_current_offset_x_y()

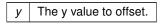
Offsets the current x value by the x origin of the current page.

Parameters

```
x The x value to offset.
```

void set_current_offset_x (int x) { m_current_x = $x + m_scroll_offset_x$; } Useful y calculation. Offsets the current y value by the y origin of the current page.

Parameters



void set_current_offset_y (int y) { m_current_y = $y + m_scroll_offset_y$; } Useful x and y calculation. Offsets the current x and y values by the x and y origin of the current page.

Parameters

X	The y value to offset.
У	The y value to offset.

13.90.2.59 adding()

```
bool seq64::seqroll::adding ( ) const [inline], [private]
```

13.90.2.60 selecting()

```
bool seq64::seqroll::selecting ( ) const [inline], [private]
```

13.90.2.61 growing()

```
bool seq64::seqroll::growing ( ) const [inline], [private]
```

13.90.2.62 normal_action()

```
bool seq64::seqroll::normal_action ( ) const [inline], [private]
```

Returns

Returns true if one of those five flags are set.

13.90.2.63 select_action()

```
bool seq64::seqroll::select_action ( ) const [inline], [private]
```

Returns

Returns true if one of those four flags are set.

13.90.2.64 drop_action()

```
bool seq64::seqroll::drop_action ( ) const [inline], [private]
```

Returns

Returns true if one of those two flags are set.

13.90.2.65 moving()

```
bool seq64::seqroll::moving ( ) const [inline], [private]
```

13.90.2.66 on_realize()

```
void seq64::seqroll::on_realize ( ) [private]
```

13.90.2.67 on_expose_event()

Parameters

ev The expose event to process.

Returns

Always returns true.

13.90.2.68 on_button_press_event()

Parameters

ev The expose event to process.

Returns

Returns the result of the Seq24 interaction or the Fruity interaction, that is, the return value of Seq24Seq \leftarrow RollInput::on_button_press_event() or FruitySeqRollInput::on_button_press_event().

13.90.2.69 on_button_release_event()

```
bool seq64::seqroll::on_button_release_event (
    GdkEventButton * ev ) [private]
```

This function checks the "rc" interaction-method option, and calls the forwarding function for the seq24 or the fruity interaction method. Might be a good case to prefer inheritance and not try to support changing the interaction method without a restart of Sequencer64.

Parameters

ev The button release event to process.

Returns

Returns the return value of Seq24SeqRollInput::on_button_release_event() or FruitySeqRollInput::on_ \leftarrow button_release_event().

13.90.2.70 on_motion_notify_event()

Parameters

ev The motion-notification event to process.

Returns

Returns the return value of Seq24SeqRollInput::on_motion_notify_event() or FruitySeqRollInput::on_motion \leftarrow _notify_event().

13.90.2.71 on_focus_in_event()

Sets the GDK HAS_FOCUS flag. Parameter "ev" is the event-focus event, not used.

Returns

Always returns false.

13.90.2.72 on_focus_out_event()

Resets the GDK HAS_FOCUS flag. Parameter "ev" is the event-focus event, not used.

Returns

Always returns false.

13.90.2.73 on_key_press_event()

The start/end key may be the same key (i.e. SPACEBAR). Allow toggling when the same key is mapped to both triggers (i.e. SPACEBAR).

Concerning the usage of the arrow keys in this function: This code is reached, but has no visible effect. Why? I think they were meant to move the point for playback. We may have a bug with our new handling of triggers (unlikely), or maybe these depend upon the proper playback mode. In any case, the old functionality is preserved. However, if there are notes selected, then these keys support selection movement.

Since the Up and Down arrow keys are used for movement, we'd have to check selection status before trying to use them to move up and down in the piano roll, in smaller steps than the new Page-Up and Page-Down key support.

Parameters

```
ev The key-press event to process.
```

Returns

Returns true if the key-press was handled.

I think we should be able to move and remove notes while playing, which is already supported using the mouse.

```
if (! perf().is_playing)
```

13.90.2.74 on_scroll_event()

This scroll event only handles basic scrolling without any modifier keys such as the Ctrl or Shift masks. The sequedit class handles that fun stuff.

Note that this function seems to duplicate the functionality of seqkeys::on_scroll_event(). Do we really need both?
Which one do we need?

Parameters

```
ev The scroll event to process.
```

Returns

Returns true if the scroll event was handled.

13.90.2.75 on_size_allocate()

Calls the base-class version of this function and sets m_window_x and m_window_y to the width and height of the allocation parameter. Then calls update_sizes().

Parameters

a The GDK allocation event object.

13.90.2.76 on_leave_notify_event()

Calls m_seqkeys_wid.set_hint_state(false). Parameter "ev" is the event-crossing event, not used.

Returns

Always returns false.

13.90.2.77 on_enter_notify_event()

Calls m_seqkeys_wid.set_hint_state(true). Parameter "ev" is the event-crossing event, not used.

Returns

Always returns false.

13.90.3 Friends And Related Function Documentation

13.90.3.1 FruitySeqRollInput

```
friend class FruitySeqRollInput [friend]
```

We've absorbed the Seg24SegRollInput class functionality back into segroll, to save code.

13.90.4 Field Documentation

13.90.4.1 m_horizontal_adjust

```
Gtk::Adjustment& seq64::seqroll::m_horizontal_adjust [private]
```

It would be good to be able to avoid this access!

Change Note layk 2016-10-17 Issue #46. No need for this declaration now, due to the fix in seqedit.

friend class sequedit; We need direct access to the horizontal scrollbar if we want to be able to make it follow the progress bar.

13.90.4.2 m_vertical_adjust

```
Gtk::Adjustment& seq64::seqroll::m_vertical_adjust [private]
```

13.90.4.3 m_old

```
rect seq64::seqroll::m_old [private]
```

13.90.4.4 m selected

```
rect seq64::seqroll::m_selected [private]
```

```
13.90.4.5 m_seq
sequence& seq64::seqroll::m_seq [private]
13.90.4.6 m_seqkeys_wid
seqkeys& seq64::seqroll::m_seqkeys_wid [private]
13.90.4.7 m_fruity_interaction
FruitySeqRollInput seq64::seqroll::m_fruity_interaction [private]
13.90.4.8 m_pos
int seq64::seqroll::m_pos [private]
Need to clarify what exactly this member is used for.
13.90.4.9 m_zoom
int seq64::seqroll::m_zoom [private]
13.90.4.10 m_snap
int seq64::seqroll::m_snap [private]
Same meaning as for the event-bar grid. This value is the denominator of the note size used for the snap.
13.90.4.11 m_ppqn
int seq64::seqroll::m_ppqn [private]
Supports values other than the default of 192.
13.90.4.12 m_note_length
int seq64::seqroll::m_note_length [private]
Used in the seq24seqroll module only.
```

```
13.90.4.13 m_scale
int seq64::seqroll::m_scale [private]
13.90.4.14 m_chord
int seq64::seqroll::m_chord [private]
13.90.4.15 m_key
int seq64::seqroll::m_key [private]
13.90.4.16 m_adding
bool seq64::seqroll::m_adding [private]
This flag was moved from both the fruity and the seq24 seqroll classes.
13.90.4.17 m_selecting
bool seq64::seqroll::m_selecting [private]
13.90.4.18 m_moving
bool seq64::seqroll::m_moving [private]
13.90.4.19 m_moving_init
bool seq64::seqroll::m_moving_init [private]
Used in the fruity and seq24 mouse-handling modules.
13.90.4.20 m_growing
```

bool seq64::seqroll::m_growing [private]

```
13.90.4.21 m_painting
bool seq64::seqroll::m_painting [private]
Used in the fruity and seq24 mouse-handling modules.
13.90.4.22 m_paste
bool seq64::seqroll::m_paste [private]
13.90.4.23 m_is_drag_pasting
bool seq64::seqroll::m_is_drag_pasting [private]
Used in the fruity mouse-handling module.
13.90.4.24 m_is_drag_pasting_start
bool seq64::seqroll::m_is_drag_pasting_start [private]
Used in the fruity mouse-handling module.
13.90.4.25 m_justselected_one
bool seq64::seqroll::m_justselected_one [private]
Used in the fruity mouse-handling module.
13.90.4.26 m move delta x
int seq64::seqroll::m_move_delta_x [private]
13.90.4.27 m_move_delta_y
int seq64::seqroll::m_move_delta_y [private]
13.90.4.28 m_move_snap_offset_x
```

int seq64::seqroll::m_move_snap_offset_x [private]

```
13.90.4.29 m_progress_x
int seq64::seqroll::m_progress_x [private]
13.90.4.30 m_scroll_offset_ticks
int seq64::seqroll::m_scroll_offset_ticks [private]
13.90.4.31 m_scroll_offset_key
int seq64::seqroll::m_scroll_offset_key [private]
13.90.4.32 m_scroll_offset_x
int seq64::seqroll::m_scroll_offset_x [private]
13.90.4.33 m_scroll_offset_y
int seq64::seqroll::m_scroll_offset_y [private]
13.90.4.34 m_transport_follow
bool seq64::seqroll::m_transport_follow [private]
13.90.4.35 m_trans_button_press
bool seq64::seqroll::m_trans_button_press [private]
13.90.4.36 m_background_sequence
```

int seq64::seqroll::m_background_sequence [private]

13.90.4.37 m_drawing_background_seq

bool seq64::seqroll::m_drawing_background_seq [private]

13.90.4.38 m_status

```
midibyte seq64::seqroll::m_status [private]
```

Used in set_background_sequence(), change_horz(), change_vert(), reset().... Never set to true, except in seq24, let's just comment it out for now. It hasn't been used in sequencer64 for awhile now.

bool m_ignore_redraw; The current status/event selected in the seqedit. Not used in seqroll at present.

13.90.4.39 m cc

midibyte seq64::seqroll::m_cc [private]

Not used in seqroll at present.

13.91 seq64::seqtime Class Reference

This class implements the piano time, whatever that is.

Inheritance diagram for seq64::seqtime:



Public Member Functions

seqtime (sequence &seq, perform &p, int zoom, Gtk::Adjustment &hadjust, int ppqn=SEQ64_USE_DEFA
 ULT_PPQN)

Principal constructor.

virtual ∼seqtime ()

Let's provide a do-nothing virtual destructor.

· void reset ()

Sets the scroll offset tick and x values, updates the sizes and the pixmap, and resets the window.

• void redraw ()

Very similar to the reset() function, except it doesn't update the sizes.

void set_zoom (int zoom)

Sets the zoom to the given value and resets the window.

Private Member Functions

```
void draw_pixmap_on_window ()
```

Draws the pixmap on the window.

- void draw_progress_on_window ()
- void update_pixmap ()

Updates the pixmap.

void change horz ()

Changes the scrolling horizontal offset, updates the pixmap, and forces a redraw.

void update_sizes ()

Updates the pixmap to a new size and queues up a draw operation.

• bool idle_progress ()

Simply returns true.

• void on_realize ()

Called when the window is drawn.

bool on_expose_event (GdkEventExpose *a_ev)

Implements the on-expose event handler.

• void on_size_allocate (Gtk::Allocation &)

Implements the on-size-allocate event handler.

• bool on_button_press_event (GdkEventButton *)

Implements the on-button-press event handler.

bool on_button_release_event (GdkEventButton *)

Implements the on-button-release event handler.

Private Attributes

- sequence & m seq
- int m_scroll_offset_ticks
- int m_scroll_offset_x
- int m_zoom

one pixel == m_zoom ticks

• int m_ppqn

Additional Inherited Members

13.91.1 Constructor & Destructor Documentation

```
13.91.1.1 seqtime()
```

In the constructor you can only allocate colors; get_window() returns 0 because the window is not yet realized>

```
13.91.1.2 \sim seqtime()
```

```
virtual seq64::seqtime::~seqtime ( ) [inline], [virtual]
```

13.91.2 Member Function Documentation

```
13.91.2.1 reset()
```

```
void seq64::seqtime::reset ( )
```

Basically identical to seqevent::reset().

13.91.2.2 redraw()

```
void seq64::seqtime::redraw ( )
```

13.91.2.3 set_zoom()

Parameters

zoom The desired zoom value, assumed to be validated already. See the seqedit::set_zoom() function.

13.91.2.4 draw_pixmap_on_window()

```
void seq64::seqtime::draw_pixmap_on_window ( ) [private]
```

13.91.2.5 draw_progress_on_window()

```
void seq64::seqtime::draw_progress_on_window ( ) [private]
```

13.91.2.6 update_pixmap()

```
void seq64::seqtime::update_pixmap ( ) [private]
```

When the zoom is at 32 ticks per pixel, there is a thick bar for every measure, and a measure number and major time division every 4 measures.at the default PPQN of 192.

A major line is a line that has a measure number in the timeline. The number of measures in a major line is 1 for zooms from 1:1 to 1:8; 2 for zoom 1:16; 4 for zoom 1:32; 8 for zoom 1:64 (new); and 16 for zoom 1:128. Zooms 1:64 and above look good only for high PPQN values.

We calculate the measure length in 32nd notes. This value is, of course, 32, when the time signature is 4/4. Then calculate measures/line. "measures_per_major" is more like "measures per major line". With a higher zoom than 32, this calculation yields a floating-point exception if m_zoom

32, so we rearrange the calculation and hope that it still works out the

same for smaller values.

Stazed:

At 32, a bar every measure.

zoom	32	16	8	4	1
ml					
1	128				
2	64				
4	32	16	8		
8	16m	8	4	2	1
16	8m	4	2	1	1
32	4m	2	1	1	1
64	2m	1	1	1	1
128	1m	1	1	1	1

Todo Sizing needs to be controlled by font parameters. Instead of 19 or 20, estimate the width of 3 letters. Instead of 9 pixels down, use the height of the segtime and the height of a character.

13.91.2.7 change_horz()

```
void seq64::seqtime::change_horz ( ) [private]
```

Simply returns false.

```
13.91.2.8 update_sizes()
void seq64::seqtime::update_sizes ( ) [private]
13.91.2.9 idle_progress()
bool seq64::seqtime::idle_progress ( ) [inline], [private]
13.91.2.10 on_realize()
void seq64::seqtime::on_realize ( ) [private]
Call the base-class version of this function first. Then addition resources are allocated.
13.91.2.11 on_expose_event()
bool seq64::seqtime::on_expose_event (
             GdkEventExpose * a_ev ) [private]
13.91.2.12 on_size_allocate()
void seq64::seqtime::on_size_allocate (
            Gtk::Allocation & a ) [private]
13.91.2.13 on_button_press_event()
bool seq64::seqtime::on_button_press_event (
              GdkEventButton * ) [inline], [private]
Simply returns false.
13.91.2.14 on_button_release_event()
bool seq64::seqtime::on_button_release_event (
             GdkEventButton * ) [inline], [private]
```

13.91.3 Field Documentation

```
13.91.3.1 m_seq
sequence& seq64::seqtime::m_seq [private]

13.91.3.2 m_scroll_offset_ticks
int seq64::seqtime::m_scroll_offset_ticks [private]

13.91.3.3 m_scroll_offset_x
int seq64::seqtime::m_scroll_offset_x [private]

13.91.3.4 m_zoom
int seq64::seqtime::m_zoom [private]

13.91.3.5 m_ppqn
int seq64::seqtime::m_ppqn [private]
```

13.92 seq64::sequence Class Reference

The sequence class is firstly a receptable for a single track of MIDI data read from a MIDI file or edited into a pattern.

Public Types

```
    enum select_action_e {
        e_select,
        e_select_one,
        e_is_selected,
        e_would_select,
        e_deselect,
        e_toggle_selection,
        e_remove_one }
```

This enumeration is used in selecting events and note.

Public Member Functions

• sequence (int ppqn=SEQ64_USE_DEFAULT_PPQN)

Principal constructor.

∼sequence ()

A rote destructor.

void partial_assign (const sequence &rhs)

A cut-down version of principal assignment operator.

event_list & events ()

'Getter' function for member m_events Non-const version.

· const event_list & events () const

'Getter' function for member m_events Const version.

• bool any_selected_notes () const

'Getter' function for member m_events.any_selected_notes()

• const triggers::List & triggerlist () const

'Getter' function for member m_triggers This is the const version.

triggers::List & triggerlist ()

'Getter' function for member m_triggers

int get_trigger_count () const

Gets the trigger count, useful for exporting a sequence.

- void set trigger paste tick (midipulse tick)
- midipulse get_trigger_paste_tick () const
- int number () const

'Getter' function for member m_seq_number

void number (int seqnum)

'Setter' function for member m_seq_number This setter will set the sequence number only if it has not already been set.

• void modify ()

A convenience function that we have to put here so that the m_parent pointer can be used without an additional include-file in the sequence.hpp module.

int event_count () const

Returns the number of events stored in m_events.

• void set_hold_undo (bool hold)

Modifies the undo-hold container.

int get_hold_undo () const

'Getter' function for member m_events_undo_hold.count()

• void set_have_undo ()

'Setter' function for member m_have_undo

bool have_undo () const

'Getter' function for member m_have_undo

• void set have redo ()

'Setter' function for member m_have_redo No reliable way to "unmodify" the performance here.

bool have_redo () const

'Getter' function for member m_have_redo

void push undo (bool hold=false)

Pushes the event-list into the undo-list or the upcoming undo-hold-list.

• void pop_undo ()

If there are items on the undo list, this function pushes the event-list into the redo-list, puts the top of the undo-list into the event-list, pops from the undo-list, calls verify_and_link(), and then calls unselect.

void pop_redo ()

If there are items on the redo list, this function pushes the event-list into the undo-list, puts the top of the redo-list into the event-list, pops from the redo-list, calls verify_and_link(), and then calls unselect.

void push_trigger_undo ()

Calls triggers::push_undo() with locking.

void pop_trigger_undo ()

Calls triggers::pop_undo() with locking.

void pop_trigger_redo ()

Calls triggers::pop_redo() with locking.

void set_name (const std::string &name)

Sets the sequence name member, m_name.

void set_name (char *name)

Sets the sequence name member, m name.

- void set_measures (int lengthmeasures)
- int get measures ()
- int get_ppqn () const

'Getter' function for member m_ppgn Provided as a convenience for the editable_events class.

void set_beats_per_bar (int beatspermeasure)

'Setter' function for member m_time_beats_per_measure

int get_beats_per_bar () const

'Getter' function for member m time beats per measure

void set_beat_width (int beatwidth)

'Setter' function for member m_time_beat_width

int get_beat_width () const

'Getter' function for member m time beat width

midipulse measures_to_ticks (int measures=1) const

A convenience function for calculating the number of ticks in the given number of measures.

void clocks per metronome (int cpm)

'Setter' function for member m_clocks_per_metronome

int clocks_per_metronome () const

'Getter' function for member m_clocks_per_metronome

void set_32nds_per_quarter (int tpq)

'Setter' function for member m_32nds_per_quarter

• int get_32nds_per_quarter () const

'Getter' function for member m_32nds_per_quarter

void us_per_quarter_note (long upqn)

'Setter' function for member m_us_per_quarter_note

• long us_per_quarter_note () const

'Getter' function for member m_us_per_quarter_note

void set_rec_vol (int rec_vol)

'Setter' function for member m_rec_vol If this velocity is greater than zero, then m_note_on_velocity will be set as well.

void set_song_mute (bool mute)

'Setter' function for member m_song_mute This function also calls set_dirty_mp() to make sure that the perfnames panel is updated to show the new mute status of the sequence.

void toggle_song_mute ()

'Setter' function for member m_song_mute This function toogles the song muting status.

• bool get_song_mute () const

'Getter' function for member m song mute

void apply_song_transpose ()

Applies the transpose value held by the master MIDI buss object, if non-zero, and if the sequence is set to be transposable.

· void set transposable (bool flag)

'Setter' function for member m_transposable Changing this flag modifies the sequence and performance.

bool get_transposable () const

'Getter' function for member m_transposable

• const char * get_name () const

'Getter' function for member m_name pointer

const std::string & name () const

'Getter' function for member m_name

void set_editing (bool edit)

'Setter' function for member m_editing

bool get_editing () const

'Getter' function for member m_editing

void set_raise (bool edit)

'Setter' function for member m_raise

• bool get_raise (void) const

'Getter' function for member m_raise

void set_length (midipulse len=0, bool adjust_triggers=true, bool verify=true)

Sets the length (m_length) and adjusts triggers for it, if desired.

• midipulse get_length () const

'Getter' function for member m_length

midipulse get_last_tick ()

Returns the last tick played, and is used by the editor's idle function.

void set_last_tick (midipulse tick)

'Setter' function for member m_last_tick This function used to be called "set_orig_tick()", now renamed to match up with get_last_tick().

midipulse mod_last_tick ()

Some MIDI file errors and other things can lead to an m_length of 0, which causes arithmetic errors when m_last_tick is modded against it.

void set_playing (bool)

Sets the playing state of this sequence.

• bool get_playing () const

'Getter' function for member m_playing

void toggle_playing ()

Toggles the playing status of this sequence.

• void toggle_queued ()

'Setter' function for member m_queued and m_queued_tick Toggles the queued flag and sets the dirty-mp flag.

void off_queued ()

'Setter' function for member m_queued Turns off (resets) the queued flag and sets the dirty-mp flag.

• void on_queued ()

'Setter' function for member m_queued Turns on (sets) the queued flag and sets the dirty-mp flag.

bool get_queued () const

'Getter' function for member m_queued

• midipulse get_queued_tick () const

'Getter' function for member m_queued_tick

bool check_queued_tick (midipulse tick) const

Helper function for perform.

void set_recording (bool)

'Setter' function for member m_recording and m_notes_on

bool get_recording () const

'Getter' function for member m_recording

· void set snap tick (int st)

'Setter' function for member m_snap_tick

void set_quantized_rec (bool qr)

'Setter' function for member m_quantized_rec

bool get_quantized_rec () const

'Getter' function for member m quantized rec

· void set thru (bool)

'Setter' function for member m_thru

• bool get_thru () const

'Getter' function for member m_thru

bool is dirty main ()

Returns the value of the dirty main flag, and sets that flag to false (i.e.

· bool is_dirty_edit ()

Returns the value of the dirty edit flag, and sets that flag to false.

bool is_dirty_perf ()

Returns the value of the dirty performance flag, and sets that flag to false.

bool is_dirty_names ()

Returns the value of the dirty names (heh heh) flag, and sets that flag to false.

void set_dirty_mp ()

Sets the dirty flags for names, main, and performance.

void set_dirty ()

Call set_dirty_mp() and then sets the dirty flag for editing.

· midibyte get_midi_channel () const

'Getter' function for member m_midi_channel

• bool is smf 0 () const

Returns true if this sequence is an SMF 0 sequence.

· void set_midi_channel (midibyte ch, bool user_change=false)

Sets the m_midi_channel number>

· void print () const

Prints a list of the currently-held events.

· void print_triggers () const

Prints a list of the currently-held triggers.

void play (midipulse tick, bool playback_mode)

The play() function dumps notes starting from the given tick, and it pre-buffers ahead.

void play_queue (midipulse tick, bool playbackmode)

Provides encapsulation for a series of called used in perform::play().

void add_note (midipulse tick, midipulse len, int note, bool paint=false, int velocity=SEQ64_PRESERVE_V

ELOCITY)

Adds a note of a given length and note value, at a given tick location.

bool add_event (const event &er)

Adds an event to the internal event list in a sorted manner.

void add_chord (int chord, midipulse tick, midipulse len, int note)

Adds a chord of a given length and note value, at a given tick location.

• void add event (midipulse tick, midibyte status, midibyte d0, midibyte d1, bool paint=false)

Adds a event of a given status value and data values, at a given tick location.

bool append_event (const event &er)

An alternative to add_event() that does not sort the events, even if the event list is implemented by an std::list.

· void sort events ()

Calls event_list::sort().

void add_trigger (midipulse tick, midipulse len, midipulse offset=0, bool adjust_offset=true)

Adds a trigger.

void split_trigger (midipulse tick)

Splits a trigger.

• void grow_trigger (midipulse tick_from, midipulse tick_to, midipulse len)

Grows a trigger.

void del_trigger (midipulse tick)

Deletes a trigger, that brackets the given tick, from the trigger-list.

• bool get_trigger_state (midipulse tick)

Checks the list of triggers against the given tick.

• bool select_trigger (midipulse tick)

Checks the list of triggers against the given tick.

· triggers::List get_triggers () const

Returns a copy of the triggers for this sequence.

• bool unselect_triggers ()

Unselects all triggers.

bool intersect triggers (midipulse position, midipulse &start, midipulse &ender)

This function examines each trigger in the trigger list.

bool intersect_notes (midipulse position, midipulse position_note, midipulse &start, midipulse &ender, int ¬e)

This function examines each note in the event list.

bool intersect_events (midipulse posstart, midipulse posend, midibyte status, midipulse &start)

This function examines each non-note event in the event list.

• void del selected trigger ()

Deletes the first selected trigger that is found.

void cut_selected_trigger ()

Copies and deletes the first selected trigger that is found.

void copy_selected_trigger ()

First, this function clears any unpasted middle-click tick setting.

void paste_trigger (midipulse paste_tick=SEQ64_NO_PASTE_TRIGGER)

If there is a copied trigger, then this function grabs it from the trigger clipboard and adds it.

bool move_selected_triggers_to (midipulse tick, bool adjust_offset, triggers::grow_edit_t which=triggers::G←
 ROW_MOVE)

Moves selected triggers as per the given parameters.

· midipulse selected trigger start ()

Gets the last-selected trigger's start tick.

midipulse selected_trigger_end ()

Gets the selected trigger's end tick.

midipulse get_max_trigger ()

Get the ending value of the last trigger in the trigger-list.

void move_triggers (midipulse start_tick, midipulse distance, bool direction)

Moves triggers in the trigger-list.

· void copy triggers (midipulse start tick, midipulse distance)

Copies triggers to another location.

void clear_triggers ()

Clears the whole list of triggers.

midipulse get_trigger_offset () const

'Getter' function for member m_trigger_offset

void set_midi_bus (char mb, bool user_change=false)

Sets the MIDI buss/port number to dump MIDI data to.

· char get midi bus () const

'Getter' function for member m_bus

void set_master_midi_bus (mastermidibus *mmb)

'Setter' function for member m_masterbus Do we need to call set_dirty_mp() here? It doesn't affect any user-interface elements.

• int select_note_events (midipulse tick_s, int note_h, midipulse tick_f, int note_l, select_action_e action)

Selects events in range provided: tick start, note high, tick end, and note low.

• int select_events (midipulse tick_s, midipulse tick_f, midibyte status, midibyte cc, select_action_e action)

Select all events in the given range, and returns the number selected.

• int select_events (midibyte status, midibyte cc, bool inverse=false)

Select all events with the given status, and returns the number selected.

- int select events (midipulse tick s, midipulse tick f, midibyte status)
- int select event handle (midipulse tick s, midipulse tick f, midibyte status, midibyte cc, int data s)

Use selected note ons if any.

int select_linked (long tick_s, long tick_f, midibyte status)

Used with seqevent when selecting Note On or Note Off, this function will select the opposite linked event.

int select_even_or_odd_notes (int note_len, bool even)

Selects every other note.

void select_all_notes (bool inverse=false)

New convenience function.

• int get_num_selected_notes () const

Counts the selected notes in the event list.

int get_num_selected_events (midibyte status, midibyte cc) const

Counts the selected events, with the given status, in the event list.

• void select all ()

Selects all events, unconditionally.

void copy_selected ()

Copies the selected events.

void cut selected (bool copyevents=true)

Cuts the selected events.

void paste_selected (midipulse tick, int note)

Pastes the selected notes (and only note events) at the given tick and the given note value.

void get_selected_box (midipulse &tick_s, int ¬e_h, midipulse &tick_f, int ¬e_l)

Returns the 'box' of the selected items.

void get_clipboard_box (midipulse &tick_s, int ¬e_h, midipulse &tick_f, int ¬e_l)

Returns the 'box' of the clipboard items.

· midipulse adjust timestamp (midipulse t, bool isnoteoff)

A new function to consolidate the adjustment of timestamps in a pattern.

midipulse trim_timestamp (midipulse t)

A new function to consolidate the adjustment of timestamps in a pattern.

midipulse clip_timestamp (midipulse ontime, midipulse offtime)

A new function to consolidate the growth/shrinkage of timestamps in a pattern.

void move_selected_notes (midipulse deltatick, int deltanote)

Removes and adds selected notes in position.

bool stream_event (event &ev)

Streams the given event.

bool change_event_data_range (midipulse tick_s, midipulse tick_f, midibyte status, midibyte cc, int d_s, int d f)

Changes the event data range.

 void change_event_data_lfo (double value, double range, double speed, double phase, wave_type_t wave, midibyte status, midibyte cc)

Modifies data events according to the parameters active in the LFO window (Ifownd).

void increment_selected (midibyte status, midibyte)

Increments events the match the given status and control values.

· void decrement selected (midibyte status, midibyte)

Decrements events the match the given status and control values.

void grow_selected (midipulse deltatick)

The original description was "Moves note off event." But this also gets called when simply selecting a second note via a ctrl-left-click, even in seq24.

void stretch_selected (midipulse deltatick)

Performs a stretch operation on the selected events.

• bool remove marked ()

Removes marked events.

bool mark_selected ()

Marks the selected events.

void remove_selected ()

Removes selected events.

void unpaint_all ()

Unpaints all events in the event-list.

· void unselect ()

Deselects all events, unconditionally.

• void verify and link ()

This function verifies state: all note-ons have a note-off, and it links note-offs with their note-ons.

void link_new ()

Links a new event.

void zero_markers ()

Resets everything to zero.

void play_note_on (int note)

Plays a note from the piano roll on the main bus on the master MIDI buss.

void play_note_off (int note)

Turns off a note from the piano roll on the main bus on the master MIDI buss.

void off_playing_notes ()

Sends a note-off event for all active notes.

void stop (bool song_mode=false)

Provides a helper function simplify and speed up perform :: reset_sequences().

void pause (bool song mode=false)

A pause version of stop().

void reset_draw_marker ()

This refreshes the play marker to the last tick.

void reset_draw_trigger_marker ()

Sets the draw-trigger iterator to the beginning of the trigger list.

draw_type_t get_next_note_event (midipulse *tick_s, midipulse *tick_f, int *note, bool *selected, int *velocity)

Each call to seqdata() fills the passed references with a events elements, and returns true.

• bool get_minmax_note_events (int &lowest, int &highest)

A new function provided so that we can find the minimum and maximum notes with only one (not two) traversal of the

bool get_next_event (midibyte status, midibyte cc, midipulse *tick, midibyte *d0, midibyte *d1, bool *selected, int evtype=EVENTS ALL)

Get the next event in the event list that matches the given status and control character.

• bool get_next_event (midibyte *status, midibyte *cc)

Get the next event in the event list.

bool get next trigger (midipulse *tick on, midipulse *tick off, bool *selected, midipulse *tick offset)

Get the next trigger in the trigger list, and set the parameters based on that trigger.

• void quantize_events (midibyte status, midibyte cc, midipulse snap_tick, int divide, bool linked=false)

Grabs the specified events, puts them into a list, quantizes them against the snap ticks, and merges them in to the event container.

• void push_quantize (midibyte status, midibyte cc, midipulse snap_tick, int divide, bool linked=false)

A new convenience function.

void transpose notes (int steps, int scale)

Transposes notes by the given steps, in accordance with the given scale.

- · void shift notes (midipulse ticks)
- void multiply_pattern (double multiplier)
- midibyte musical_key () const

'Getter' function for member m_musical_key

void musical_key (int key)

'Setter' function for member m musical key

midibyte musical_scale () const

'Getter' function for member m_musical_scale

void musical_scale (int scale)

'Setter' function for member m_musical_scale

• int background_sequence () const

'Getter' function for member m_background_sequence

void background sequence (int bs)

'Setter' function for member m_background_sequence Only partial validation at present, we do not want the upper limit to be hard-wired at this time.

· void show events () const

A member function to dump a summary of events stored in the event-list of a sequence.

void copy_events (const event_list &newevents)

Copies an external container of events into the current container, effectively replacing all of its events.

midipulse note_off_margin () const

'Getter' function for member m_note_length

Private Types

typedef std::stack< event_list > EventStack

Provides a stack of event-lists for use with the undo and redo facility.

Private Member Functions

- sequence & operator= (const sequence &rhs)
- bool event_in_range (const event &e, midibyte status, midipulse tick_s, midipulse tick f) const

A convenience function used a couple of times.

void set parent (perform *p)

'Setter' function for member m_parent Sets the "parent" of this sequence, so that it can get some extra information about the performance.

void put_event_on_bus (event &ev)

Takes an event that this sequence is holding, and places it on the MIDI buss.

void set_trigger_offset (midipulse trigger_offset)

Sets m_trigger_offset and wraps it to m_length.

void adjust_trigger_offsets_to_length (midipulse newlen)

Adjusts trigger offsets to the length specified for all triggers, and undo triggers.

- · midipulse adjust_offset (midipulse offset)
- · void remove (event_list::iterator i)

 $A\ helper\ function,\ which\ does\ not\ lock/unlock,\ so\ it\ is\ unsafe\ to\ call\ without\ supplying\ an\ iterator\ from\ the\ event-list.$

· void remove (event &e)

A helper function, which does not lock/unlock, so it is unsafe to call without supplying an iterator from the event-list.

· void remove all ()

Clears all events from the event container.

· bool channel match (const event &e) const

Checks to see if the event's channel matches the sequence's nominal channel.

Private Attributes

perform * m_parent

For pause support, we need a way for the sequence to find out if JACK transport is active.

· event list m events

This list holds the current pattern/sequence events.

· triggers m_triggers

The triggers associated with the sequence, used in the performance/song editor.

· event_list m_events_undo_hold

Provides a list of event actions to undo for the Stazed LFO and segdata support.

· bool m_have_undo

A stazed flag indicating that we have some undo information.

bool m_have_redo

A stazed flag indicating that we have some redo information.

EventStack m_events_undo

Provides a list of event actions to undo.

EventStack m_events_redo

Provides a list of event actions to redo.

· event list::iterator m iterator draw

An iterator for drawing events.

• bool m_channel_match

A new feature for recording, based on a "stazed" feature.

• midibyte m midi channel

Contains the proper MIDI channel for this sequence.

midibyte m_bus

Contains the proper MIDI bus number for this sequence.

· bool m song mute

Provides a flag for the song playback mode muting.

bool m_transposable

Indicate if the sequence is transposable or not.

· int m notes on

Provides a member to hold the polyphonic step-edit note counter.

mastermidibus * m_masterbus

Provides the master MIDI buss which handles the output of the sequence to the proper buss and MIDI channel.

int m_playing_notes [SEQ64_MIDI_NOTES_MAX]

Provides a "map" for Note On events.

bool m_was_playing

Indicates if the sequence was playing.

bool m_playing

True if sequence playback currently is in progress for this sequence.

· bool m recording

True if sequence recording currently is in progress for this sequence.

bool m_quantized_rec

True if recording in quantized mode.

• bool m thru

True if recording in MIDI-through mode.

bool m_queued

True if the events are queued.

· bool m dirty main

These flags indicate that the content of the sequence has changed due to recording, editing, performance management, or even (?) a name change.

bool m_dirty_edit

Provides the main is-edited flag.

bool m_dirty_perf

Provides performance dirty flagflag.

· bool m_dirty_names

Provides the names dirtiness flag.

bool m_editing

Indicates that the sequence is currently being edited.

· bool m raise

Used in segmenu and segedit.

std::string m name

Provides the name/title for the sequence.

midipulse m_last_tick

These members manage where we are in the playing of this sequence, including triggering.

• midipulse m_queued_tick

Provides the next tick to play?

• midipulse m_trigger_offset

Provides the trigger offset.

• const int m_maxbeats

This constant provides the scaling used to calculate the time position in ticks (pulses), based also on the PPQN value.

int m_ppqn

Holds the PPQN value for this sequence, so that we don't have to rely on a global constant value.

int m_seq_number

A new member so that the sequence number is carried along with the sequence.

· midipulse m length

Holds the length of the sequence in pulses (ticks).

• midipulse m_snap_tick

The size of snap in units of pulses (ticks).

int m_time_beats_per_measure

Provides the number of beats per bar used in this sequence.

· int m time beat width

Provides with width of a beat.

• int m_clocks_per_metronome

Augments the beats/bar and beat-width with the additional values included in a Time Signature meta event.

int m_32nds_per_quarter

Augments the beats/bar and beat-width with the additional values included in a Time Signature meta event.

long m_us_per_quarter_note

Augments the beats/bar and beat-width with the additional values included in a Tempo meta event.

int m_rec_vol

The volume to be used when recording.

int m_note_on_velocity

The Note On velocity used.

int m_note_off_velocity

The Note Off velocity used.

midibyte m_musical_key

Holds a copy of the musical key for this sequence, which we now support writing to this sequence.

midibyte m_musical_scale

Holds a copy of the musical scale for this sequence, which we now support writing to this sequence.

• int m_background_sequence

Holds a copy of the background sequence number for this sequence, which we now support writing to this sequence.

• mutex m_mutex

Provides locking for the sequence.

• const midipulse m_note_off_margin

Provides the number of ticks to shave off of the end of painted notes.

Static Private Attributes

• static event_list m_events_clipboard

A static clipboard for holding pattern/sequence events.

Friends

- class perform
- class triggers

13.92.1 Detailed Description

More members than you can shake a stick at.

13.92.2 Member Typedef Documentation

13.92.2.1 EventStack

```
typedef std::stack<event_list> seq64::sequence::EventStack [private]
```

13.92.3 Member Enumeration Documentation

```
13.92.3.1 select_action_e
```

```
enum seq64::sequence::select_action_e
```

Se the select_note_events() and select_events() functions.

Enumerator

e_select	Selection in progress.		
e_select_one	To select a single event.		
e_is_selected	The events are selected.		
e_would_select	The events would be selected.		
e_deselect	To deselect event under the cursor.		
e_toggle_selection	Toggle selection under cursor.		
e_remove_one	To remove one note under the cursor.		

13.92.4 Constructor & Destructor Documentation

Parameters

ppqn | Provides the PPQN parameter to perhaps alter the default PPQN value of this sequence.

```
13.92.4.2 \simsequence()
seq64::sequence::\simsequence ( )
```

13.92.5 Member Function Documentation

```
13.92.5.1 operator=()
```

13.92.5.2 partial_assign()

We're replacing that incomplete function (many members are not assigned) with the more accurately-named partial_assign() function.

It did not assign them all, so we created this partial_assign() function to do this work, and replaced operator =() with this function in client code.

Threadsafe

Parameters

rhs Provides the source of the new member values.

```
13.92.5.3 events() [1/2]
event_list& seq64::sequence::events ( ) [inline]
13.92.5.4 events() [2/2]
const event_list& seq64::sequence::events ( ) const [inline]
13.92.5.5 any_selected_notes()
bool seq64::sequence::any_selected_notes ( ) const [inline]
13.92.5.6 triggerlist() [1/2]
const triggers::List& seq64::sequence::triggerlist ( ) const [inline]
13.92.5.7 triggerlist() [2/2]
triggers::List& seq64::sequence::triggerlist ( ) [inline]
13.92.5.8 get_trigger_count()
int seq64::sequence::get_trigger_count ( ) const [inline]
13.92.5.9 set_trigger_paste_tick()
void seq64::sequence::set_trigger_paste_tick (
             midipulse tick ) [inline]
```

```
13.92.5.10 get_trigger_paste_tick()
midipulse seq64::sequence::get_trigger_paste_tick ( ) const [inline]
13.92.5.11 number() [1/2]
int seq64::sequence::number ( ) const [inline]
13.92.5.12 number() [2/2]
void seq64::sequence::number (
              int seqnum ) [inline]
13.92.5.13 modify()
void seq64::sequence::modify ( )
One minor issue is how can we unmodify the performance? We'd need to keep a count/stack of modifications over
all sequences in the performance. Probably not practical, in general. We will probably keep track of the modification
of the buss (port) and channel numbers, as per GitHub Issue #47...
13.92.5.14 event_count()
int seq64::sequence::event_count ( ) const
Note that only playable events are counted in a sequence. If a sequence class function provides a mutex, call
m_events.count() instead.
Threadsafe
Returns
     Returns m_events.count().
13.92.5.15 set_hold_undo()
```

Parameters

hold

Threadsafe

If true, then the events in the m_events container are added to the m_events_undo_hold container. Otherwise, that container is cleared.

```
13.92.5.16 get_hold_undo()
int seq64::sequence::get_hold_undo ( ) const [inline]
13.92.5.17 set_have_undo()
void seq64::sequence::set_have_undo ( ) [inline]
13.92.5.18 have_undo()
bool seq64::sequence::have_undo ( ) const [inline]
13.92.5.19 set_have_redo()
void seq64::sequence::set_have_redo ( ) [inline]
13.92.5.20 have_redo()
bool seq64::sequence::have_redo ( ) const [inline]
13.92.5.21 push_undo()
void seq64::sequence::push_undo (
             bool hold = false )
```

Parameters

hold

A new parameter for the stazed undo/redo support, not yet used. If true, then the events go into the undo-hold-list.

```
13.92.5.22 pop_undo()

void seq64::sequence::pop_undo ( )
```

We would like to be able to set perform's modify flag to false here, but other sequences might still be in a modified state. We could add a modify flag to sequence, and falsify that flag here. Something to think about.

Threadsafe

```
13.92.5.23 pop_redo()
void seq64::sequence::pop_redo ( )
Threadsafe
13.92.5.24 push_trigger_undo()
void seq64::sequence::push_trigger_undo ( )
Threadsafe
13.92.5.25 pop_trigger_undo()
void seq64::sequence::pop_trigger_undo ( )
Threadsafe
13.92.5.26 pop_trigger_redo()
void seq64::sequence::pop_trigger_redo ( )
Threadsafe
13.92.5.27 set_name() [1/2]
void seq64::sequence::set_name (
             const std::string & name )
```

```
13.92.5.28 set_name() [2/2]
void seq64::sequence::set_name (
             char * name )
13.92.5.29 set_measures()
void seq64::sequence::set_measures (
             int lengthmeasures )
13.92.5.30 get_measures()
int seq64::sequence::get_measures ( )
13.92.5.31 get_ppqn()
int seq64::sequence::get_ppqn ( ) const [inline]
13.92.5.32 set_beats_per_bar()
void seq64::sequence::set_beats_per_bar (
             int beatspermeasure )
Threadsafe
Parameters
 beatspermeasure
                    The new setting of the beats-per-bar value.
```

```
13.92.5.33 get_beats_per_bar()
```

```
int seq64::sequence::get_beats_per_bar ( ) const [inline]
```

```
13.92.5.34 set_beat_width()
void seq64::sequence::set_beat_width (
             int beatwidth )
Threadsafe
Parameters
 beatwidth
             The new setting of the beat width value.
13.92.5.35 get_beat_width()
int seq64::sequence::get_beat_width ( ) const [inline]
Threadsafe
13.92.5.36 measures_to_ticks()
midipulse seq64::sequence::measures_to_ticks (
              int measures = 1 ) const [inline]
13.92.5.37 clocks_per_metronome() [1/2]
void seq64::sequence::clocks_per_metronome (
             int cpm ) [inline]
13.92.5.38 clocks_per_metronome() [2/2]
int seq64::sequence::clocks_per_metronome ( ) const [inline]
13.92.5.39 set_32nds_per_quarter()
```

void seq64::sequence::set_32nds_per_quarter (
 int tpq) [inline]

```
13.92.5.40 get_32nds_per_quarter()
int seq64::sequence::get_32nds_per_quarter ( ) const [inline]
13.92.5.41 us_per_quarter_note() [1/2]
void seq64::sequence::us_per_quarter_note (
             long upqn ) [inline]
13.92.5.42 us_per_quarter_note() [2/2]
long seq64::sequence::us_per_quarter_note ( ) const [inline]
13.92.5.43 set_rec_vol()
void seq64::sequence::set_rec_vol (
             int recvol )
Threadsafe
Parameters
 recvol
         The new setting of the recording volume setting. It is used only if it ranges from 0 to
         SEQ64_MAX_NOTE_ON_VELOCITY, or is set to SEQ64_PRESERVE_VELOCITY.
```

```
13.92.5.46 get_song_mute()
bool seq64::sequence::get_song_mute ( ) const [inline]
13.92.5.47 apply_song_transpose()
void seq64::sequence::apply_song_transpose ( )
13.92.5.48 set_transposable()
void seq64::sequence::set_transposable (
              bool flag )
Note that when a sequence is being read from a MIDI file, it will not yet have a parent, so we have to check for that
before setting the perform modify flag.
13.92.5.49 get_transposable()
bool seq64::sequence::get_transposable ( ) const [inline]
13.92.5.50 get_name()
const char* seq64::sequence::get_name ( ) const [inline]
Deprecated
13.92.5.51 name()
const std::string& seq64::sequence::name ( ) const [inline]
13.92.5.52 set_editing()
void seq64::sequence::set_editing (
             bool edit ) [inline]
```

```
13.92.5.53 get_editing()
```

```
bool seq64::sequence::get_editing ( ) const [inline]
```

13.92.5.54 set_raise()

```
void seq64::sequence::set_raise (
          bool edit ) [inline]
```

13.92.5.55 get_raise()

13.92.5.56 set_length()

```
void seq64::sequence::set_length (
          midipulse len = 0,
          bool adjust_triggers = true,
          bool verify = true )
```

This function is called in seqedit::apply_length(), when the user selects a sequence length in measures. This function is also called when reading a MIDI file.

There's an issue, though. If the application is compiled to use the original std::list container for MIDI events, that implementation sorts the container after every event insertion. If the application is compiled to used the std::map container (to speed up the reading of large MIDI files *greatly*), sorting happens automatically. But, if we use the original std::list implementation, but leave the sorting until later (to speed up the reading of large MIDI files *greatly*), then the verify_and_link() call that happens with every new event happens before the events are sorted, and the result is elongated notes showing up in the pattern slot in the main window. Therefore, we need a way to skip the verification when reading a MIDI file, and do the verification only after all events are read.

That function calculates the length in ticks:

```
L = M x B x 4 x P / W
L == length (ticks or pulses)
M == number of measures
B == beats per measure
P == pulses per quarter-note
W == beat width in beats per measure

For our "b4uacuse" MIDI file, M can be about 100 measures, B is 4, P can be 192 (but we want to support higher values), and W is 4.
So L = 100 * 4 * 4 * 192 / 4 = 76800 ticks. Seems small.
```

len	The length value to be set. If it is smaller than ppqn/4, then it is set to that value, unless zero, in which case m_length is used and does not change. It also sets the length value the sequence's triggers.	
adjust_triggers	If true, m_triggers.adjust_offsets_to_length() is called. The value defaults to true.	
verify	This new parameter defaults to true. If true, verify_and_link() is called. Otherwise, it is not, and the caller should call this function with the default value after reading all the events.	

```
13.92.5.57 get_length()
midipulse seq64::sequence::get_length ( ) const [inline]

13.92.5.58 get_last_tick()
midipulse seq64::sequence::get_last_tick ( )
```

If m_length is 0, this function returns m_last_tick - m_trigger_offset, to avoid an arithmetic exception. Should we return 0 instead?

Note that seqroll calls this function to help get the location of the progress bar. What does perfedit do?

This function replaces the "m_last_tick % m_length", returning m_last_tick if m_length is 0 or 1.

```
13.92.5.61 set_playing() void seq64::sequence::set_playing ( bool <math>p )
```

When playing, and the sequencer is running, notes get dumped to the ALSA buffers.

р

Provides the playing status to set. True means to turn on the playing, false means to turn it off, and turn off any notes still playing.

```
13.92.5.62 get_playing()
bool seq64::sequence::get_playing ( ) const [inline]
13.92.5.63 toggle_playing()
void seq64::sequence::toggle_playing ( ) [inline]
How exactly does this differ from toggling the mute status?
13.92.5.64 toggle_queued()
void seq64::sequence::toggle_queued ( )
Also calculates the queued tick based on m_last_tick.
Threadsafe
13.92.5.65 off_queued()
void seq64::sequence::off_queued ( )
Do we need to set m_queued_tick as in toggle_queued()? Currently not used.
Threadsafe
13.92.5.66 on_queued()
void seq64::sequence::on_queued ( )
Do we need to set m_queued_tick as in toggle_queued()? Currently not used.
Threadsafe
13.92.5.67 get_queued()
bool seq64::sequence::get_queued ( ) const [inline]
```

```
13.92.5.68 get_queued_tick()
midipulse seq64::sequence::get_queued_tick ( ) const [inline]
13.92.5.69 check_queued_tick()
bool seq64::sequence::check_queued_tick (
             midipulse tick ) const [inline]
13.92.5.70 set_recording()
void seq64::sequence::set_recording (
             bool r )
Threadsafe
13.92.5.71 get_recording()
bool seq64::sequence::get_recording ( ) const [inline]
13.92.5.72 set_snap_tick()
void seq64::sequence::set_snap_tick (
             int st )
Threadsafe
13.92.5.73 set_quantized_rec()
void seq64::sequence::set_quantized_rec (
             bool qr )
Threadsafe
13.92.5.74 get_quantized_rec()
bool seq64::sequence::get_quantized_rec ( ) const [inline]
```

Returns

Returns the dirty status.

```
13.92.5.75 set_thru()
void seq64::sequence::set_thru (
              bool r )
Threadsafe
13.92.5.76 get_thru()
bool seq64::sequence::get_thru ( ) const [inline]
13.92.5.77 is_dirty_main()
bool seq64::sequence::is_dirty_main ( )
resets it). This flag signals that a redraw is needed from recording.
Threadsafe
Returns
     Returns the dirty status.
13.92.5.78 is_dirty_edit()
bool seq64::sequence::is_dirty_edit ( )
The m_dirty_edit flag is set by the function set_dirty().
Threadsafe
Returns
     Returns the dirty status.
13.92.5.79 is_dirty_perf()
bool seq64::sequence::is_dirty_perf ( )
Threadsafe
```

```
13.92.5.80 is_dirty_names()
bool seq64::sequence::is_dirty_names ( )
```

Not sure that we need to lock a boolean on modern processors.

Threadsafe

Returns

Returns the dirty status.

```
13.92.5.81 set_dirty_mp()
void seq64::sequence::set_dirty_mp ( )
```

These flags are meant for causing user-interface refreshes, not for performance modification.

m_dirty_names is set to false in is_dirty_names(); m_dirty_names is set to false in is_dirty_main(); m_dirty_names is set to false in is_dirty_perf().

Not threadsafe

ch	The MIDI channel to set as the channel number for this sequence.
user_change	If true (the default value is false), the user has decided to change this value, and we might need to modify the perform's dirty flag, so that the user gets prompted for a change, This is a response to GitHub issue #47, where channel changes do not cause a prompt to save the sequence.

This function is called by the sequencer thread, performance. The tick comes in as global tick. It turns the sequence off after we play in this frame.

Note

With pause support, the progress bar for the pattern/sequence editor does what we want: pause with the pause button, and rewind with the stop button. Works with JACK, with issues, but we'd like to have the stop button do a rewind in JACK, too.

The trigger calculations have been offloaded to the triggers::play() function. It's return value and side-effects tell if there's a change in playing based on triggers and tells the ticks that bracket it.

Parameters

end_tick	Provides the current end-tick value. The tick comes in as a global tick.
playback_mode	Provides how playback is managed. True indicates that it is performance/song-editor playback, controlled by the set of patterns and triggers set up in that editor, and saved with the song in seq24 format. False indicates that the playback is controlled by the main window, in live mode.

13.92.5.89 play_queue()

Starts the playing of a pattern/sequence. This function just has the sequence dump its events. It ignores the sequence if it has no playable MIDI events.

Change Note ca 2016-10-12 Issue #39. Removed the check for a non-zero event count. This lets the seqroll show the progress bar in motion.

Parameters

tick		Provides the tick/pulse from which to start playing.
pla	ybackmode	Indicates if the playback is in live mode (false) or song mode (true).

13.92.5.90 add_note()

It adds a single note-on / note-off pair.

The paint parameter indicates if we care about the painted event, so then the function runs though the events and deletes the painted ones that overlap the ones we want to add.

Also note that push_undo() is not incorporated into this function, for the sake of speed.

Here, we could ignore events not on the sequence's channel, as an option. We have to be careful because this function can be used in painting notes.

Stazed:

```
http://www.blitter.com/~russtopia/MIDI/~jglatt/tech/midispec.htm

Note Off: The first data is the note number. There are 128 possible notes on a MIDI device, numbered 0 to 127 (where Middle C is note number 60). This indicates which note should be released. The second data byte is the velocity, a value from 0 to 127. This indicates how quickly the note should be released (where 127 is the fastest). It's up to a MIDI device how it uses velocity information. Often velocity will be used to tailor the VCA release time. MIDI devices that can generate Note Off messages, but don't implement velocity features, will transmit Note Off messages with a preset velocity of 64.
```

Also, we now see that seq24 never used the recording-velocity member (m_rec_vol). We use it to modify the new m_note_on_velocity member if the user changes it in the seqedit window.

tick	The time destination of the new note, in pulses.
len	The duration of the new note, in pulses.
note	The pitch destination of the new note.
paint	If true, repaint the whole set of events, in order to be left with a clean view of the inserted event. The default is false.
velocity	If not set to SEQ64_PRESERVE_VELOCITY, the velocity of the note is set to this value. Otherwise, it is hard-wired to the stored note-on velocity. The name of this macro is counter-intuitive here. Currently, the note-off velocity is HARD-WIRED!

Then it reset the draw-marker and sets the dirty flag.

Currently, when reading a MIDI file [see the midifile::parse() function], only the main events (notes, after-touch, pitch, program changes, etc.) are added with this function. So, we can rely on reading only playable events into a sequence. Well, actually, certain meta-events are also read, to obtain channel, buss, and more settings. Also read for a sequence, if the global-sequence flag is not set, are the new key, scale, and background sequence parameters.

This module (sequencer) adds all of those events as well, but it can surely add other events. We should assume that any events added by sequencer are playable/usable.

Here, we could ignore events not on the sequence's channel, as an option. We have to be careful because this function can be used in painting events.

Threadsafe

Warning

This pushing (and, in writing the MIDI file, the popping), causes events with identical timestamps to be written in reverse order. Doesn't affect functionality, but it's puzzling until one understands what is happening. Actually, this is true only in Seq24, we've fixed that behavior for Sequencer64.

Parameters

er Provide a reference to the event to be added; the event is copied into the events cont	ainer.
---	--------

Returns

Returns true if the event was added.

13.92.5.92 add_chord()

If SEQ64_STAZED_CHORD_GENERATOR is not defined, it devolves to add_note().

Todo Add the ability to preserve the incoming velocity.

Threadsafe

Parameters

chord	If greater than 0 (and less than c_chord_number), a chord (multiple notes) will be generated using this chord in the c_chord_table[] array. Otherwise, only a single note will be added.	
tick	The time destination of the new note, in pulses.	
len	The duration of the new note, in pulses.	
note	The pitch destination of the new note.	

The paint parameter indicates if we care about the painted event, so then the function runs though the events and deletes the painted ones that overlap the ones we want to add.

Threadsafe

tick	The time destination of the event.
status	The type of event to add.
d0	The first data byte for the event.
d1	The second data byte for the event (if needed).
paint	If true, the inserted event is marked for painting.

13.92.5.94 append_event()

This function is meant mainly for reading the MIDI file, to save a lot of time.

Parameters

er Provide a reference to the event to be added; the event is copied into the events container.

Returns

Returns true if the event was added.

13.92.5.95 sort_events()

```
void seq64::sequence::sort_events ( ) [inline]
```

13.92.5.96 add_trigger()

A pass-through function that calls triggers::add(). See that function for more details.

Threadsafe

Parameters

tick	The time destination of the trigger.
len	The duration of the trigger.
offset	The performance offset of the trigger.
fixoffset	If true, adjust the offset.

13.92.5.97 split_trigger()

This is the public overload of split_trigger.

Threadsafe

Parameters

13.92.5.98 grow_trigger()

See triggers::grow() for more information.

Parameters

tickfrom	The desired from-value back which to expand the trigger, if necessary.
tickto	The desired to-value towards which to expand the trigger, if necessary.
len	The additional length to append to tickto for the check.

Threadsafe

13.92.5.99 del_trigger()

See triggers::remove().

Threadsafe

Parameters

tick Provides the tick to be used for finding the trigger to be erased.

13.92.5.100 get_trigger_state()

If any trigger is found to bracket that tick, then true is returned.

tick Provides the tick of interest.

Returns

Returns true if a trigger is found that brackets the given tick.

13.92.5.101 select_trigger()

If any trigger is found to bracket that tick, then true is returned, and the trigger is marked as selected.

Parameters

tick Provides the tick of interest.

Returns

Returns true if a trigger is found that brackets the given tick; this is the return value of m_triggers.select().

13.92.5.102 get_triggers()

```
triggers::List seq64::sequence::get_triggers ( ) const
```

This function is basically a threadsafe version of sequence:: trigger list().

Returns

Returns of copy of m_triggers.triggerlist().

13.92.5.103 unselect_triggers()

```
bool seq64::sequence::unselect_triggers ( )
```

Returns

Returns the m_triggers.unselect() return value.

13.92.5.104 intersect_triggers()

If the given position is between the current trigger's tick-start and tick-end values, the these values are copied to the start and end parameters, respectively, and then we exit. See triggers::intersect().

Threadsafe

Parameters

position	The position to examine.
start	The destination for the starting tick of the matching trigger.
ender	The destination for the ending tick of the matching trigger.

Returns

Returns true if a trigger was found whose start/end ticks contained the position. Otherwise, false is returned, and the start and end return parameters should not be used.

13.92.5.105 intersect_notes()

If the given position is between the current notes on and off time values, values, the these values are copied to the start and end parameters, respectively, the note value is copied to the note parameter, and then we exit.

Threadsafe

Parameters

ſ		position	The position to examine.
		position_note	I think this is the note value we might be looking for ???
	out	start	The destination for the starting timestamp of the matching note.
	out	ender	The destination for the ending timestamp of the matching note.
	out	note	The destination for the note of the matching event. Why is this an int value???

Returns

Returns true if a event was found whose start/end ticks contained the position. Otherwise, false is returned, and the start and end return parameters should not be used.

13.92.5.106 intersect_events()

If the given position is between the current notes's timestamp-start and timestamp-end values, the these values are copied to the posstart and posend parameters, respectively, and then we exit.

Threadsafe

Parameters

posstart	The starting position to examine.
posend	The ending position to examine.
status	The desired status value.
start	The destination for the starting timestamp of the matching trigger.

Returns

Returns true if a event was found whose start/end timestamps contained the position. Otherwise, false is returned, and the start and end return parameters should not be used.

```
13.92.5.107 del_selected_trigger()

void seq64::sequence::del_selected_trigger ( )

13.92.5.108 cut_selected_trigger()

void seq64::sequence::cut_selected_trigger ( )

13.92.5.109 copy_selected_trigger()

void seq64::sequence::copy_selected_trigger ( )

Then it copies the first selected trigger that is found.

13.92.5.110 paste_trigger()
```

void seq64::sequence::paste_trigger (

Why isn't this protected by a mutex? We will enable this if anything bad happens, such as a deadlock, or corruption, that we can prove happens here.

midipulse paste_tick = SEQ64_NO_PASTE_TRIGGER)

paste_tick	A new parameter that provides the tick for pasting, or SEQ64_NO_PASTE_TRIGGER (-1) if there	1
	is none.	

13.92.5.111 move_selected_triggers_to()

Threadsafe

Parameters

tick	The tick at which the trigger starts.
adjustoffset	Set to true if the offset is to be adjusted.
which	Selects which movement will be done, as discussed above.

Returns

Returns the value of triggers::move_selected(), which indicate that the movement could be made. Used in Seq24PerfInput::handle_motion_key().

13.92.5.112 selected_trigger_start()

```
midipulse seq64::sequence::selected_trigger_start ( )
```

Threadsafe

Returns

Returns the tick_start() value of the last-selected trigger. If no triggers are selected, then -1 is returned.

13.92.5.113 selected_trigger_end()

```
midipulse seq64::sequence::selected_trigger_end ( )
```

Threadsafe

Returns

Returns the tick_end() value of the last-selected trigger. If no triggers are selected, then -1 is returned.

13.92.5.114 get_max_trigger()

```
midipulse seq64::sequence::get_max_trigger ( )
```

Threadsafe

Returns

Returns the maximum trigger value.

13.92.5.115 move_triggers()

Note the dependence on the m_length member being kept in sync with the parent's value of m_length.

Threadsafe

Parameters

starttick	The current location of the triggers.
distance	The distance away from the current location to which to move the triggers.
direction	If true, the triggers are moved forward. If false, the triggers are moved backward.

13.92.5.116 copy_triggers()

start	tick	The current location of the triggers.
dista	nce	The distance away from the current location to which to copy the triggers.

```
13.92.5.117 clear_triggers()

void seq64::sequence::clear_triggers ( )

Threadsafe

13.92.5.118 get_trigger_offset()

midipulse seq64::sequence::get_trigger_offset ( ) const [inline]

13.92.5.119 set_midi_bus()

void seq64::sequence::set_midi_bus ( char mb, bool user_change = false )
```

Threadsafe

Parameters

mb	The MIDI buss to set as the buss number for this sequence. Also called the "MIDI port" number.
user_change	If true (the default value is false), the user has decided to change this value, and we might need to modify the perform's dirty flag, so that the user gets prompted for a change, This is a response to GitHub issue #47, where buss changes do not cause a prompt to save the sequence.

mmb	Provides a pointer to the master MIDI buss for this sequence. This should be a reference, but isn't, nor is it
	checked.

13.92.5.122 select_note_events()

Be aware the the event::is_note() function is used, and that it includes Aftertouch events, which generally need to stick with their Note On counterparts.

If a "note" event is detected, then we skip it. This is necessary since channel pressure and control change use d0 for seqdata, and d0 is returned by get_note(). This causes note selection to occasionally select them when their seqdata values are within range of the tick selection. So therefore we want only Note Ons and Note Offs.

Note

The continuation below ("continue") is necessary since channel pressure and control change use d0 for sequata [which is returned by get_note()]. This causes seqroll note selection to occasionally select them when their sequata values are within the range of tick selection. So only, note ons and offs. What about Aftertouch? We have the event::is_note() function for that.

Parameters

tick_s	The starting tick.
note⊷	The highest note selected.
_h	
tick_f	The ending, or finishing, tick.
note⊷	The lowest note selected.
_/	
action	The action to perform on the selection.

Returns

Returns the number of notes selected.

```
midipulse tick_f,
midibyte status,
midibyte cc,
select_action_e action )
```

Note that there is also an overloaded version of this function.

Threadsafe

Parameters

tick⊷	The start time of the selection.
_s	
tick←	The finish time of the selection.
_f	
status	The desired event in the selection.
сс	The desired control-change in the selection, if the event is a control-change.
action	The desired selection action.

Returns

Returns the number of events selected.

Note that there is also an overloaded version of this function.

Threadsafe

Warning

This used to be a void function, so it just returns 0 for now.

Parameters

status	Provides the status value to be selected.
СС	If the status is EVENT_CONTROL_CHANGE, then data byte 0 must match this value.
inverse	If true, invert the selection.

Returns

Always returns 0.

```
13.92.5.125 select_events() [3/3]
```

13.92.5.126 select_event_handle()

```
int seq64::sequence::select_event_handle (
    midipulse tick_s,
    midipulse tick_f,
    midibyte status,
    midibyte cc,
    int dats )
```

Parameters

tick⊷	Provides the starting tick.
_s	
tick←	Provides the ending (finishing) tick.
_f	
status	Provides the desired MIDI event to be selected.
cc	Provides the desired MIDI control value to be selected.
dats	Provides the center of a small data value range of plus or minus 2.

Returns

Returns the number of events selected.

13.92.5.127 select_linked()

tick←	Provides the starting tick.
_s	
tick←	Provides the ending (finishing) tick.
_f	
status	Provides the desired MIDI event to be selected.

Returns

Returns the number of notes selected.

13.92.5.128 select_even_or_odd_notes()

Enabled only if USE_STAZED_ODD_EVEN_SELECTION is defined.

Parameters

note_len	The desired note lengths for the selection.
even	True if we want the even notes.

Returns

Returns the number of notes selected.

13.92.5.129 select_all_notes()

What about Aftertouch events? I think we need to select them as well in seqedit, so let's add that selection here as well.

Parameters

inverse	If set to true (the default is false), then this causes the selection to be inverted.
---------	---

13.92.5.130 get_num_selected_notes()

```
int seq64::sequence::get_num_selected_notes ( ) const
```

Threadsafe

Returns

Returns m_events.count_selected_notes().

13.92.5.131 get_num_selected_events()

If the event is a control change (CC), then it must also match the given CC value.

Threadsafe

Parameters

status	The desired kind of event to count.
cc	The desired control-change to count, if the event is a control-change.

Returns

Returns m_events.count_selected_events().

```
13.92.5.132 select_all()

void seq64::sequence::select_all ( )

Threadsafe

13.92.5.133 copy_selected()

void seq64::sequence::copy_selected ( )
```

This function also has the danger, discovered by user 0rel, of events being modified after being added to the clipboard. So we add his reconstruction fix here as well. To summarize the steps:

```
-# Clear the m_events_clipboard. NO! If we have no events to
   copy to the clipboard, we do not want to clear it. This kills
   cut-and-paste functionality.
-# Add all selected events in this clipboard to the sequence.
-# Normalize the timestamps of the events in the clip relative to the
   timestamp of the first selected event. (Is this really needed?)
-# Reconstruct/reconstitute the m_events_clipboard.
```

This process is a bit easier to manage than erase/insert on events because std::multimap has no erase() function that returns the next valid iterator. Also, we use a local clipboard first, to save on copying. We've enhanced the error-checking, too.

Finally, note that m_events_clipboard is a static member of sequence, so:

```
-# Copying can be done between sequences.
-# Access to it needs to be protected by a mutex.
```

Threadsafe

13.92.5.134 cut_selected()

Pushes onto the undo stack, may copy the events, marks the selected events, and removes them. Now also sets the dirty flag so that the caller doesn't have to. Also raises the modify flag on the parent perform object.

copvevents	If true, copy the selected events before marking and removing them.
00/0/01/01	··· ··· ··· ·· · · · · · · · · · · ·

13.92.5.135 paste selected()

Also, we've moved external calls to push undo() into this function. The caller shouldn't have to do that.

The event_keys used to access/sort the multimap event_list is not updated after changing timestamp/rank of the stored events. Regenerating all key/value pairs before merging them solves this issue, so that the order of events in the sequence will be preserved. This action is not needed for moving or growing events. Nor is it needed if the old std::list implementation of the event container is compiled in. However, it is needed in any operation that modifies the timestamp of an event inside the container:

```
- copy_selected()
- paste_selected()
- quantize_events() TODO TODO TODO!
```

The alternative to reconstructing the map is to erase-and-insert the events modified in the code above, rather than just tweaking their values, which have an effect on sorting for the event-map implementation. However, multimap does not provide an erase() function that returns the next valid iterator, which would complicate this method of operation. So we're inclined to stick with this solution.

There was an issue with copy/pasting a whole sequence. The pasted events did not go to their destination, but overlayed the original events. This bugs also occurred in Seq24 0.9.2. It occurs with the allofarow.mid file when doing Ctrl-A Ctrl-C Ctrl-V Move-Mouse Left-Click. It turns out the original code was checking only the first event to see if it was a Note event. For sequences that started with a Control Change or Program Change (or other non-Note events), the highest note was never modified, and none of the note events were adjusted.

Finally, we only want to transpose note events (i.e. alter m_data[0]), and not other kinds of events. We still need to figure out what to do with aftertouch, though. Currently likely to be covered by the processing of the note that it accompanies.

Threadsafe

tick	The time destination for the paste. This represents the "x" coordinate of the upper left corner of the paste-box. It will be converted to an offset, for example pasting every event 48 ticks forward from the original copy.
note	The note/pitch destination for the paste. This represents the "y" coordinate of the upper left corner of the paste-box. It will be converted to an offset, for example pasting every event 7 notes higher than the original copy.

13.92.5.136 get_selected_box()

Note the common-code betweem this function and get_clipboard_box(). Also note we could return a boolean indicating if the return values were filled in.

Threadsafe

Parameters

out	tick_s	Side-effect return reference for the start time.
out	note⊷	Side-effect return reference for the high note.
	_h	
out	tick_f	Side-effect return reference for the finish time.
out out	tick_f note←	Side-effect return reference for the finish time. Side-effect return reference for the low note.

13.92.5.137 get_clipboard_box()

Note the common-code betweem this function and get_selected_box(). Also note we could return a boolean indicating if the return values were filled in.

Threadsafe

Parameters

out	tick_s	Side-effect return reference for the start time.
out	note⊷	Side-effect return reference for the high note.
	_h	
out	tick_f	Side-effect return reference for the finish time.
out	note⊷	Side-effect return reference for the low note.
	_1	

13.92.5.138 adjust_timestamp()

```
midipulse seq64::sequence::adjust_timestamp (
```

```
midipulse t,
bool isnoteoff)
```

- If the timestamp plus the delta is greater that m_length, we do round robin magic.
- If the timestamp is greater than m_length, then it is wrapped around to the beginning.
- If the timestamp equals m_length, then it is set to 0, and later, trimmed.
- If the timestamp is less than 0, then it is set to the end.

Taken from similar code in move_selected_notes() and grow_selected(). Be careful using this function.

Parameters

t	Provides the timestamp to be adjusted based on m_length.
isnoteoff	Used for "expanding" the timestamp from 0 to just less than m_length, if necessary. Should be set to true only for Note Off events; it defaults to false, which means to wrap the events around the end of the sequence if necessary, and is used only in movement, not in growth.

Returns

Returns the adjusted timestamp.

```
13.92.5.139 trim_timestamp()
```

Similar to adjust_timestamp, but it doesn't have an *isnoteoff* parameter.

Parameters

t Provides the timestamp to be adjusted based on m_length.

Returns

Returns the adjusted timestamp.

13.92.5.140 clip_timestamp()

If the new (off) timestamp is less than the on-time, it is clipped to the snap value. If it is greater than the length of the sequence, then it is clipped to the sequence length. No wrap-around.

ontime	Provides the original time, which limits the amount of negative adjustment that can be done.
offtime	Provides the timestamp to be adjusted and clipped.

Returns

Returns the adjusted timestamp.

13.92.5.141 move_selected_notes()

Also currently moves any other events in the range of the selection.

Also, we've moved external calls to push_undo() into this function. The caller shouldn't have to do that.

Another thing this function does is wrap-around when movement occurs. Any events (except Note Off) that will start just after the END of the pattern will be wrapped around to the beginning of the pattern.

Fixed:

Select all notes in a short pattern that starts at time 0 and has non-note events starting at time 0 (see contrib/midi/allofarow.mid); move them with the right arrow, and move them back with the left arrow; then view in the event editor, and see that the non-Note events have not moved back, and in fact move way too far to the right, actually to near the END marker. We've fixed that in the new adjust_timestamp() function.

This function checks for any marked events in seq24, but now we make sure the event is a Note On or Note Off event before dealing with it. We now handle properly events like Program Change, Control Change, and Pitch Wheel. Remember that Aftertouch is treated like a note, as it has velocity. For non-Notes, event::get_note() returns m data[0], and we don't want to adjust that.

Note

We leave a small gap where mark_selected() locks and unlocks, then we lock again. This should only be an issue if moving notes while the sequence is playing.

delta_tick	Provides the amount of time to move the selected notes. Note that it also applies to events. Note-Off events are expanded to m_length if their timestamp would be 0. All other events will wrap around to 0.	
delta_note	Provides the amount of pitch to move the selected notes. This value is applied only to Note (On and Off) events. Also, if this value would bring a note outside the range of 0 to 127, that note is not changed and the event is not moved.	

13.92.5.142 stream_event()

The event's timestamp is adjusted, if needed. If recording:

```
If the pattern is playing, the event is added.
If the pattern is playing and quantized record is in force, the note's timestamp is altered.
If not playing, but the event is a Note On or Note Off, we add it and keep track of it.
```

If MIDI Thru is enabled, the event is put on the buss.

We are adding a feature where events are rejected if their channel doesn't match that of the sequence. This has been a complaint of some people. Could modify the add_event() and add_note() functions, but better to do it here for comprehensive event support. Also have to make sure the event-channel is preserved before this function is called, and also need to make sure that the channel is appended on both playback and in saving of the MIDI file.

We are also adding the usage, at last, of the m_rec_vol member, including the "Free" menu entry in seqedit, which sets the velocity to SEQ64_PRESERVE_VELOCITY (-1).

Todo When we feel like debugging, we will replace the global is-playing call with the parent perform's is-running call.

Threadsafe

Parameters

```
ev Provides the event to stream.
```

Returns

Returns true if the event's channel matched that of this sequence, and the channel-matching feature was set to true. Also returns true if we're not using channel-matching. A return value of true means the event should be saved.

13.92.5.143 change_event_data_range()

Changes only selected events, if any.

Threadsafe

Let t ==the current tick value; t ==tick start value; t ==tick finish value; t ==ti

If this were an interpolation formula it would be:

Something is not quite right; to be investigated.

Parameters

tick←	Provides the starting tick value.
_s	
tick⊷	Provides the ending tick value.
_f	
status	Provides the event status that is to be changed.
СС	Provides the event control value.
data⊷	Provides the starting data value.
_s	
data⊷	Provides the finishing data value.
_f	

Returns

Returns true if the data was changed.

13.92.5.144 change_event_data_lfo()

value	Provides the base value for the event data value. Ranges from 0 to 127 in increments of 0.1. This amount is added to the result of the wave_func() calculation.
range	Provides the range for the event data value. Ranges from 0 to 127 in increments of 0.1.
speed	Provides the inverse periodicity (?) for the modifications. Ranges from 0 to 16 in increments of 0.01. Not sure what units this value is in.
phase	The phase of the event modification. Ranges from 0 to 1 (what units?) in increments of 0.01.
wave	The wave type to apply. Ranges from 1 to 5.
status	The status value for the events to modify.
СС	Provides the control-change value for Control Change events that are to be modified. Generated by Doxygen

13.92.5.145 increment_selected()

```
void seq64::sequence::increment_selected (
            midibyte astat,
            midibyte )
```

The supported statuses are:

```
EVENT_NOTE_ON
```

- EVENT_NOTE_OFF
- EVENT_AFTERTOUCH
- EVENT_CONTROL_CHANGE
- EVENT_PITCH_WHEEL EVENT_PROGRAM_CHANGE
- EVENT_CHANNEL_PRESSURE

Threadsafe

Parameters

```
The desired event.
astat
```

Parameter "acontrol", the desired control-change, is unused. This might be a bug, or at least a missing feature.

13.92.5.146 decrement_selected()

```
void seq64::sequence::decrement_selected (
            midibyte astat,
            midibyte )
```

The supported statuses are:

- · One-byte messages
 - EVENT_PROGRAM_CHANGE
 - EVENT_CHANNEL_PRESSURE
- · Two-byte messages
 - EVENT_NOTE_ON
 - EVENT_NOTE_OFF
 - EVENT AFTERTOUCH
 - EVENT_CONTROL_CHANGE
 - EVENT_PITCH_WHEEL

Threadsafe

astat	The desired event.
uotut	THE aconca event.

Parameter "acontrol", the desired control-change, is unused. This might be a bug, or at least a missing feature.

13.92.5.147 grow selected()

And, though it doesn't move Note Off events, it does reconstruct them.

This function is called when doing a ctrl-left mouse move on the selected notes or when using ctrl-left-arrow or ctrl-right-arrow to shrink or stretch the selected notes. Using the mouse allows pretty much any amount of growth or shrinkage, but use the arrow keys limits the changes to the current snap value.

This function grows/shrinks only Note On events that are marked and linked. If an event is not linked, this function now ignores the event's timestamp, rather than risk a segfault on a null pointer. Compare this function to the stretch_selected() and move_selected_notes() functions.

This function would strip out non-Notes, but now it at least preserves them and moves them, to try to preserve their relative position re the notes.

In any case, we want to mark the original off-event for deletion, otherwise we get duplicate off events, for example in the "Begin/End" pattern in the test.midi file.

This function now tries to prevent pathological growth, such as trying to shrink the notes to zero length or less, or stretch them beyond the length of the sequence. Otherwise we get weird and unexpected results. Also, we've moved external calls to push_undo() into this function. The caller shouldn't have to do that.

A comment on terminology: The user "selects" notes, while the sequencer "marks" notes. The first thing this function does is mark all the selected notes.

Threadsafe

Parameters

delta An offset for each linked event's timestamp.

13.92.5.148 stretch_selected()

This should move a note off event, according to old comments, but it doesn't seem to do that. See the grow_\circ
selected() function. Rather, it moves any event in the selection.

Also, we've moved external calls to push_undo() into this function. The caller shouldn't have to do that.

Threadsafe

delta_tick	Provides the amount of time to stretch the selected notes.
------------	--

13.92.5.149 remove_marked()

```
bool seq64::sequence::remove_marked ( )
```

Note how this function forwards the call to m_event.remove_marked().

Threadsafe

Returns

Returns true if at least one event was removed.

13.92.5.150 mark_selected()

```
bool seq64::sequence::mark_selected ( )
```

Threadsafe

Returns

Returns true if there were any events that got marked.

13.92.5.151 remove_selected()

```
void seq64::sequence::remove_selected ( )
```

This is a new convenience function to fold in the push_undo() and mark_selected() calls. It makes the process slightly faster, as well.

Threadsafe Also makes the whole process threadsafe.

13.92.5.152 unpaint_all()

```
void seq64::sequence::unpaint_all ( )
```

Threadsafe

13.92.5.153 unselect()

```
void seq64::sequence::unselect ( )
```

```
13.92.5.154 verify_and_link()

void seq64::sequence::verify_and_link ( )

Threadsafe

13.92.5.155 link_new()

void seq64::sequence::link_new ( )

Threadsafe

13.92.5.156 zero_markers()

void seq64::sequence::zero_markers ( ) [inline]
```

This function is used when the sequencer stops. This function currently sets m_last_tick = 0, but we would like to avoid that if doing a pause, rather than a stop, of playback.

It flushes a note to the midibus to preview its sound, used by the virtual piano.

Threadsafe

Parameters

note The note to play. It is not checked for range validity, for the sake of speed.

```
13.92.5.158 play_note_off()

void seq64::sequence::play_note_off (
    int note )
```

Threadsafe

Parameters

note The note to turn off. It is not checked for range validity, for the sake of speed.

```
13.92.5.159 off_playing_notes()
void seq64::sequence::off_playing_notes ( )
```

This function does not bother checking if m_masterbus is a null pointer.

Threadsafe

In Live mode, the user controls playback, while in Song mode, JACK or the performance/song editor controls playback. This function used to be called "reset()".

Parameters

cana mada	True if song mode is on. This can mean that JACK transport is not in control of playback.	
sona moae	True il sono mode is on. Tris can mean mai JACN hansbort is noi in comioi di biavback.	

13.92.5.161 pause()

It still includes the note-shutoff capability to prevent notes from lingering. Note that we do not call set_playing(false)... it disarms the sequence, which we do not want upon pausing.

```
13.92.5.162 reset_draw_marker()
void seq64::sequence::reset_draw_marker ( )
```

It resets the draw marker so that calls to get_next_note_event() will start from the first event.

Threadsafe

```
13.92.5.163 reset_draw_trigger_marker()
void seq64::sequence::reset_draw_trigger_marker ( )
```

```
13.92.5.164 get_next_note_event()
```

When it has no more events, returns a false.

Note that, before the first call to draw a sequence, the $reset_draw_marker()$ function must be called, to reset m_\leftarrow iterator_draw.

out	tick_s	Provides a pointer destination for the start time.
out	tick_f	Provides a pointer destination for the finish time.
out	note	Provides a pointer destination for the note pitch value Probably should be a midibyte value.
out	selected	Provides a pointer destination for the selection status of the note.
out	velocity	Provides a pointer destination for the note velocity. Probably should be a midibyte value.

13.92.5.165 get_minmax_note_events()

Todo For efficency, we should calculate this only when the event set changes, and save the results and return them if good.

Threadsafe

Parameters

lowest	A reference parameter to return the note with the lowest value. if there are no notes, then it is set to SEQ64_MIDI_COUNT_MAX-1.
highest	A reference parameter to return the note with the highest value. if there are no notes, then it is set to -1.

Returns

If there are no notes in the list, then false is returned, and the results should be disregarded.

13.92.5.166 get_next_event() [1/2]

Then set the rest of the parameters parameters using that event. If the status is the new value EVENT_ANY, then any event will be obtained.

Note the usage of event::is_desired_cc_or_not_cc(status, cc, *d0); Either we have a control change with the right CC or it's a different type of event.

status	The type of event to be obtained. The special value EVENT_ANY can be provided so that no event
	statuses are filtered.
CC	The continuous controller value that might be desired.
tick	A pointer return value for the tick value of the next event found.
d0	A pointer return value for the first data value of the event.
d1	A pointer return value for the second data value of the event.
selected	A pointer return value for the is-selected status of the event.
evtype	A stazed parameter for picking either all event or unselected events.

Then set the status and control character parameters using that event.

Parameters

status	Provides a pointer to the MIDI status byte to be set, as a way to retrieve the event.
СС	The return pointer for the control value.

13.92.5.168 get_next_trigger()

13.92.5.169 quantize_events()

One confusing things is why the original versions of the events don't seem to be deleted.

status	Indicates the type of event to be quantized.
СС	The desired control-change to count, if the event is a control-change.
snap_tick	Provides the maximum amount to move the events. Actually, events are moved to the previous or next snap_tick value depend on whether they are halfway to the next one or not.
divide	A rough indicator of the amount of quantization. The only values used in the application are either 1 ("quantize") or 2 ("tighten"). The latter value reduces the amount of change slightly.
linked	False by default, this parameter indicates if marked events are to be relinked, as far as we can tell.

13.92.5.170 push_quantize()

See the sequence::quantize_events() function for more information. This function just does locking and a push-undo before calling that function.

Parameters

status	The kind of event to quantize, such as Note On, or the event type selected in the pattern editor's data pane.
СС	The control-change value to quantize, again as selected in the pattern editor's data pane. For Note Ons, this value should be set to 0.
snap_tick	The number of ticks to use for quantizing the events. Usually, this is the snap value selected in the pattern editor.
divide	Provides a division value, usually either 1 ("quantize") or 2 ("tighten").
linked	Set this value to true for tightening notes. The default value of this parameter is false.

13.92.5.171 transpose_notes()

If the scale value is 0, this is "no scale", which is the chromatic scale, where all 12 notes, including sharps and flats, are part of the scale.

Also, we've moved external calls to push_undo() into this function. The caller shouldn't have to do that.

Note

We noticed (ca 2016-06-10) that MIDI aftertouch events need to be transposed, but are not being transposed here. Assuming they are selectable (another question!), the test for note-on and note-off is not sufficient, and so has been replaced by a call to event::is_note_msg().

steps	The number of steps to transpose the notes.
scale	The scale to make the notes adhere to while transposing.

```
13.92.5.172 shift_notes()
void seq64::sequence::shift_notes (
            midipulse ticks )
13.92.5.173 multiply_pattern()
void seq64::sequence::multiply_pattern (
            double multiplier )
13.92.5.174 musical_key() [1/2]
midibyte seq64::sequence::musical_key ( ) const [inline]
13.92.5.175 musical_key() [2/2]
void seq64::sequence::musical_key (
             int key ) [inline]
13.92.5.176 musical_scale() [1/2]
midibyte seq64::sequence::musical_scale ( ) const [inline]
13.92.5.177 musical_scale() [2/2]
void seq64::sequence::musical_scale (
             int scale ) [inline]
```

Disabling the sequence number (setting it to SEQ64_SEQUENCE_LIMIT) is valid.

Compare this function to the remove_all() function. Copying the container is a lot of work, but fairly fast, even with an std::multimap as the container.

Threadsafe Note that we had to consolidate the replacement of all the events in the container in order to prevent the "Save to Sequence" button in the eventedit object from causing the application to segfault. It would segfault when the mainwand timer callback would fire, causing updates to the sequence's slot pixmap, which would then try to access deleted events. Part of the issue was that note links were dropped when copying the events, so now we call verify_and_link() to hopefully reconstitute the links.

Parameters

newevents Provides the container of MIDI events that will completely replace the current container. Normally this container is supplied by the event editor, via the eventslots class.

```
13.92.5.182 note_off_margin()
```

midipulse seq64::sequence::note_off_margin () const [inline]

13.92.5.183 event_in_range()

Makes if-clauses easier to read.

Parameters

е	Provides the event to be checked.
status	Provides the event type that must be matched.
tick←	The lower end of the range of timestamps that the event must fall within.
_s	
tick←	The upper end of the range of timestamps that the event must fall within.
_f	

Returns

Returns true if the event matchs all of the restrictions noted.

13.92.5.184 set_parent()

Remember that m_parent is not at all owned by the sequence. We just don't want to do all the work necessary to make it a reference, at this time.

Parameters

p A pointer to the parent, assigned only if not already assigned.

13.92.5.185 put_event_on_bus()

This function does not bother checking if m_masterbus is a null pointer.

Parameters

ev The event to put on the buss.

Threadsafe

```
13.92.5.186 set_trigger_offset()
```

If m_length is 0, then m_trigger_offset is simply set to the parameter.

Threadsafe

Parameters

trigger_offset	The full trigger offset to set.
----------------	---------------------------------

13.92.5.187 adjust_trigger_offsets_to_length()

Threadsafe

Might can get rid of this function?

Parameters

```
newlength The new length of the adjusted trigger.
```

13.92.5.188 adjust_offset()

event_list::iterator i) [private]

We no longer bother checking the pointer. If it is bad, all hope is lost. If the event is a note off, and that note is currently playing, then send a note off.

Not threadsafe

Provides the iterator to the event to remove from the event list.

Finds the given event in m_events, and removes the first iterator matching that. If there are events that would match after that, they remain in the container. This matches seq24 behavior.

Not threadsafe

Parameters

e Provides a reference to the event to be removed.

```
13.92.5.191 remove_all()
void seq64::sequence::remove_all ( ) [private]
```

Unsets the modified flag. (Why?) Also see the new copy_events() function.

```
13.92.5.192 channel_match()
```

Parameters

e The event whose channel nybble is to be checked.

Returns

Returns true if the channel-matching feature is enable and the channels match, or true if the channel-matching feature is turned off.

13.92.6 Friends And Related Function Documentation

13.92.6.1 perform

```
friend class perform [friend]
```

13.92.6.2 triggers

```
friend class triggers [friend]
```

13.92.7 Field Documentation

13.92.7.1 m_events_clipboard

```
event_list seq64::sequence::m_events_clipboard [static], [private]
```

Being static allows for copy/paste between patterns.

13.92.7.2 m_parent

```
perform* seq64::sequence::m_parent [private]
```

We can use the rc_settings flag(s), but JACK could be disconnected. We could use a reference here, but, to avoid modifying the midifile class as well, we use a pointer. It is set in perform::add_sequence(). This member would also be using for passing modification status to the parent, so that the GUI code doesn't have to do it.

```
13.92.7.3 m_events
```

```
event_list seq64::sequence::m_events [private]
```

It used to be called m list events, but a map implementation is now available, and is the default.

13.92.7.4 m_triggers

```
triggers seq64::sequence::m_triggers [private]
```

13.92.7.5 m_events_undo_hold

```
event_list seq64::sequence::m_events_undo_hold [private]
```

Changed, of course, from std::list<event> to the sequence::Events typedef.

```
Events m_events_undo_hold;
```

```
13.92.7.6 m_have_undo
```

```
bool seq64::sequence::m_have_undo [private]
```

13.92.7.7 m_have_redo

```
bool seq64::sequence::m_have_redo [private]
```

Previously, unlike the perfedit, the sequedit did not provide a redo facility.

13.92.7.8 m_events_undo

```
EventStack seq64::sequence::m_events_undo [private]
```

13.92.7.9 m_events_redo

```
EventStack seq64::sequence::m_events_redo [private]
```

13.92.7.10 m_iterator_draw

```
event_list::iterator seq64::sequence::m_iterator_draw [private]
```

13.92.7.11 m_channel_match

```
bool seq64::sequence::m_channel_match [private]
```

If true (not yet the default), then the seqedit window will record only MIDI events that match its channel. The old behavior is preserved if this variable is set to false.

13.92.7.12 m_midi_channel

```
midibyte seq64::sequence::m_midi_channel [private]
```

However, if this value is EVENT_NULL_CHANNEL (0xFF), then this sequence is an SMF 0 track, and has no single channel.

```
13.92.7.13 m_bus
midibyte seq64::sequence::m_bus [private]
13.92.7.14 m_song_mute
bool seq64::sequence::m_song_mute [private]
13.92.7.15 m_transposable
bool seq64::sequence::m_transposable [private]
A potential feature from stazed's seq32 project. Now it is an actual, configurable feature.
13.92.7.16 m_notes_on
int seq64::sequence::m_notes_on [private]
13.92.7.17 m_masterbus
mastermidibus* seq64::sequence::m_masterbus [private]
13.92.7.18 m_playing_notes
int seq64::sequence::m_playing_notes[SEQ64_MIDI_NOTES_MAX] [private]
It is used when muting, to shut off the notes that are playing.
13.92.7.19 m_was_playing
bool seq64::sequence::m_was_playing [private]
13.92.7.20 m_playing
bool seq64::sequence::m_playing [private]
```

```
13.92.7.21 m_recording
bool seq64::sequence::m_recording [private]
13.92.7.22 m_quantized_rec
bool seq64::sequence::m_quantized_rec [private]
13.92.7.23 m_thru
bool seq64::sequence::m_thru [private]
13.92.7.24 m_queued
bool seq64::sequence::m_queued [private]
13.92.7.25 m_dirty_main
bool seq64::sequence::m_dirty_main [private]
Provides the main dirtiness flag.
13.92.7.26 m_dirty_edit
bool seq64::sequence::m_dirty_edit [private]
13.92.7.27 m_dirty_perf
bool seq64::sequence::m_dirty_perf [private]
13.92.7.28 m_dirty_names
```

bool seq64::sequence::m_dirty_names [private]

```
13.92.7.29 m_editing
bool seq64::sequence::m_editing [private]
13.92.7.30 m_raise
bool seq64::sequence::m_raise [private]
It allows a sequence editor window to pop up if not already raised, in seqedit::timeout().
13.92.7.31 m_name
std::string seq64::sequence::m_name [private]
13.92.7.32 m_last_tick
midipulse seq64::sequence::m_last_tick [private]
Provides the last tick played.
13.92.7.33 m_queued_tick
midipulse seq64::sequence::m_queued_tick [private]
13.92.7.34 m_trigger_offset
midipulse seq64::sequence::m_trigger_offset [private]
13.92.7.35 m_maxbeats
const int seq64::sequence::m_maxbeats [private]
Hardwired to c_maxbeats at present.
13.92.7.36 m_ppqn
int seq64::sequence::m_ppqn [private]
```

```
13.92.7.37 m_seq_number
```

```
int seq64::sequence::m_seq_number [private]
```

This number is set in the perform::install_sequence() function.

```
13.92.7.38 m_length
```

```
midipulse seq64::sequence::m_length [private]
```

This value should be a power of two when used as a bar unit.

```
13.92.7.39 m_snap_tick
```

```
midipulse seq64::sequence::m_snap_tick [private]
```

It starts out as the value m_ppqn / 4.

13.92.7.40 m_time_beats_per_measure

```
int seq64::sequence::m_time_beats_per_measure [private]
```

Defaults to 4. Used by the sequence editor to mark things in correct time on the user-interface.

```
13.92.7.41 m_time_beat_width
```

```
int seq64::sequence::m_time_beat_width [private]
```

Defaults to 4, which means the beat is a quarter note. A value of 8 would mean it is an eighth note. Used by the sequence editor to mark things in correct time on the user-interface.

```
13.92.7.42 m_clocks_per_metronome
```

```
int seq64::sequence::m_clocks_per_metronome [private]
```

This value provides the number of MIDI clocks between metronome clicks. The default value of this item is 24. It can also be read from some SMF 1 files, such as our hymne.mid example.

```
13.92.7.43 m_32nds_per_quarter
```

```
int seq64::sequence::m_32nds_per_quarter [private]
```

This value provides the number of notated 32nd notes in a MIDI quarter note (24 MIDI clocks). The usual (and default) value of this parameter is 8; some sequencers allow this to be changed.

```
13.92.7.44 m_us_per_quarter_note
```

```
long seq64::sequence::m_us_per_quarter_note [private]
```

This value can be extracted from the beats-per-minute value (mastermidibus::m_beats_per_minute), but here we set it to 0 by default, indicating that we don't want to write it. Otherwise, it can be read from a MIDI file, and saved here to be restored later.

```
13.92.7.45 m_rec_vol
```

```
int seq64::sequence::m_rec_vol [private]
```

```
13.92.7.46 m_note_on_velocity
```

```
int seq64::sequence::m_note_on_velocity [private]
```

Currently set to SEQ64_DEFAULT_NOTE_ON_VELOCITY. If the recording velocity (m_rec_vol) is non-zero, this value will be set to the desired recording velocity. A "stazed" feature.

```
13.92.7.47 m_note_off_velocity
```

```
int seq64::sequence::m_note_off_velocity [private]
```

Currently set to SEQ64_DEFAULT_NOTE_OFF_VELOCITY, and currently unmodifiable. A "stazed" feature.

```
13.92.7.48 m_musical_key
```

```
midibyte seq64::sequence::m_musical_key [private]
```

If the value is SEQ64_KEY_OF_C, then there is no musical key to be set.

```
13.92.7.49 m_musical_scale
```

```
midibyte seq64::sequence::m_musical_scale [private]
```

If the value is the enumeration value c_scale_off, then there is no musical scale to be set.

```
13.92.7.50 m_background_sequence
```

```
int seq64::sequence::m_background_sequence [private]
```

If the value is greater than max_sequence(), then there is no background sequence to be set.

```
13.92.7.51 m_mutex
```

```
mutex seq64::sequence::m_mutex [mutable], [private]
```

Made mutable for use in certain locked getter functions.

13.92.7.52 m_note_off_margin

```
const midipulse seq64::sequence::m_note_off_margin [private]
```

Also used when the user attempts to shrink a note to zero (or less than zero) length.

13.93 seq64::trigger Class Reference

This class hold a single trigger for a sequence object.

Public Member Functions

• trigger ()

Initializes the trigger structure.

bool operator< (const trigger &rhs)

This operator compares only the m_tick_start members.

• midipulse length () const

'Getter' function for member m_tick_end and m_tick_start.

midipulse tick_start () const

'Getter' function for member m_tick_start

void tick start (midipulse s)

 ${\it 'Setter' function for member m_tick_start}$

• void increment_tick_start (midipulse s)

'Setter' function for member m_tick_start

void decrement_tick_start (midipulse s)

'Setter' function for member m_tick_start

• midipulse tick_end () const

'Getter' function for member m_tick_end

• void tick_end (midipulse e)

'Setter' function for member m_tick_end

• void increment_tick_end (midipulse s)

'Setter' function for member m_tick_end

void decrement_tick_end (midipulse s)

'Setter' function for member m_tick_end

• midipulse offset () const

'Getter' function for member m_offset

void offset (midipulse o)

'Setter' function for member m_offset

· void increment_offset (midipulse s)

'Setter' function for member m offset

void decrement_offset (midipulse s)

'Setter' function for member m_offset

• bool selected () const

'Getter' function for member m selected

void selected (bool s)

'Setter' function for member m_selected

Private Attributes

midipulse m_tick_start

Provides the starting tick for this trigger.

• midipulse m_tick_end

Provides the ending tick for this trigger.

• midipulse m_offset

Provides the offset for this trigger.

• bool m_selected

Indicates that the trigger is part of a selection.

13.93.1 Detailed Description

This class is used in playback, and is contained in the triggers class.

13.93.2 Constructor & Destructor Documentation

```
13.93.2.1 trigger()
seq64::trigger::trigger ( ) [inline]
```

13.93.3 Member Function Documentation

```
13.93.3.1 operator<()
```

Parameters

```
rhs The "right-hand side" of the less-than operation.
```

Returns

Returns true if m_tick_start is less than rhs's.

13.93.3.2 length()

```
midipulse seq64::trigger::length ( ) const [inline]
```

We've seen that some of the calculations of trigger length are wrong, being 1 tick less than the true length of the trigger in pulses. This function calculates trigger length the correct way.

```
13.93.3.3 tick_start() [1/2]
midipulse seq64::trigger::tick_start ( ) const [inline]
13.93.3.4 tick_start() [2/2]
void seq64::trigger::tick_start (
             midipulse s ) [inline]
13.93.3.5 increment_tick_start()
void seq64::trigger::increment_tick_start (
             midipulse s ) [inline]
13.93.3.6 decrement_tick_start()
void seq64::trigger::decrement_tick_start (
             midipulse s ) [inline]
13.93.3.7 tick_end() [1/2]
midipulse seq64::trigger::tick_end ( ) const [inline]
13.93.3.8 tick_end() [2/2]
void seq64::trigger::tick_end (
             midipulse e ) [inline]
```

```
13.93.3.9 increment_tick_end()
void seq64::trigger::increment_tick_end (
            midipulse s ) [inline]
13.93.3.10 decrement_tick_end()
void seq64::trigger::decrement_tick_end (
             midipulse s ) [inline]
13.93.3.11 offset() [1/2]
midipulse seq64::trigger::offset ( ) const [inline]
13.93.3.12 offset() [2/2]
void seq64::trigger::offset (
             midipulse o ) [inline]
13.93.3.13 increment_offset()
void seq64::trigger::increment_offset (
            midipulse s ) [inline]
13.93.3.14 decrement_offset()
void seq64::trigger::decrement_offset (
             midipulse s ) [inline]
13.93.3.15 selected() [1/2]
```

bool seq64::trigger::selected () const [inline]

```
13.93.3.16 selected() [2/2]
void seq64::trigger::selected (
             bool s ) [inline]
13.93.4 Field Documentation
13.93.4.1 m_tick_start
midipulse seq64::trigger::m_tick_start [private]
13.93.4.2 m_tick_end
midipulse seq64::trigger::m_tick_end [private]
13.93.4.3 m_offset
midipulse seq64::trigger::m_offset [private]
13.93.4.4 m_selected
bool seq64::trigger::m_selected [private]
```

13.94 seq64::triggers Class Reference

The triggers class is a receptable the triggers that can be used with a sequence object.

Public Member Functions

· triggers (sequence &parent)

Principal constructor.

∼triggers ()

A rote destructor.

• triggers & operator= (const triggers &rhs)

Principal assignment operator.

void set ppqn (int ppqn)

'Setter' function for member m_ppqn We have to set this value after construction for best safety.

void set_length (int len)

'Setter' function for member m_length We have to set this value after construction for best safety.

· const List & triggerlist () const

'Getter' function for member m_triggers This is the const version

• List & triggerlist ()

'Getter' function for member m_triggers

• void push undo ()

Pushes the list-trigger into the trigger undo-list, then flags each item in the undo-list as unselected.

• void pop_undo ()

If the trigger undo-list has any items, the list-trigger is pushed into the redo list, the top of the undo-list is coped into the list-trigger, and then pops from the undo-list.

void pop_redo ()

If the trigger redo-list has any items, the list-trigger is pushed into the undo list, the top of the redo-list is coped into the list-trigger, and then pops from the redo-list.

· void print (const std::string &seqname) const

Prints a list of the currently-held triggers.

bool play (midipulse &starttick, midipulse &endtick)

If playback-mode (song mode) is in force, that is, if using in-triggers and on/off triggers, this function handles that kind of playback.

• void add (midipulse tick, midipulse len, midipulse offset=0, bool adjustoffset=true)

Adds a trigger.

void adjust_offsets_to_length (midipulse newlen)

Adjusts trigger offsets to the length specified for all triggers, and undo triggers.

• void split (midipulse tick)

Splits the first trigger that brackets the splittick parameter.

void grow (midipulse tickfrom, midipulse tickto, midipulse length)

Grows a trigger.

· void remove (midipulse tick)

Deletes the first trigger that brackets the given tick from the trigger-list.

• bool get_state (midipulse tick)

Checks the list of triggers against the given tick.

bool select (midipulse tick)

Checks the list of triggers against the given tick.

• bool unselect ()

Unselects all triggers.

· bool intersect (midipulse position, midipulse &start, midipulse &end)

This function examines each trigger in the trigger list.

void remove_selected ()

Deletes the first selected trigger that is found.

void copy selected ()

Copies the first selected trigger that is found.

• void paste (midipulse paste_tick=SEQ64_NO_PASTE_TRIGGER)

If there is a copied trigger, then this function grabs it from the trigger clipboard and adds it.

• bool move_selected (midipulse tick, bool adjustoffset, grow_edit_t which=GROW_MOVE)

Moves selected triggers as per the given parameters.

• midipulse get_selected_start ()

Gets the selected trigger's start tick.

midipulse get_selected_end ()

Gets the selected trigger's end tick.

• midipulse get_maximum ()

Get the ending value of the last trigger in the trigger-list.

· void move (midipulse starttick, midipulse distance, bool direction)

Moves triggers in the trigger-list.

· void copy (midipulse starttick, midipulse distance)

Not sure what these diagrams are for yet.

• void clear ()

Clears the whole list of triggers.

bool next (midipulse *tick_on, midipulse *tick_off, bool *selected, midipulse *tick_offset)

Get the next trigger in the trigger list, and set the parameters based on that trigger.

trigger next trigger ()

Get the next trigger in the trigger list.

void reset_draw_trigger_marker ()

Sets the draw-trigger iterator to the beginning of the trigger list.

- · void set trigger paste tick (midipulse tick)
- midipulse get_trigger_paste_tick () const

Private Types

```
    enum grow_edit_t {
        GROW_START,
        GROW_END,
        GROW_MOVE }
```

Provides a typedef introduced by Stazed to make the trigger grow/move code easier to understand.

typedef std::list< trigger > List

Exposes the triggers type, currently needed for midi_container only.

typedef std::stack< List > Stack

Provides a stack for use with the undo/redo features of the trigger support.

Private Member Functions

· midipulse adjust_offset (midipulse offset)

Adjusts the given offset by mod'ing it with m_length and adding m_length if needed, and returning the result.

• void split (trigger &trig, midipulse splittick)

Splits the trigger given by the parameter into two triggers.

Private Attributes

• sequence & m_parent

Holds a reference to the parent sequence object that owns this trigger object.

List m triggers

This list holds the current pattern/triggers events.

• trigger m_clipboard

This item holds a single copied trigger, to be pasted later.

Stack m_undo_stack

Handles the undo list for a series of operations on triggers.

Stack m_redo_stack

Handles the redo list for a series of operations on triggers.

• List::iterator m_iterator_play_trigger

An iterator for cycling through the triggers during playback.

· List::iterator m_iterator_draw_trigger

An iterator for cycling through the triggers during drawing.

bool m_trigger_copied

Set to true if there is an active trigger in the trigger clipboard.

• midipulse m_paste_tick

The tick point for pasting.

• int m_ppqn

Holds the value of the PPQN from the parent sequence, for easy access.

• int m_length

Holds the value of the length from the parent sequence, for easy access.

Friends

- · class midi_container
- · class midifile
- · class sequence
- class Seq24PerfInput
- class FruityPerfInput

13.94.1 Member Typedef Documentation

```
13.94.1.1 List

typedef std::list<trigger> seq64::triggers::List [private]

13.94.1.2 Stack

typedef std::stack<List> seq64::triggers::Stack [private]
```

13.94.2.1 grow_edit_t

```
enum seq64::triggers::grow_edit_t [private]
```

13.94.2 Member Enumeration Documentation

Enumerator

GROW_START	Grow the start of the trigger.
GROW_END	Grow the end of the trigger.
GROW_MOVE	Move the entire trigger block.

13.94.3 Constructor & Destructor Documentation

13.94.3.1 triggers()

Parameters

parent The triggers object often needs to tell its parent sequence object what to do (such as stop playing).

13.94.3.2 ∼triggers()

```
seq64::triggers::~triggers ( )
```

13.94.4 Member Function Documentation

13.94.4.1 operator=()

Follows the stock rules for such an operator, but does a little more then just assign member values.

FIXED, BEWARE: Currently, it does not assign them all, so we should create a partial_copy() function to do this work, and use it where it is needed.

Parameters

rhs	Provides the "right-hand side" of the assignment operation.
-----	---

Returns

Returns a reference to self, for use in concatenated assignment operations.

```
13.94.4.2 set_ppqn()
void seq64::triggers::set_ppqn (
             int ppqn ) [inline]
13.94.4.3 set_length()
void seq64::triggers::set_length (
              int len ) [inline]
Also, there a chance that the length of the parent might change from time to time. Currently, only the sequence
constructor and midifile call this function.
13.94.4.4 triggerlist() [1/2]
const List& seq64::triggers::triggerlist ( ) const [inline]
13.94.4.5 triggerlist() [2/2]
List& seq64::triggers::triggerlist ( ) [inline]
13.94.4.6 push_undo()
void seq64::triggers::push_undo ( )
13.94.4.7 pop_undo()
void seq64::triggers::pop_undo ( )
13.94.4.8 pop_redo()
void seq64::triggers::pop_redo ( )
```

void seq64::triggers::print (

const std::string & seqname) const

13.94.4.9 print()

segname A tag name to accompany the print-out, for the human to re
--

13.94.4.10 play()

This is a new function for sequence::play() to call.

The for-loop goes through all the triggers, determining if there is are trigger start/end values before the *end_tick*. If so, then the trigger state is set to true (start only within the tick range) or false (end is within the tick range), and the trigger tick is set to start or end. The first start or end trigger that is past the end tick cause the search to end.

If the trigger state has changed, then the start/end ticks are passed back to the sequence, and the trigger offset is adjusted.

Parameters

start_tick	Provides the starting tick value, and returns the modified value as a side-effect.
end_tick	Provides the ending tick value, and returns the modified value as a side-effect.

Returns

Returns true if we're through playing the frame (trigger turning off), and the caller should stop the playback.

13.94.4.11 add()

```
void seq64::triggers::add (
    midipulse tick,
    midipulse len,
    midipulse offset = 0,
    bool fixoffset = true )
```

What is this?

tick	Provides the tick (pulse) time at which the trigger goes on.
len	Provides the length of the trigger. This value is actually calculated from the "on" value minus the "off" value read from the MIDI file.
offset	This value specifies the offset of the trigger. It is a feature of the c_triggers_new that c_triggers doesn't have. It is the third value in the trigger specification of the Sequencer64 MIDI file.
fixoffset	If true, the offset parameter is modified by adjust_offset() first. We think that basically makes sure it is positive.

13.94.4.12 adjust_offsets_to_length()

Parameters

newlength Pro	vides the length to which to adjust the offsets.
---------------	--

COMMON CODE?

This is the first trigger where splittick is greater than L and less than R.

Parameters

splittick	Provides the tick that must be bracketed for the split to be made.
-----------	--

13.94.4.14 grow()

This function looks for the first trigger where the tickfrom parameter is between the trigger's tick-start and tick-end values. If found then the trigger's start is moved back to tickto, if necessary, or the trigger's end is moved to tickto plus the length parameter, if necessary.

Then this new trigger is added, and the function breaks from the search loop.

	tickfrom	The desired from-value back which to expand the trigger, if necessary.
	tickto	The desired to-value towards which to expand the trigger, if necessary.
İ	len	The additional length to append to tickto for the check.

13.94.4.15 remove()

Parameters

tick Provides the tick to be examined.

13.94.4.16 get_state()

If any trigger is found to bracket that tick, then true is returned.

Parameters

Returns

Returns true if a trigger is found that brackets the given tick.

13.94.4.17 select()

If any trigger is found to bracket that tick, then true is returned, and the trigger is marked as selected.

Parameters

tick	Provides the tick of interest.

Returns

Returns true if a trigger is found that brackets the given tick.

13.94.4.18 unselect()

```
bool seq64::triggers::unselect ( )
```

Returns

Always returns false.

13.94.4.19 intersect()

If the given position is between the current trigger's tick-start and tick-end values, the these values are copied to the start and end parameters, respectively, and then we exit.

Parameters

position	The position to examine.
start	The destination for the starting tick (m_tick_start) of the matching trigger.
ender	The destination for the ending tick (m_tick_end) of the matching trigger.

Returns

Returns true if a trigger was found whose start/end ticks contained the position. Otherwise, false is returned, and the start and end return parameters should not be used.

13.94.4.20 remove_selected()

```
void seq64::triggers::remove_selected ( )
```

13.94.4.21 copy_selected()

```
void seq64::triggers::copy_selected ( )
```

13.94.4.22 paste()

It pastes at the copy end. or at the paste-tick, if supplied.

Parameters

paste_tick	Provides the optional tick at which to paste the trigger. If not set to
	SEQ64_NO_PASTE_TRIGGER, this value is used to adjust the paste offset.

13.94.4.23 move_selected()

Parameters

tick	The tick at which the trigger starts.
fixoffset	Set to true if the offset is to be adjusted.
which	Selects which movement will be done, as discussed above. See the values of the
	trigger::grow_edit_t type.

Returns

Returns true if there was room to move. Otherwise, false is returned. We need this feature to support keystoke movement of a selected trigger in the perfroll window, and keep it from continually incrementing when there can be no more movement. This causes moving the other direction to be delayed while the accumulating movement counter is used up. However, right now we can't rely on this result, and ignore it. There may be no way around this minor issue.

```
13.94.4.24 get_selected_start()
```

```
midipulse seq64::triggers::get_selected_start ( )
```

We guess this ends up selecting only one trigger, otherwise only the last selected one would effectively set the result.

Returns

Returns the tick_start() value of the last-selected trigger. If no triggers are selected, then midipulse(-1) is returned.

13.94.4.25 get_selected_end()

```
midipulse seq64::triggers::get_selected_end ( )
```

Returns

Returns the tick_end() value of the last-selected trigger. If no triggers are selected, then midipulse(-1) is returned.

13.94.4.26 get_maximum()

```
midipulse seq64::triggers::get_maximum ( )
```

Returns

Returns the tick-end for the last trigger, if available. Otherwise, 0 is returned.

13.94.4.27 move()

There's no way to optimize this by saving tick values, as they are potentially modified at each step.

Parameters

starttick	The current location of the triggers.
distance	The distance away from the current location to which to move the triggers.
direction	If true, the triggers are moved forward. If false, the triggers are moved backward.

13.94.4.28 copy()

```
void seq64::triggers::copy (
```

```
midipulse starttick,
         midipulse distance )
[
    ][ ]
. . .
... a
. . .
     play
5
     offset
3
8 10 play
L R
[
     ] [ ] [] orig
[
              ]
     [ ] [ ][] split on the R marker, shift first [ ]
     delete middle
     move ticks
         ][ ] [ ] [] split on L
     [
               [ ] [ ] increase all after L
0123456789abcdef0123456789abcdef
][ ][ ][ ][ ][
      ][][][][][][
                    ] [ ][ ]
[ ][
      4 0 7 4 2 0
4 0 1 4 6 0
                    6 2
2 6 inverse offset
 4
0
         ] [
                    ] [
                    ] [ ][ ]
     ] [
[
                    e a
2 6 inverse offset
0
  С
      c 0 1 4 6 8
                     ] [
] [ ][ ]
k g f c a 8
0 4 c
                     inverse offset
      c ghkmn
0123456789abcdefghijklmonpq
ponmlkjihgfedcba9876543210
```

Copies triggers to a point distant from a given tick.

Parameters

starttick	The current location of the triggers.
distance	The distance away from the current location to which to copy the triggers.

Ofedcba9876543210fedcba9876543210fedcba9876543210fedcba9876543210

Todo It would be a bit simpler to simply return a trigger object, wouldn't it?

Parameters

tick_on	Return value for the retrieval of the starting tick for the trigger.
tick_off	Return value for the retrieval of the ending tick for the trigger.
selected	Return value for the retrieval of the is-selected flag for the trigger.
offset	Return value for the retrieval of the offset for the trigger.

Returns

Returns true if a trigger was found. If false, the caller cannot rely on the values returned through the return parameters.

Side-effect(s) The value of the m_iterator_draw_trigger member will be altered by this call, unless pointing to the end of the triggerlist, or if there are no triggers.

```
13.94.4.31 next_trigger()

trigger seq64::triggers::next_trigger ( )
```

Returns

Returns the next trigger. If there is none, a default trigger object is returned.

```
13.94.4.32 reset_draw_trigger_marker()
```

```
void seq64::triggers::reset_draw_trigger_marker () [inline]
```

13.94.4.33 set_trigger_paste_tick()

13.94.4.34 get_trigger_paste_tick()

```
midipulse seq64::triggers::get_trigger_paste_tick ( ) const [inline]
```

13.94.4.35 adjust_offset()

Parameters

Returns

Returns the new offset. However, if m_length is 0, no change is made, and the original offset is returned.

The original trigger ends 1 tick before the splittick parameter, and the new trigger starts at splittick and ends where the original trigger ended.

Parameters

trig	Provides the original trigger, and also holds the changes made to that trigger as it is shortened, as a
	side-effect.
splittick	The position just after where the original trigger will be truncated, and the new trigger begins.

13.94.5 Friends And Related Function Documentation

```
13.94.5.1 midi_container
friend class midi_container [friend]
13.94.5.2 midifile
friend class midifile [friend]
13.94.5.3 sequence
friend class sequence [friend]
13.94.5.4 Seq24PerfInput
friend class Seq24PerfInput [friend]
13.94.5.5 FruityPerfInput
friend class FruityPerfInput [friend]
13.94.6 Field Documentation
13.94.6.1 m_parent
sequence& seq64::triggers::m_parent [private]
13.94.6.2 m_triggers
```

List seq64::triggers::m_triggers [private]

```
13.94.6.3 m_clipboard
trigger seq64::triggers::m_clipboard [private]
13.94.6.4 m_undo_stack
Stack seq64::triggers::m_undo_stack [private]
13.94.6.5 m_redo_stack
Stack seq64::triggers::m_redo_stack [private]
13.94.6.6 m_iterator_play_trigger
List::iterator seq64::triggers::m_iterator_play_trigger [private]
13.94.6.7 m_iterator_draw_trigger
List::iterator seq64::triggers::m_iterator_draw_trigger [private]
13.94.6.8 m_trigger_copied
bool seq64::triggers::m_trigger_copied [private]
13.94.6.9 m_paste_tick
midipulse seq64::triggers::m_paste_tick [private]
Set to -1 if not in force. This is a new feature from stazed's Seq32 project.
13.94.6.10 m_ppqn
int seq64::triggers::m_ppqn [private]
```

This should not change, but we have to set it after construction, and so we provide a setter for it, set_ppqn(), called by the sequence constructor.

13.94.6.11 m_length

```
int seq64::triggers::m_length [private]
```

This might change, we're not yet sure.

13.95 seq64::user_instrument Class Reference

Provides data about the MIDI instruments, readable from the "user" configuration file.

Public Member Functions

• user_instrument (const std::string &name="")

Default constructor.

user_instrument (const user_instrument &rhs)

Copy constructor.

• user_instrument & operator= (const user_instrument &rhs)

Principal assignment operator.

bool is_valid () const

'Getter' function for member m_is_valid

void set_defaults ()

Sets the default values.

• const std::string & name () const

'Getter' function for member m_instrument_def.instrument (name of instrument)

int controller_count () const

'Getter' function for member m_controller_count This function returns the number of active controllers.

• int controller_max () const

'Getter' function for member MIDI_CONTROLLER_MAX This function returns the maximum number of controllers, active or inactive.

const std::string & controller_name (int c) const

'Getter' function for member m_instrument_def.controllers[c]

bool controller_active (int c) const

'Getter' function for member m_instrument_def.controllers_active[c]

void set_controller (int c, const std::string &cname, bool isactive)

'Setter' function for member m_instrument_def.controllers[c] and .controllers_active[c] Only sets the controller values if the object is already valid.

Private Member Functions

void set_name (const std::string &instname)

'Setter' function for member m_instrument_def.instrument If the name parameter is not empty, the validity flag is set to true, otherwise it is set to false.

void copy_definitions (const user_instrument &rhs)

Copies the array members from one instance of user_instrument to this one.

Private Attributes

• bool m_is_valid

Provides a validity flag, useful in returning a reference to a bogus object for internal error-check.

• int m_controller_count

Provides the actual number of non-default controllers actually set.

• user_instrument_t m_instrument_def

The instance of the structure that this class wraps.

13.95.1 Detailed Description

Will later make the size adjustable, if it makes sense to do so.

13.95.2 Constructor & Destructor Documentation

Fills in the defaults for the instrument definition, sets its name, and provides some light validation.

Parameters

name The name of the instrument, valid only if it is not empty.

Parameters

rhs The sources of the data for the copy.

13.95.3 Member Function Documentation

```
13.95.3.1 operator=()
```

```
rhs The sources of the data for the assignment.
```

Returns

Returns a reference to this object.

```
13.95.3.2 is_valid()
bool seq64::user_instrument::is_valid ( ) const [inline]

13.95.3.3 set_defaults()

void seq64::user_instrument::set_defaults ( )
Also invalidates the object.
```

13.95.3.4 name()

```
const std::string& seq64::user_instrument::name ( ) const [inline]
```

13.95.3.5 controller_count()

```
int seq64::user_instrument::controller_count ( ) const [inline]
```

13.95.3.6 controller_max()

```
int seq64::user_instrument::controller_max ( ) const [inline]
```

Remember that the controller numbers for each MIDI instrument range from 0 to 127 (MIDI_CONTROLLER_MAX-1).

13.95.3.7 controller_name()

```
const std::string & seq64::user_instrument::controller_name (  \qquad \qquad \text{int } c \text{ ) const}
```

c The index of the desired controller.

Returns

The name of the desired controller has is returned. If the index c is out of range, or the object is not valid, then a reference to an internal, empty string is returned.

13.95.3.8 controller_active()

```
bool seq64::user_instrument::controller_active (  \quad \text{int } c \text{ ) const}
```

Parameters

c The index of the desired controller.

Returns

The status of the desired controller has is returned. If the index c is out of range, or the object is not valid, then false is returned.

13.95.3.9 set_controller()

Parameters

	С	The index of the desired controller.
	cname	The name of the controller to be set as the controller name.
isactive A flag that indicates if the		A flag that indicates if the desired controller is active.

13.95.3.10 set_name()

Too tricky?

of the instrument, valid only if it is not empty.	instname
---	----------

13.95.3.11 copy_definitions()

```
void seq64::user_instrument::copy_definitions (
    const user_instrument & rhs ) [private]
```

Does not include the validity flag.

Parameters

rhs The sources of the data for the partial copy.

13.95.4 Field Documentation

13.95.4.1 m_is_valid

```
bool seq64::user_instrument::m_is_valid [private]
```

Callers should check this flag via the is_valid() accessor before using this object. This flag is set to true when any valid member assignment occurs via a public setter call. However, setting an empty name for the instrument member will render the object invalid.

```
13.95.4.2 m_controller_count
```

```
int seq64::user_instrument::m_controller_count [private]
```

Often, the "user" configuration file has only a few out of the 128 assigned explicitly.

13.95.4.3 m_instrument_def

```
user_instrument_t seq64::user_instrument::m_instrument_def [private]
```

13.96 seq64::user_instrument_t Struct Reference

This structure corresponds to [user-instrument-N] definitions in the \sim /.seq24usr or \sim /.config/sequencer64/susr file.

Data Fields

• std::string instrument

Provides the name of the "instrument" being supported.

std::string controllers [SEQ64_MIDI_CONTROLLER_MAX]

Provides a list of up to 128 controllers (e.g.

bool controllers_active [SEQ64_MIDI_CONTROLLER_MAX]

Provides a flag that indicates if each of up to 128 controller is active and supported.

13.96.1 Field Documentation

13.96.1.1 instrument

```
std::string seq64::user_instrument_t::instrument
```

Do not confuse "instrument" with "program" here. An "instrument" is most likely a hardware MIDI sound-box (though it could be a software synthesizer as well.

13.96.1.2 controllers

```
std::string seq64::user_instrument_t::controllers[SEQ64_MIDI_CONTROLLER_MAX]
```

"Modulation"). If a controller isn't present, or if General MIDI is in force, this name might be empty.

13.96.1.3 controllers_active

```
bool seq64::user_instrument_t::controllers_active[SEQ64_MIDI_CONTROLLER_MAX]
```

If false, it might be an unsupported controller or a General MIDI device.

13.97 seq64::user_midi_bus Class Reference

Provides data about the MIDI busses, readable from the "user" configuration file.

Public Member Functions

• user_midi_bus (const std::string &name="")

Default constructor.

user_midi_bus (const user_midi_bus &rhs)

Copy constructor.

user_midi_bus & operator= (const user_midi_bus &rhs)

Principal assignment operator.

• bool is_valid () const

'Getter' function for member m_is_valid

· void set_defaults ()

Sets the default values.

· const std::string & name () const

'Getter' function for member m_midi_bus_def.alias (name of alias)

int channel_count () const

'Getter' function for member m_channel_count

• int channel max () const

'Getter' function for member SEQ64_MIDI_BUS_CHANNEL_MAX

· int instrument (int channel) const

'Getter' function for member m_midi_bus_def.instrument[channel]

void set_instrument (int channel, int instrum)

'Getter' function for member m_midi_bus_def.instrument[channel]

Private Member Functions

• void set_name (const std::string &name)

'Setter' function for member m_midi_bus_def.alias (name of alias) Also sets the validity flag according to the emptiness of the name parameter.

void copy_definitions (const user_midi_bus &rhs)

Copies the member fields from one instance of user_midi_bus to this one.

Private Attributes

· bool m_is_valid

Provides a validity flag, useful in returning a reference to a bogus object for internal error-check.

• int m_channel_count

Provides the actual number of non-default buss channels actually set.

· user_midi_bus_t m_midi_bus_def

The instance of the structure that this class wraps.

13.97.1 Detailed Description

Will later make the size adjustable, if it makes sense to do so.

13.97.2 Constructor & Destructor Documentation

name The name of the buss, valid only if it is not empty.

Parameters

rhs The sources of the data for the copy.

13.97.3 Member Function Documentation

13.97.3.1 operator=()

Parameters

rhs The sources of the data for the assignment.

Returns

Returns a reference to this object.

13.97.3.2 is_valid()

bool seq64::user_midi_bus::is_valid () const [inline]

13.97.3.3 set_defaults()

```
void seq64::user_midi_bus::set_defaults ( )
```

Also invalidates the object. All 16 of the channels are set to SEQ64_GM_INSTRUMENT_FLAG (-1).

13.97.3.4 name()

```
const std::string& seq64::user_midi_bus::name ( ) const [inline]
```

13.97.3.5 channel_count()

```
int seq64::user_midi_bus::channel_count ( ) const [inline]
```

Returns

This function returns the number of channels. Basically this value is always the same as that returned by channel_max(), but this pair of functions is consistent with the count functions in the user_instrument class.

13.97.3.6 channel_max()

```
int seq64::user_midi_bus::channel_max ( ) const [inline]
```

Returns

Returns the maximum number of MIDI buss channels. Remember that the instrument channels for each MIDI buss range from 0 to 15 (MIDI_BUS_CHANNEL_MAX-1).

13.97.3.7 instrument()

Parameters

channel	Provides the desired buss channel number.
---------	---

Returns

The instrument number of the desired buss channel is returned. If the channel number is out of range, or the object is not valid, then SEQ64_GM_INSTRUMENT_FLAG (-1) is returned.

13.97.3.8 set_instrument()

Does not alter the validity flag, just checks it.

Parameters

channel	Provides the desired buss channel number.	
instrum	Provides the instrument number to set that channel to.	

13.97.3.9 set_name()

13.97.3.10 copy_definitions()

Does not include the validity flag.

13.97.4 Field Documentation

13.97.4.1 m_is_valid

```
bool seq64::user_midi_bus::m_is_valid [private]
```

Callers should check this flag via the is_valid() accessor before using this object. This flag is set to true when any valid member assignment occurs via a public setter call.

13.97.4.2 m_channel_count

```
int seq64::user_midi_bus::m_channel_count [private]
```

Often, the "user" configuration file has only a few out of the 16 assigned explicitly.

13.97.4.3 m_midi_bus_def

```
user_midi_bus_t seq64::user_midi_bus::m_midi_bus_def [private]
```

13.98 seq64::user_midi_bus_t Struct Reference

This structure corresponds to [user-midi-bus-0] definitions in the \sim /.seq24usr ("user") file (\sim /.config/sequencer64/sequencer64.usr in the latest version of the application).

Data Fields

• std::string alias

Provides the user's desired name for the MIDI bus.

int instrument [SEQ64_MIDI_BUS_CHANNEL_MAX]

Provides an implicit list of MIDI channels from 0 to 15 (1 to 16) and the "instrument" number assigned to each channel.

13.98.1 Field Documentation

13.98.1.1 alias

```
std::string seq64::user_midi_bus_t::alias
```

For example, "2x2 A" for some kind of MIDI card or USB MIDI cable. If manual-alsa-ports is enabled, this could be something like "[0] seq24 0", and that is what should be shown in that case.

13.98.1.2 instrument

```
int seq64::user_midi_bus_t::instrument[SEQ64_MIDI_BUS_CHANNEL_MAX]
```

Note that the "instrument" is not a MIDI program number. Instead, it is the number associated with a "user-instrument" section in the "user" configuration file.

13.99 seq64::user_settings Class Reference

Holds the current values of sequence settings and settings that can modify the number of sequences and the configuration of the user-interface.

Public Member Functions

user_settings ()

Default constructor.

· user settings (const user settings &rhs)

Copy constructor.

• user_settings & operator= (const user_settings &rhs)

Principal assignment operator.

· void set defaults ()

Sets the default values.

• void normalize ()

Calculate the derived values from the already-set values.

bool add_bus (const std::string &alias)

Adds a user buss to the container, but only does so if the name parameter is not empty.

bool add instrument (const std::string &instname)

Adds a user instrument to the container, but only does so if the name parameter is not empty.

const user midi bus & bus (int index)

'Getter' function for member Unlike the non-const version this function is public.

const user_instrument & instrument (int index)

'Getter' function for member Unlike the non-const version this function is public.

int bus_count () const

'Getter' function for member m_midi_buses.size()

void set bus instrument (int index, int channel, int instrum)

'Getter' function for member m_midi_buses[index].instrument[channel] Currently this function is used, in the userfile ::parse() function.

• int bus_instrument (int buss, int channel)

'Getter' function for member m_midi_buses[buss].instrument[channel]

• const std::string & bus_name (int buss)

'Getter' function for member m_midi_buses[buss].name

· int instrument count () const

 ${\it 'Getter' function for member m_instruments.size()}$

· void set instrument controllers (int index, int cc, const std::string &ccname, bool isactive)

'Setter' function for member m_midi_instrument_defs[index].controllers, controllers_active

const std::string & instrument_name (int instrum)

'Getter' function for member m_instruments[instrument].instrument (name of instrument)

• const std::string & instrument_name (int buss, int channel)

Gets the correct instrument number from the buss and channel, and then looks up the name of the instrument.

· bool instrument controller active (int instrum, int cc)

'Getter' function for member m_instruments[instrument].controllers_active[controller]

bool controller_active (int buss, int channel, int cc)

A convenience function so that the caller doesn't have to get the instrument number from the bus_instrument() member function.

const std::string & instrument_controller_name (int instrum, int cc)

'Getter' function for member m_instruments[instrument].controllers_active[controller]

• const std::string & controller name (int buss, int channel, int cc)

'Getter' function for member m_instruments[instrument].controllers_active[controller] A convenience function so that the caller doesn't have to get the instrument number from the bus_instrument() member function.

int grid_style () const

'Getter' function for member m_grid_style Checks for normal style.

· bool grid is normal () const

'Getter' function for member m_grid_style Checks for normal style.

• bool grid_is_white () const

'Getter' function for member m_grid_style Checks for the white style.

bool grid_is_black () const

'Getter' function for member m_grid_style Checks for the black style.

• int grid brackets () const

'Getter' function for member m_grid_brackets

• int mainwnd_rows () const

'Getter' function for member m_mainwnd_rows

int mainwnd_cols () const

'Getter' function for member m_mainwnd_cols

• int seqs_in_set () const

'Getter' function for member m_seqs_in_set, dependent member

int gmute_tracks () const

'Getter' function for member m_gmute_tracks, dependent member

int max_sets () const

'Getter' function for member m max sets

• int max_sequence () const

'Getter' function for member m_max_sequence, dependent member

int text_x () const

'Getter' function for member m_text_x, not user modifiable, not saved

int text_y () const

'Getter' function for member m_text_y, not user modifiable, not saved

• int seqchars x () const

'Getter' function for member m_seqchars_x, not user modifiable, not saved

int seqchars_y () const

'Getter' function for member m_seqchars_y, not user modifiable, not saved

• int seqarea_x () const

'Getter' function for member m_seqarea_x, not user modifiable, not saved

int seqarea_y () const

'Getter' function for member m_seqarea_y, not user modifiable, not saved

• int seqarea_seq_x () const

 ${\it 'Getter' function for member m_seqarea_seq_x, not user modifiable, not saved}$

• int seqarea_seq_y () const

'Getter' function for member m_seqarea_seq_y, not user modifiable, not saved

int mainwid_border () const

'Getter' function for member m_mainwid_border

• int mainwid_spacing () const

'Getter' function for member m_mainwid_spacing

int mainwid_x () const

'Getter' function for member m_mainwid_x, dependent member

int mainwid_y () const

'Getter' function for member m_mainwid_y, dependent member

• int control_height () const

'Getter' function for member m_control_height

• int zoom () const

'Getter' function for member m_current_zoom

• void zoom (int value)

'Setter' function for member m_current_zoom This value is not modified unless the value parameter is between 1 and 512, inclusive.

· bool global seq feature () const

'Getter' function for member m_global_seq_feature_save

void global_seq_feature (bool flag)

'Setter' function for member m_global_seq_feature_save

• int seqedit_scale () const

'Getter' function for member m segedit scale

· void segedit scale (int scale)

'Setter' function for member m_seqedit_scale

• int seqedit_key () const

'Getter' function for member m_seqedit_key

void seqedit_key (int key)

'Setter' function for member m_seqedit_key

• int seqedit_bgsequence () const

'Getter' function for member m_seqedit_bgsequence

• void seqedit_bgsequence (int seqnum)

'Setter' function for member m_seqedit_bgsequence Note that SEQ64_IS_LEGAL_SEQUENCE() allows the SE← Q64_SEQUENCE LIMIT (0x800 = 2048) value, to turn off the use of a background sequence.

bool use_new_font () const

'Getter' function for member m_use_new_font

bool allow_two_perfedits () const

'Getter' function for member m_allow_two_perfedits

• int perf_h_page_increment () const

'Getter' function for member m_h_perf_page_increment

int perf_v_page_increment () const

'Getter' function for member m_v_perf_page_increment

• int progress_bar_colored () const

'Getter' function for member m_progress_bar_colored

· bool progress_bar_thick () const

'Getter' function for member m_progress_bar_thick

· bool inverse colors () const

Accessor m_inverse_colors

int window_redraw_rate () const

'Getter' function for member m window redraw rate ms

· bool use_more_icons () const

'Getter' function for member m_use_more_icons

bool save_user_config () const

'Getter' function for member m_save_user_config

void save_user_config (bool flag)

'Setter' function for member m_save_user_config

• int midi_ppqn () const

'Getter' function for member m_midi_ppqn

int midi_beats_per_bar () const

'Getter' function for member m_midi_beats_per_measure

· midibpm midi beats per minute () const

'Getter' function for member m_midi_beats_per_minute

int midi_beat_width () const

'Getter' function for member m_midi_beat_width

· char midi buss override () const

'Getter' function for member m_midi_buss_override

int velocity_override () const

'Getter' function for member m_velocity_override

· int bpm precision () const

'Getter' function for member m_bpm_precision

midibpm bpm_step_increment () const

'Getter' function for member m_bpm_step_increment

midibpm bpm_page_increment () const

'Getter' function for member m_bpm_page_increment

• int min zoom () const

'Getter' function for member mc_min_zoom

• int max_zoom () const

'Getter' function for member mc_max_zoom

• int baseline ppqn () const

'Getter' function for member mc_baseline_ppqn

· void use_new_font (bool flag)

'Setter' function for member m_use_new_font

void allow_two_perfedits (bool flag)

Sets the value of allowing two perfedits to be created and shown to the user.

void perf_h_page_increment (int inc)

Sets the horizontal page increment size for the horizontal scrollbar of a perfedit window.

void perf v page increment (int inc)

Sets the vertical page increment size for the vertical scrollbar of a perfedit window.

void progress bar colored (int palcode)

'Setter' function for member m_progress_bar_colored

void progress_bar_thick (bool flag)

'Setter' function for member m_progress_bar_thick

void inverse_colors (bool flag)

'Setter' function for member m_inverse_colors

void window_redraw_rate (int ms)

'Setter' function for member m window redraw rate ms

· void use more icons (bool flag)

'Setter' function for member m_use_more_icons

void midi_ppqn (int ppqn)

'Setter' function for member m_midi_ppqn This value can be set from 96 to 19200 (this upper limit will be determined by what Sequencer64 can actually handle).

void midi_buss_override (char buss)

'Setter' function for member m_midi_buss_override This value can be set from 0 to 31.

• void velocity_override (int vel)

'Setter' function for member m_velocity_override

void bpm_precision (int precision)

'Setter' function for member m_bpm_precision

void bpm_step_increment (midibpm increment)

'Setter' function for member m_bpm_step_increment

void bpm_page_increment (midibpm increment)

'Setter' function for member m_bpm_page_increment

Protected Member Functions

void grid brackets (int thickness)

'Getter' function for member m_grid_brackets

void grid_style (int gridstyle)

'Setter' function for member m_grid_style

• void mainwnd rows (int value)

'Setter' function for member m_mainwnd_rows This value is not modified unless the value parameter is between 4 and 8 inclusive.

· void mainwnd_cols (int value)

'Setter' function for member m_mainwnd_cols This value is not modified unless the value parameter is between 8 and 10, inclusive.

• void max_sets (int value)

'Setter' function for member m_max_sets This value is not modified unless the value parameter is between 32 and 64, inclusive.

void text x (int value)

'Setter' function for member m_text_x This value is not modified unless the value parameter is between 6 and 6, inclusive.

void text y (int value)

'Setter' function for member m_text_y This value is not modified unless the value parameter is between 12 and 12, inclusive.

void seqchars_x (int value)

'Setter' function for member m_seqchars_x This affects the size or crampiness of a pattern slot, and for now we will hardwire it to 15.

void seqchars_y (int value)

'Setter' function for member m_seqchars_y This affects the size or crampiness of a pattern slot, and for now we will hardwire it to 5.

void segarea_x (int value)

'Setter' function for member m_seqarea_x

void segarea_y (int value)

'Setter' function for member m_segarea_y

void seqarea_seq_x (int value)

'Setter' function for member m segarea seg x

void segarea seg y (int value)

'Setter' function for member m_segarea_seg_y

• void mainwid_border (int value)

'Setter' function for member m_mainwid_border This value is not modified unless the value parameter is between 0 and 3, inclusive.

void mainwid spacing (int value)

'Setter' function for member m_mainwid_spacing This value is not modified unless the value parameter is between 2 and 6, inclusive.

void control_height (int value)

'Setter' function for member m_control_height This value is not modified unless the value parameter is between 0 and 4. inclusive.

void dump_summary ()

Provides a debug dump of basic information to help debug a surprisingly intractable problem with all busses having the name and values of the last buss in the configuration.

• void midi_beats_per_bar (int beatsperbar)

'Setter' function for member m_midi_beats_per_measure This value can be set from 1 to 16.

void midi_beats_per_minute (midibpm beatsperminute)

'Setter' function for member m_midi_beats_minute This value can be set from 20 to 500.

· void midi beat width (int beatwidth)

'Setter' function for member m_midi_beatwidth This value can be set to any power of 2 in the range from 1 to 16.

Private Types

```
    enum mainwid_grid_style_t {
        grid_style_normal,
        grid_style_white,
        grid_style_black,
        grid_style_max }
```

typedef std::vector< user_midi_bus > Busses

[user-midi-bus-definitions]

- typedef std::vector< user_midi_bus >::iterator BussIterator
- typedef std::vector< user midi bus >::const iterator BussConstIterator
- typedef std::vector< user_instrument > Instruments

[user-instrument-definitions]

- typedef std::vector< user_instrument >::iterator InstrumentIterator
- typedef std::vector< user_instrument >::const_iterator InstrumentConstIterator

Private Member Functions

· user midi bus & private bus (int buss)

'Getter' function for member m_midi_buses[index] (internal function) If the index is out of range, then an invalid object is returned.

• user instrument & private instrument (int instrum)

'Getter' function for member m_instruments[index] If the index is out of range, then a invalid object is returned.

Private Attributes

· Busses m midi buses

Provides data about the MIDI busses, readable from the "user" configuration file.

Instruments m_instruments

Provides data about the MIDI instruments, readable from the "user" configuration file.

mainwid_grid_style_t m_grid_style

[user-interface-settings]

• int m_grid_brackets

Specify drawing brackets (like the old Seg24) or a solid box.

int m_mainwnd_rows

Number of rows in the Patterns Panel.

int m_mainwnd_cols

Number of columns in the Patterns Panel.

int m_max_sets

Maximum number of screen sets that can be supported.

· int m mainwid border

These control sizes.

- int m_mainwid_spacing
- · int m_control_height

This constants seems to be created for a future purpose, perhaps to reserve space for a new bar on the mainwid pane.

• int m current zoom

Provides the initial zoom value, in units of ticks per pixel.

bool m_global_seq_feature_save

If true, this value provide a bit of backward-compatibility with the global key/scale/background-sequence persistence feature.

int m_seqedit_scale

Replaces sequence is loaded into the sequence editor.

int m_seqedit_key

Replaces seqedit::m_initial_key as the repository for the key to apply when a sequence is loaded into the sequence editor.

int m_seqedit_bgsequence

Replaces sequelit::m_initial_sequence as the repository for the background sequence to apply when a sequence is loaded into the sequence editor.

bool m_use_new_font

Sets the usage of the font.

· bool m_allow_two_perfedits

Enables the usage of two perfedit windows, for added convenience in editing multi-set songs.

• int m_h_perf_page_increment

Allows a changed to the page size for the horizontal scroll bar.

• int m v perf page increment

Allows a changed to the page size for the vertical scroll bar.

· int m_progress_bar_colored

If set, makes progress bars have the "progress_color()", instead of black.

· bool m_progress_bar_thick

If set, makes progress bars thicker than 1 pixel...

bool m_inverse_colors

If set, use an alternate, neo-inverse color palette.

int m_window_redraw_rate_ms

Provides the global setting for redraw rate of windows.

· bool m use more icons

Another [user-interface-settings] item.

• int m_text_x

Constants for the mainwid class.

- int m_text_y
- · int m seqchars x

Constants for the mainwid class.

- · int m_seqchars_y
- int m_midi_ppqn

Provides the universal PPQN setting for the duration of this setting.

int m_midi_beats_per_measure

Provides the universal and unambiguous MIDI value for beats per measure, also called "beats per bar" (BPB).

int m_midi_beats_per_minute

Provides the universal and unambiguous MIDI value for beats per minute (BPM).

int m_midi_beat_width

Provides the universal MIDI value for beats width (BW).

char m_midi_buss_override

Provides a universal override of the buss number for all sequences, for the purpose of convenience of of testing.

• int m_velocity_override

Sets the default velocity for note adding.

• int m_bpm_precision

Sets the precision of the BPM (beats-per-minute) setting.

• midibpm m_bpm_step_increment

The step increment value for BPM, regardless of the decimal precision.

midibpm m_bpm_page_increment

This is the larger increment for paging the BPM.

· int m total segs

The maximum number of patterns supported is given by the number of patterns supported in the panel (32) times the maximum number of sets (32), or 1024 patterns.

int m_seqs_in_set

Number of patterns/sequences in the Patterns Panel, also known as a "set" or "screen set".

• int m gmute tracks

Number of group-mute tracks that can be supported, which is m_seqs_in_set squared, or 1024.

• int m_max_sequence

The maximum number of patterns supported is given by the number of patterns supported in the panel (32) times the maximum number of sets (32), or 1024 patterns.

• int m segarea x

The m_seqarea_x and m_seqarea_y constants are derived from the width and heights of the default character set, and the number of characters in width, and the number of lines, in a pattern/sequence box.

- · int m_seqarea_y
- int m_seqarea_seq_x

Area of what? Doesn't look at all like it is based on the size of characters.

- int m_seqarea_seq_y
- · int m_mainwid_x

The width of the main pattern/sequence grid, in pixels.

- · int m_mainwid_y
- · bool m_save_user_config

Provides a temporary variable that can be set from the command line to cause the "user" state to be saved into the "user" configuration file.

const int mc_min_zoom

Provides the minimum zoom value, currently a constant.

const int mc_max_zoom

Provides the maximum zoom value, currently a constant.

const int mc_baseline_ppqn

Permanent storage for the baseline, default PPQN used by Seq24.

Friends

· class userfile

13.99.1 Detailed Description

These settings will eventually be made part of the "user" settings file.

13.99.2 Member Typedef Documentation

13.99.2.1 Busses

```
typedef std::vector<user_midi_bus> seq64::user_settings::Busses [private]
```

Internal type for the container of user_midi_bus objects. Sorry about the "confusion" about "bus" versus "buss". See Google for arguments about it.

13.99.2.2 BussIterator

```
typedef std::vector<user_midi_bus>::iterator seq64::user_settings::BussIterator [private]
```

13.99.2.3 BussConstiterator

typedef std::vector<user_midi_bus>::const_iterator seq64::user_settings::BussConstIterator
[private]

13.99.2.4 Instruments

typedef std::vector<user_instrument> seq64::user_settings::Instruments [private]

Internal type for the container of user_instrument objects.

13.99.2.5 InstrumentIterator

typedef std::vector<user_instrument>::iterator seq64::user_settings::InstrumentIterator [private]

13.99.2.6 InstrumentConstiterator

 $\label{typedef} \begin{tabular}{ll} typedef std::vector < user_instrument > ::const_iterator seq 64::user_settings::Instrument Const \leftarrow Iterator [private] \end{tabular}$

13.99.3 Member Enumeration Documentation

13.99.3.1 mainwid_grid_style_t

enum seq64::user_settings::mainwid_grid_style_t [private]

Enumerator

grid_style_normal	
The grid background color is white. This style better fits displaying the white-on-black	
sequence numbers. The box is drawn with brackets on either side.	
grid_style_black	The grid background color is black.
grid_style_max	Marks the end of the list, and is an illegal

13.99.4 Constructor & Destructor Documentation

```
13.99.4.1 user_settings() [1/2]
seq64::user_settings::user_settings ( )
13.99.4.2 user_settings() [2/2]
seq64::user_settings::user_settings (
             const user_settings & rhs )
13.99.5 Member Function Documentation
13.99.5.1 operator=()
user_settings & seq64::user_settings::operator= (
             const user_settings & rhs )
13.99.5.2 set_defaults()
void seq64::user_settings::set_defaults ( )
For the m_midi_buses and m_instruments members, this function can only iterate over the current size of the
vectors. But the default size is zero!
13.99.5.3 normalize()
void seq64::user_settings::normalize ( )
Should we normalize the BPM increment values here, in case they are irregular?
13.99.5.4 add_bus()
bool seq64::user_settings::add_bus (
```

const std::string & alias)

```
13.99.5.5 add_instrument()
bool seq64::user_settings::add_instrument (
            const std::string & instname )
13.99.5.6 bus()
const user_midi_bus& seq64::user_settings::bus (
             int index ) [inline]
Cannot append the const specifier.
13.99.5.7 instrument()
const user_instrument& seq64::user_settings::instrument (
             int index ) [inline]
Cannot append the const specifier.
13.99.5.8 bus_count()
int seq64::user_settings::bus_count ( ) const [inline]
13.99.5.9 set_bus_instrument()
void seq64::user_settings::set_bus_instrument (
             int index,
             int channel,
             int instrum )
13.99.5.10 bus_instrument()
int seq64::user_settings::bus_instrument (
             int buss,
             int channel ) [inline]
13.99.5.11 bus_name()
const std::string& seq64::user_settings::bus_name (
```

int buss) [inline]

```
13.99.5.12 instrument_count()
int seq64::user_settings::instrument_count ( ) const [inline]
13.99.5.13 set_instrument_controllers()
void seq64::user\_settings::set\_instrument\_controllers (
             int index,
             int cc,
             const std::string & ccname,
             bool isactive )
13.99.5.14 instrument_name() [1/2]
const std::string& seq64::user_settings::instrument_name (
            int instrum ) [inline]
13.99.5.15 instrument_name() [2/2]
const std::string& seq64::user_settings::instrument_name (
             int buss,
             int channel ) [inline]
13.99.5.16 instrument_controller_active()
bool seq64::user_settings::instrument_controller_active (
             int instrum,
             int cc ) [inline]
13.99.5.17 controller_active()
bool seq64::user_settings::controller_active (
             int buss,
             int channel,
```

It also has a shorter name.

int cc) [inline]

```
13.99.5.18 instrument_controller_name()
const std::string& seq64::user_settings::instrument_controller_name (
             int instrum,
             int cc ) [inline]
13.99.5.19 controller_name()
const std::string& seq64::user_settings::controller_name (
             int buss,
             int channel,
             int cc ) [inline]
It also has a shorter name.
13.99.5.20 grid_style() [1/2]
int seq64::user_settings::grid_style ( ) const [inline]
13.99.5.21 grid_is_normal()
bool seq64::user_settings::grid_is_normal ( ) const [inline]
13.99.5.22 grid_is_white()
bool seq64::user_settings::grid_is_white ( ) const [inline]
13.99.5.23 grid_is_black()
bool seq64::user_settings::grid_is_black ( ) const [inline]
13.99.5.24 grid_brackets() [1/2]
```

int seq64::user_settings::grid_brackets () const [inline]

```
13.99.5.25 mainwnd_rows() [1/2]
int seq64::user_settings::mainwnd_rows ( ) const [inline]
13.99.5.26 mainwnd_cols() [1/2]
int seq64::user_settings::mainwnd_cols ( ) const [inline]
13.99.5.27 seqs_in_set()
int seq64::user_settings::seqs_in_set ( ) const [inline]
13.99.5.28 gmute_tracks()
int seq64::user_settings::gmute_tracks ( ) const [inline]
13.99.5.29 max_sets() [1/2]
int seq64::user_settings::max_sets ( ) const [inline]
13.99.5.30 max_sequence()
int seq64::user_settings::max_sequence ( ) const [inline]
13.99.5.31 text_x() [1/2]
int seq64::user_settings::text_x ( ) const [inline]
13.99.5.32 text_y() [1/2]
int seq64::user_settings::text_y ( ) const [inline]
```

```
13.99.5.33 seqchars_x() [1/2]
int seq64::user_settings::seqchars_x ( ) const [inline]
13.99.5.34 seqchars_y() [1/2]
int seq64::user_settings::seqchars_y ( ) const [inline]
13.99.5.35 seqarea_x() [1/2]
int seq64::user_settings::seqarea_x ( ) const [inline]
13.99.5.36 seqarea_y() [1/2]
int seq64::user_settings::seqarea_y ( ) const [inline]
13.99.5.37 seqarea_seq_x() [1/2]
int seq64::user_settings::seqarea_seq_x ( ) const [inline]
13.99.5.38 seqarea_seq_y() [1/2]
int seq64::user_settings::seqarea_seq_y ( ) const [inline]
13.99.5.39 mainwid_border() [1/2]
int seq64::user_settings::mainwid_border ( ) const [inline]
13.99.5.40 mainwid_spacing() [1/2]
int seq64::user_settings::mainwid_spacing ( ) const [inline]
```

```
13.99.5.41 mainwid_x()
int seq64::user\_settings::mainwid\_x ( ) const [inline]
13.99.5.42 mainwid_y()
int seq64::user_settings::mainwid_y ( ) const [inline]
13.99.5.43 control_height() [1/2]
int seq64::user_settings::control_height ( ) const [inline]
13.99.5.44 zoom() [1/2]
int seq64::user_settings::zoom ( ) const [inline]
13.99.5.45 zoom() [2/2]
void seq64::user_settings::zoom (
             int value )
The default value is 2. Note that 0 is allowed as a special case, which allows the default zoom to be adjusted when
the PPQN value is different from the default.
13.99.5.46 global_seq_feature() [1/2]
bool seq64::user_settings::global_seq_feature ( ) const [inline]
13.99.5.47 global_seq_feature() [2/2]
void seq64::user_settings::global_seq_feature (
            bool flag ) [inline]
```

```
13.99.5.48 seqedit_scale() [1/2]
int seq64::user_settings::seqedit_scale ( ) const [inline]
13.99.5.49 seqedit_scale() [2/2]
void seq64::user_settings::seqedit_scale (
             int scale ) [inline]
13.99.5.50 seqedit_key() [1/2]
int seq64::user_settings::seqedit_key ( ) const [inline]
13.99.5.51 seqedit_key() [2/2]
void seq64::user_settings::seqedit_key (
             int key ) [inline]
13.99.5.52 seqedit_bgsequence() [1/2]
int seq64::user_settings::seqedit_bgsequence ( ) const [inline]
13.99.5.53 seqedit_bgsequence() [2/2]
void seq64::user_settings::seqedit_bgsequence (
             int seqnum ) [inline]
13.99.5.54 use_new_font() [1/2]
bool seq64::user_settings::use_new_font ( ) const [inline]
```

```
13.99.5.55 allow_two_perfedits() [1/2]
bool seq64::user_settings::allow_two_perfedits ( ) const [inline]
13.99.5.56 perf_h_page_increment() [1/2]
int seq64::user_settings::perf_h_page_increment ( ) const [inline]
13.99.5.57 perf_v_page_increment() [1/2]
\verb|int seq64:: user_settings::perf_v_page_increment ( ) const [inline]|\\
13.99.5.58 progress_bar_colored() [1/2]
int seq64::user_settings::progress_bar_colored ( ) const [inline]
13.99.5.59 progress_bar_thick() [1/2]
bool seq64::user_settings::progress_bar_thick ( ) const [inline]
13.99.5.60 inverse_colors() [1/2]
bool seq64::user_settings::inverse_colors ( ) const [inline]
13.99.5.61 window_redraw_rate() [1/2]
int seq64::user_settings::window_redraw_rate ( ) const [inline]
13.99.5.62 use_more_icons() [1/2]
bool seq64::user_settings::use_more_icons ( ) const [inline]
```

```
13.99.5.63 save_user_config() [1/2]
bool seq64::user_settings::save_user_config ( ) const [inline]
13.99.5.64 save_user_config() [2/2]
void seq64::user_settings::save_user_config (
            bool flag ) [inline]
13.99.5.65 grid_brackets() [2/2]
void seq64::user_settings::grid_brackets (
             int thickness ) [inline], [protected]
13.99.5.66 grid_style() [2/2]
void seq64::user_settings::grid_style (
             int gridstyle ) [protected]
13.99.5.67 mainwnd_rows() [2/2]
void seq64::user_settings::mainwnd_rows (
              int value ) [protected]
The default value is 4. Dependent values are recalculated after the assignment.
13.99.5.68 mainwnd_cols() [2/2]
void seq64::user\_settings::mainwnd\_cols (
             int value ) [protected]
The default value is 8. Dependent values are recalculated after the assignment.
```

The default value is 32. Dependent values are recalculated after the assignment.

13.99.5.69 max_sets() [2/2]

void seq64::user_settings::max_sets (

int value) [protected]

The default value is 6. Dependent values are recalculated after the assignment. This value is currently restricted, until we can code up a bigger font.

The default value is 12. Dependent values are recalculated after the assignment. This value is currently restricted, until we can code up a bigger font.

```
13.99.5.72 seqchars_x() [2/2]
void seq64::user_settings::seqchars_x (
           int value ) [protected]
13.99.5.73 seqchars_y() [2/2]
void seq64::user_settings::seqchars_y (
             int value ) [protected]
13.99.5.74 seqarea_x() [2/2]
void seq64::user_settings::seqarea_x (
             int value ) [protected]
13.99.5.75 seqarea_y() [2/2]
void seq64::user_settings::seqarea_y (
             int value ) [protected]
13.99.5.76 seqarea_seq_x() [2/2]
void seq64::user_settings::seqarea_seq_x (
             int value ) [protected]
```

```
13.99.5.77 seqarea_seq_y() [2/2]
void seq64::user_settings::seqarea_seq_y (
             int value ) [protected]
13.99.5.78 mainwid_border() [2/2]
void seq64::user_settings::mainwid_border (
             int value ) [protected]
The default value is 0. Dependent values are recalculated after the assignment.
13.99.5.79 mainwid_spacing() [2/2]
void seq64::user_settings::mainwid_spacing (
             int value ) [protected]
The default value is 2. Dependent values are recalculated after the assignment.
13.99.5.80 control_height() [2/2]
void seq64::user_settings::control_height (
             int value ) [protected]
The default value is 0. Dependent values are recalculated after the assignment.
13.99.5.81 dump_summary()
void seq64::user_settings::dump_summary ( ) [protected]
Does its work only if PLATFORM DEBUG and SEQ64 USE DEBUG OUTPUT are defined. Only enabled in
emergencies :-D.
13.99.5.82 midi_ppqn() [1/2]
int seq64::user_settings::midi_ppqn ( ) const [inline]
13.99.5.83 midi_beats_per_bar() [1/2]
int seq64::user_settings::midi_beats_per_bar ( ) const [inline]
```

```
13.99.5.84 midi_beats_per_minute() [1/2]
midibpm seq64::user_settings::midi_beats_per_minute ( ) const [inline]
13.99.5.85 midi_beat_width() [1/2]
int seq64::user_settings::midi_beat_width ( ) const [inline]
13.99.5.86 midi_buss_override() [1/2]
char seq64::user_settings::midi_buss_override ( ) const [inline]
13.99.5.87 velocity_override() [1/2]
int seq64::user_settings::velocity_override ( ) const [inline]
13.99.5.88 bpm_precision() [1/2]
int seq64::user_settings::bpm_precision ( ) const [inline]
13.99.5.89 bpm_step_increment() [1/2]
midibpm seq64::user_settings::bpm_step_increment ( ) const [inline]
13.99.5.90 bpm_page_increment() [1/2]
midibpm seq64::user_settings::bpm_page_increment ( ) const [inline]
13.99.5.91 min_zoom()
int seq64::user_settings::min_zoom ( ) const [inline]
```

```
13.99.5.92 max_zoom()
int seq64::user_settings::max_zoom ( ) const [inline]
13.99.5.93 baseline_ppqn()
int seq64::user_settings::baseline_ppqn ( ) const [inline]
13.99.5.94 use_new_font() [2/2]
void seq64::user_settings::use_new_font (
            bool flag ) [inline]
13.99.5.95 allow_two_perfedits() [2/2]
void seq64::user_settings::allow_two_perfedits (
             bool flag ) [inline]
13.99.5.96 perf_h_page_increment() [2/2]
void seq64::user_settings::perf_h_page_increment (
              int inc )
This value ranges from 1 (the original value, really too small for a "page" operation) to 6 (which is 24 measures, the
same as the typical width of the perfroll)
13.99.5.97 perf_v_page_increment() [2/2]
void seq64::user_settings::perf_v_page_increment (
```

This value ranges from 1 (the original value, really too small for a "page" operation) to 18 (which is 18 tracks, slightly more than the typical height of the perfroll)

int inc)

```
13.99.5.99 progress_bar_thick() [2/2]
void seq64::user_settings::progress_bar_thick (
            bool flag ) [inline]
13.99.5.100 inverse_colors() [2/2]
void seq64::user_settings::inverse_colors (
             bool flag ) [inline]
13.99.5.101 window_redraw_rate() [2/2]
void seq64::user_settings::window_redraw_rate (
             int ms ) [inline]
13.99.5.102 use_more_icons() [2/2]
void seq64::user_settings::use_more_icons (
             bool flag ) [inline]
13.99.5.103 midi_ppqn() [2/2]
void seq64::user_settings::midi_ppqn (
             int value )
The default value is 192.
13.99.5.104 midi_buss_override() [2/2]
void seq64::user_settings::midi_buss_override (
             char buss )
```

The default value is -1, which means that there is no buss override. It provides a way to override the buss number for smallish MIDI files. It replaces the buss-number read from the file. This option is turned on by the –bus option, and is merely a convenience feature for the quick previewing of a tune. (It's called "developer laziness".)

```
13.99.5.106 bpm_precision() [2/2]
void seq64::user_settings::bpm_precision (
            int precision )
13.99.5.107 bpm_step_increment() [2/2]
void seq64::user_settings::bpm_step_increment (
             midibpm increment )
13.99.5.108 bpm_page_increment() [2/2]
void seq64::user_settings::bpm_page_increment (
             midibpm increment )
13.99.5.109 midi_beats_per_bar() [2/2]
void seq64::user_settings::midi_beats_per_bar (
             int value ) [protected]
The default value is 4.
13.99.5.110 midi_beats_per_minute() [2/2]
void seq64::user_settings::midi_beats_per_minute (
             midibpm value ) [protected]
The default value is 120.
13.99.5.111 midi_beat_width() [2/2]
void seq64::user_settings::midi_beat_width (
             int bw ) [protected]
The default value is 4.
13.99.5.112 private_bus()
user_midi_bus & seq64::user_settings::private_bus (
             int index ) [private]
```

This invalid object has an empty alias, and all the instrument numbers are -1.

```
13.99.5.113 private_instrument()
```

This invalid object has an empty(), instrument name, false for all controllers_active[] values, and empty controllers[] string values.

13.99.6 Friends And Related Function Documentation

```
13.99.6.1 userfile
friend class userfile [friend]
```

13.99.7 Field Documentation

13.99.7.2 m_instruments

```
13.99.7.1 m_midi_buses

Busses seq64::user_settings::m_midi_buses [private]
```

Since this object is a vector, its size is adjustable.

```
Instruments seq64::user_settings::m_instruments [private]
```

The size is adjustable, and grows as objects are added.

```
13.99.7.3 m_grid_style
```

mainwid_grid_style_t seq64::user_settings::m_grid_style [private]

These are not labelled, but are present in the "user" configuration file in the following order:

```
-# grid-style
-# grid-brackets
-# mainwnd-rows
-# mainwnd-cols
-# max-set
-# mainwid-border
-# control-height
-# zoom
-# global-seg-feature
-# use-new-font
-# allow-two-perfedits
-# perf-h-page-increment
-# perf-v-page-increment
-# progress-bar-colored (new)
-# progress-bar-thick (new)
-# window-redraw-rate-ms (new)
```

Specifies the current grid style.

13.99.7.4 m_grid_brackets

```
int seq64::user_settings::m_grid_brackets [private]
```

0 = no brackets, 1 and above is the thickness of the brakets. 1 is the normal thickness of the brackets, 2 is a two-pixel thickness, and so on.

13.99.7.5 m_mainwnd_rows

```
int seq64::user_settings::m_mainwnd_rows [private]
```

The current value is 4, and if changed, many other values depend on it. Together with m_mainwnd_cols, this value fixes the patterns grid into a 4 x 8 set of patterns known as a "screen set". We would like to be able to change this value from 4 to 8, and maybe allow the values of 5, 6, and 7 as well. But if we could just get 8 working, then well would Sequencer64 deserve the 64 in its name.

13.99.7.6 m_mainwnd_cols

```
int seq64::user_settings::m_mainwnd_cols [private]
```

The current value is 4, and probably won't change, since other values depend on it. Together with m_mainwnd_rows, this value fixes the patterns grid into a 4 x 8 set of patterns known as a "screen set".

13.99.7.7 m_max_sets

```
int seq64::user_settings::m_max_sets [private]
```

Basically, that the number of times the Patterns Panel can be filled. 32 sets can be created. Although this value is part of the "user" configuration file, it is likely that it will never change. Rather, the number of sequences per set would change. We'll see.

13.99.7.8 m_mainwid_border

```
int seq64::user_settings::m_mainwid_border [private]
```

We'll try changing them and see what happens. Increasing these value spreads out the pattern grids a little bit and makes the Patterns panel slightly bigger. Seems like it would be useful to make these values user-configurable.

13.99.7.9 m_mainwid_spacing

```
int seq64::user_settings::m_mainwid_spacing [private]
```

13.99.7.10 m_control_height

```
int seq64::user_settings::m_control_height [private]
```

But it is used only in this header file, to define m_mainwid_y, but doesn't add anything to that value.

13.99.7.11 m_current_zoom

```
int seq64::user_settings::m_current_zoom [private]
```

The original default value was 32 ticks per pixel, but larger PPQN values need higher values, and we will have to adapt the default zoom to the PPQN value. Also, the zoom can never be zero, as it can appear as the divisor in scaling equations.

13.99.7.12 m_global_seq_feature_save

```
bool seq64::user_settings::m_global_seq_feature_save [private]
```

In this feature, applying one of these three changes to a sequence causes them to also be applied to sequences that are subsequently opened for editing. However, we improve on this feature by allowing the changes to be saved in the global, proprietary part of the saved MIDI file.

If false, the user can still save the key/scale/background-sequence values with each individual sequence, so they can be different.

This value will be true by default, unless changed in the "user" configuration file.

13.99.7.13 m_seqedit_scale

```
int seq64::user_settings::m_seqedit_scale [private]
```

Its default value is c_scale_off. Although this value is now stored in the user_settings class, it always comes from the currently loaded MIDI file, if present. If m_global_seq_feature_save is true, this variable is stored in the "proprietary" track at the end of the file, under the control tag c_musicscale, and will be applied to any sequence that is edited. If m_global_seq_feature_save is false, this variable is stored, if used, in the meta-data for the sequence to which it applies, and, again, is tagged with the control tag c_musicscale.

13.99.7.14 m_seqedit_key

```
int seq64::user_settings::m_seqedit_key [private]
```

Its default value is SEQ64_KEY_OF_C. Although this value is now stored in the user_settings class, it always comes from the currently loaded MIDI file, if present. If m_global_seq_feature_save is true, this variable is stored in the "proprietary" track at the end of the file, under the control tag c_musickey, and will be applied to any sequence that is edited. If m_global_seq_feature_save is false, this variable is stored, if used, in the meta-data for the sequence to which it applies, and, again, is tagged with the control tag c_musickey.

13.99.7.15 m_seqedit_bgsequence

```
int seq64::user_settings::m_seqedit_bgsequence [private]
```

Its default value is SEQ64_SEQUENCE_LIMIT. Although this value is now stored in the user_settings class, it always comes from the currently loaded MIDI file, if present. If m_global_seq_feature_save is true, this variable is stored, if it has a valid (but not "legal") value, in the "proprietary" track at the end of the file, under the control tag c_backsequence, and will be applied to any sequence that is edited. If m_global_seq_feature_save is false, this variable is stored, if used, in the meta-data for the sequence to which it applies, and, again, is tagged with the control tag c_backsequence.

```
13.99.7.16 m_use_new_font
```

```
bool seq64::user_settings::m_use_new_font [private]
```

By default, in normal mode, the new font is used. In legacy mode, the old font is used.

13.99.7.17 m_allow_two_perfedits

```
bool seq64::user_settings::m_allow_two_perfedits [private]
```

Defaults to true.

13.99.7.18 m_h_perf_page_increment

```
int seq64::user_settings::m_h_perf_page_increment [private]
```

The value used to be hardwired to 1 (in four-measure units), now it defaults to 4 (16 measures at a time). The value of 1 is already covered by the scrollbar arrows.

13.99.7.19 m_v_perf_page_increment

```
int seq64::user_settings::m_v_perf_page_increment [private]
```

The value used to be hardwired to 1 (in single-track units), now it defaults to 8. The value of 1 is already covered by the scrollbar arrows.

13.99.7.20 m_progress_bar_colored

```
int seq64::user_settings::m_progress_bar_colored [private]
```

This value is no longer hardwired in the gui_palette_gtk2 module to be red. Now we want to let the color select from a slightly large palette. We chande this from a boolean to an integer to allow the selection of more colors.

13.99.7.21 m_progress_bar_thick

```
bool seq64::user_settings::m_progress_bar_thick [private]
```

2 pixels. It isn't useful to support anything thicker.

13.99.7.22 m_inverse_colors

```
bool seq64::user_settings::m_inverse_colors [private]
```

Not all colors are reversed, though.

13.99.7.23 m_window_redraw_rate_ms

```
int seq64::user_settings::m_window_redraw_rate_ms [private]
```

Not all windows use this yet. The default is 40 ms (c_redraw_ms, which is 20 ms in Windows builds)), but some windows originally used 25 ms, so beware of side-effects.

13.99.7.24 m_use_more_icons

```
bool seq64::user_settings::m_use_more_icons [private]
```

If set to 1, icons will be used for more buttons. This setting affects only a few buttons so far, such as the buttons at the top of the main window.

13.99.7.25 m_text_x

```
int seq64::user_settings::m_text_x [private]
```

The m_text_x and m_text_y constants help define the "seqarea" size. It looks like these two values are the character width (x) and height (y) in pixels. Thus, these values would be dependent on the font chosen. But that, currently, is hard-wired. See the m_font_6_12[] array for the default font specification.

However, please not that font files are not used. Instead, the fonts are provided by two pixmaps in the src/pixmap directory: font_b.xpm (black lettering on a white background) and font_w.xpm (white lettering on a black background).

We have added black-on-yellow and yellow-on-black versions of the fonts, to support the highlighting of pattern boxes if they are empty of actual MIDI events.

We have also added a set of four new font files that are roughly the same size, and are treated as the same size, but look smooth and less like a DOS-era font.

The font module does not use these values directly, but does define some similar variables that differ slightly between the two styles of font. There are a lot of tricks and hard-wired places to fix before further work can be done with fonts in Sequencer64.

13.99.7.26 m_text_y

```
int seq64::user_settings::m_text_y [private]
```

13.99.7.27 m_seqchars_x

```
int seq64::user_settings::m_seqchars_x [private]
```

The m_seqchars_x and m_seqchars_y constants help define the "seqarea" size. These look like the number of characters per line and the number of lines of characters, in a pattern/sequence box.

```
13.99.7.28 m_seqchars_y
```

```
int seq64::user_settings::m_seqchars_y [private]
```

13.99.7.29 m_midi_ppqn

```
int seq64::user_settings::m_midi_ppqn [private]
```

This variable replaces the global ppqn. The default value of this setting is 192 parts-per-quarter-note (PPQN). There is still a lot of work to get a different PPQN to work properly in speed of playback, scaling of the user interface, and other issues. Note that this value can be changed by the still-experimental –ppqn option. There is one remaining trace of the global, though: DEFAULT_PPQN.

```
13.99.7.30 m_midi_beats_per_measure
```

```
int seq64::user_settings::m_midi_beats_per_measure [private]
```

This variable will replace the global beats per measure. The default value of this variable is SEQ64_DEFAULT_← BEATS_PER_MEASURE (4). For external access, we will call this value "beats per bar", abbreviate it "BPB", and use "bpb" in any accessor function names. Now, although it applies to the whole session, we should be able to continue seq24's tradition of allowing each sequence to have its own time signature. Also, there are a number of places where the number 4 appears and looks like it might be a hardwired BPB value, either for MIDI purposes or for drawing the piano-roll grids. So we might need a couple different versions of this variable.

```
13.99.7.31 m_midi_beats_per_minute
```

```
int seq64::user_settings::m_midi_beats_per_minute [private]
```

This variable will replace the global beats per minute. The default value of this variable is DEFAULT_BPM (120). This variable should apply to the whole session; there's probably no way to support a diffent tempo for each sequence. But we shall see. For external access, we will call this value "beats per minute", abbreviate it "BPM", and use "bpm" in any accessor function names.

```
13.99.7.32 m midi beat width
```

```
int seq64::user_settings::m_midi_beat_width [private]
```

This variable will replace the global beat_width. The default value of this variable is DEFAULT_BEAT_WIDTH (4). Now, although it applies to the whole session, we should be able to continue seq24's tradition of allowing each sequence to have its own time signature. Also, there are a number of places where the number 4 appears and looks like it might be a hardwired BW value, either for MIDI purposes or for drawing the user-interface. So we might need a couple different versions of this variable. For external access, we will call this value "beat width", abbreviate it "BW", and use "bw" in any accessor function names.

```
13.99.7.33 m_midi_buss_override
```

```
char seq64::user_settings::m_midi_buss_override [private]
```

This variable replaces the global buss-override variable, and is set via the command-line option –bus.

13.99.7.34 m_velocity_override

```
int seq64::user_settings::m_velocity_override [private]
```

The value SEQ64_PRESERVE_VELOCITY (-1) preserves the velocity of incoming notes, so that nuances in live playing can be preserved. The popup-menu for the "Vol" button in the seqedit window shows this value as the "Free" menu entry. The rest of the values in the menu show a few select velocities, but any velocity from 0 to 127 can be entered here. Of course, 0 is not recommended.

13.99.7.35 m_bpm_precision

```
int seq64::user_settings::m_bpm_precision [private]
```

The original value was effectively 0, but we need to be able to support the following values:

```
0. The legacy default.
```

- One decimal place in the BPM spinner.
 Two decimal places in the BPM spinner.

13.99.7.36 m bpm step increment

```
midibpm seq64::user_settings::m_bpm_step_increment [private]
```

The default value is the legacy value, 1, for a BPM precision value of 0. The default value is 0.1 if one decimal place of precision is in force, and 0.01 if two decimal places of precision is in force. This is the increment that is performed in the BPM field of the main window when the arrow-buttons are clicked, the up/down arrow keys are pressed, or the BPM MIDI controls are processed.

13.99.7.37 m_bpm_page_increment

```
midibpm seq64::user_settings::m_bpm_page_increment [private]
```

Currently, the only way to use this increment is to click in the BPM field of the main window and then use the Page-Up and Page-Down keys.

13.99.7.38 m_total_seqs

```
int seq64::user_settings::m_total_seqs [private]
```

It is basically the same value as m max sequence by default. It is a derived value, and not stored in the "user" file. We might make it equal to the maximum number of sequences the currently-loaded MIDI file.

```
m_total_seqs = m_seqs_in_set * m_max_sets;
```

```
13.99.7.39 m_seqs_in_set
```

```
int seq64::user_settings::m_seqs_in_set [private]
```

This value is $4 \times 8 = 32$ by default.

Warning

Currently implicit/explicit in a number of the "rc" file and rc_settings. Would probably want the left 32 or the first 32 items in the main window only to be subject to keystroke control. This value is calculated by the normalize() function, and is *not* part of the "user" configuration file.

13.99.7.40 m_gmute_tracks

```
int seq64::user_settings::m_gmute_tracks [private]
```

This value is *not* part of the "user" configuration file; it is calculated by the normalize() function.

13.99.7.41 m_max_sequence

```
int seq64::user_settings::m_max_sequence [private]
```

It is a derived value, and not stored in the "user" file.

```
m_max_sequence = m_seqs_in_set * m_max_sets;
```

```
13.99.7.42 m_seqarea_x
```

```
int seq64::user_settings::m_seqarea_x [private]
```

Compare these two constants to m_seqarea_seq_x(y), which was in mainwid.h, but is now in this file.

```
13.99.7.43 m_seqarea_y
```

```
int seq64::user_settings::m_seqarea_y [private]
```

13.99.7.44 m_seqarea_seq_x

```
int seq64::user_settings::m_seqarea_seq_x [private]
```

These are used only in the mainwid module.

```
13.99.7.45 m_seqarea_seq_y
int seq64::user_settings::m_seqarea_seq_y [private]
13.99.7.46 m_mainwid_x
int seq64::user_settings::m_mainwid_x [private]
Affected by the m_mainwid_border and m_mainwid_spacing values.
 c_mainwid_x =
     (c_seqarea_x + c_mainwid_spacing) * c_mainwnd_cols -
         c_mainwid_spacing + c_mainwid_border * 2
 );
13.99.7.47 m_mainwid_y
int seq64::user_settings::m_mainwid_y [private]
13.99.7.48 m_save_user_config
bool seq64::user_settings::m_save_user_config [private]
file.
```

Normally, this state is not saved. It is not saved because there is currently no user-interface for editing it, and because it can pick up some command-line options, and it is not right to have them written to the "user" configuration

(The "rc" configuration file is a different case, having historically always been saved, and having a number of command-line options, such as JACK settings that should generally be permanent on a given system.)

Anyway, this flag can be set by the -user-save option. This setting is never saved. But note that, if no "user" configuration file is found, it is then saved anyway.

```
13.99.7.49 mc_min_zoom
const int seq64::user_settings::mc_min_zoom [private]
It's value is 1.
13.99.7.50 mc_max_zoom
const int seq64::user_settings::mc_max_zoom [private]
```

It's value was 32, but is now 512, to allow for better presentation of high PPQN valued sequences.

13.99.7.51 mc_baseline_ppqn

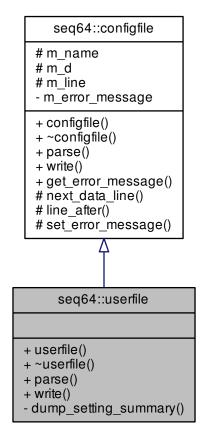
```
const int seq64::user_settings::mc_baseline_ppqn [private]
```

This value is necessary in order to keep user-interface elements stable when different PPQNs are used. It is set to DEFAULT_PPQN.

13.100 seq64::userfile Class Reference

Supports the user's \sim /.config/sequencer64/sequencer64.usr and \sim /.seq24usr configuration file.

Inheritance diagram for seq64::userfile:



Public Member Functions

- userfile (const std::string &a_name)
 Principal constructor.
- \sim userfile ()

A rote destructor needed for a derived class.

• bool parse (perform &a_perf)

Parses a "usr" file, filling in the given perform object.

bool write (const perform &a_perf)

This function just returns false, as there is no "perform" information in the user-file yet.

Private Member Functions

• void dump_setting_summary ()

Provides a debug dump of basic information to help debug a surprisingly intractable problem with all busses having the name and values of the last buss in the configuration.

Additional Inherited Members

13.100.1 Constructor & Destructor Documentation

Parameters

name Provides the full file path specification to the configuration file.

```
13.100.1.2 \simuserfile()
```

```
seq64::userfile::~userfile ( )
```

13.100.2 Member Function Documentation

13.100.2.1 parse()

This function opens the file as a text file (line-oriented).

Parameters

a_perf The performance object, currently unused.
--

Returns

Returns true if the parsing succeeded.

Implements seq64::configfile.

13.100.2.2 write()

Parameters

a_perf The performance object, currently unused.

Returns

Returns true if the writing succeeded.

Implements seq64::configfile.

```
13.100.2.3 dump_setting_summary()
```

```
void seq64::userfile::dump_setting_summary ( ) [private]
```

Does work only if PLATFORM_DEBUG is defined; see the user_settings class.

Index

~AbstractPerfInput	seq64::mastermidibus, 458
seq64::AbstractPerfInput, 134	∼midi alsa
~automutex	 seq64::midi_alsa, 468
seq64::automutex, 138	~midi_alsa_info
~busarray	seq64::midi_alsa_info, 477
seq64::busarray, 140	∼midi api
~businfo	seq64::midi_api, 483
seq64::businfo, 150	~midi_container
~configfile	seq64::midi_container, 493
seq64::configfile, 163	~midi_in_jack
~editable_event	seq64::midi_in_jack, 509
seq64::editable_event, 171	~midi info
~editable_events	seq64::midi_info, 514
seq64::editable_events, 182	~midi_jack
~event	seq64::midi_jack, 526
seq64::event, 193	~midi_jack_data
~event_list	seq64::midi_jack_data, 536
seg64::event list, 214	~midi jack info
~eventedit	seq64::midi_jack_info, 540
seq64::eventedit, 228	~midi_list
~eventslots	seq64::midi_list, 548
seq64::eventslots, 244	~midi_out_jack
~gui_assistant	seq64::midi_out_jack, 559
seq64::gui_assistant, 276	~midi_queue
~gui_assistant_gtk2	seq64::midi_queue, 565
seq64::gui_assistant_gtk2, 279	~midi_splitter
~gui_drawingarea_gtk2	seq64::midi_splitter, 570
seq64::gui_drawingarea_gtk2, 284	~midi_vector
~gui_palette_gtk2	seq64::midi_vector, 579
seq64::gui_palette_gtk2, 302	~midibase
~gui_window_gtk2	seq64::midibase, 587
seq64::gui_window_gtk2, 312	~midibus
~jack_assistant	seq64::midibus, 607
seq64::jack_assistant, 318	~midifile
~keys_perform	seq64::midifile, 619
seq64::keys_perform, 352	~optionsfile
~keys_perform_gtk2	seq64::optionsfile, 646
seq64::keys_perform_gtk2, 376	~perfedit
~lfownd	seq64::perfedit, 654
seq64::Ifownd, 392	~perfnames
~maintime	seq64::perfnames, 670
seq64::maintime, 397	~perform
~mainwid	seq64::perform, 690
seq64::mainwid, 404	~perfroll
~mainwnd	seq64::perfroll, 768
seq64::mainwnd, 421	~perftime
~mastermidibase	seq64::perftime, 786
seq64::mastermidibase, 440	~rterror
~mastermidibus	seq64::rterror, 819

~rtmidi	seq64::user_settings, 1093
	add chord
seq64::rtmidi, 823 ∼rtmidi_in	_
	seq64::seqroll, 950
seq64::rtmidi_in, 831	seq64::sequence, 1010
~rtmidi_info	add_clock
seq64::rtmidi_info, 841	seq64::perform, 745
~rtmidi_out	add_event
seq64::rtmidi_out, 850	seq64::midi_container, 496
\sim seqdata	seq64::sequence, 1010, 1011
seq64::seqdata, 861	add_extended_keys_page
\sim seqedit	seq64::options, 643
seq64::seqedit, 878	add_input
\sim seqevent	seq64::perform, 745
seq64::seqevent, 906	seq64::rtmidi_info, 842
\sim seqkeys	add instrument
seq64::seqkeys, 921	seq64::user_settings, 1093
\sim seqmenu	add_jack_sync_page
seq64::seqmenu, 934	seq64::options, 643
~segroll	add_keyboard_page
seq64::seqroll, 949	seq64::options, 643
~seqtime	
seq64::seqtime, 978	add_long
~sequence	seq64::midi_container, 496
seq64::sequence, 993	add_midi_clock_page
∼triggers	seq64::options, 642
seq64::triggers, 1060	add_midi_input_page
~userfile	seq64::options, 643
	add_mouse_page
seq64::userfile, 1119	seq64::options, 643
about_dialog	add_note
seq64::mainwnd, 426	seq64::seqroll, 950
AbstractPerfInput	seq64::sequence, 1009
seq64::AbstractPerfInput, 134	add_output
action	seq64::rtmidi_info, 842
seq64::midi_control, 501	add_sequence
activate	seq64::perform, 700
seq64::businfo, 152	add short
seq64::jack_assistant, 322	seq64::midi container, 496
seq64::mastermidibase, 450	add_trigger
•	seq64::midifile, 625
seq64::mastermidibus, 462	seq64::sequence, 1012
seq64::perform, 734	add variable
activate_adding	seq64::midi container, 495
seq64::AbstractPerfInput, 135	adding
seq64::FruityPerfInput, 268	seq64::seqroll, 964
seq64::Seq24PerfInput, 853	• • •
active	adj_callback_bpm
seq64::businfo, 150	seq64::mainwnd, 422
seq64::midi_control, 502	adj_callback_ss
add	seq64::mainwnd, 422
seq64::busarray, 140, 141	adjust_offset
seq64::editable_events, 185	seq64::sequence, 1043
seq64::event_list, 215	seq64::triggers, 1070
seq64::midi_jack_info, 544	adjust_offsets_to_length
seq64::midi_port_info, 561	seq64::triggers, 1063
seq64::midi_queue, 565	adjust_timestamp
seq64::triggers, 1062	seq64::sequence, 1026
add_bus	adjust_trigger_offsets_to_length
seq64::midi_info, 519	seq64::sequence, 1043
seq64::rtmidi_info, 842	adjustment_dummy
· · · · · · · · · · · · · · · · · · ·	- ·

seq64, 102	api_flush
alias	seq64::mastermidibase, 451
seq64::user_midi_bus_t, 1083	seq64::mastermidibus, 461, 463
align_selection	seq64::midi_alsa, 472
seq64::seqroll, 961	seq64::midi_alsa_info, 479
all_notes_off	seq64::midi_api, 486
seq64::perform, 736	seq64::midi_info, 516
allocate	seq64::midi_jack, 532
seq64::midi_queue, 566	seq64::midi_jack_info, 542
allow click edit	seq64::midibase, 599
seq64::rc_settings, 801, 805	seq64::midibus, 610
allow_mod4_mode	seq64::rtmidi, 826
seq64::rc_settings, 801, 805	seq64::rtmidi_info, 845
allow_snap_split	api_get_bus_name
seq64::rc_settings, 801, 805	seq64::midi_api, 487
allow_two_perfedits	api_get_midi_event
seq64::user_settings, 1100, 1106	seq64::mastermidibase, 452
analyze	seq64::mastermidibus, 459, 462
seq64::editable_event, 176	seq64::midi_alsa, 470
any_group_unmutes	seq64::midi_alsa_info, 477
seq64::perform, 731	seq64::midi_api, 485
any_selected_notes	seq64::midi_in_jack, 510
seq64::event_list, 220	seq64::midi_info, 516
seq64::sequence, 994	seq64::midi_jack, 531
api_clock	seq64::midi_jack_info, 541
seq64::mastermidibase, 452	seq64::midibase, 598
seq64::midi_alsa, 472	seq64::midibus, 612
seq64::midi_api, 487	seq64::rtmidi, 825
seq64::midi_jack, 533	seq64::rtmidi_info, 845
seq64::midibase, 600	api_get_port_name
seq64::midibus, 611, 613	seq64::midi_api, 487
seq64::rtmidi, 824	seq64::midi_jack, 534
api_connect	api_init
seq64::midi_api, 484	seq64::mastermidibase, 450
seq64::midi_info, 516	seq64::mastermidibus, 459, 463
seq64::midi_jack, 529	api_init_clock
seq64::midi_jack_info, 541	seq64::mastermidibase, 451
seq64::midibus, 611	api_init_in
seq64::rtmidi, 823	seq64::midi_alsa, 469
seq64::rtmidi_info, 846	seq64::midi_api, 484
api_continue_from	seq64::midi_jack, 530
seq64::mastermidibase, 451	seq64::midibase, 599
seq64::mastermidibus, 462	seq64::midibus, 608, 611
seq64::midi_alsa, 472	seq64::rtmidi, 825
seq64::midi_api, 486	api_init_in_sub
seq64::midi_jack, 532	seq64::midi_alsa, 470
seq64::midibase, 599	seq64::midi_api, 485
seq64::midibus, 610, 613	seq64::midi_jack, 531
seq64::rtmidi, 824	seq64::midibase, 598
api_data	seq64::midibus, 609, 612
seq64::rtmidi_in_data, 836	seq64::rtmidi, 825
api_deinit_in	api_init_out
seq64::midi_alsa, 470	seq64::midi_alsa, 469
seq64::midi_api, 485	seq64::midi_api, 484
seq64::midi_jack, 531	seq64::midi_jack, 529
seq64::midibase, 598	seq64::midibase, 599
seq64::midibus, 609, 612	seq64::midibus, 608, 612
seq64::rtmidi, 825	seq64::rtmidi, 825

	04 1 11 004
api_init_out_sub	seq64::rtmidi, 824
seq64::midi_alsa, 469	seq64::rtmidi_info, 845
seq64::midi_api, 484	api_start
seq64::midi_jack, 530	seq64::mastermidibase, 450
seq64::midibase, 598	seq64::mastermidibus, 461
seq64::midibus, 608, 612	seq64::midi_alsa, 472
seq64::rtmidi, 825	seq64::midi_api, 486
api_is_more_input	seq64::midi_jack, 533
seq64::mastermidibase, 452	seq64::midibase, 600
seq64::mastermidibus, 459, 462	seq64::midibus, 610, 613
api_play	seq64::rtmidi, 824
seq64::midi_alsa, 471	api_stop
seq64::midi_api, 485	seq64::mastermidibase, 451
seq64::midi_jack, 532	seq64::mastermidibus, 461
seq64::midibase, 598	seq64::midi_alsa, 472
seq64::midibus, 609, 614	seq64::midi_api, 486
seg64::rtmidi, 823	seq64::midi_jack, 533
api_poll_for_midi	seq64::midibase, 600
seq64::mastermidibase, 452	seq64::midibase, 600 seq64::midibus, 611, 613
seq64::mastermidibus, 459, 462	seq64::rtmidi, 824
•	•
seq64::midi_alsa, 470	api_sysex
seq64::midi_alsa_info, 478	seq64::midi_alsa, 471
seq64::midi_api, 485	seq64::midi_api, 486
seq64::midi_in_jack, 509	seq64::midi_jack, 532
seq64::midi_info, 516	seq64::midibase, 599
seq64::midi_jack, 531	seq64::midibus, 610
seq64::midi_jack_info, 541	seq64::rtmidi, 826
seq64::midibase, 597	app_client_name
seq64::midibus, 613	seq64::rc_settings, 804
seq64::rtmidi, 826	app_name
seq64::rtmidi_info, 845	seq64::midi_info, 515
api_port_start	seq64::rtmidi_info, 844
api_port_start seq64::mastermidibase, 452	seq64::rtmidi_info, 844 append
. — —	-
seq64::mastermidibase, 452	append
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463	append seq64::event_list, 216
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516	append seq64::event_list, 216 append_event seq64::sequence, 1011
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_api, 487	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_info, 515	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_info, 515 seq64::midi_jack, 533	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_info, 515 seq64::midi_jack, 533 seq64::midi_jack_info, 542	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_info, 515 seq64::midi_jack, 533 seq64::midi_jack_info, 542 seq64::rtmidi, 824	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_jack, 533 seq64::midi_jack, 533 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708 array
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_jack, 533 seq64::midi_jack, 533 seq64::midi_jack_info, 542 seq64::rtmidi, 824 seq64::rtmidi_info, 845 api_set_ppqn	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708 array seq64::midi_message, 554
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_jack, 533 seq64::midi_jack, 533 seq64::midi_jack_info, 542 seq64::rtmidi, 824 seq64::rtmidi_info, 845 api_set_ppqn seq64::mastermidibase, 451	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708 array seq64::midi_message, 554 at
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_jack_533 seq64::midi_jack_info, 542 seq64::rtmidi, 824 seq64::rtmidi, 824 seq64::rtmidi_info, 845 api_set_ppqn seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708 array seq64::midi_message, 554 at seq64::midi_message, 553
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_jack_info, 515 seq64::midi_jack_info, 542 seq64::midi_jack_info, 542 seq64::rtmidi, 824 seq64::rtmidi_info, 845 api_set_ppqn seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708 array seq64::midi_message, 554 at seq64::midi_message, 553 at_bpm_dn
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_jack, 533 seq64::midi_jack, 533 seq64::midi_jack_info, 542 seq64::rtmidi, 824 seq64::rtmidi, 824 seq64::mastermidibase, 451 seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa_info, 478	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708 array seq64::midi_message, 554 at seq64::midi_message, 553 at_bpm_dn seq64::keys_perform, 365
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_info, 515 seq64::midi_jack, 533 seq64::midi_jack, 533 seq64::midi_jack_info, 542 seq64::rtmidi, 824 seq64::rtmidi, 824 seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa_info, 478 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_alsa_info, 478 seq64::midi_api, 487	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708 array seq64::midi_message, 554 at seq64::midi_message, 553 at_bpm_dn seq64::keys_perform, 365 at_bpm_up
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_jack, 533 seq64::midi_jack, 533 seq64::rtmidi_jack, info, 542 seq64::rtmidi, 824 seq64::rtmidi_info, 845 api_set_ppqn seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa_info, 478 seq64::midi_alsa_info, 478 seq64::midi_api, 487 seq64::midi_api, 487 seq64::midi_info, 515	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708 array seq64::midi_message, 554 at seq64::midi_message, 553 at_bpm_dn seq64::keys_perform, 365 at_bpm_up seq64::keys_perform, 365
seq64::mastermidibase, 452 seq64::mastermidibus, 462, 463 seq64::midi_alsa_info, 478 seq64::midi_info, 516 seq64::midi_jack_info, 542 seq64::rtmidi_info, 845 api_set_beats_per_minute seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_info, 515 seq64::midi_jack, 533 seq64::midi_jack, 533 seq64::midi_jack_info, 542 seq64::rtmidi, 824 seq64::rtmidi, 824 seq64::mastermidibase, 451 seq64::mastermidibus, 461, 463 seq64::midi_alsa_info, 478 seq64::midi_alsa, 473 seq64::midi_alsa_info, 478 seq64::midi_alsa_info, 478 seq64::midi_api, 487	append seq64::event_list, 216 append_event seq64::sequence, 1011 append_sysex seq64::event, 202 application_name seq64::rc_settings, 804 apply_length seq64::seqedit, 882 apply_song_transpose seq64::mainwnd, 425 seq64::perform, 704 seq64::sequence, 1001 armed_saved seq64::perform, 708 array seq64::midi_message, 554 at seq64::midi_message, 553 at_bpm_dn seq64::keys_perform, 365 at_bpm_up

at_fast_forward	baseline_ppqn
seq64::keys_perform, 368	seq64::user_settings, 1106
at_follow_transport	beat_width
seq64::keys_perform, 368	seq64::midi_timing, 575
at_group_learn	beats
seq64::keys_perform, 367	seq64::midi_measures, 551
at_group_off	beats_per_measure
seq64::keys_perform, 367	seq64::midi_timing, 575
at_group_on	beats_per_minute
seq64::keys_perform, 367	seq64::midi_timing, 574, 575
at_keep_queue	begin
seq64::keys_perform, 366	seq64::editable_events, 184
at_menu_mode	seq64::event_list, 215
seq64::keys_perform, 367	bg_color
at_pattern_edit	seq64::gui_palette_gtk2, 305, 306
seq64::keys_perform, 368	black
at_pause	seq64::gui_palette_gtk2, 303
seq64::keys_perform, 367	black_key
at_pointer_position	seq64::gui_palette_gtk2, 305
seq64::keys_perform, 368	black_paint
at_queue	seq64::gui_palette_gtk2, 305
seq64::keys_perform, 366	blue
at_replace	seq64::gui_palette_gtk2, 305
seq64::keys_perform, 366	bpm
at_rewind	seq64::midi_info, 515
seq64::keys_perform, 368	seq64::midibase, 589
at_screenset_dn	seq64::rtmidi_info, 845
seq64::keys_perform, 366	bpm_dn
at_screenset_up	seq64::keys_perform, 353
seq64::keys_perform, 366	bpm_from_tempo_us
at_set_playing_screenset	seq64, 79
seq64::keys_perform, 366	bpm_page_increment
at_show_ui_sequence_key	seq64::user_settings, 1105, 1108
seq64::keys_perform, 369	bpm_precision
at_show_ui_sequence_number	seq64::user_settings, 1105, 1107
seq64::keys_perform, 369	bpm_step_increment
at_snapshot_1	seq64::user_settings, 1105, 1108
seq64::keys_perform, 366	bpm_up
at_snapshot_2	seq64::keys_perform, 352
seq64::keys_perform, 366	build_details
at_song_mode	seq64, 87
seq64::keys_perform, 367	build_info_dialog
at_start	seq64::mainwnd, 426
seq64::keys_perform, 367	bus
at_stop	seq64::busarray, 141
seq64::keys_perform, 369	seq64::businfo, 150, 152
at_tap_bpm	seq64::user_settings, 1094
seq64::keys_perform, 368	bus_container
at_toggle_jack	seq64::midi_info, 519
seq64::keys_perform, 367	bus_count
at_toggle_mutes	seq64::user_settings, 1094
seq64::keys_perform, 368	bus instrument
auto_option_save	seq64::user_settings, 1094
seq64::rc_settings, 801, 805	bus_name
automutex	seq64::midibase, 588, 597
seq64::automutex, 137	seq64::user_settings, 1094
background_sequence	
	busarray 140
seq64::sequence, 1040, 1041	seq64::busarray, 140

seq64::businfo, 153	c_midi_control_bpm_page_dn
businfo	seq64, 120
seq64::businfo, 149, 150	c_midi_control_bpm_page_up
BussConstIterator	seq64, 120
seq64::user_settings, 1091	c_midi_control_bpm_up
BussIterator	seq64, 118
seq64::user_settings, 1091	c_midi_control_mod_glearn
bussbyte	seq64, 119
seq64, 64	c_midi_control_mod_gmute
Busses	seq64, 119
seq64::user_settings, 1091	c_midi_control_mod_queue
button	seq64, 119
seq64::click, 158	c_midi_control_mod_replace
seq64::options, 639	seq64, 118
button_press	c_midi_control_mod_snapshot
seq64::seqroll, 961	seq64, 118
button_press_initial	c_midi_control_play_ss
seq64::seqroll, 960	seq64, 119
button release	c_midi_control_playback
seq64::seqroll, 961	
3343 111334.311, 331	seq64, 119
c_backsequence	c_midi_control_record
seq64, 117	seq64, 119
c_bpmtag	c_midi_control_solo
seq64, 116	seq64, 119
c_chord_number	c_midi_control_ss_dn
seq64, 123	seq64, 118
c_chord_size	c_midi_control_ss_up
seq64, 123	seq64, 118
c_chord_table	c_midi_control_thru
seq64, 123	seq64, 120
c_chord_table_text	c_midi_controls
seq64, 123	seq64, 119
c_chord_text	c_midi_controls_extended
seq64, 123	seq64, 120
c_controller_names	c_midi_track_ctrl
seq64, 109	seq64, 117
c interval text	c midibus
	seq64, 114
seq64, 123	c_midibus_input_size
c_key_text	seq64, 114
seq64, 123	c_midibus_output_size
c_mainwid_x	seq64, 114
seq64, 130	c_midibus_sysex_chunk
c_mainwid_y	
seq64, 130	seq64, 114
c_max_busses	c_midich
seq64, 124	seq64, 115
c_max_instruments	c_midiclocks
seq64, 124	seq64, 116
c_midi_control_16	c_midictrl
seq64, 120	seq64, 116
c_midi_control_17	c_music_scales
seq64, 120	seq64, 68
c_midi_control_18	c_musickey
seq64, 120	seq64, 117
c_midi_control_19	c_musicscale
seq64, 120	seq64, 117
c_midi_control_bpm_dn	c_mutegroups
seq64, 118	seq64, 116
1 /	1 7 -

c_notes	seq64::perfroll, 772
seq64, 116	seq64::seqkeys, 924
c_perf_bp_mes	seq64::seqroll, 959
seq64, 117	channel_count
c_perf_bw	seq64::user_midi_bus, 1081
seq64, 117	channel_match
c_scales_policy	seq64::sequence, 1044
seq64, 121	channel_max
c_scales_text	seq64::user_midi_bus, 1081
seq64, 122	channel_string
c_scales_transpose_dn	seq64::editable_event, 176
seq64, 122	char_height
c_scales_transpose_up	seq64::font, 261
seq64, 121	char_width
c_status_queue	seq64::font, 261
seq64, 129	CharList
c_status_replace	seq64::midi_list, 547
seq64, 129	CharVector
c_status_snapshot	seq64::midi_vector, 579
seq64, 129	check_channel
c_timesig	seq64::event, 195
seq64, 116	check_queued_tick
c_transpose	seq64::sequence, 1005
seq64, 117	checklen
c_triggers	seq64::midifile, 624
seq64, 116	choose_file
c_triggers_new	seq64::mainwnd, 428
seq64, 116	choose_ppqn
calculate_base_sizes	seq64, 100
seq64::mainwid, 409	clamp
seq64::mainwid, 409 cancel_callback	clamp seq64, 105, 106
•	•
cancel_callback	seq64, 105, 106
cancel_callback seq64::midi_api, 489	seq64, 105, 106 clamp_track
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832	seq64, 105, 106 clamp_track seq64::perform, 743
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category	seq64, 105, 106 clamp_track seq64::perform, 743 clear
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_linfo, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699 clear_flags
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::eventedit, 232	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::eventedit, 232 seq64::seqedit, 888	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962 clear_link
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::eventedit, 232 seq64::seqedit, 888 change_horz	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962 clear_link seq64::event, 203
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::eventedit, 232 seq64::seqedit, 888 change_horz seq64::perfroll, 772	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962 clear_link seq64::event, 203 clear_links
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::eventedit, 232 seq64::seqedit, 888 change_horz seq64::perfroll, 772 seq64::perftime, 787	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::trinidi_info, 842 seq64::trinidi_info, 842 seq64::trinidi_info, 842 seq64::perform, 699 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962 clear_link seq64::event_list, 219
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::eventedit, 232 seq64::eventedit, 232 seq64::perffoll, 772 seq64::perfftime, 787 seq64::sequent, 910	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::trinidi_info, 842 seq64::trinidi_info, 842 seq64::perform, 699 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962 clear_link seq64::event_list, 219 clear_old seq64::seqroll, 962
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::eventedit, 232 seq64::eventedit, 232 seq64::seqedit, 888 change_horz seq64::perffine, 787 seq64::seqdata, 864 seq64::seqvent, 910 seq64::seqroll, 959	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_list, 515 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962 clear_link seq64::event, 203 clear_links seq64::event_list, 219 clear_old seq64::seqroll, 962 clear_selected
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::eventedit, 232 seq64::eventedit, 232 seq64::perffoll, 772 seq64::perfftime, 787 seq64::sequent, 910	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962 clear_link seq64::event_list, 219 clear_old seq64::seqroll, 962 clear_selected seq64::seqroll, 962
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::sequentedit, 232 seq64::seqedit, 888 change_horz seq64::perfroll, 772 seq64::seqdata, 864 seq64::seqvent, 910 seq64::seqroll, 959 seq64::seqtime, 979 change_vert	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_list, 548 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962 clear_link seq64::event_list, 219 clear_old seq64::seqroll, 962 clear_selected seq64::seqroll, 962 clear_seqente_triggers
cancel_callback seq64::midi_api, 489 seq64::rtmidi_in, 832 category seq64::editable_event, 173 category_string seq64::editable_event, 173 category_t seq64::editable_event, 169 cc_match seq64::event, 200 change_event_data_lfo seq64::sequence, 1030 change_event_data_range seq64::sequence, 1029 change_focus seq64::eventedit, 232 seq64::seqedit, 888 change_horz seq64::perfroll, 772 seq64::perftime, 787 seq64::seqdata, 864 seq64::seqvent, 910 seq64::seqroll, 959 seq64::seqtime, 979	seq64, 105, 106 clamp_track seq64::perform, 743 clear seq64::editable_events, 186 seq64::event_list, 217 seq64::midi_container, 495 seq64::midi_info, 515 seq64::midi_port_info, 562 seq64::midi_vector, 581 seq64::rtmidi_info, 842 seq64::triggers, 1068 clear_all seq64::perform, 699 clear_flags seq64::seqroll, 962 clear_link seq64::event_list, 219 clear_old seq64::seqroll, 962 clear_selected seq64::seqroll, 962

2220442224222 1010	andia disatan
seq64::sequence, 1019	config_directory
clear_window	seq64::rc_settings, 804, 809
seq64::gui_drawingarea_gtk2, 285	config_filename
click	seq64::rc_settings, 804, 809
seq64::click, 156	config_filename_alt
client	seq64::rc_settings, 804, 810
seq64::jack_assistant, 327	config_filespec
client_handle	seq64::rc_settings, 800
seq64::midi_jack, 526, 527	configfile
seq64::midi_jack_info, 540, 543	seq64::configfile, 163
client_name	connect
seq64::jack_assistant, 327	seq64::midi_jack_info, 544
client_open	connect_name
seq64::jack_assistant, 327	seq64::midi_info, 518
client_uuid	seq64::midi_port_info, 563
seq64::jack_assistant, 327	seq64::midibase, 588
clip_timestamp	connect_port
seq64::sequence, 1027	seq64::midi_jack, 528
clock	const_iterator
seq64::busarray, 142	seq64::editable_events, 181
seq64::businfo, 153	seq64::event_list, 214
seq64::mastermidibase, 450	seq64::seqmenu, 933
seq64::midibase, 596	container
clock_callback_mod	seq64::midi_message, 553
seq64::options, 641	continue_from
clock_callback_off	seq64::busarray, 142
seq64::options, 640	seq64::businfo, 153
clock_callback_on	seq64::mastermidibase, 443
seq64::options, 641	seq64::midibase, 596
clock_e	continue_sysex
seq64, 68	seq64::rtmidi_in_data, 835
clock_mod_callback	control_height
seq64::options, 641	seq64::user_settings, 1099, 1104
clock_tick_duration_bogus	controller_active
seq64, 81	seq64::user_instrument, 1076
clock_ticks_from_ppqn	seq64::user_settings, 1095
seq64, 82	controller_count
clocks_per_metronome	seq64::user_instrument, 1075
seq64::perform, 693	controller_max
seq64::sequence, 999	seq64::user_instrument, 1075
close_client	controller_name
seq64::midi_jack, 528	
· —	seq64::user_instrument, 1075
close_out	seq64::user_settings, 1096
close_out seq64::eventedit, 232	seq64::user_settings, 1096 controllers
close_out seq64::eventedit, 232 close_port	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727 Color	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773 convert_sel_box_to_rect
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727 Color seq64::font, 259	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773 convert_sel_box_to_rect seq64::seqroll, 957
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727 Color seq64::font, 259 seq64::gui_palette_gtk2, 301	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773 convert_sel_box_to_rect seq64::seqroll, 957 convert_t
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727 Color seq64::font, 259 seq64::gui_palette_gtk2, 301 combine_bytes	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773 convert_sel_box_to_rect seq64::seqroll, 957 convert_t seq64::seqevent, 911
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727 Color seq64::font, 259 seq64::gui_palette_gtk2, 301 combine_bytes seq64::perform, 696	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773 convert_sel_box_to_rect seq64::seqroll, 957 convert_t seq64::seqevent, 911 convert_tn
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727 Color seq64::font, 259 seq64::gui_palette_gtk2, 301 combine_bytes seq64::perform, 696 complete_paste	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773 convert_sel_box_to_rect seq64::seqroll, 957 convert_t seq64::seqevent, 911 convert_tn seq64::seqroll, 956
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727 Color seq64::font, 259 seq64::gui_palette_gtk2, 301 combine_bytes seq64::perform, 696 complete_paste seq64::seqroll, 954	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773 convert_sel_box_to_rect seq64::seqroll, 957 convert_t seq64::seqevent, 911 convert_tn seq64::seqroll, 956 convert_tn_box_to_rect
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727 Color seq64::font, 259 seq64::gui_palette_gtk2, 301 combine_bytes seq64::perform, 696 complete_paste seq64::seqroll, 954 condition_var	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773 convert_sel_box_to_rect seq64::seqroll, 957 convert_t seq64::seqevent, 911 convert_tn seq64::seqroll, 956 convert_tn_box_to_rect seq64::seqroll, 957
close_out seq64::eventedit, 232 close_port seq64::midi_jack, 528 collapse seq64::perfedit, 658 seq64::perform, 727 Color seq64::font, 259 seq64::gui_palette_gtk2, 301 combine_bytes seq64::perform, 696 complete_paste seq64::seqroll, 954	seq64::user_settings, 1096 controllers seq64::user_instrument_t, 1078 controllers_active seq64::user_instrument_t, 1078 convert_drop_xy seq64::perfroll, 773 convert_sel_box_to_rect seq64::seqroll, 957 convert_t seq64::seqevent, 911 convert_tn seq64::seqroll, 956 convert_tn_box_to_rect

sep84:sep4rioll, 771 sep84:seqdata, 864 sep64:seqvent, 910 convert_xy sep64:seqroll, 956 convert_y sep64:seqroll, 956 convert_y sep64:seqvent, 950 copy sep64:seqvent, 950 copy_definitions sep64:seqfiggers, 1067 copy_definitions sep64:sequence, 1016 copy_selected sep64:sequence, 1011 sep64:sequence, 1014 sep64:sequence, 1016 copy_selected sep64:sequence, 1016 cark_blue sep64:gui_palette_gtk2, 303 dark_grey sep64:sequence, 1016 cark_blue sep64:gui_palette_gtk2, 303 dark_grey sep64:sequence, 1016 cark_blue sep64:gui_palette_gtk2, 303 dark_grey sep64:sequence, 1016 cark_grey sep64:sequence, 1016 cark_blue sep64:gui_palette_gtk2, 303 dark_grey sep64:sequence, 1016 cark_grey sep64:sequence, 1016 cark_blue sep64:gui_palette_gtk2, 303 dark_grey sep64:sequence, 1016 cark_grey sep64:sequence, 1016 cark		
sep64::seqvent, 910 convert_xy	seq64::perfroll, 771	current_seq
convert_xy seq64::perfroll, 771 seq64::seqroll, 956 convert_y seq64::perframes, 670 seq64::perframes, 670 seq64::perfedit, 658 seq64::perfedit, 658 seq64::perform, 727 seq64::riggers, 1067 copy_ definitions seq64::sequence, 1041 copy_ selected seq64::sequence, 1041 copy_ selected seq64::sequence, 1041 copy_ selected seq64::sequence, 1041 copy_ selected seq64::sequence, 1065 copy_ selected seq64::sequence, 1065 copy_ selected seq64::sequence, 1016 copy_ triggers seq64::sequence, 1016 copy_ triggers seq64::sequence, 1018 count seq64::midi_ message, 554 seq64::midi_ queue, 565 seq64::midi_ queue, 565 seq64::midi_ queue, 565 seq64::midi_ queue, 565 seq64::went_list, 221 count_selected_ events seq64::perform, 757 count_selected_ events seq64::perform, 764 create_ lash, driver seq64::perform, 744 create_ menu_ image seq64::perform, 744 decrement_ data2 seq64::perform, 716 decrement_ data2 seq64::perform, 717 decrement_ control seq64::perform, 717 decrement_ control seq64::perform, 717 decrement_ seq64::perform, 717 decrement_ idex, start seq64::perform, 717 decrement_ idex seq64::perform, 716 decrement_ idex seq64::perform, 716 decrement_ idex seq64::perform,	· · · · · · · · · · · · · · · · · · ·	•
seq64::perfroll, 771 seq64::seqventslots, 249 seq64::perframes, 670 seq64::perframes, 670 seq64::perform, 727 seq64::perform, 727 seq64::perform, 727 seq64::perform, 727 seq64::perform, 727 seq64::sequence, 1016 dark_blue seq64::gui_palette_gtk2, 303 dark_oyan seq64::gui_palette_gtk2, 304 dark_green seq64::gui_palette_gtk2,		_
seq64::seqroll, 956 convert_y seq64::perfnames, 670 seq64::perfnames, 670 seq64::perfnames, 670 seq64::perfnames, 670 seq64::perform, 727 seq64::perform, 727 seq64::perform, 727 seq64::sequence, 1067 copy_definitions seq64::perform, 727 seq64::sequence, 1041 copy_selected copy_events seq64::sequence, 1041 copy_selected_triggers seq64::sequence, 1041 copy_selected_triggers seq64::sequence, 1041 copy_selected_triggers seq64::sequence, 1024 seq64::sequence, 1024 seq64::sequence, 1065 copy_triggers seq64::sequence, 1066 copy_triggers seq64::sequence, 1018 count seq64::busarray, 141 seq64::busarray, 141 seq64::celilable_events, 185 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::went_list, 221 count_selected_notes seq64::perform, 774 create_master_bus seq64::perform, 744 create_menu_image seq64::perform, 774 decrement_data2 seq64::went, 201 decrement_offset seq64::perform, 774 decrement_selected seq64::perform, 777 decrement_selected seq64::perform, 776 decrement_tick_send seq64::perform, 777 decrement_selected seq64::perform, 776 decrement_tick_send seq64::perform, 776 decrement_tick_send seq64::perform, 776 decrement_tick_send seq64::perform, 775 decrement_selected seq64::perform		
convert_y		-
seq64::eventslots, 249 seq64::sequence, 1024 cut_selected_trigger cut_selected_trigger seq64::perfedit, 658 seq64::perform, 727 seq64::perform, 727 dark_blue seq64::perform, 727 dark_cyan seq64::user_indi_bus, 1082 dark_cyan copy_definitions seq64::gui_palette_glk2, 304 seq64::user_indi_bus, 1082 dark_green copy_selected seq64::gui_palette_glk2, 303 dark_green seq64::gui_palette_glk2, 304 dark_green seq64::gui_palette_glk2, 304 dark_green seq64::gui_palette_glk2, 304 dark_green seq64::gui_palette_glk2, 303 dark_green seq64::gui_palette_glk2, 304 dark_green seq64::gui_palette_glk2, 303 dark_green seq64::gui_palette_glk2, 303 <t< td=""><td></td><td></td></t<>		
seq84::performes, 670 seq64::perdett, 588 seq64::perform, 727 seq64::triggers, 1067 seq64::perform, 727 seq64::triggers, 1067 seq64::user_instrument, 1077 seq64::user_instrument, 1077 seq64::user_instrument, 1077 seq64::user_instrument, 1077 seq64::user_instrument, 1077 seq64::user_instrument, 1077 seq64::sequence, 1041 dark_green seq64::gui_palette_gtk2, 303 dark_grey seq64::gui_palette_gtk2, 304 dark_grey seq64::gui_palette_gtk2, 304 dark_grey seq64::gui_palette_gtk2, 304 dark_grey seq64::gui_palette_gtk2, 304 dark_magenta seq64::gui_palette_gtk2, 303 dark_red seq64::gui_palette_gtk2, 304 dark_magenta seq64::gui_palette_gtk2, 303 dark_red seq64::gui_palette_gtk2, 303 dark_red seq64::gui_palette_gtk2, 304 dark_magenta seq64::gui_palette_gtk2, 303 dark_red		
seq64::seqkeys, 923 seq64::perfedit, 658	•	
copy seq64::perfedit, 658 dark_blue seq64::gui_palette_gtk2, 303 dark_cyan seq64::user_inistrument, 1077 seq64::user_inistrument, 1077 seq64::user_inistrument, 1077 seq64::gui_palette_gtk2, 304 dark_green seq64::gui_palette_gtk2, 304 dark_green seq64::gui_palette_gtk2, 303 dark_green seq64::gui_palette_gtk2, 304 dark_magenta dark_magenta seq64::gui_palette_gtk2, 304 dark_magenta dark_magenta seq64::gui_palette_gtk2, 304 dark_magenta dark_green seq64::gui_palette_gtk2, 304 dark_magenta dark_green seq64::gui_palette_gtk2, 304 dark_magenta dark_green seq64::gui_palette_gtk2, 303 dark_reen seq64::gui_palette_gtk2, 303 dark_reen seq64::gui_palette_gtk2, 303 <		
seq64::perform, 727 seq64::user_inid_pus, 1067 copy_definitions seq64::user_inid_bus, 1082 copy_events seq64::sequence, 1041 copy_selected seq64::sequence, 1024 seq64::sequence, 1016 copy_trigger seq64::sequence, 1016 copy_trigger seq64::sequence, 1018 count seq64::event_list, 215 seq64::mid_queue, 565 seq64::mid_queue, 565 seq64::event_list, 221 count_selected_neus seq64::event_list, 221 count_selected_neus seq64::perform, 722 count_selected_neus seq64::event_list, 221 count_selected_neus seq64::event_list, 221 count_selected_neus seq64::event_list, 221 count_selected_neus seq64::event_list, 221 count_selected_neus seq64::perform, 744 create_master_bus seq64::seqedit, 887 create_master_bus seq64::seqedit, 886 create_inidid_pusfer seq64::seqedit, 886 create_seqedit seq64::seqedit, 886 c		seq64::sequence, 1016
seq64::perform, 727 seq64::triggers, 1067 copy_definitions seq64::user_instrument, 1077 seq64::user_inidi_bus, 1082 seq64::gui_palette_gtk2, 304 copy_events seq64::gui_palette_gtk2, 304 seq64::user_inidi_bus, 1082 seq64::gui_palette_gtk2, 304 copy_events seq64::gui_palette_gtk2, 304 seq64::sequence, 1041 dark_magenta copy_selected seq64::gui_palette_gtk2, 304 seq64::sequence, 1016 seq64::gui_palette_gtk2, 304 copy_selected_trigger seq64::gui_palette_gtk2, 304 seq64::sequence, 1016 seq64::gui_palette_gtk2, 304 copy_selected_trigger seq64::gui_palette_gtk2, 304 seq64::sequence, 1016 seq64::gui_palette_gtk2, 304 copy_selected_trigger seq64::gui_palette_gtk2, 304 dark_magenta seq64::gui_palette_gtk2, 304 dark_roang seq64::gui_palette_gtk2, 304 dark_magenta seq64::gui_palette_gtk2, 304 dark_roang seq64::gui_palette_gtk2, 304 dark_roang seq64::gui_palette_gtk2, 304 dark_roang seq64::gui_palette_gtk2, 303 dark_roang seq64::gui_palette_gtk2		dark blue
dark_cyan seq64::ririggers, 1067 seq64::ririggers, 1067 seq64::risequence, 1041 seq64::sequence, 1041 seq64::sequence, 1024 seq64::sequence, 1065 seq64::sequence, 1065 seq64::sequence, 1065 seq64::sequence, 1065 seq64::sequence, 1016 seq64::sequence, 1016 seq64::sequence, 1018 seq64::sequence, 1018 seq64::sequence, 1018 seq64::sequence, 1018 seq64::sequence, 1018 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_message, 554 seq64::event_list, 215 seq64::event_list, 221 count_selected_events seq64::event_list, 220 create_jack_client seq64::event_list, 220 create_jack_olient seq64::seqedit, 887 create_menu image seq64::seqedit, 887 create_menu seq64::seqedit, 886 create_ringbuffer seq64::seqedit, 886 create_ringbuffer seq64::sequenu, 937 current_event seq64::seqedit seq64::sequenu, 937 current_event seq64::seqedit seq64::sequenu, 937 current_event seq64::seqedit, 244 current_screen_set_notepad seq64::eventslots, 251 decirit seq64::sequenu, 937 current_event seq64::eventslots, 244 current_screen_set_notepad seq64::gack_assistant, 321 seq64::gack_as		
seq64::user_instrument, 1077 seq64::user_initrument, 1077 seq64::sequence, 1041 dark_grey seq64::sequence, 1024 seq64::sequence, 1024 seq64::sequence, 1065 copy_triggers seq64::perform, 722 seq64::perform, 722 seq64::perform, 722 seq64::busarray, 141 seq64::ditable_events, 185 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_queue, 566 seq64::midi_queue, 566 seq64::midi_queue, 566 seq64::midi_queue, 566 seq64::midi_queue, 566 seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_lack_client seq64::perform, 716 decrement_bottom seq64::eventslots, 252 decrement_current seq64::perform, 716 decrement_clata1 seq64::eventslots, 252 decrement_clata1 seq64::eventslots, 252 decrement_data2 seq64::event, 201 decrement_data2 seq64::event, 201 decrement_flata2 seq64::event, 201 decreme		
copy_centil dark_green seq64::user_midi_bus, 1082 seq64::guj_palette_gtk2, 303 copy_events seq64::guj_palette_gtk2, 304 seq64::sequence, 1041 seq64::guj_palette_gtk2, 304 copy_selected seq64::guj_palette_gtk2, 304 seq64::sequence, 1055 seq64::guj_palette_gtk2, 303 copy_selected_trigger seq64::guj_palette_gtk2, 303 seq64::sequence, 1016 seq64::guj_palette_gtk2, 303 copy_triggers seq64::guj_palette_gtk2, 303 seq64::sequence, 1018 seq64::guj_palette_gtk2, 303 count seq64::sequence, 1018 count seq64::sequence, 1018 count seq64::seqeft.:gui_palette_gtk2, 303 datk_orange seq64::gui_palette_gtk2, 303 data_seq64::sequence, 1018 seq64::gui_palette_gtk2, 303 data_seq64::sequence, 1018 seq64::gui_palette_gtk2, 303 data_seq64::sequence, 1018 seq64::midi_gui_palette_gtk2, 303		_ ·
seq64::user_midi_bus, 1082 dark_grey copy_events seq64::sequence, 1041 copy_selected seq64::sequence, 1024 seq64::sequence, 1024 seq64::gui_palette_gtk2, 304 dark_magenta seq64::gui_palette_gtk2, 304 dark_range seq64::gui_palette_gtk2, 303 dark_red seq64::gui_palette_gtk2, 303 dark_red<		
copy_events seq64::sequence, 1041 copy_selected seq64::sequence, 1024 seq64::sequence, 1065 copy_selected_trigger seq64::sequence, 1016 copy_triggers seq64::sequence, 1016 copy_triggers seq64::sequence, 1018 count seq64::sequence, 105 seq64::midi_control, 502 data_string seq64::seqitable_event, 176 deactivate seq64::seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::midi_queue, 566 decrement_beats_per_minute seq64::perform, 716 decrement_bottom seq64::perform, 716 decrement_current seq64::perform, 716 decrement_data1 seq64::perform, 744 create_lash_driver seq64::perform, 744 create_menu_image seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_rigui_palette_gtk2, 304 dark_rad seq64::qui_palette_gtk2, 303 data seq64::perform, 502 data_string seq64::perform, 176 deactivate seq64::businfo, 152 dealtocate seq64::businfo, 152 dealtocate seq64::perform, 716 decrement_bottom seq64::perform, 716 decrement_data1 seq64::perform, 716 decrement_data1 seq64::perform, 717 decrement_data1 seq64::perform, 717 decrement_selected seq64::requence, 1031 decrement_tick_end seq64::requence, 1031 decrement_tick_end seq64::requence, 1031 decrement_tick_end seq64::requence, 1055 decrement_tick_end seq64::requence, 1056 decrement_tick_end seq64::requenc		_ -
seq64::sequence, 1041 copy_selected seq64::sequence, 1024 seq64::sequence, 1065 copy_selected_trigger seq64::sequence, 1016 copy_triggers seq64::sequence, 1016 copy_triggers seq64::sequence, 1018 count seq64::sequence, 1018 count seq64::sequence, 1018 count seq64::sequence, 1018 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 221 count_selected_notes seq64::event_list, 221 count_selected_indes seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64, 94 create_lash_driver seq64, 97 create_master_bus seq64::seqedit, 887 create_menu_image seq64::seqedit, 886 create_menu_image seq64::seqedit, 886 create_menu_image seq64::seqedit, 886 create_ingbuffer seq64::seqedit, 886 create_ingbuffer seq64::seqedit, 886 create_ingbuffer seq64::seqedit, 886 create_ingbuffer seq64::seqedit seq64::seqenenu, 937 current_event seq64::eventslots, 251 decrement_tick_end seq64::event_lots, 251 decrement_tick_end seq64::event_lots, 251 decrement_tops seq64::eventslots, 251 decrement_tick_end seq64::eventslots, 251 decrement_tops seq64::eventslots, 251 decrement_tick_end seq64::eventslots, 251 decrement_tops seq64::eventslots, 251 decrement_tops seq64::eventslots, 251 decrement_tops seq64::eventslots, 251 decrement_tick_end seq64::eventslots, 251 deinit seq64::eventslots, 251 deinit seq64::eventslots, 251 deinit seq64::eventslots, 251 deinit seq64::eventslots, 251		
copy_selected dark_magental seq64::sequence, 1024 seq64::gui_palette_gtk2, 304 dark_orange seq64::gui_palette_gtk2, 303 copy_selected_trigger dark_red seq64::sequence, 1016 seq64::gui_palette_gtk2, 303 copy_triggers data seq64::sequence, 1018 seq64::gui_palette_gtk2, 303 count seq64::sequence, 1018 count seq64::midi_control, 502 data seq64::midi_queue, 566 seq64::midi_squeue, 565 seq64::midi_queue, 566 seq64::midi_gueue, 565 <t< td=""><td></td><td>seq64::gui_palette_gtk2, 304</td></t<>		seq64::gui_palette_gtk2, 304
seq64::sequence, 1024 seq64::striggers, 1065 copy_selected_trigger seq64::sequence, 1016 copy_triggers seq64::sequence, 1018 count seq64::sequence, 1018 count seq64::sevent_list, 215 seq64::sevent_list, 215 seq64::midi_queue, 565 seq64::midi_gueue, 565 seq64::midi_gueue, 565 seq64::midi_gueue, 565 seq64::event_list, 221 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64, 94 create_lash_driver seq64, 97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 886 create_menu_image seq64::seqedit, 886 create_ingbuffer seq64::midi_jack, 528 create_seqedit seq64::midi_queue, 566 decrement_beats_per_minute seq64::perform, 716 decrement_beats_per_minute seq64::eventslots, 252 decrement_current seq64::eventslots, 252 decrement_current seq64::event, 201 decrement_data1 seq64::event, 201 decrement_offset seq64::perform, 717 decrement_screenset seq64::perform, 717 decrement_screenset seq64::perform, 717 decrement_screenset seq64::perform, 717 decrement_screenset seq64::purrent_seq64::purrent_seq64::purrent_seq64::perform, 717 decrement_screenset seq64::rigger, 1055 decrement_lick_latat seq64::rigger, 1055 decrement_lick_start seq64::rigger, 1054 decrement_lop seq64::purrent_screen_set_notepad		
seq64::triggers, 1065 copy_selected_trigger seq64::sequence, 1016 copy_friggers seq64::perform, 722 seq64::sequence, 1018 count seq64::busarray, 141 seq64::ditable_events, 185 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::event_list, 221 count_selected_notes seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64, 94 create_lash_driver seq64, 97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::midi_jack, 528 create_seqedit seq64::perform, 717 decrement_tick_end seq64::perform, 717 decrement_tick_end seq64::sequence, 1031 decrement_tick_start seq64::reventslots, 251 decrement_top seq64:sequence, 1031 decrement_top seq64::reventslots, 251 deinit seq64::reventslots, 251 decrement_top seq64::reventslots, 251 decrement_top seq64::reventslots, 251 decrement_top seq64::reventslots, 251 deinit	• • —	seq64::gui_palette_gtk2, 304
copy_selected_trigger seq64::sequence, 1016 copy_triggers seq64::perform, 722 seq64::sequence, 1018 count seq64::busarray, 141 seq64::editable_events, 185 seq64::midi_message, 554 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::event_list, 221 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64::perform, 744 create_lash_driver seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::perform, 717 decrement_seqe64::perform, 717 decrement_seq64::perform, 717 decrement_tick_seqfed: seq64::perform, 715 decrement_tick_seqfed: seq64::perform, 715 decrement_tick_seqfed: seq64::perform, 717 decrement_tick_seqfed: seq64::perform, 715 decrement_tick_seqfed: seq64::perform, 71	• •	
seq64::sequence, 1016 copy_triggers seq64::perform, 722 seq64::sequence, 1018 count seq64::busarray, 141 seq64::event, 176 seq64::midi_message, 554 seq64::midi_queue, 565 seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64::perform, 744 create_master_bus seq64::perform, 744 create_menu_image seq64::perform, 744 create_menuuimage seq64::perform, 744 create_menus seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_seqedit seq64::seqedit, 886 create_seqedit seq64::seqenenu, 937 current_event seq64::eventslots, 251 deallocate seq64::businfo, 152 deallocate seq64::businfo, 152 deallocate seq64::midi_queue, 566 decrement_bottom seq64::perform, 716 decrement_bottom seq64::perform, 716 decrement_current seq64::eventslots, 252 decrement_data1 seq64::event, 201 decrement_data2 seq64::event, 201 decrement_offset seq64::event, 201 decrement_offset seq64::perform, 744 decrement_seqed::event, 201 decrement_seqed::event, 201 decrement_seqed::event, 201 decrement_seqed::event, 201 decrement_seqed::event, 201 decrement_sequence, 1055 decrement_seqed::event, 201 decrement_seqeded seq64::rigger, 1055 decrement_tick_end seq64::rigger, 1055 decrement_tick_start seq64::rigger		
copy_triggers seq64::perform, 722 seq64::perform, 723 seq64::perform, 724 seq64::perform, 725 seq64::perform, 726 seq64::perform, 716 decrement_beats_per_minute seq64::perform, 716 decrement_bottom seq64::perform, 716 decrement_beats_per_minute seq64::perform, 716 decrement_bottom seq64::perform, 716 decrement_beats_per_minute seq64::perform, 716 decrement_sequence, 1031 decrement_sequence, 1031 decrement_tick_end seq64::perform, 717 decrement_tick_end seq64::perform, 717 decrement_tick_end seq64::perform, 715 decre		
seq64::perform, 722 seq64::sequence, 1018 count seq64::busarray, 141 seq64::editable_events, 185 seq64::event_list, 215 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64: 94 create_lash_driver seq64: 97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menu seq64::seqedit, 886 create_menus seq64::seqedit, 886 create_menus seq64::seqedit, 886 create_ringbuffer seq64::seqedit seq64::seqedit seq64::seqedit seq64::seqmenu, 937 current_event seq64::eventslots, 251 decrement_corrent seq64::perform, 717 decrement_screenset seq64::midi_queue, 566 decrement_bottom seq64::eventslots, 252 deallocate seq64::event_bottom seq64::eventslots, 252 decrement_corrent seq64::eventslots, 252 decrement_data1 seq64::event, 201 decrement_offset seq64::event, 201 decrement_screenset seq64::eventslots, 258 decrement_screenset seq64::eventslots, 258 decrement_tick_end seq64::eventslots, 251 decrement_top seq64::eventslots, 251 deinit seq64::eventslots, 251 deinit seq64::pack_assistant, 321	· · · · · · · · · · · · · · · · · · ·	seq64::gui_palette_gtk2, 303
seq64::sequence, 1018 count seq64::busarray, 141 seq64::editable_events, 185 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_message, 554 seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64, 94 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menu_image seq64::seqedit, 886 create_ringbuffer seq64::seqedit, 886 create_seqedit seq64::events, 186 cdat_seqent seq64::eventslots, 258 decrement_current seq64::perform, 774 decrement_offset seq64::perform, 717 decrement_selected seq64::perform, 717 decrement_selected seq64::perform, 717 decrement_selected seq64::perform, 717 decrement_selected seq64::midi_qoueue, 566 deactivate seq64::midi_queue, 566 decrement_beats_per_minute seq64::perform, 716 decrement_beats_per_minute seq64::perform, 716 decrement_beats_per_minute seq64::eventslots, 252 decrement_current seq64::eventslots, 252 decrement_current seq64::eventslots, 252 decrement_data1 seq64::eventslots, 252 decrement_data1 seq64::event, 201 decrement_data2 create_ment_offset seq64::event, 201 decrement_offset seq64::event, 201 decrement_seqefit seq64::perform, 717 decrement_selected seq64::perform, 716 decrement_data1 seq64::perfor		
count seq64::busarray, 141 seq64::editable_events, 185 seq64::event_list, 215 seq64::midi_queue, 565 seq64::midi_queue, 565 seq64::midi_gueue, 565 seq64::event_list, 221 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64, 94 create_lash_driver seq64, 97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menu_image seq64::seqedit, 886 create_ringbuffer seq64::seqedit, 886 create_seqedit seq64::seqmenu, 937 current_event seq64::seqnenu, 937 current_event seq64::eventslots, 252 decrement_data1 seq64::event, 201 decrement_offset seq64::event, 201 decrement_offset seq64::seqedit, 886 decrement_screenset seq64::event, 201 decrement_ticket seq64::editable_event, 176 deactivate seq64::midi_queue, 566 decrement_beats_per_minute seq64::eventslots, 252 decrement_current seq64::eventslots, 252 decrement_data1 seq64::event, 201 decrement_offset seq64::event, 201 decrement_offset seq64::event, 201 decrement_screenset seq64::equence, 1055 decrement_selected seq64::sequence, 1031 decrement_tick_end seq64::sequence, 1055 decrement_tick_start seq64::rigger, 1055 decrement_tick_start seq64::rigger, 1055 decrement_tick_start seq64::rigger, 1055 decrement_tick_start seq64::rigger, 1054 decrement_lop seq64::eventslots, 244 current_screen_set_notepad		seq64::event, 205
seq64::busarray, 141 seq64::editable_events, 185 seq64::event_list, 215 seq64::midi_message, 554 seq64::midi_queue, 565 seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64.94 create_lash_driver seq64.97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_queue, 566 deactivate seq64::midi_queue, 566 decrement_beats_per_minute seq64::perform, 716 decrement_bottom seq64::eventslots, 252 decrement_current seq64::eventslots, 252 decrement_data1 seq64::eventslots, 252 decrement_data2 seq64::event, 201 decrement_data2 seq64::event, 201 decrement_offset seq64::seqedit, 887 decrement_screenset seq64::perform, 717 decrement_screenset seq64::sequence, 1031 decrement_tick_end seq64::ringger, 1055 decrement_tick_end seq64::ringger, 1055 decrement_tick_start seq64::ringger, 1054 decrement_tick_start seq64::ringger, 1054 decrement_tick_start seq64::ringger, 1054 decrement_top	•	seq64::midi_control, 502
seq64::editable_events, 185 seq64::event_list, 215 seq64::midi_message, 554 seq64::midi_queue, 565 seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64.94 create_lash_driver seq64.97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::seqqedit seq64::seqmenu, 937 current_event seq64::eventslots, 228 decrement_current seq64::perform, 774 decrement_data2 seq64::perform, 774 decrement_screenset seq64::sequence, 1031 decrement_selected seq64::sequence, 1031 decrement_tick_end seq64::sequence, 1055 decrement_tick_end seq64::sequence, 1054 decrement_tick_start seq64::seqdeitseleevents, 186 current_index seq64::eventslots, 244 current_screen_set_notepad		data_string
seq64::event_list, 215 seq64::midi_message, 554 seq64::midi_queue, 565 seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64::event_list, 220 create_lash_driver seq64::perform, 744 create_menu_image seq64::event_value create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::seqmenu, 937 current_event seq64::eventslots, 252 decrement_data1 seq64::sequent_data1 decrement_data2 seq64::event, 201 decrement_offset seq64::seqedit, 886 decrement_selected seq64::seqedit, 886 create_ringbuffer seq64::seqedit seq64::seqmenu, 937 current_event seq64::editable_events, 186 current_index seq64::eventslots, 244 current_screen_set_notepad seq64::pack_assistant, 321	•	seq64::editable_event, 176
seq64::midi_message, 554 seq64::midi_queue, 565 seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64, 94 create_lash_driver seq64, 97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_qack, 528 create_seqedit seq64::midi_queue, 566 decrement_beats_per_minute seq64::perform, 716 decrement_bottom seq64::perform, 252 decrement_current seq64::eventslots, 252 decrement_data1 seq64::eventslots, 252 decrement_data1 seq64::event_data1 seq64::event_data2 seq64::event_data2 seq64::event_data2 seq64::reate_mater_bus seq64::perform, 744 decrement_offset seq64::rigger, 1055 decrement_screenset seq64::perform, 717 decrement_screenset seq64::perform, 717 decrement_screenset seq64::sequence, 1031 decrement_tick_end seq64::seqedit seq64::midi_jack, 528 decrement_tick_end seq64::rigger, 1055 decrement_tick_start seq64::editable_events, 186 current_event seq64::editable_events, 186 current_index seq64::eventslots, 244 current_screen_set_notepad seq64::jack_assistant, 321	•	deactivate
seq64::midi_queue, 565 seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64, 94 create_lash_driver seq64::perform, 744 create_master_bus seq64::perform, 744 create_menu_image seq64::event_offset create_menus seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::seqmenu, 937 current_event seq64::eventslots, 252 decrement_current seq64::eventslots, 252 decrement_data1 seq64::event, 201 decrement_data2 seq64::event, 201 decrement_offset seq64::rigger, 1055 decrement_screenset seq64::seqedit, 886 decrement_screenset seq64::sequence, 1031 decrement_tick_end seq64::rigger, 1055 decrement_tick_start seq64::rigger, 1054 decrement_top seq64::rigger, 1054 decrement_top seq64::eventslots, 244 current_screen_set_notepad seq64::pack_assistant, 321	· —	seq64::businfo, 152
seq64::midi_splitter, 571 count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64::event_data1 create_lash_driver seq64::perform, 744 create_menu_image seq64::event_seq64::event, 201 seq64::event_master_bus seq64::event, 201 decrement_offset create_menu_image seq64::event, 201 decrement_offset create_menus seq64::perform, 744 create_menus seq64::perform, 717 seq64::seqedit, 887 create_menus seq64::perform, 717 seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::seqmenu, 937 current_event seq64::ditable_events, 186 current_index seq64::perform, 1954 decrement_tick_start seq64::eventslots, 251 decrement_top seq64::eventslots, 251 decrement_top seq64::perform, 715 seq64::eventslots, 244 current_screen_set_notepad decrement_top seq64::perform, 715 decrement_tick_start seq64::rigger, 1055 decrement_tick_start seq64::rigger, 1054 decrement_top seq64::eventslots, 251 deinit seq64::perform, 715		deallocate
count_selected_events seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64::eventslots, 252 seq64::eventslots, 252 create_lash_driver seq64::event, 201 seq64::perform, 744 decrement_offset create_menu_image seq64::seqedit, 887 decrement_screenset create_menus seq64::perform, 717 seq64::seqedit, 886 decrement_selected create_ringbuffer seq64::midi_jack, 528 decrement_tick_end seq64::seqmenu, 937 decrement_tick_start seq64::seqmenu, 937 decrement_tick_start seq64::ditable_events, 186 decrement_top seq64::eventslots, 244 decrement_screen_set_notepad seq64::jack_assistant, 321	. —.	seq64::midi_queue, 566
seq64::event_list, 221 count_selected_notes seq64::event_list, 220 create_jack_client seq64::eventslots, 252 seq64, 94 create_lash_driver seq64::event, 201 seq64, 97 create_master_bus seq64::event, 201 seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::sequence, 1031 seq64::midi_jack, 528 create_seqedit seq64::ringger, 1055 seq64::sequenu, 937 create_seq64::sequenue, 1055 seq64::sequenue, 1055 seq64::ringger, 1055 decrement_selected create_seqedit seq64::sequence, 1031 seq64::ringger, 1055 decrement_tick_end create_seq64::ringger, 1055 seq64::ringger, 1055 decrement_tick_start seq64::ringger, 1054 decrement_top seq64::eventslots, 251 decrement_top seq64::eventslots, 251 deinit current_screen_set_notepad		decrement_beats_per_minute
count_selected_notes seq64::event_list, 220 create_jack_client seq64, 94 create_lash_driver seq64::event, 201 seq64, 97 create_master_bus seq64::event, 201 seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::sequent, 2031 seq64::midi_jack, 528 create_seqedit seq64::midi_jack, 528 create_seqedit seq64::seqmenu, 937 current_event seq64::event, 201 decrement_offset decrement_offset seq64::perform, 717 decrement_screenset decrement_screenset decrement_screenset decrement_screenset decrement_selected seq64::sequence, 1031 seq64::sequence, 1031 decrement_tick_end create_seqedit seq64::rigger, 1055 decrement_tick_start current_event seq64::rigger, 1054 decrement_top seq64::eventslots, 244 deinit current_screen_set_notepad seq64::jack_assistant, 321		seq64::perform, 716
seq64::event_list, 220 create_jack_client seq64, 94 create_lash_driver seq64, 97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::menu_image seq64::menu_tick_end seq64::menu_tick_start seq64::ditable_events, 186 current_index seq64::eventslots, 244 current_screen_set_notepad decrement_current seq64::event, 201 decrement_offset seq64::rigger, 1055 decrement_screenset seq64::requence, 1055 decrement_selected seq64::rigger, 1055 decrement_tick_end seq64::rigger, 1055 decrement_tick_end seq64::rigger, 1055 decrement_tick_start seq64::rigger, 1054 decrement_top seq64::eventslots, 251 deinit seq64::jack_assistant, 321	· —	decrement_bottom
create_jack_client seq64, 94 create_lash_driver seq64, 97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::midi_jack, 528 create_seqedit seq64::seqmenu, 937 current_event seq64::editable_events, 186 current_screen_set_notepad cereate_seqedit, 201 decrement_offset seq64::rigger, 1055 decrement_screenset seq64::perform, 717 decrement_selected seq64::sequence, 1031 decrement_tick_end seq64::rigger, 1055 decrement_tick_start seq64::rigger, 1055 decrement_tick_start seq64::rigger, 1054 decrement_top seq64::eventslots, 251 deinit seq64::jack_assistant, 321		seq64::eventslots, 252
seq64, 94 create_lash_driver seq64::event, 201 seq64, 97 create_master_bus seq64::event, 201 seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::seqmenu, 937 current_event seq64::eventslots, 244 current_screen_set_notepad decrement_data2 seq64::event, 201 decrement_offset seq64::trigger, 1055 decrement_screenset seq64::reqnent_screenset seq64::sequence, 1031 decrement_tick_end seq64::trigger, 1055 decrement_tick_start seq64::trigger, 1054 decrement_top seq64::eventslots, 251 deinit seq64::jack_assistant, 321	· —	decrement_current
create_lash_driver seq64, 97 create_master_bus seq64::event, 201 seq64::event, 201 seq64::event, 201 seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::seqmenu, 937 current_event seq64::eventslots, 244 current_screen_set_notepad seq64::jack_assistant, 321	 -	seq64::eventslots, 252
seq64, 97 create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::seqmenu, 937 current_event seq64::editable_events, 186 current_screen_set, 105 seq64::editable_set, 186 current_screen_set, 105 decrement_screenset seq64::ringpuffer seq64::sequence, 1031 decrement_tick_end seq64::ringger, 1055 decrement_tick_start seq64::ringger, 1055 decrement_tick_start seq64::ringger, 1054 decrement_top seq64::eventslots, 244 decrement_top seq64::eventslots, 244 current_screen_set_notepad seq64::jack_assistant, 321	•	decrement_data1
create_master_bus seq64::perform, 744 create_menu_image seq64::seqedit, 887 create_menus seq64::seqedit, 886 create_ringbuffer seq64::midi_jack, 528 create_seqedit seq64::seqmenu, 937 current_event seq64::editable_events, 186 current_screen_set seq64::requence, 1031 decrement_tick_end seq64::trigger, 1055 decrement_tick_start seq64::trigger, 1054 decrement_tick_start seq64::editable_events, 186 current_index seq64::eventslots, 244 current_screen_set_notepad seq64::jack_assistant, 321		•
seq64::perform, 744 create_menu_image	•	decrement_data2
create_menu_image seq64::trigger, 1055 seq64::seqedit, 887 decrement_screenset create_menus seq64::perform, 717 seq64::seqedit, 886 decrement_selected create_ringbuffer seq64::midi_jack, 528 decrement_tick_end create_seqedit seq64::trigger, 1055 seq64::seqmenu, 937 decrement_tick_start current_event seq64::trigger, 1054 seq64::editable_events, 186 decrement_top current_index seq64::eventslots, 244 current_screen_set_notepad seq64::jack_assistant, 321		•
seq64::seqedit, 887 create_menus	• •	-
create_menus seq64::perform, 717 seq64::seqedit, 886 decrement_selected create_ringbuffer seq64::sequence, 1031 seq64::midi_jack, 528 decrement_tick_end create_seqedit seq64::trigger, 1055 seq64::seqmenu, 937 decrement_tick_start current_event seq64::trigger, 1054 seq64::editable_events, 186 decrement_top current_index seq64::eventslots, 241 current_screen_set_notepad seq64::jack_assistant, 321	-	
seq64::seqedit, 886 create_ringbuffer seq64::sequence, 1031 seq64::midi_jack, 528 create_seqedit seq64::seqmenu_tick_end seq64::trigger, 1055 seq64::seqmenu, 937 decrement_tick_start current_event seq64::ditable_events, 186 current_index seq64::eventslots, 244 current_screen_set_notepad decrement_selected seq64::sequence, 1031 decrement_tick_end seq64::trigger, 1055 decrement_tick_start seq64::eventslots, 251 deinit seq64::eventslots, 251	·	-
create_ringbuffer seq64::sequence, 1031 seq64::midi_jack, 528 decrement_tick_end create_seqedit seq64::trigger, 1055 seq64::seqmenu, 937 decrement_tick_start current_event seq64::trigger, 1054 seq64::editable_events, 186 decrement_top current_index seq64::eventslots, 251 seq64::eventslots, 244 deinit current_screen_set_notepad seq64::jack_assistant, 321		• •
seq64::midi_jack, 528 create_seqedit	·	-
create_seqedit seq64::trigger, 1055 seq64::seqmenu, 937 decrement_tick_start current_event seq64::trigger, 1054 seq64::editable_events, 186 decrement_top current_index seq64::eventslots, 251 seq64::eventslots, 244 deinit current_screen_set_notepad seq64::jack_assistant, 321	— -	
seq64::seqmenu, 937 current_event	•	
current_eventseq64::trigger, 1054seq64::editable_events, 186decrement_topcurrent_indexseq64::eventslots, 251seq64::eventslots, 244deinitcurrent_screen_set_notepadseq64::jack_assistant, 321	- ·	
seq64::editable_events, 186 current_index seq64::eventslots, 244 current_screen_set_notepad decrement_top seq64::eventslots, 251 deinit seq64::jack_assistant, 321	·	<u> </u>
current_index seq64::eventslots, 251 seq64::eventslots, 244 deinit current_screen_set_notepad seq64::jack_assistant, 321	_	
seq64::eventslots, 244 deinit current_screen_set_notepad seq64::jack_assistant, 321	•	_ ·
current_screen_set_notepad seq64::jack_assistant, 321	_	•
	•	
seq64::perform, /30 deinit_in	·	
	seqo4::periorm, 730	aeinii_in

seq64::midibase, 594	seq64::eventslots, 250
deinit_jack_transport	draw_events_on
seq64::perform, 739	seq64::seqdata, 864
del_selected_trigger	seq64::seqevent, 910
seq64::sequence, 1016	seg64::segroll, 958
del_trigger	draw_events_on_pixmap
seq64::sequence, 1013	seq64::seqdata, 865
delete_api	seq64::seqevent, 908
seq64::rtmidi, 828	seq64::seqroll, 952
seq64::rtmidi info, 847	draw key
delete_current_event	seq64::seqkeys, 923
seq64::eventslots, 247	draw_line
delete_current_sequence	seq64::gui_drawingarea_gtk2, 286–288
seq64::seqmenu, 936	draw_line_on_pixmap
delete_lash_driver	seq64::gui_drawingarea_gtk2, 286, 287
seq64, 98	draw_line_on_window
delete_sequence	seq64::seqdata, 862
seq64::perform, 701	draw_marker_on_sequence
delta_time_us_to_ticks	seg64::mainwid, 406
seq64, 80	draw_normal_rectangle_on_pixmap
device_ignore	seq64::gui drawingarea gtk2, 293
seq64::rc_settings, 803, 807	draw_pixmap_on_window
device_ignore_num	seq64::mainwid, 405
seq64::rc_settings, 803, 807	seq64::perftime, 789
disconnect	seq64::seqdata, 865
seq64::midi_jack_info, 544	seq64::seqevent, 908
display_name	seq64::seqtime, 978
seq64::midibase, 588, 597	draw_progress
divisions	seq64::perfroll, 770
seq64::midi_measures, 551	draw_progress_on_window
do_action	seq64::perftime, 787
seq64::seqedit, 887	seq64::seqroll, 953
do_input	seq64::seqtime, 978
seq64::rtmidi_in_data, 835	draw_rectangle
done	seq64::gui drawingarea gtk2, 290–292
seq64::midi_container, 494	draw_rectangle_on_pixmap
seq64::midi_list, 548	seq64::gui_drawingarea_gtk2, 292, 293
seq64::midi_vector, 580	draw_selection_on_window
double_ticks_from_ppqn	seq64::seqevent, 908
seq64, 82	seq64::seqroll, 953
draw_all	draw sequence
seq64::perfroll, 770	seq64::perfnames, 670
draw_area	draw_sequence_on
seq64::seqkeys, 923	seq64::perfroll, 772
draw background	draw_sequence_on_pixmap
seq64::perftime, 787	seq64::mainwid, 407
• •	•
seq64::seqevent, 908 draw_background_on	draw_sequence_pixmap_on_window seq64::mainwid, 408
seq64::perfroll, 772	•
draw_background_on_pixmap	draw_sequences seq64::perfedit, 659
seq64::seqroll, 952	seq64::perfnames, 670
draw_drawable	draw_sequences_on_pixmap
seq64::gui_drawingarea_gtk2, 294	seq64::mainwid, 408
draw_drawable_row	•
seq64::perfroll, 772	draw_type_t
·	seq64, 69 dref
draw_event	
seq64::eventslots, 249	seq64::editable_events, 184, 185
draw_events	seq64::event_list, 218

drop_action	EVENT_MIDI_SYSEX
seq64::seqroll, 965	seq64, 111
drop_event	EVENT_MIDI_TUNE_SELECT
seq64::seqevent, 909	seq64, 112
drop x	EVENT_NOTE_OFF
seq64::gui_drawingarea_gtk2, 285	
	seq64, 110
drop_y	EVENT_NOTE_ON
seq64::gui_drawingarea_gtk2, 285	seq64, 110
dump_midi_input	EVENT_NULL_CHANNEL
seq64::mastermidibase, 445	seq64, 113
dump_setting_summary	EVENT_PITCH_WHEEL
seq64::userfile, 1120	seq64, 110
dump_summary	EVENT_PROGRAM_CHANGE
seq64::user_settings, 1104	seq64, 110
	EVENT STATUS BIT
EVENT_AFTERTOUCH	seq64, 109
seq64, 110	•
EVENT ANY	EVENTS_ALL
seq64, 109	seq64, 114
EVENT_CHANNEL_PRESSURE	EVENTS_UNSELECTED
seq64, 110	seq64, 114
EVENT CLEAR CHAN MASK	edit_action_t
	seq64, 69
seq64, 113	edit_callback_notepad
EVENT_CONTROL_CHANGE	seq64::mainwnd, 422
seq64, 110	edit_field_has_focus
EVENT_GET_CHAN_MASK	seq64::mainwnd, 429
seq64, 113	•
EVENT_MIDI_ACTIVE_SENS	editable_event
seq64, 113	seq64::editable_event, 170, 171
EVENT_MIDI_CLOCK	editable_events
seq64, 112	seq64::editable_events, 182
EVENT_MIDI_CONTINUE	seq64::event_list, 222
seq64, 112	empty
EVENT_MIDI_META	seq64::event_list, 215
	seq64::midi_message, 554
seq64, 113	seq64::midi queue, 565
EVENT_MIDI_QUARTER_FRAME	end
seq64, 111	
EVENT_MIDI_RESET	seq64::editable_events, 184
seq64, 113	seq64::event_list, 215
EVENT_MIDI_SONG_F4	enqueue_draw
seq64, 111	seq64::eventedit, 229
EVENT MIDI SONG F5	seq64::eventslots, 249
seq64, 111	seq64::perfedit, 654
EVENT MIDI SONG F9	seq64::perfnames, 670
seq64, 112	seq64::perfroll, 773
EVENT_MIDI_SONG_FD	seq64::perftime, 787
seq64, 113	enregister
•	seq64::perform, 695
EVENT_MIDI_SONG_POS	• •
seq64, 111	enregister_peer
EVENT_MIDI_SONG_SELECT	seq64::perfedit, 655
seq64, 111	enregister_perfedits
EVENT_MIDI_START	seq64::mainwnd, 424
seq64, 112	errdump
EVENT_MIDI_STOP	seq64::midifile, 632
seq64, 112	error
EVENT_MIDI_SYSEX_CONTINUE	seq64::midi_api, 488
seq64, 112	seq64::midi_info, 518
EVENT_MIDI_SYSEX_END	error_is_fatal
seq64, 112	seq64::midifile, 621
3040 i, i i L	ooqo minamo, oz i

error_message	seq64::perform, 697
seq64, 88	fast_forward
seq64::jack_assistant, 330	seq64::keys_perform, 360
seq64::midifile, 621	seq64::perfedit, 656
seq64::optionsfile, 648	seq64::perform, 699
event	ff_rw_button_t
seq64::event, 192	seq64::perform, 690
event_count	ff_rw_type
seq64::eventslots, 244	seq64::perform, 698
seq64::sequence, 995	fg_color
event_edit	seq64::gui_palette_gtk2, 306
seq64::keys_perform, 358	file_access
event_in_range	seq64, 89
seq64::sequence, 1041	file accessible
event_key	seq64, 90
seq64::event_list::event_key, 210	file_executable
event list	seq64, 91
seq64::event_list, 214	file_exists
event_name	seq64, 89
seq64::editable_event::name_value_t, 638	file_exit
event value	
seq64::editable_event::name_value_t, 638	seq64::mainwnd, 428
EventStack	file_import_dialog
seq64::sequence, 992	seq64::mainwnd, 426
eventedit	file_is_directory
seq64::eventedit, 227	seq64, 91
seq64::eventslots, 255	file_new
Events	seq64::mainwnd, 425
seq64::editable_events, 181	file_open
seq64::eunable_events, 101 seq64::event_list, 213	seq64::mainwnd, 425
_	file_readable
events	seq64, 90
seq64::editable_events, 184	file_save
seq64::event_list, 221 seq64::sequence, 994	seq64::mainwnd, 425
•	file_save_as
EventsPair	seq64::mainwnd, 427
seq64::editable_events, 181	file_writable
seq64::event_list, 213	seq64, 90
eventslots	filename
seq64::editable_events, 187	seq64::rc_settings, 803, 808
seq64::eventedit, 235	fill
seq64::eventslots, 243	seq64::midi_container, 493
expand	fill_background_pixmap
seq64::perfedit, 658	seq64::perfroll, 769
seq64::perform, 727	fill background window
extract_bus_name	seq64::mainwid, 405
seq64, 84	fill_meta_track_end
extract_names	seq64::midi container, 497
seq64::midi_jack_info, 544	fill_proprietary
extract_port_name	seq64::midi_container, 497
seq64, 85	fill seg name
extract_port_names	seq64::midi_container, 497
seq64, 84	
extract_timing_numbers	fill_seq_number
seq64, 71	seq64::midi_container, 496
EE DW !	fill_time_sig_and_tempo
FF_RW_timeout	seq64::midi_container, 497
seq64, 105	fill_top_bar
seq64::perform, 697	seq64::seqedit, 885
FF rewind	filter_by_channel

seq64::mastermidibase, 440, 441	seq64::perform, 693
seq64::perform, 694	seq64::sequence, 999
seq64::rc_settings, 802, 807	get_all_port_info
filter_callback	seq64::midi_alsa_info, 479
seq64::options, 641	seq64::midi_info, 519
finish	seq64::midi_jack_info, 543
seq64::perform, 702	seq64::rtmidi_info, 844
first_message	get_api
seq64::rtmidi_in_data, 835	seq64::rtmidi, 827, 828
flush	get_api_info
seq64::mastermidibase, 444	seq64::rtmidi_info, 846
seq64::midibase, 595	get_beat_width
follow_progress	seq64::jack_assistant, 319
seq64::perfroll, 770	seq64::perform, 693
seq64::seqroll, 954 follow transport	seq64::sequence, 999
- ·	get_beats_per_bar
seq64::keys_perform, 360 font	seq64::perform, 692
	seq64::sequence, 998
seq64::font, 259 font render	get_beats_per_measure
seq64, 102	seq64::jack_assistant, 320
force draw	get_beats_per_minute
seq64::gui drawingarea gtk2, 285	seq64::jack_assistant, 320
seq64::seqevent, 909	seq64::mastermidibase, 441
seq64::seqkeys, 922	seq64::perform, 705
seq64::seqroll, 954	get_bpm
format_timestamp	seq64::mastermidibase, 441
seq64::editable_event, 175	get_bus_id
front	seq64::midi_info, 517
seq64::midi_queue, 566	seq64::midi_port_info, 562
FruityPerfInput	seq64::midibase, 588
seq64::FruityPerfInput, 265	seq64::rtmidi, 826
seq64::perfroll, 777	seq64::rtmidi_info, 842
seq64::triggers, 1071	get_bus_index
FruitySeqEventInput	seq64::midibase, 588
seq64::FruitySeqEventInput, 269	get_bus_name
seq64::seqevent, 914	seq64::midi_info, 517
FruitySeqRollInput	seq64::midi_port_info, 562
seq64::FruitySeqRollInput, 272	seq64::rtmidi, 826
seq64::seqkeys, 927	seq64::rtmidi_info, 843
seq64::seqroll, 970	get_channel
full	seq64::event, 195
seq64::midi_queue, 565	get_client
full_port_count	seq64::midi_alsa, 468
seq64::midi_info, 515	seq64::midibus, 608 get_clipboard_box
seq64::rtmidi, 827	seq64::sequence, 1026
seq64::rtmidi_info, 843	
e Processing and Processing	get_clock
g_midi_control_limit	seq64::busarray, 144
seq64, 121	seq64::mastermidibase, 449 seq64::midibase, 590
g_rc_settings	get_clock_mod
seq64, 130	seq64::midibase, 593
g_user_settings	•
seq64, 130	get_compiled_api
get	seq64::rtmidi_info, 841 get_current_jack_position
seq64::midi_container, 495 seq64::midi_list, 548	seq64, 96
seq64::midi_vector, 580	•
get_32nds_per_quarter	seq64::jack_assistant, 334 seq64::perform, 750
gor_oziluo_pei_qualtel	sequ r penonn, 750

get_current_sequence	seq64::sequence, 1018
seq64::seqmenu, 935	get_maximum
get_data	seq64::triggers, 1067
seq64::event, 201	get_measures
get_editing	seq64::seqedit, 882
seq64::sequence, 1001	seq64::sequence, 998
get_error_message	get_message
seq64::configfile, 164	seq64::rterror, 820
get_follow_transport	get_midi_bus
seq64::jack_assistant, 326	seq64::sequence, 1019
seq64::perform, 698	get_midi_bus_name
get_group_mute_state	seq64::busarray, 144
seq64::perform, 710	get_midi_channel
get_hold_undo	seq64::sequence, 1007
seq64::sequence, 996	get_midi_event
get_input	seq64::busarray, 147
seq64::busarray, 146	seq64::mastermidibase, 447
seq64::mastermidibase, 448	seq64::midibase, 593
seq64::midi_info, 517	get_midi_in_bus_name
seq64::midi_port_info, 562	seq64::mastermidibase, 446
seq64::midibase, 591	get_midi_out_bus_name
seq64::perform, 746	seq64::mastermidibase, 445
seq64::rtmidi_info, 843	get_minmax_note_events
get_jack_client_info	seq64::sequence, 1037
seq64::jack_assistant, 328	get_name
get_jack_mode	seq64::sequence, 1001
seq64::jack_assistant, 325	get_next_event
get_jack_pos	seq64::sequence, 1037, 1038
seq64::jack_assistant, 325	get_next_note_event
get_jack_stop_tick	seq64::sequence, 1035
seq64::jack_assistant, 325	get_next_trigger
get_jack_tick	seq64::sequence, 1038
seq64::jack_assistant, 325	get_note
seq64::perform, 702	seq64::event, 205
get_key_events	get_note_velocity
seq64::keys_perform, 362	seq64::event, 206
seq64::perform, 712	get_num_in_buses
get_key_events_rev	seq64::mastermidibase, 440
seq64::keys_perform, 363	get_num_out_buses
seq64::perform, 712	seq64::mastermidibase, 440
get_key_groups	get_num_selected_events
seq64::keys_perform, 362	seq64::sequence, 1023
seq64::perform, 712	get_num_selected_notes
get_key_groups_rev	seq64::sequence, 1023
seq64::keys_perform, 363	get_offset
seq64::perform, 712	seq64::perform, 711
get_keys	get_playing
seq64::keys_perform, 352	seq64::sequence, 1004
get_last_tick	get_playing_screenset
seq64::sequence, 1003	seq64::perform, 725
get_left_tick	get_port
seq64::perform, 703	seq64::midi_alsa, 469
get_length	seq64::midibus, 608
seq64::sequence, 1003	get_port_count
get_linked	seq64::midi_info, 516
seq64::event, 203	seq64::midi_port_info, 562
get_max_trigger	seq64::rtmidi, 827
seq64::perform, 727	seq64::rtmidi_info, 843

get_port_id	get_system
seq64::midi_info, 517	seq64::midi_info, 517
seq64::midi_port_info, 562	seq64::midi_port_info, 563
seq64::midibase, 589	seq64::rtmidi_info, 844
seq64::rtmidi, 827	get thru
seq64::rtmidi_info, 843	seq64::sequence, 1006
get_port_name	get_tick
seq64::midi_info, 517	seq64::perform, 702
seq64::midi_port_info, 562	get_timestamp
seq64::rtmidi, 827	seq64::event, 195
seq64::rtmidi_info, 843	get_toggle_jack
get_ppqn	seq64::perfedit, 655
seq64::jack_assistant, 319	seq64::perform, 696
seq64::mastermidibase, 441	get_transposable
seq64::sequence, 998	seq64::sequence, 1001
get_quantized_rec	get_transpose
seq64::sequence, 1005	seq64::perform, 705
get_queue_number	get_trigger_count
seq64::midi_port_info, 563	seq64::sequence, 994
get_queued	get_trigger_offset
seq64::sequence, 1004	seq64::sequence, 1019
get_queued_tick	get_trigger_paste_tick
seq64::sequence, 1004	seq64::sequence, 994
get_raise	seq64::triggers, 1070
seq64::sequence, 1002	get_trigger_state
get_rank	seq64::sequence, 1013
seq64::event, 207	get_triggers
get_recording	seq64::sequence, 1014
seq64::sequence, 1005	get_version
get_right_tick	seq64::rtmidi_info, 841
seq64::perform, 704	get_virtual
get_screen_set_notepad	seq64::midi_info, 517
seq64::perform, 730	seq64::midi_port_info, 563
get screenset	seq64::rtmidi_info, 844
seq64::perform, 725	getType
get_selected_box	seq64::rterror, 820
seg64::segroll, 958	global_queue
seq64::sequence, 1025	seq64::midi info, 518, 519
	seq64::rtmidi_info, 844
get_selected_end	· —
seq64::triggers, 1067	global_seq_feature
get_selected_start	seq64::user_settings, 1099
seq64::triggers, 1066	gmute_tracks
get_sequence	seq64::user_settings, 1097
seq64::mastermidibase, 441	green
seq64::perform, 718, 719	seq64::gui_palette_gtk2, 305
seq64::seqmenu, 935	grey
get_song_mute	seq64::gui_palette_gtk2, 304
seq64::sequence, 1000	grid_brackets
get_start_tick	seq64::user_settings, 1096, 1102
seq64::perform, 703	grid_is_black
get_state	seq64::user_settings, 1096
seq64::triggers, 1064	grid_is_normal
get_status	seq64::user_settings, 1096
seq64::event, 200	grid_is_white
get_sysex	seq64::user_settings, 1096
seq64::event, 202	grid_style
get_sysex_size	seq64::user_settings, 1096, 1102
seq64::event, 203	group_learn

seq64::keys_perform, 357	handle_motion_key
group_off	seq64::AbstractPerfInput, 135
seq64::keys_perform, 356, 357	seq64::FruityPerfInput, 268
group_on	seq64::Seq24PerfInput, 854
seq64::keys_perform, 356	handle_save
grow	seq64::eventedit, 233
seq64::perfedit, 658	handle_signal
seq64::triggers, 1063	seq64::mainwnd, 422
grow_edit_t	have_redo
seq64::triggers, 1059	seq64::perform, 726
grow_selected	seq64::sequence, 996
seq64::sequence, 1032	have_undo
grow_selected_notes	seq64::perform, 726
seq64::seqroll, 960	seq64::sequence, 996
grow_trigger	height
seq64::sequence, 1013	seq64::gui_drawingarea_gtk2::rect, 817
growing	seq64::rect, 817
seq64::seqroll, 965	help_check
gs_mainwid_pointer	seq64, 85
seq64, 130	seq64::rc_settings, 811
gs perfedit pointer 0	highlight
seq64, 131	seq64::perform, 718
•	home_config_directory
gs_perfedit_pointer_1	
seq64, 131	seq64::rc_settings, 810
Gtk, 49	horizontal_adjust
gtk_drawarea_init	seq64::perfroll, 773
seq64::gui_drawingarea_gtk2, 295	seq64::seqedit, 881
gui	seq64::seqroll, 954
seq64::perform, 694	horizontal_set
gui_assistant	seq64::perfroll, 774
seq64::gui_assistant, 276	seq64::seqedit, 881
gui_assistant_gtk2	idlo progress
seq64::gui_assistant_gtk2, 278	idle_progress seq64::maintime, 397
gui_drawingarea_gtk2	
seq64::gui_drawingarea_gtk2, 283	seq64::perftime, 789
gui_palette_gtk2	seq64::seqroll, 958
seq64::gui_palette_gtk2, 302	seq64::seqtime, 980
gui_window_gtk2	idle_redraw
seq64::gui_window_gtk2, 311	seq64::seqdata, 862
le and le and all	seq64::seqevent, 909
handle_cancel	seq64::seqroll, 958
seq64::eventedit, 233	ignore_flags
handle_close	seq64::rtmidi_in_data, 834, 835
seq64::eventedit, 232	in_range
seq64::seqedit, 888	seq64::midi_control, 504
handle_config	increment
seq64::lash, 388	seq64::midi_splitter, 570
handle_delete	increment_beats_per_minute
seq64::eventedit, 232	seq64::perform, 717
handle_event	increment_bottom
seq64::lash, 388	seq64::eventslots, 253
handle_insert	increment_current
seq64::eventedit, 232	seq64::eventslots, 252
handle_midi_control	increment_data1
seq64::perform, 729	seq64::event, 201
handle_midi_control_ex	increment_data2
seq64::perform, 730	seq64::event, 201
handle_modify	increment_offset
seq64::eventedit, 233	seq64::trigger, 1055

increment_screenset	seq64::midi_api, 489
seq64::perform, 718	input_func
increment_selected	seq64::perform, 709
seq64::sequence, 1031	input_ports
increment_size	seq64::midi_info, 514
seq64::perfroll, 769	input_thread_func
seq64::perftime, 787	seq64, 99
increment_tick_end	seq64::perform, 747
seq64::trigger, 1054	insert_event
increment_tick_start	seq64::eventslots, 245, 246
seq64::trigger, 1054	install_sequence
increment_top	seq64::perform, 742
seq64::eventslots, 252	install_signal_handlers
info_message	seq64::mainwnd, 428
seq64, 88	instrument
seq64::jack_assistant, 330	seq64::user_instrument_t, 1078
init	seq64::user_midi_bus, 1081
seq64::font, 260	seq64::user_midi_bus_t, 1083
seq64::jack_assistant, 321	seq64::user_settings, 1094
seq64::lash, 388	instrument_controller_active
seq64::mastermidibase, 440	seq64::user_settings, 1095
init_before_show	instrument_controller_name
seq64::perfedit, 654	seq64::user_settings, 1095
seq64::perfroll, 769	instrument_count
init clock	seq64::user_settings, 1094
seq64::busarray, 142	instrument_name
seq64::businfo, 151–153	seq64::user_settings, 1095
seq64::mastermidibase, 443	InstrumentConstIterator
seq64::midibase, 596	seq64::user_settings, 1092
init in	InstrumentIterator
seq64::midibase, 594	seq64::user_settings, 1092
init_in_sub	Instruments
seq64::midibase, 594	seq64::user_settings, 1092
init_input	interaction_method
seq64::businfo, 151, 152	seq64::rc_settings, 803, 808
init_jack_transport	interaction_method_t
seq64::perform, 738	seq64, 68
init out	intersect
seq64::midibase, 594	seq64::triggers, 1065
init out sub	intersect_events
seq64::midibase, 594	seq64::sequence, 1016
initialize	intersect notes
seq64::busarray, 141	seq64::sequence, 1015
seq64::businfo, 151	intersect_triggers
SEOD4DUSINIO. 131	
seq64::midi_splitter, 570	seq64::sequence, 1014
seq64::midi_splitter, 570 initialize_buses	seq64::sequence, 1014 invalid_key
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445	seq64::sequence, 1014 invalid_key seq64, 97
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized seq64::businfo, 151	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active seq64::midi_control, 502
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized seq64::businfo, 151 inner_start	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active seq64::midi_control, 502 inverse_colors
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized seq64::businfo, 151 inner_start seq64::perform, 743	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active seq64::midi_control, 502 inverse_colors seq64::user_settings, 1101, 1107
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized seq64::businfo, 151 inner_start seq64::perform, 743 inner_stop	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active seq64::midi_control, 502 inverse_colors seq64::user_settings, 1101, 1107 is
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized seq64::businfo, 151 inner_start seq64::perform, 743 inner_stop seq64::perform, 743	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active seq64::midi_control, 502 inverse_colors seq64::user_settings, 1101, 1107 is seq64::keystroke, 384
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized seq64::businfo, 151 inner_start seq64::perform, 743 inner_stop seq64::perform, 743 input	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active seq64::midi_control, 502 inverse_colors seq64::user_settings, 1101, 1107 is seq64::keystroke, 384 is_active
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized seq64::businfo, 151 inner_start seq64::perform, 743 inner_stop seq64::perform, 743 input seq64::mastermidibase, 450	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active seq64::midi_control, 502 inverse_colors seq64::user_settings, 1101, 1107 is seq64::keystroke, 384 is_active seq64::perform, 704
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized seq64::businfo, 151 inner_start seq64::perform, 743 inner_stop seq64::perform, 743 input seq64::mastermidibase, 450 input_callback	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active seq64::midi_control, 502 inverse_colors seq64::user_settings, 1101, 1107 is seq64::keystroke, 384 is_active seq64::perform, 704 is_adding
seq64::midi_splitter, 570 initialize_buses seq64::mastermidibase, 445 initialized seq64::businfo, 151 inner_start seq64::perform, 743 inner_stop seq64::perform, 743 input seq64::mastermidibase, 450	seq64::sequence, 1014 invalid_key seq64, 97 inverse_active seq64::midi_control, 502 inverse_colors seq64::user_settings, 1101, 1107 is seq64::keystroke, 384 is_active seq64::perform, 704

seq64::AbstractPerfInput, 136	is marked
is_black_key	seq64::event, 204
seq64::seqkeys, 924	is master
is_channel_msg	seq64::jack_assistant, 319
seq64::event, 195	is_middle
is_control_status	seq64::click, 157
seq64::perform, 691	is_modified
is_ctrl_key	seq64::event_list, 216
seq64, 102–104	seq64::perform, 691, 739
is_ctrl_shift_key	seq64::seqmenu, 934, 935
seq64, 104	is_more_input
is_current_seq_active	seq64::mastermidibase, 447
seq64::seqmenu, 935	is_mseq_valid
is_current_seq_in_edit	seq64::perform, 742
seq64::seqmenu, 935	is_no_modifier
is_delete	seq64, 103
seq64::keystroke, 384	is_note
is_desired_cc_or_not_cc	seq64::event, 206
seq64::event, 197	is_note_msg
is_dirty_edit	seq64::event, 196
seq64::perform, 723	is_note_off
seq64::sequence, 1006	seq64::event, 206
is_dirty_main	is_note_off_recorded
seq64::perform, 723	seq64::event, 207
seq64::sequence, 1006	is_note_on
is_dirty_names	seq64::event, 206
seq64::perform, 724	is_null_midipulse
seq64::sequence, 1006	seq64, 98
is_dirty_perf	is_one_byte_msg
seq64::perform, 724	seq64::event, 196
	-
seq64::sequence, 1006 is_dumping	is_output_port seq64::midibase, 590
	•
seq64::mastermidibase, 441	is_painted
is_edit_sequence	seq64::event, 204
seq64::perform, 692	is_pattern_playing
seq64::seqmenu, 935	seq64::perform, 695, 741
is_exportable	is_port_open
seq64::perform, 724	seq64::midi_api, 487
is_group_learning	seq64::rtmidi, 826
seq64::perform, 733	is_press
is_input_port	seq64::click, 157
seq64::midi_api, 484	seq64::keystroke, 384
seq64::midibase, 589, 590	is_realized
is_input_system_port	seq64::gui_window_gtk2, 312
seq64::mastermidibase, 448	is_right
seq64::perform, 746	seq64::click, 157
is_inverse	is_running
seq64::gui_palette_gtk2, 303	seq64::jack_assistant, 319
is_jack_master	seq64::perform, 695
seq64::perform, 695	is_save
is_jack_running	seq64::mainwnd, 428
seq64::perform, 695	is_screenset_valid
is_left	seq64::perform, 740
seq64::click, 157	is_selected
is_letter	seq64::event, 204
seq64::keystroke, 384	is_seq_valid
is_linked	seq64::perform, 741
seq64::event, 203	is_sequence_in_edit

seq64::perform, 701	jack_shutdown_callback
is_shift_key	seq64, <mark>93</mark>
seq64, 103, 104	seq64::jack_assistant, 332
is_smf_0	jack_sync_callback
seq64::perform, 718	seq64, <mark>92</mark>
seq64::sequence, 1007	seq64::jack_assistant, 332
is_strict_note_msg	seq64::perform, 748
seq64::event, 197	jack_timebase_callback
is_super_key	seq64, <mark>93</mark>
seq64, 104	seq64::jack_assistant, 333
is_sysex_special_id	seq64::perform, 749
seq64::midifile, 633	jack_transport_callback
is_system_port	seq64, 94
seq64::busarray, 146	seq64::jack_assistant, 332
seq64::midi_api, 484	seq64::perform, 748
seq64::midibase, 590	jf_bit
is_two_byte_msg	seq64::jack_status_pair_t, 340
seq64::event, 196	jf_meaning
is_valid	seq64::jack_status_pair_t, 341
seq64::user_instrument, 1075	js_clock_tick
seq64::user midi bus, 1080	seq64::jack_scratchpad, 339
is_virtual_port	js_current_tick
seq64::midi api, 484	seq64::jack_scratchpad, 339
seg64::midibase, 589	js_delta_tick_frac
iterator	seq64::jack_scratchpad, 340
seq64::editable_events, 181	js_dumping
seq64::event_list, 214	seq64::jack_scratchpad, 339
seq64::seqmenu, 933	js_init_clock
	seq64::jack_scratchpad, 339
jack_assistant	js_jack_stopped
seq64::jack_assistant, 318	seq64::jack_scratchpad, 339
seq64::perform, 746	js_looping
jack_data	seq64::jack_scratchpad, 339
seq64::midi_jack, 526	js_playback_mode
jack_dialog	seq64::jack_scratchpad, 339
seq64::mainwnd, 426	js_ticks_converted
jack_dummy_callback	seq64::jack_scratchpad, 340
seq64, 100	js_ticks_converted_last
jack frame rate	seq64::jack scratchpad, 340
seq64::jack_assistant, 326	js_ticks_delta
jack_idle_connect	seq64::jack_scratchpad, 340
seq64::gui_assistant, 276	js_total_tick
seq64::gui assistant gtk2, 279	seq64::jack_scratchpad, 339
jack_message_bit_bucket	Sequ4jack_Scratchpau, 339
seq64, 109	keep_queue
jack_process_io	seq64::keys_perform, 354
seq64, 109	Key
seq64::midi_jack_info, 544	seq64::editable events, 181
jack_process_rtmidi_input	key
seq64, 107	seq64::keystroke, 384
jack_process_rtmidi_output	key_name
seq64, 108	seq64::keys_perform, 364
jack_session_callback	seq64::keys_perform_gtk2, 376
seq64, 96	seq64::perform, 711
seq64::jack_assistant, 334	key_press_event
jack_session_uuid	seq64::perftime, 790
seq64::rc_settings, 803, 808	keybindentry
jack_shutdown	seq64::keybindentry, 342
seq64::perform, 748	seq64::perform, 746

keys	seq64::keys_perform_transfer, 379
seq64::gui_assistant, 277	kpt_song_mode
seq64::perform, 694	seq64::keys_perform_transfer, 380
keys_perform	kpt_start
seq64::keys_perform, 351	seq64::keys_perform_transfer, 379
keys_perform_gtk2	kpt_stop
seq64::keys_perform_gtk2, 376	seq64::keys_perform_transfer, 379
keystroke	kpt_tap_bpm
seq64::keystroke, 382, 383	seq64::keys_perform_transfer, 380
keyval_name	kpt_toggle_jack
seq64, 97	seq64::keys_perform_transfer, 380
keyval_normalize	kpt_toggle_mutes
seq64, 97	seq64::keys_perform_transfer, 381
kpt_bpm_dn	lash
seq64::keys_perform_transfer, 378	
kpt_bpm_up	seq64::lash, 387 lash driver
seq64::keys_perform_transfer, 378	seq64, 98
kpt_event_edit	lash support
seq64::keys_perform_transfer, 380	seq64::rc_settings, 801, 805
kpt_fast_forward	lash_support_callback
seq64::keys_perform_transfer, 381	seq64::options, 642
kpt_follow_transport	lash_timeout_connect
seq64::keys_perform_transfer, 381	seq64::gui_assistant, 276
kpt_group_learn	seq64::gui_assistant_gtk2, 279
seq64::keys_perform_transfer, 378	last_used_dir
kpt_group_off	seq64::rc_settings, 804, 808
seq64::keys_perform_transfer, 378	launch
kpt_group_on	seq64::perform, 699
seq64::keys_perform_transfer, 378	launch_input_thread
kpt_keep_queue	seq64::perform, 738
seq64::keys_perform_transfer, 379	launch_output_thread
kpt_menu_mode	seq64::perform, 738
seq64::keys_perform_transfer, 380	learn toggle
kpt_pattern_edit	seq64::mainwnd, 424
seq64::keys_perform_transfer, 380	seq64::perform, 716
kpt_pause	left_right_size
seq64::keys_perform_transfer, 380	seq64::perform, 704
kpt_pointer_position	legacy format
seq64::keys_perform_transfer, 381	seq64::rc_settings, 801, 805
kpt_queue	length
seq64::keys_perform_transfer, 379	seq64::trigger, 1053
kpt_replace	lfownd
seq64::keys_perform_transfer, 379	seq64::lfownd, 392
kpt_rewind	seq64::seqdata, 868
seq64::keys_perform_transfer, 381	light_grey
kpt_screenset_dn	seq64::gui_palette_gtk2, 304
seq64::keys_perform_transfer, 378	line_after
kpt_screenset_up	seq64::configfile, 164
seq64::keys_perform_transfer, 378	line_color
kpt_set_playing_screenset	seq64::gui_palette_gtk2, 303
seq64::keys_perform_transfer, 378	line_count
kpt_show_ui_sequence_key	seq64::eventslots, 244
seq64::keys_perform_transfer, 379	line_increment
kpt_show_ui_sequence_number	seq64::eventslots, 244
seq64::keys_perform_transfer, 380	line_maximum
kpt_snapshot_1	seq64::eventslots, 244
seq64::keys_perform_transfer, 379	link
kpt_snapshot_2	seq64::event, 203

link_new	seq64::rc_settings, 812
seq64::event_list, 218	m_allow_two_perfedits
seq64::sequence, 1034	seq64::user_settings, 1112
List	m_alsa_seq
seq64::triggers, 1059	seq64::mastermidibus, 464
load_events	seq64::midi_alsa_info, 480
seq64::editable_events, 183	m_api_data
seq64::eventslots, 245	seq64::rtmidi_in_data, 838
load_inverse_palette	m_app_client_name
seq64::gui_palette_gtk2, 302	seq64::rc_settings, 816
lock	m_app_name
seq64::mutex, 637	seq64::midi_info, 521
log2_time_sig_value	m_application_name
seq64, 78	seq64::rc_settings, 816
log_main_sequence	m_armed_progress_color
seq64::midi_splitter, 570	seq64::mainwid, 412
long_options	m_armed_saved
seq64, 124	seq64::perform, 751
lookup_keyevent_key	m_armed_statuses
seq64::keys_perform, 363	seq64::perform, 751
seq64::perform, 713	m_auto_option_save
lookup_keyevent_seq	seq64::rc_settings, 812
seq64::keys_perform, 363	m_b_on_c_pixmap
seq64::perform, 713	seq64::font, 263
lookup_keygroup_group	m_b_on_y_pixmap
seq64::keys_perform, 364	seq64::font, 262
seq64::perform, 714	m_back
lookup_keygroup_key	seq64::midi_queue, 567
seq64::keys_perform, 363	m_background
seq64::perform, 714	seq64::gui_drawingarea_gtk2, 296
m_32nds_per_quarter	m_background_sequence
seq64::perform, 756	seq64::seqroll, 974
seq64::sequence, 1050	seq64::sequence, 1051
m_4bar_offset	m_background_x
seq64::perfroll, 780	seq64::perfroll, 779
seq64::perftime, 791	m_bar_width
m FF RW button type	seq64::maintime, 398
seq64::perform, 751	m_beat_length
m active	seq64::perfroll, 780
seq64::businfo, 154	m_beat_width
seq64::midi_control, 504	seq64::jack_assistant, 337
m adding	seq64::maintime, 398
seq64::AbstractPerfInput, 136	seq64::midi_timing, 576
seq64::Seq24SeqEventInput, 857	seq64::perform, 755
seq64::seqroll, 972	m_beats
m_adding_pressed	seq64::midi_measures, 552
seq64::AbstractPerfInput, 136	m_beats_per_bar
m_adjust_bpm	seq64::perform, 755
seq64::mainwnd, 433	m_beats_per_measure
m_adjust_load_offset	seq64::jack_assistant, 337
seq64::mainwnd, 434	seq64::midi_timing, 576
m_adjust_ss	m_beats_per_minute
seq64::mainwnd, 433	seq64::jack_assistant, 338
m_allow_click_edit	seq64::mastermidibase, 455
seq64::rc_settings, 813	seq64::midi_timing, 576
m_allow_mod4_mode	m_bg_color
seq64::rc_settings, 812	seq64::gui_palette_gtk2, 309
m_allow_snap_split	m_bgsequence

seq64::seqedit, 892	seq64::seqedit, 898
m_black	m_button_chord
seq64::gui_palette_gtk2, 306	seq64::seqedit, 900
m_black_pixmap seq64::font, 262	m_button_collapse seq64::perfedit, 664
m_blk_key	m_button_copy
seq64::gui_palette_gtk2, 308	seq64::perfedit, 664
m_blk_paint	m_button_data
seq64::gui_palette_gtk2, 308	seq64::seqedit, 901
m blue	m button del
seq64::gui_palette_gtk2, 308	seq64::eventedit, 236
m bottbox	m_button_down
seq64::eventedit, 236	seq64::mainwid, 412
m_bottom_iterator	m_button_expand
seq64::eventslots, 257	seq64::perfedit, 664
m_box_height	m_button_follow
seq64::maintime, 399	seq64::perfedit, 665
m_box_less_pill	m_button_grow
seq64::maintime, 399	seq64::perfedit, 664
m_box_width	m_button_ins
seq64::maintime, 399	seq64::eventedit, 236
m_bpm	m_button_jack
seq64::midi_info, 521	seq64::mainwnd, 433
seq64::midibase, 601	seq64::perfedit, 664
seq64::perfedit, 666	m_button_jack_connect
seq64::perform, 755	seq64::options, 644
m_bpm_page_increment	m_button_jack_disconnect
seq64::user_settings, 1115	seq64::options, 644
m_bpm_precision	m_button_jack_master
seq64::user_settings, 1115	seq64::options, 644
m_bpm_step_increment	m_button_jack_master_cond
seq64::user_settings, 1115	seq64::options, 644
m_bus	m_button_jack_transport
seq64::businfo, 153	seq64::options, 644
seq64::sequence, 1046	m_button_key
m_bus_announce	seq64::seqedit, 900
seq64::mastermidibase, 454	m_button_learn
m_bus_container	seq64::mainwnd, 433
seq64::midi_info, 520	m_button_length
m_bus_id	seq64::seqedit, 899
seq64::midibase, 601	m_button_lfo
m_bus_index	
0.4	seq64::seqedit, 897
seq64::midibase, 601	m_button_loop
m_bus_name	m_button_loop seq64::perfedit, 664
m_bus_name seq64::midibase, 602	m_button_loop seq64::perfedit, 664 m_button_modify
m_bus_name seq64::midibase, 602 m_button	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236
m_bus_name seq64::midibase, 602 m_button seq64::click, 159	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm seq64::perfedit, 665	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899 m_button_ok
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm seq64::perfedit, 665 seq64::seqedit, 901	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899 m_button_ok seq64::options, 644
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm seq64::perfedit, 665 seq64::seqedit, 901 m_button_bus	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899 m_button_ok seq64::options, 644 m_button_perfedit
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm seq64::perfedit, 665 seq64::seqedit, 901 m_button_bus seq64::seqedit, 898	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899 m_button_ok seq64::options, 644 m_button_perfedit seq64::mainwnd, 433
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm seq64::perfedit, 665 seq64::seqedit, 901 m_button_bus seq64::seqedit, 898 m_button_bw	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899 m_button_ok seq64::options, 644 m_button_perfedit seq64::mainwnd, 433 m_button_play
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm seq64::perfedit, 665 seq64::seqedit, 901 m_button_bus seq64::seqedit, 898 m_button_bw seq64::perfedit, 665	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899 m_button_ok seq64::options, 644 m_button_perfedit seq64::mainwnd, 433 m_button_play seq64::mainwnd, 433
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm seq64::perfedit, 665 seq64::seqedit, 901 m_button_bus seq64::seqedit, 898 m_button_bw seq64::perfedit, 665 seq64::perfedit, 665 seq64::seqedit, 901	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899 m_button_ok seq64::options, 644 m_button_perfedit seq64::mainwnd, 433 m_button_play seq64::mainwnd, 433 seq64::perfedit, 663
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm seq64::perfedit, 665 seq64::seqedit, 901 m_button_bus seq64::seqedit, 898 m_button_bw seq64::perfedit, 665 seq64::seqedit, 901 m_button_cancel	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899 m_button_ok seq64::options, 644 m_button_perfedit seq64::mainwnd, 433 m_button_play seq64::mainwnd, 433 seq64::perfedit, 663 m_button_quantize
m_bus_name seq64::midibase, 602 m_button seq64::click, 159 m_button_bpm seq64::perfedit, 665 seq64::seqedit, 901 m_button_bus seq64::seqedit, 898 m_button_bw seq64::perfedit, 665 seq64::perfedit, 665 seq64::seqedit, 901	m_button_loop seq64::perfedit, 664 m_button_modify seq64::eventedit, 236 m_button_note_length seq64::seqedit, 899 m_button_ok seq64::options, 644 m_button_perfedit seq64::mainwnd, 433 m_button_play seq64::mainwnd, 433 seq64::perfedit, 663

seq64::seqedit, 901	seq64::perfnames, 674
m_button_redo	m_chord
seq64::perfedit, 664	seq64::seqedit, 892
seq64::seqedit, 898	seq64::seqroll, 972
m_button_save	m_client
seq64::eventedit, 237	seq64::lash, 389
m_button_scale	m_client_name
seq64::seqedit, 900	seq64::midi_in_jack, 510
m_button_sequence	seq64::midi_port_info::port_info_t, 793
seq64::seqedit, 898	m_client_number
m button snap	seq64::midi_port_info::port_info_t, 793
seq64::perfedit, 663	m clip mask
seq64::seqedit, 899	seq64::font, 263
m_button_stop	m_clipboard
seq64::mainwnd, 433	seq64::seqmenu, 941
seq64::perfedit, 663	seq64::triggers, 1071
m button tools	m_clock_mod
seg64::segedit, 898	seq64::midibase, 600
m_button_undo	m_clock_type
seq64::perfedit, 664	seq64::midibase, 601
seq64::seqedit, 897	m_clocks_per_metronome
m_button_xpose	seq64::perform, 755
seq64::perfedit, 663	seq64::sequence, 1050
m_button_zoom	m cond
seq64::seqedit, 899	seq64::condition_var, 161
m bw	m_condition_var
seq64::perfedit, 666	seq64::perform, 760
m_bytes	m_config_directory
seq64::midi_message, 554	seq64::rc_settings, 815
m_c_on_b_pixmap	m_config_filename
seq64::font, 263	seq64::rc_settings, 815
m_call_seq_edit	m_config_filename_alt
seq64::mainwnd, 435	seq64::rc_settings, 815
m_call_seq_eventedit	m connected
seq64::mainwnd, 435	seq64::midi_api, 490
m_category	m_container
seq64::editable_event, 178	seq64::busarray, 148
m_cc	m_continue_sysex
seq64::seqdata, 869	seq64::rtmidi_in_data, 838
seq64::seqevent, 917	m control height
seg64::segroll, 975	seq64::user_settings, 1110
m_cell_h	m control status
seq64::font, 261	seq64::perform, 758
m_cell_w	m_controller_count
seq64::font, 261	seq64::user instrument, 1077
m_channel	m_current_event
seq64::event, 208	seq64::editable_events, 187
m_channel_count	m_current_index
seq64::user_midi_bus, 1082	seq64::eventslots, 257
m channel match	m_current_iterator
seq64::sequence, 1046	seq64::eventslots, 257
m_char_list	m_current_seq
seq64::midi_list, 549	seq64::seqmenu, 941
seq64::midifile, 635	m_current_x
m_char_vector	seq64::FruityPerfInput, 268
seq64::midi_vector, 581	seq64::gui_drawingarea_gtk2, 297
m_char_w	m_current_y
seq64::eventslots, 256	seq64::FruityPerfInput, 268
•	

seq64::gui_drawingarea_gtk2, 297	seq64::seqdata, 870
m_current_zoom	m_drawing_background_seq
seq64::user_settings, 1110	seq64::seqroll, 974
m_d	m_drop_sequence
seq64::configfile, 165	seq64::perfroll, 781
m_data	m_drop_tick
seq64::event, 208	seq64::perfroll, 781
seq64::midi_control, 504	m_drop_tick_trigger_offset
seq64::midifile, 634	seq64::perfroll, 781
m_dest_addr_client	m_drop_x
seq64::midi_alsa, 474	seq64::gui_drawingarea_gtk2, 297
seq64::midibus, 614	m_drop_y
m_dest_addr_port	seq64::gui_drawingarea_gtk2, 298
seq64::midi_alsa, 474	m_dumping_input
seq64::midibus, 614	seq64::mastermidibase, 455
m_device_ignore	m_edit_sequence
seq64::rc_settings, 814	seq64::perform, 759
m_device_ignore_num	m_editbox
seq64::rc_settings, 814	seq64::eventedit, 236
m_dirty_edit	m_editing
seq64::sequence, 1048	seq64::sequence, 1048
m_dirty_main	m_editing_cc
seq64::sequence, 1048	seq64::seqedit, 902
m_dirty_names	m_editing_status
seq64::sequence, 1048	seq64::seqedit, 902
m_dirty_perf	m_effective_tick
seq64::sequence, 1048	seq64::Seq24PerfInput, 855
m_disable_reported	m_entry_bpm
seq64::midifile, 634	seq64::perfedit, 665
m_display_name	seq64::seqedit, 901
seq64::midibase, 602	m_entry_bus
m_divisions	seq64::seqedit, 898
seq64::midi_measures, 552	m_entry_bw
m_divs_per_beat	seq64::perfedit, 665
seq64::perfroll, 779	seq64::seqedit, 901
m_dk_blue	m_entry_channel
seq64::gui_palette_gtk2, 307	seq64::seqedit, 899
m_dk_cyan	m_entry_chord
seq64::gui_palette_gtk2, 307	seq64::seqedit, 900
m_dk_green	m_entry_data
seq64::gui_palette_gtk2, 306	seq64::seqedit, 901
m_dk_grey	m_entry_ev_data_0
seq64::gui_palette_gtk2, 308	seq64::eventedit, 238
m_dk_magenta	m_entry_ev_data_1
seq64::gui_palette_gtk2, 307	seq64::eventedit, 238
m_dk_orange	m_entry_ev_name
seq64::gui_palette_gtk2, 307	seq64::eventedit, 238
m_dk_red	m_entry_ev_timestamp
seq64::gui_palette_gtk2, 306	seq64::eventedit, 238
m_do_input	m_entry_key
seq64::rtmidi_in_data, 837	seq64::seqedit, 900
m_dont_reset_ticks	m_entry_length
seq64::perform, 758	seq64::seqedit, 900
m_drag_handle	m_entry_name
seq64::seqdata, 870	seq64::seqedit, 902
m_drag_paste_start_pos	m_entry_note_length
seq64::FruitySeqRollInput, 274	seq64::seqedit, 899
m_dragging	m_entry_notes

seq64::mainwnd, 434	seq64::midi api, 490
m_entry_scale	m_first_message
seq64::seqedit, 900	seq64::rtmidi_in_data, 838
·	
m_entry_sequence	m_flash_height
seq64::seqedit, 898	seq64::maintime, 399
m_entry_snap	m_flash_width
seq64::perfedit, 663	seq64::maintime, 399
seq64::seqedit, 899	m_flash_x
m_entry_xpose	seq64::maintime, 399
seq64::perfedit, 663	m_follow_transport
m_entry_zoom	seq64::jack_assistant, 337
seq64::seqedit, 899	m_font_h
m_erase_painting	seq64::font, 261
seq64::FruitySeqRollInput, 274	m_font_w
m_error_callback	seq64::font, 261
seq64::midi_api, 490	m_foreground
m_error_callback_user_data	seq64::gui_drawingarea_gtk2, 297
seq64::midi_api, 490	m format timestamp
	_
m_error_is_fatal	seq64::editable_event, 178
seq64::midifile, 634	m_front
m_error_message	seq64::midi_queue, 567
seq64::configfile, 165	m_fruity_interaction
seq64::midifile, 634	seq64::perfroll, 782
m_error_string	seq64::seqevent, 915
seq64::midi_api, 490	seq64::seqroll, 971
seq64::midi_info, 521	m_gc
m_event_container	seq64::gui_drawingarea_gtk2, 296
seq64::eventslots, 255	m_global_bgsequence
m_event_count	seq64::midifile, 635
seq64::eventslots, 256	m_global_queue
m_eventedit	seq64::midi_info, 521
seq64::seqmenu, 941	m_global_seq_feature_save
m_events	seq64::user_settings, 1111
seq64::editable_events, 187	m_gmute_tracks
seq64::event_list, 222	seq64::user_settings, 1116
seq64::sequence, 1045	m_green
m_events_clipboard	seq64::gui_palette_gtk2, 308
seq64::sequence, 1045	m_grey
m_events_redo	seq64::gui_palette_gtk2, 308
seq64::sequence, 1046	m_grid_brackets
m_events_undo	seq64::user_settings, 1109
seq64::sequence, 1046	m_grid_style
m_events_undo_hold	seq64::user_settings, 1109
seq64::sequence, 1045	m_grow_direction
m_eventslots	seq64::perfroll, 782
seq64::eventedit, 235	m_growing
m_excell_FF_RW	seq64::perfroll, 782
seq64::perform, 751	seq64::seqevent, 916
• •	
m_fg_color	seq64::seqroll, 972
seq64::gui_palette_gtk2, 309	m_gui_support
m_file_size	seq64::perform, 761
seq64::midifile, 634	m_h_page_increment
m_filename	seq64::perfroll, 778
seq64::rc_settings, 815	m_h_perf_page_increment
m_filter_by_channel	seq64::user_settings, 1112
seq64::mastermidibase, 455	m_hadjust
seq64::rc_settings, 814	seq64::gui_drawingarea_gtk2, 296
m_first_error_occurred	seq64::perfedit, 662

seq64::seqedit, 896	seq64::seqedit, 891
m_has_link	m_initial_zoom
seq64::event, 208	seq64::seqedit, 891
m_have_button_press	m_initialized
seq64::perfroll, 780	seq64::businfo, 154
m_have_focus	m input
seq64::eventedit, 239	seq64::midi_info, 520
seq64::seqedit, 902	m_input_data
m_have_redo	seq64::midi_api, 490
seq64::perform, 760	m_input_port_name
seq64::sequence, 1046	seq64::midi_alsa, 474
m_have_undo	seq64::midi_aisa, 474 seq64::midibus, 615
	•
seq64::perform, 760	m_inputing
seq64::sequence, 1045	seq64::midibase, 601
m_hbox	seq64::perform, 754
seq64::lfownd, 393	m_instrument_def
seq64::perfedit, 665	seq64::user_instrument, 1077
seq64::seqedit, 897	m_instruments
m_hbox2	seq64::user_settings, 1109
seq64::seqedit, 897	m_interaction
m_hint_key	seq64::perfroll, 782
seq64::seqkeys, 928	m_interaction_method
m_hint_state	seq64::rc_settings, 814
seq64::seqkeys, 928	m_inverse_active
m_hlbox	seq64::midi_control, 504
seq64::perfedit, 665	m_inverse_colors
m_horizontal_adjust	seq64::user_settings, 1112
seq64::seqroll, 970	m_is_drag_pasting
m_hscroll	seq64::FruitySeqEventInput, 271
seq64::perfedit, 662	seq64::seqroll, 973
m_hscroll_new	m_is_drag_pasting_start
seq64::seqedit, 896	seq64::FruitySeqEventInput, 271
m_htopbox	seq64::seqroll, 973
seq64::eventedit, 235	m_is_input
m_ignore_flags	seq64::midi_port_info::port_info_t, 794
seq64::rtmidi_in_data, 837	m_is_input_port
m_image_play	seg64::midibase, 602
seq64::mainwnd, 432	m is inverse
seq64::perfedit, 663	seq64::gui_palette_gtk2, 306
m_image_transpose	m_is_lash_supported
seq64::seqedit, 894	seq64::lash, 389
m_in_thread	m_is_modified
seq64::perform, 754	seq64::event_list, 222
m_in_thread_launched	seq64::perform, 760
seq64::perform, 754	m_is_pattern_playing
m_inbus_array	seq64::perform, 754
seq64::mastermidibase, 454	
•	m_is_press
m_info_api	seq64::click, 158
seq64::rtmidi_info, 848	seq64::keystroke, 386
m_init_clock	m_is_realized
seq64::businfo, 154	seq64::gui_window_gtk2, 314
m_init_input	m_is_running
seq64::businfo, 154	seq64::mainwnd, 434
m_initial_chord	seq64::perfedit, 666
seq64::seqedit, 891	m_is_system
m_initial_note_length	seq64::midi_port_info::port_info_t, 794
seq64::seqedit, 891	m_is_system_port
m_initial_snap	seq64::midibase, 602

m_is_valid	seq64::jack_assistant, 336
seq64::user_instrument, 1077	seq64::perform, 757
seq64::user_midi_bus, 1082	m_jack_transport_state
m_is_virtual	seq64::jack_assistant, 336
seq64::midi_port_info::port_info_t, 794	m_jack_transport_state_last
m_is_virtual_port	seq64::jack_assistant, 336
seq64::midibase, 602	m_jsession_ev
m_iterator_draw	seq64::jack_assistant, 336
seq64::sequence, 1046	m_justselected_one
m_iterator_draw_trigger	seq64::FruitySeqEventInput, 271
seq64::triggers, 1072	seq64::seqroll, 973
m_iterator_play_trigger	m_key
seq64::triggers, 1072	seq64::keybindentry, 343
m_jack_asst	seq64::keystroke, 386
seq64::perform, 760	seq64::seqedit, 892
	seq64::seqkeys, 928
m_jack_buffmessage	
seq64::midi_jack_data, 537	seq64::seqroll, 972
m_jack_buffsize	m_key_bpm_dn
seq64::midi_jack_data, 537	seq64::keys_perform, 370
m_jack_client	m_key_bpm_up
seq64::jack_assistant, 335	seq64::keys_perform, 370
seq64::midi_jack_data, 537	m_key_event_edit
seq64::midi_jack_info, 545	seq64::keys_perform, 373
m_jack_client_name	m_key_events
seq64::jack_assistant, 335	seq64::keys_perform, 370
m_jack_client_uuid	m_key_events_rev
seq64::jack_assistant, 335	seq64::keys_perform, 370
m_jack_data	m_key_fast_forward
seq64::midi_jack, 535	seq64::keys_perform, 373
m_jack_frame_current	m_key_follow_transport
seq64::jack_assistant, 335	seq64::keys_perform, 373
m_jack_frame_last	m_key_group_learn
seq64::jack_assistant, 335	seq64::keys_perform, 372
m_jack_frame_rate	m_key_group_off
seq64::jack_assistant, 337	seq64::keys_perform, 372
m_jack_info	m_key_group_on
seq64::midi_jack, 535	seq64::keys_perform, 372
m_jack_lasttime	m_key_groups
seq64::midi_jack_data, 537	seq64::keys perform, 370
· — —	
m_jack_master	m_key_groups_rev
seq64::jack_assistant, 336	seq64::keys_perform, 370
m_jack_parent	m_key_keep_queue
seq64::jack_assistant, 335	seq64::keys_perform, 371
m_jack_port	m_key_menu_mode
seq64::midi_jack_data, 537	seq64::keys_perform, 372
m_jack_ports	m_key_pattern_edit
seq64::midi_jack_info, 545	seq64::keys_perform, 373
m_jack_pos	m_key_pause
seq64::jack_assistant, 336	seq64::keys_perform, 372
m_jack_rtmidiin	m_key_pointer_position
seq64::midi_jack_data, 537	seq64::keys_perform, 373
m_jack_running	m_key_queue
seq64::jack_assistant, 336	seq64::keys_perform, 371
m_jack_session_uuid	m_key_replace
seq64::rc_settings, 815	seq64::keys_perform, 371
m_jack_stop_tick	m_key_rewind
seq64::jack_assistant, 337	seq64::keys_perform, 373
m_jack_tick	m_key_screenset_dn

seq64::keys_perform, 371	seq64::sequence, 1049
m_key_screenset_up	m_last_tick_x
seq64::keys_perform, 371	seq64::mainwid, 413
	•
m_key_set_playing_screenset seq64::keys_perform, 371	m_last_used_dir
	seq64::rc_settings, 815
m_key_show_ui_sequence_key	m_lasttick
seq64::keys_perform, 370	seq64::midibase, 602
m_key_show_ui_sequence_number	m_left_marker_tick
seq64::keys_perform, 370	seq64::perftime, 792
m_key_snapshot_1	m_left_tick
seq64::keys_perform, 371	seq64::perform, 756
m_key_snapshot_2	m_legacy_format
seq64::keys_perform, 371	seq64::rc_settings, 812
m_key_song_mode	m_length
seq64::keys_perform, 372	seq64::sequence, 1050
m_key_start	seq64::triggers, 1072
seq64::keys_perform, 372	m_lfo_wnd
m_key_stop	seq64::seqedit, 897
seq64::keys_perform, 374	m_line
m_key_tap_bpm	seq64::configfile, 165
seq64::keys_perform, 373	m_line_color
m_key_toggle_jack	seq64::gui_palette_gtk2, 309
seq64::keys_perform, 372	m_line_count
m_key_toggle_mutes	seq64::eventslots, 256
seq64::keys_perform, 373	m_line_maximum
m_keying	seq64::eventslots, 256
seq64::seqkeys, 928	m_line_overlap
m_keying_note	seq64::eventslots, 257
seq64::seqkeys, 928	m_linked
m_keys_perform	seq64::event, 208
seq64::gui_assistant, 277	m_local_addr_client
m_label_category	seq64::midi_alsa, 474
seq64::eventedit, 238	seq64::midi_disa, 77
m_label_channel	m local addr port
seq64::eventedit, 237	seq64::midi_alsa, 474
m_label_ev_count	seq64::midibus, 615
	•
seq64::eventedit, 237	m_looping
m_label_modified	seq64::perform, 755
seq64::eventedit, 238	m_lt_grey
m_label_ppqn	seq64::gui_palette_gtk2, 308
seq64::eventedit, 237	m_main_cursor
m_label_right	seq64::mainwnd, 432
seq64::eventedit, 238	m_main_time
m_label_seq_name	seq64::mainwnd, 432
seq64::eventedit, 237	m_main_wid
m_label_spacer	seq64::mainwnd, 432
seq64::eventedit, 237	m_mainperf
m_label_time_fmt	seq64::gui_drawingarea_gtk2, 297
seq64::eventedit, 238	seq64::gui_window_gtk2, 314
m_label_time_sig	seq64::options, 643
seq64::eventedit, 237	seq64::seqmenu, 941
m_lash_args	m_mainwid_border
seq64::lash, 389	seq64::mainwid, 414
m_lash_support	seq64::user_settings, 1110
seq64::rc_settings, 812	m_mainwid_spacing
m_last_playing	seq64::mainwid, 414
seq64::mainwid, 413	seq64::user_settings, 1110
m_last_tick	m_mainwid_x

seq64::mainwid, 414	m_menu_edit
seq64::user_settings, 1117	seq64::mainwnd, 431
m_mainwid_y	m_menu_file
seq64::mainwid, 414	seq64::mainwnd, 431
seq64::user_settings, 1117	m_menu_help
m_mainwnd_cols	seq64::mainwnd, 431
seq64::mainwid, 413	m_menu_key
seq64::user_settings, 1110	seq64::seqedit, 895
m_mainwnd_rows	m_menu_length
seq64::mainwid, 413	seq64::seqedit, 894
seq64::user_settings, 1110	m_menu_midibus
m_manual_alsa_ports	seq64::seqedit, 894
seq64::rc_settings, 814	m_menu_midich
m_marked	seq64::seqedit, 894
seq64::event, 209	m_menu_mode
m_master_bus	seq64::mainwnd, 434
seq64::perform, 756	m_menu_note_length
m_master_clocks	seq64::seqedit, 894
seq64::mastermidibase, 454	m_menu_rec_vol
seq64::perform, 756	seq64::seqedit, 895
m_master_info	m_menu_scale
seq64::midi_api, 489	seq64::seqedit, 895
seq64::midibus, 615	m_menu_sequences
m_master_inputs	seq64::seqedit, 895
seq64::mastermidibase, 454	m_menu_snap
seq64::perform, 756	seq64::perfedit, 662
m_masterbus	seq64::seqedit, 894
seq64::sequence, 1047	m_menu_tools
m_max_busses	seq64::seqedit, 893
seq64::mastermidibase, 453	m_menu_view
m_max_sequence	seq64::mainwnd, 431
seq64::user_settings, 1116	m_menu_xpose
m_max_sets	seq64::perfedit, 663
seq64::mainwid, 414	m_menu_zoom
seq64::perform, 759	seq64::seqedit, 894
seq64::user_settings, 1110	m_menubar
m_max_value	seq64::mainwnd, 431
seq64::midi_control, 505	seq64::seqedit, 893
m_maxbeats	m_message
seq64::sequence, 1049	seq64::rterror, 820
m_measure_length	seq64::rtmidi_in_data, 837
seq64::perfroll, 780	m_midi_api
seq64::perftime, 792	seq64::rtmidi, 828
m_measures	m_midi_beat_width
seq64::midi_measures, 551	seq64::user_settings, 1114
seq64::seqedit, 892	m_midi_beats_per_measure
m_menu	seq64::user_settings, 1114
seq64::seqmenu, 941	m_midi_beats_per_minute
m_menu_bpm	seq64::user_settings, 1114
seq64::perfedit, 666	m_midi_bus_def
seq64::seqedit, 895	seq64::user_midi_bus, 1082
m_menu_bw	m_midi_buses
seq64::perfedit, 666	seq64::user_settings, 1109
seq64::seqedit, 895	m_midi_buss_override
m menu chords	seq64::user_settings, 1114
seq64::seqedit, 895	m_midi_cc_off
m_menu_data	seq64::perform, 758
seq64::seqedit, 895	m_midi_cc_on
1	

seq64::perform, 758	m_mute_group
m_midi_cc_toggle	seq64::perform, 751
seq64::perform, 758	m_mute_group_selected
m_midi_channel	seq64::perform, 752
seq64::sequence, 1046	m_mutex
m_midi_handle	seq64::mastermidibase, 455
seq64::midi_info, 521	seq64::midibase, 603
m_midi_info	seq64::midifile, 634
seq64::rtmidi, 828	seq64::sequence, 1051
m_midi_master	m_mutex_lock
seq64::mastermidibus, 464	seq64::mutex, 637
m_midi_mode_input	m_name
seq64::midi_info, 520	seq64::configfile, 165
m_midi_parameters	seq64::midifile, 634
seq64::editable_events, 187	seq64::sequence, 1049
m_midi_ppqn	m_name_category
seq64::user_settings, 1114	seq64::editable_event, 178
m_midiclockpos	m_name_channel
seq64::perform, 758	seq64::editable_event, 179
m_midiclockrunning	m_name_data
seq64::perform, 757	seq64::editable_event, 179
m_midiclocktick	m_name_meta
seq64::perform, 757	seq64::editable_event, 178
m_min_value	m_name_seqspec
seq64::midi_control, 504	seq64::editable_event, 179
m_mode_group	m_name_status
seq64::perform, 752	seq64::editable_event, 178
m_mode_group_learn	m_name_timestamp
seq64::perform, 752	seq64::editable_event, 178
m_modified	m_namebox_w
seq64::seqmenu, 942	seq64::perfnames, 674
m_modifier	m_names_chars
seq64::click, 159	seq64::perfnames, 674
seq64::keystroke, 386	m_names_x
m_move_delta_x	seq64::perfnames, 674
seq64::seqroll, 973	m_names_y
m_move_delta_y	seq64::perfnames, 674
seq64::seqroll, 973	seq64::perfroll, 779
m_move_snap_offset_x	m_new_format
seq64::seqevent, 917	seq64::midifile, 635
seq64::seqroll, 973	m_note_length
m_moving seq64::mainwid, 412	seq64::seqedit, 892 seq64::seqroll, 971
•	•
seq64::perfroll, 782	m_note_off_margin
seq64::seqevent, 916	seq64::sequence, 1052 m note off velocity
seq64::seqroll, 972	
m_moving_init	seq64::sequence, 1051
seq64::seqevent, 916	m_note_on_velocity
seq64::seqroll, 972	seq64::sequence, 1051
m_moving_seq	m_notebook
seq64::mainwid, 412	seq64::options, 644
m_multi_client	m_notes_on
seq64::midi_jack, 535	seq64::sequence, 1047
seq64::midi_jack_info, 545	m_notify
m_musical_key	seq64::perform, 761
seq64::sequence, 1051	m_num_poll_descriptors
m_musical_scale	seq64::mastermidibus, 464
seq64::sequence, 1051	seq64::midi_alsa_info, 480

m_number_h	seq64::midi_api, 489
seq64::seqdata, 869	m_pass_sysex
m_number_offset_y	seq64::rc_settings, 813
seq64::seqdata, 869	m_paste
m_number_w	seq64::seqevent, 917
seq64::seqdata, 869	seq64::seqroll, 973
m_numbers	m_paste_tick
seq64::seqdata, 869	seq64::triggers, 1072
m_offset	m_peer_perfedit
seq64::font, 262	seq64::perfedit, 661
seq64::perform, 758	m_perf
seq64::trigger, 1056	seq64::keybindentry, 344
m_old	m_perf_edit
seq64::seqdata, 869	seq64::mainwnd, 432
seq64::seqevent, 915	m_perf_edit_2
seq64::seqroll, 970	seq64::mainwnd, 432
m_old_progress_ticks	m_perf_scale_x
seq64::perfroll, 780	seq64::perfroll, 779
m_old_seq	seq64::perftime, 792
seq64::mainwid, 412	m perfnames
m one measure	seq64::perfedit, 662
seq64::perform, 756	m_perform
m_options	seq64::lash, 389
seq64::mainwnd, 432	m_perfroll
m_optsbox	seq64::perfedit, 662
seq64::eventedit, 236	m_perftime
m_orange	seq64::perfedit, 662
seq64::gui_palette_gtk2, 307	m_phase
m_out_thread	seq64::lfownd, 394
seq64::perform, 753	m_pill_width
m_out_thread_launched	seq64::maintime, 399
seq64::perform, 754	•
• •	m_pixmap seq64::font, 262
m_outbus_array seq64::mastermidibase, 454	•
•	seq64::gui_drawingarea_gtk2, 296
m_output	m_playback_mode
seq64::midi_info, 520	seq64::perform, 755
m_outputing	m_playing
seq64::perform, 754	seq64::sequence, 1047
m_padded_h	m_playing_notes
seq64::font, 262	seq64::sequence, 1047
m_page_factor	m_playing_screen
seq64::perfroll, 778	seq64::perform, 752
m_pager_index	m_playscreen_offset
seq64::eventslots, 257	seq64::perform, 752
m_painted	m_poll_descriptors
seq64::event, 209	seq64::mastermidibus, 464
m_painting	seq64::midi_alsa_info, 480
seq64::seqevent, 916	m_port_container
seq64::seqroll, 972	seq64::midi_port_info, 563
m_parent	m_port_count
seq64::editable_event, 178	seq64::midi_port_info, 563
seq64::eventslots, 255	m_port_id
seq64::perfnames, 673	seq64::midibase, 601
seq64::perfroll, 778	m_port_name
seq64::perftime, 791	seq64::midi_port_info::port_info_t, 793
seq64::sequence, 1045	seq64::midibase, 602
seq64::triggers, 1071	m_port_number
m_parent_bus	seq64::midi_port_info::port_info_t, 793

m_pos	seq64::sequence, 1049
seq64::midifile, 634	m range
seq64::seqroll, 971	seq64::lfownd, 394
m_position_for_get	m_rank
seq64::midi_container, 500	seq64::event list::event key, 211
m_pp_eighth	m_rec_vol
seq64::seqedit, 893	seq64::sequence, 1051
m_pp_sixteenth	m_recording
seq64::seqedit, 893	seq64::sequence, 1047
m_pp_whole	m red
seq64::seqedit, 893	seq64::gui_palette_gtk2, 307
• • •	m_redo_stack
m_ppqn seq64::jack_assistant, 337	seq64::triggers, 1072
seq64::maintime, 400	m_redo_vect
seq64::mainwnd, 431	
seq64::mastermidibase, 454	seq64::perform, 760
•	m_redraw_period_ms seq64::gui window gtk2, 314
seq64::midi_info, 521	
seq64::midi_splitter, 572	m_remote_port_name
seq64::midi_timing, 576	seq64::midi_jack, 535
seq64::midibase, 601	m_reposition
seq64::midifile, 635	seq64::perform, 751
seq64::perfedit, 666	m_reveal_alsa_ports
seq64::perform, 755	seq64::rc_settings, 814
seq64::perfroll, 778	m_right_marker_tick
seq64::perftime, 791	seq64::perftime, 792
seq64::seqedit, 893	m_right_tick
seq64::seqevent, 915	seq64::perform, 757
seq64::seqroll, 971	m_rightbox
seq64::seqtime, 981	seq64::eventedit, 236
seq64::sequence, 1049	m_ring
seq64::triggers, 1072	seq64::midi_queue, 567
m_print_keys	m_ring_size
seq64::rc_settings, 814	seq64::midi_queue, 567
m_priority	m_roll_length_ticks
seq64::rc_settings, 813	seq64::perfroll, 781
m_progress_bar_colored	m_rt_midi
seq64::user_settings, 1112	seq64::midibus, 615
m_progress_bar_thick	m_running
seq64::user_settings, 1112	seq64::perform, 754
m_progress_color	m_safety_mutex
seq64::gui_palette_gtk2, 309	seq64::automutex, 138
m_progress_height	m_save_user_config
seq64::mainwid, 415	seq64::user_settings, 1117
m progress x	m_scale
seg64::segroll, 973	seq64::seqedit, 892
m_quantized_rec	seq64::seqkeys, 928
seq64::sequence, 1048	seq64::seqroll, 971
m_queue	m_scale_phase
seq64::mastermidibase, 454	seq64::lfownd, 393
seq64::midibase, 601	m_scale_range
seq64::rtmidi_in_data, 837	seq64::lfownd, 393
m_queue_number	m_scale_speed
seq64::midi_port_info::port_info_t, 794	seq64::lfownd, 393
m_queued	m_scale_value
seq64::sequence, 1048	seq64::lfownd, 393
m_queued_tick	m_scale_wave
seq64::sequence, 1049	seq64::lfownd, 393
m_raise	m_screen_set_notepad

seq64::perform, 758	seq64::user_settings, 1116
m_screenset	m_seqarea_y
seq64::mainwid, 413	seq64::mainwid, 413
seq64::perform, 759	seq64::user_settings, 1116
m_screenset_offset	m_seqchars_x
seq64::mainwid, 415	seq64::user_settings, 1113
m_screenset_slots	m_seqchars_y
seq64::mainwid, 415	seq64::user_settings, 1113
m_scroll_offset_key	m_seqdata
seq64::seqkeys, 927	seq64::lfownd, 393
seq64::seqroll, 974	m_seqdata_wid
m_scroll_offset_ticks	seq64::seqedit, 896
seq64::seqdata, 868	seq64::seqevent, 916
seq64::seqevent, 916	m_seqedit
seq64::seqroll, 974	seq64::seqmenu, 941
seq64::seqtime, 981	m_seqedit_bgsequence
m_scroll_offset_x	seq64::user_settings, 1111
seq64::seqdata, 869	m_seqedit_key
seq64::seqevent, 916	seq64::user_settings, 1111
seq64::seqroll, 974	m_seqedit_scale
seq64::seqtime, 981	seq64::user_settings, 1111
m_scroll_offset_y	m_seqevent_wid
seq64::seqkeys, 928	seq64::seqedit, 896
seq64::seqroll, 974	m_seqkeys_wid
m_selected	seq64::seqedit, 896
seq64::event, 209	seq64::seqroll, 971
seq64::seqevent, 915	m_seqroll_wid
seq64::seqroll, 970	seq64::seqedit, 897
seq64::trigger, 1056	m_seqs
m_selecting	seq64::perform, 752
seq64::seqevent, 916	m_seqs_active
seq64::seqroll, 972	seq64::perform, 752
m_seq	m_seqs_in_set
seq64::eventedit, 239	seq64::perfnames, 675
seq64::eventslots, 255	seq64::perform, 759
seq64::lfownd, 393	seq64::user_settings, 1115
seq64::mastermidibase, 455	m_seqtime_wid
seq64::midi_alsa, 474	seq64::seqedit, 896
seq64::midibus, 614	m_sequence
seq64::seqdata, 868	seq64::editable_events, 187
seq64::seqedit, 893	seq64::midi_container, 500
seq64::seqevent, 915	m_sequence_active
seq64::seqevent, 915 seq64::seqkeys, 927	m_sequence_active seq64::perfnames, 675
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782 seq64::seqevent, 915	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782 seq64::seqevent, 915 m_seq_number	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759 m_sequence_max
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782 seq64::seqevent, 915 m_seq_number seq64::sequence, 1049	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759 m_sequence_max seq64::perfnames, 675
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782 seq64::seqevent, 915 m_seq_number seq64::sequence, 1049 m_seqarea_seq_x	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759 m_sequence_max seq64::perfnames, 675 seq64::perform, 759
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782 seq64::seqevent, 915 m_seq_number seq64::sequence, 1049 m_seqarea_seq_x seq64::mainwid, 413	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759 m_sequence_max seq64::perfnames, 675 seq64::perform, 759 seq64::perforl, 781
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782 seq64::seqevent, 915 m_seq_number seq64::sequence, 1049 m_seqarea_seq_x seq64::mainwid, 413 seq64::user_settings, 1116	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759 m_sequence_max seq64::perfnames, 675 seq64::perform, 759 seq64::perforll, 781 m_sequence_offset
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782 seq64::seqevent, 915 m_seq_number seq64::sequence, 1049 m_seqarea_seq_x seq64::mainwid, 413 seq64::user_settings, 1116 m_seqarea_seq_y	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759 m_sequence_max seq64::perfnames, 675 seq64::perforll, 781 m_sequence_offset seq64::perfnames, 675
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782 seq64::seqevent, 915 m_seq_number seq64::sequence, 1049 m_seqarea_seq_x seq64::mainwid, 413 seq64::user_settings, 1116 m_seqarea_seq_y seq64::mainwid, 414	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759 m_sequence_max seq64::perfnames, 675 seq64::perform, 759 seq64::perfroll, 781 m_sequence_offset seq64::perfnames, 675 seq64::perfnames, 675 seq64::perfnames, 675
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::seqevent, 915 m_seq_number seq64::sequence, 1049 m_seqarea_seq_x seq64::mainwid, 413 seq64::user_settings, 1116 m_seqarea_seq_y seq64::mainwid, 414 seq64::user_settings, 1116	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759 m_sequence_max seq64::perfnames, 675 seq64::perform, 759 seq64::perfroll, 781 m_sequence_offset seq64::perfnames, 675 seq64::perfnames, 675 seq64::perfnames, 675
seq64::seqevent, 915 seq64::seqkeys, 927 seq64::seqroll, 970 seq64::seqtime, 981 m_seq24_interaction seq64::perfroll, 782 seq64::seqevent, 915 m_seq_number seq64::sequence, 1049 m_seqarea_seq_x seq64::mainwid, 413 seq64::user_settings, 1116 m_seqarea_seq_y seq64::mainwid, 414	m_sequence_active seq64::perfnames, 675 seq64::perfroll, 781 m_sequence_count seq64::perform, 759 m_sequence_high seq64::perform, 759 m_sequence_max seq64::perfnames, 675 seq64::perform, 759 seq64::perfroll, 781 m_sequence_offset seq64::perfnames, 675 seq64::perfnames, 675 seq64::perfnames, 675

24	
seq64::eventslots, 256	seq64::perform, 757
seq64::perfnames, 674 m_show_midi	m_stats seq64::rc_settings, 813
seq64::rc_settings, 813	m_status
m show octave letters	seq64::event, 208
seq64::seqkeys, 928	seq64::midi_control, 504
m showbox	seq64::seqdata, 869
seq64::eventedit, 236	seq64::seqevent, 917
m_sigpipe	seq64::seqroll, 975
seq64::mainwnd, 431	m_sysex
m size	seq64::event, 208
seq64::midi_queue, 567	m_sysex_size
m_size_box_w	seq64::event, 208
seq64::perfroll, 779	m table
m slot	seq64::eventedit, 235
seq64::keybindentry, 344	seq64::perfedit, 661
m_slots_chars	seg64::segedit, 897
seq64::eventslots, 255	m_text_size_x
m_slots_x	seq64::mainwid, 414
seg64::eventslots, 256	m_text_size_y
m_slots_y	seq64::mainwid, 414
seq64::eventslots, 256	m text x
m_smf0_channels	seq64::user_settings, 1113
seq64::midi_splitter, 573	m text y
m_smf0_channels_count	seq64::user_settings, 1113
seq64::midi_splitter, 572	m_thru
m_smf0_main_sequence	seq64::sequence, 1048
seq64::midi_splitter, 573	m_tick
m_smf0_seq_number	seq64::maintime, 400
seq64::midi_splitter, 573	seq64::perform, 757
m_smf0_splitter	m tick end
seq64::midifile, 635	seq64::trigger, 1056
m snap	m_tick_offset
seq64::perfedit, 666	seq64::perftime, 791
seq64::perfedit, 778	m_tick_start
seq64::perftime, 792	seq64::trigger, 1056
seq64::seqedit, 892	m_ticks_per_bar
seq64::seqevent, 915	seq64::perfroll, 779
seq64::seqroll, 971	m_time_beat_width
m_snap_tick	seq64::sequence, 1050
seq64::sequence, 1050	m_time_beats_per_measure
m_song_mute	seq64::sequence, 1050
seq64::sequence, 1047	m_timearea_y
m_song_start_mode	seq64::perftime, 792
seq64::perform, 750	• •
· ·	m_timeout_connect
m_speed	seq64::mainwnd, 434
seq64::lfownd, 394	m_timestamp
m_spinbutton_bpm	seq64::event, 207
seq64::mainwnd, 433	seq64::event_list::event_key, 211
m_spinbutton_load_offset	seq64::midi_message, 555
seq64::mainwnd, 434	m_toggle_jack
m_spinbutton_ss	seq64::jack_assistant, 337
seq64::mainwnd, 434	m_toggle_play
m_standard_bpm	seq64::seqedit, 901
seq64::perfedit, 666	m_toggle_q_rec
m_start_from_perfedit	seq64::seqedit, 902
seq64::perform, 750	m_toggle_record
m_starting_tick	seq64::seqedit, 902

m_toggle_thru	m user data
seq64::seqedit, 902	seq64::rtmidi_in_data, 838
m_toggle_transpose	m_user_filename
seq64::seqedit, 894	seq64::rc_settings, 815
m tooltips	m_user_filename_alt
seq64::mainwnd, 431	seq64::rc_settings, 815
seq64::options, 643	m_using_callback
seq64::perfedit, 665	seq64::rtmidi_in_data, 838
seq64::seqedit, 900	m_v_page_increment
m_top_index	seq64::perfroll, 778
seq64::eventslots, 257	m_v_perf_page_increment
m top iterator	seq64::user settings, 1112
seq64::eventslots, 257	m_vadjust
m_total_seqs	seq64::eventedit, 235
seq64::user_settings, 1115	seq64::gui_drawingarea_gtk2, 296
m_tracks_mute_state	seq64::perfedit, 662
seq64::perform, 751	seq64::seqedit, 896
m_trans_button_press	m_value
seq64::perfroll, 780	seq64::lfownd, 394
seq64::seqroll, 974	m_vbox
m_transport_follow	seq64::seqedit, 897
seq64::perfroll, 780	m_vector_sequence
seq64::seqroll, 974	seq64::mastermidibase, 455
m_transposable	m_velocity_override
seq64::sequence, 1047	
·	seq64::user_settings, 1114
m_transpose	m_vertical_adjust
seq64::perform, 753	seq64::seqroll, 970
m_trigger_copied	m_vscroll
seq64::triggers, 1072	seq64::eventedit, 235
m_trigger_offset	seq64::perfedit, 662
seq64::sequence, 1049	m_vscroll_new
m_triggers	seq64::seqedit, 896
seq64::sequence, 1045	m_was_active_edit
seq64::triggers, 1071	seq64::perform, 753
m_type	m_was_active_main
seq64::keybindentry, 343	seq64::perform, 753
seq64::rterror, 820	m_was_active_names
m_undo_stack	seq64::perform, 753
seq64::triggers, 1072	m_was_active_perf
m_undo_vect	seq64::perform, 753
seq64::perform, 760	m_was_playing
m_us_per_quarter_note	seq64::sequence, 1047
seq64::perform, 756	m_wave
seq64::sequence, 1050	seq64::lfownd, 394
m_use_default_ppqn	m_wave_name
seq64::midi_splitter, 572	seq64::Ifownd, 394
seq64::midifile, 635	m_white
m_use_jack_polling	seq64::gui_palette_gtk2, 307
seq64::mastermidibus, 465	m_white_pixmap
m_use_more_icons	seq64::font, 262
seq64::user_settings, 1113	m_wht_key
m_use_new_font	seq64::gui_palette_gtk2, 309
seq64::font, 261	m_wht_paint
seq64::user_settings, 1111	seq64::gui_palette_gtk2, 308
m_usemidiclock	m_window
seq64::perform, 757	seq64::gui_drawingarea_gtk2, 296
m_user_callback	m_window_redraw_rate_ms
seq64::rtmidi_in_data, 838	seq64::user_settings, 1112

m_window_x	make_clock
seq64::gui_drawingarea_gtk2, 297	seq64::event, 205
seq64::gui_window_gtk2, 314	make_directory
m_window_y	seq64, 91
seq64::gui_drawingarea_gtk2, 297	make_section_name
seq64::gui_window_gtk2, 314	seq64, 101
m_with_jack_master	manual_alsa_ports
seq64::rc_settings, 813	seq64::rc_settings, 803, 807
m_with_jack_master_cond	mark
seq64::rc_settings, 813	seq64::event, 204
m_with_jack_midi	mark_all
seq64::rc_settings, 814	seq64::event_list, 220
m_with_jack_transport	mark_out_of_range
seq64::rc_settings, 813	seq64::event_list, 219
m_x	mark_selected
seq64::click, 158	seq64::event_list, 219
m_xy_offset	seq64::sequence, 1033
seq64::perfnames, 674	master_bus
m_y	seq64::perform, 694
seq64::click, 159	master_info
m y on b pixmap	seq64::midi_api, 488
seq64::font, 262	master_midi_mode
m_yellow	seq64::midi api, 488
seq64::gui palette gtk2, 307	mastermidibase
m_zoom	seq64::mastermidibase, 439
seq64::perfroll, 779	mastermidibus
seq64::seqdata, 868	seq64::mastermidibus, 458
seq64::seqedit, 892	seq64::midibase, 600
seq64::seqevent, 915	seq64::midibus, 614
seq64::seqroll, 971	seq64::rtmidi_info, 847
seq64::seqtime, 981	match
maintime	seq64::midi_control, 503
seq64::maintime, 397	seg64::midibase, 589
mainwid	max_active_set
seq64::mainwid, 404	seq64::perform, 738
mainwid_border	max sequence
seq64::user_settings, 1098, 1104	seq64::user_settings, 1097
mainwid_grid_style_t	max sets
seq64::user settings, 1092	seq64::user_settings, 1097, 1102
mainwid_spacing	max_value
seq64::user settings, 1098, 1104	seq64::midi_control, 503
mainwid x	max_zoom
seq64::user_settings, 1098	seq64::user settings, 1105
mainwid y	mc_baseline_ppqn
seq64::user_settings, 1099	seq64::user settings, 1117
mainwnd	. – •
seq64::maintime, 398	mc_max_zoom seq64::user settings, 1117
•	. –
seq64::mainwid, 412	mc_min_zoom
seq64::mainwnd, 420	seq64::user_settings, 1117
seq64::perform, 746	measures
seq64::rc_settings, 811	seq64::midi_measures, 550
seq64::seqmenu, 940	measures_to_ticks
mainwnd_cols	seq64, 83
seq64::user_settings, 1097, 1102	seq64::sequence, 999
mainwnd_key_event	measurestring_to_pulses
seq64::perform, 721	seq64, 74
mainwnd_rows	menu_mode
seq64::user_settings, 1096, 1102	seq64::keys_perform, 359

merge	seq64::mastermidibus, 464
seq64::event_list, 217	seq64::midi_jack, 534
message	seq64::midi_jack_info, 540
seq64::rtmidi_in_data, 834	midi_list
message_concatenate	seq64::midi_list, 547
seq64, 88	midi_measures
meta_string	seq64::midi_measures, 550
seq64::editable_event, 176	midi_measures_to_pulses
midi_alsa	seq64, 74
seq64::midi_alsa, 468	midi_message
midi_alsa_info	seq64::midi_message, 553
seq64::mastermidibase, 453	midi_mode
seq64::mastermidibus, 464	seq64::midi_info, 514
seq64::midi_alsa_info, 476	seq64::rtmidi_info, 841, 842
midi_api	midi_out_alsa
seq64::midi_api, 483	seq64::midi_out_alsa, 557
midi_api_name	midi_out_jack
seq64, 106	seq64::midi_out_jack, 559
midi beat width	midi port info
seq64::user_settings, 1105, 1108	seq64::midi port info, 561
midi beats per bar	midi ppqn
seq64::user_settings, 1104, 1108	seq64::user settings, 1104, 1107
midi_beats_per_minute	midi_probe
seq64::user_settings, 1104, 1108	seq64, 106
midi_buss_override	midi_queue
seq64::user_settings, 1105, 1107	seq64::midi_queue, 565
midi container	midi_splitter
seq64::event_list, 222	seq64::event_list, 222
seq64::midi_container, 493	seq64::midi_splitter, 569
seq64::triggers, 1070	midi_timing
midi_control	seq64::midi_timing, 574
seq64::midi_control, 502	midi_vector
midi_control_event	seq64::midi_vector, 579
seq64::perform, 729	midibase
midi_control_off	seq64::midibase, 586
seq64::perform, 728	midibpm
midi_control_on	seq64, 65
seq64::perform, 728	midibus
midi_control_toggle	seq64::midibus, 606, 607
seq64::perform, 727	seq64::rtmidi, 828
midi_handle	seq64::rtmidi_info, 847
seq64::midi_info, 514, 519	midibyte
midi_in_alsa	seq64, 64
seq64::midi_in_alsa, 507	midifile
midi_in_jack	seq64::event_list, 222
seq64::midi_in_jack, 509	seq64::midi_container, 499
midi_info	seq64::midifile, 618
seq64::midi_info, 514	seq64::perform, 746
midi_input_callback	seq64::triggers, 1071
seq64, 109	midilong
midi_input_test	seq64, 65
seq64, 107	midipulse
midi_jack	seq64, 65
seq64::midi_jack, 524	midishort
seq64::midi_jack_info, 544	seq64, 64
midi_jack_data	millisleep
seq64::midi_jack_data, 536	seq64, 98
midi_jack_info	min

seq64, 107	multi_client
min_value	seq64::midi_jack, 526
seq64::midi_control, 503	seq64::midi_jack_info, 540 multiply pattern
min_zoom	seq64::sequence, 1040
seq64::user_settings, 1105 mod control	musical_key
seg64::click, 158	seq64::sequence, 1040
seq64::keystroke, 385	musical_scale
mod_control_shift	seq64::sequence, 1040
seq64::click, 158	mute_all_tracks
seg64::keystroke, 385	seq64::perform, 708
mod last tick	seq64::seqmenu, 940
seq64::sequence, 1003	mute_group_offset
mod_super	seq64::perform, 741
seq64::click, 158	mute_group_tracks
seq64::keystroke, 385	seq64::perform, 731
mod_timestamp	mute_op_t
seq64::event, 198	seq64::perform, 689
modifier	mute_screenset
seq64::click, 158	seq64::perform, 709
seq64::keystroke, 385	mutex
modify	seq64::mutex, 637
seq64::perform, 691	name
seq64::sequence, 995	seq64::sequence, 1001
modify_current_event	seq64::user_instrument, 1075
seq64::eventslots, 247	seq64::user_midi_bus, 1080
motion_notify seq64::seqroll, 962	name_change_callback
mouse_action	seq64::seqedit, 884
seq64::seqedit, 888	name_to_value
mouse_action_e	seq64::editable_event, 172
seq64, 69	nc_midi_port_info
mouse_click_edit_callback	seq64::midi_info, 519
seq64::options, 642	new_current_sequence seq64::seqmenu, 935
mouse fruity callback	new file
seq64::options, 642	seq64::mainwnd, 428
mouse_mod4_callback	new open error dialog
seq64::options, 642	seq64::mainwnd, 427
mouse_seq24_callback	new sequence
seq64::options, 642	seq64::perform, 700
mouse_snap_split_callback	seq64::seqmenu, 936
seq64::options, 642	next
move	seq64::triggers, 1069
seq64::triggers, 1067	next_data_line
move_selected	seq64::configfile, 163
seq64::triggers, 1066	next_trigger
move_selected_notes	seq64::triggers, 1069
seq64::seqroll, 959	non_cc_match
seq64::sequence, 1028	seq64::event, 200
move_selected_triggers_to seq64::sequence, 1017	normal_action
move_selection_box	seq64::seqroll, 965
seq64::seqroll, 959	normalize seq64::user_settings, 1093
move_triggers	note_off_length
seq64::perform, 722	seq64::seqroll, 950
seq64::sequence, 1018	note_off_margin
moving	seq64::sequence, 1041
seq64::seqroll, 965	number
3040 1110041011, 000	

off_playing_notes	seq64::sequence, 995	seq64::seqdata, 865
off playing notes seq84:sequence, 1034 off_queued seq84:sequence, 1004 off_sequences seq84:sequence, 1004 off_sequences seq84:seqrine, 1055 offset seq84::rigger, 1055 seq84::rigger, 1055 seq84::rigger, 1055 seq84::seqrine, 1055 seq84::seqrine, 1055 seq84::ruityPerInput, 125 seq84::seqrine, 1055 seq84::rigger, 1055 seq84::rigySeqRentinput, 270 seq84::rigySeqRentinput, 270 seq84::seqrentinput, 270 seq84::seqrentinput, 270 seq84::seqrentinput, 270 seq84::riggering, 980 on_button_ress_eqwints, 1056 seq84::seqrentinput, 270 seq84::rigySeqRentinput, 270 seq84::riggering, 980 on_button_relass_eqwints, 1056 seq84::seqrentinput, 270 seq84::riggering, 1056 seq84::seqrentinput, 270 seq84::riggering, 1056 seq84::seqrentinput, 270 seq84::riggering, 1056 seq84::seqrentinput, 275 seq84::seq94:		·
seq64::sequence, 1034 off_queued seq64::sequence, 1004 off_sequences seq64::sequence, 1005 offset seq64::seqiser, 1055 on_button_press_event seq64::hostractPerInput, 265 seq64::seq24SeqEventInput, 270 seq64::seq24SeqEventInput, 285 seq64::seq24SeqEventInput, 856 sep64::seq24SeqEventInput, 856 sep64::seq4:seq4it, 889 seq64::seq4:it, 234 seq64::seq4it, 889 seq64::seqvent, 913 seq64::seq4it, 889 seq64::seqvent, 912 seq64::seqtent, 912 seq64::seqtine, 980 on_button_release_event seq64::seqvent, 912 seq64::seqtine, 980 on_delete_event seq64::seqvent, 912 seq64::seqvent, 913 seq64::seqvent, 914 seq64::seqvent, 914 seq64::seqvent, 915 seq64::seqvent, 916 seq64::seqvent, 917 seq64::seqvent, 918 se	off_playing_notes	•
off_queued	seq64::sequence, 1034	
seq64::sequence, 1004 off_sequences	off_queued	·
off_sequences	seq64::sequence, 1004	•
seq64::perform, 735 offset seq64:trigger, 1055 on_button_press_event seq64::AbstractPerfinput, 135 seq64::FruityPerfinput, 265 seq64::FruityPerfinput, 270 seq64::FruitySeqEventinput, 270 seq64::Seq24SeqEventinput, 852 seq64::seq24SeqEventinput, 856 seq64::seq18::seq24SeqEventinput, 856 seq64::perfroil, 775 seq64::perfroil, 775 seq64::perfroil, 775 seq64::seqvent, 912 seq64::seqventlinput, 270 seq64::perfroil, 775 seq64::perfroil, 661 seq64::seqvent, 912 seq64::seqvent, 912 seq64::seqvent, 912 seq64::perfroil, 775 seq64::perfroil, 776 seq64::perfroil, 775 seq64::perfqoil, 796 seq64::perfqoil, 796 seq64::perfqoil, 796 seq64::perfqoil, 796 seq		
offset	· ·	•
on_button_press_event		•
seq64::AbstractPerfinput, 135 seq64::FruitySeqFeventinput, 270 seq64::FruitySeqFeventinput, 273 seq64::Seq24SeqEventinput, 856 seq64::seq24SeqEventinput, 856 seq64::seqevent, 910 seq64::perfinames, 672 seq64::perfinames, 672 seq64::perfinames, 672 seq64::seqevent, 910 seq64::seqevent, 910 seq64::seqevent, 910 seq64::seqevent, 911 seq64::seqevent, 912 seq64::seqvent, 916 seq64::seqvent, 917 seq64::seqvent, 918 seq64::seqvent	,	seq64::perfroll, 776
seq64::FruitySeqEventInput, 270 seq64::FruitySeqEventInput, 273 seq64::Seq24Perfinput, 852 seq64::Seq24Perfinput, 856 seq64::seqrelli, 234 seq64::seqrelli,		seq64::seqedit, 889
seq64::FruitySeqEventInput, 270 seq64::FruitySeqRollInput, 273 seq64::Seq24PsqLinput, 852 seq64::Seq24PsqLinput, 856 seq64::seq24PsqLinput, 856 seq64::seqnoll, 775 seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 776 seq64::seqvent, 912 seq64::seqvent, 912 seq64::seqvent, 912 seq64::seqine, 980 on_button_release_event seq64::Seq24PerfInput, 273 seq64::Seq24Perflinput, 273 seq64::seq24Perflinput, 273 seq64::seq4SeqEventlnput, 273 seq64::seq4SeqEventlnput, 273 seq64::seq4SeqEventlnput, 273 seq64::seq4SeqEventlnput, 273 seq64::seq4SeqEventlnput, 853 seq64::seq4seqEventlnput, 856 seq64::seqtonl, 966 seq64::seqtonl, 966 seq64::seq1seq1seq1seq1seq1seq1seq1seq1seq1seq1	·	·
seq64::FruitySeqRollInput, 273 seq64::Seq24Perfinput, 856 seq64::eventslots, 253 seq64::mainwid, 410 seq64::perfroll, 775 seq64::seqtent, 912 seq64::seqtent, 912 seq64::seqtent, 980 on_button_release event seq64::FruitySeqRollInput, 273 seq64::perfrollput, 135 seq64::perfylinput, 266 seq64::seq24Perfinput, 135 seq64::perfroll, 776 seq64::perfoll, 966 seq64::FruitySeqRollInput, 273 seq64::perfollput, 273 seq64::perfollput, 853 seq64::seq24Perfinput, 856 seq64::seqede, seqed-individed and seq64::perfoll seqefe, seq	•	seq64::seqroll, 967
seq64::Seq24Perfinput, 852 seq64::Seq24SeqEventlinput, 856 seq64::eventslots, 253 seq64::mainwid, 410 seq64::perfroll, 775 seq64::perfroll, 775 seq64::seqdata, 866 seq64::seqvent, 912 seq64::seqtime, 980 on_button_release_event seq64::seq24Perfinput, 270 seq64::perfroll, 276 seq64::ruityPerfinput, 273 seq64::seq24Perfinput, 273 seq64::seq24Perfinput, 273 seq64::seq24Perfinput, 273 seq64::seq24Perfinput, 273 seq64::seq24Perfinput, 275 seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 776 seq64::perfoll, 776 seq64::perfroll, 776 seq64::perfroll, 776 seq64::perfro	· · · · · · · · · · · · · · · · · · ·	on_focus_out_event
seq64::Seq24SeqEventInput, 856 seq64::seventslots, 253 seq64::mainwid, 410 seq64::perfroil, 775 seq64::perfroil, 775 seq64::perfroil, 775 seq64::seqdata, 866 seq64::seqvent, 912 seq64::seqroil, 966 seq64::seqtime, 980 on_button_release_event seq64::Seq24SeqEventInput, 270 seq64::perfroil, 775 seq64::perfroil, 776 seq64::seq24SeqEventInput, 270 seq64::seq24SeqEventInput, 273 seq64::seq24SeqEventInput, 856 seq64::perfroil, 775 seq64::perfroil, 775 seq64::perfroil, 775 seq64::perfroil, 775 seq64::seqotala, 866 seq64::seqidala, 866 seq64::fruitySeqFoollinput, 270 seq64::seq24SeqEventInput, 853 seq64::seq4::perfroil, 775 seq64::perfroil, 775 seq64::perfroil, 775 seq64::seqvent, 912 seq64::seqotala, 866 seq64::seqvent, 912 seq64::seqotala, 866 seq64::seqvent, 912 seq64::seqvent, 913 seq64::seqvent, 918 seq64::seqvent, 919 seq64::perfoil, 661 seq64::perfoil, 66	· · · · · · · · · · · · · · · · · · ·	seq64::eventedit, 234
seq64::mainwiid, 410 seq64::perfrimes, 672 seq64::perfrime, 790 seq64::seqdeta, 866 seq64::seqevent, 912 seq64::seqevent, 912 seq64::seqevent, 92 seq64::seqevent, 980 on_button_release_event seq64::FruitySeqRollinput, 270 seq64::FruitySeqRollinput, 273 seq64::perfrime, 790 seq64::fruitySeqRollinput, 273 seq64::perfrime, 980 on_betton_release_event seq64::AbstractPerfinput, 266 seq64::FruitySeqRollinput, 270 seq64::FruitySeqRollinput, 273 seq64::perfrime, 790 seq64::seqventslots, 253 seq64::mainwiid, 410 seq64::perfrime, 790 seq64::perfrime, 790 seq64::perfrime, 790 seq64::perfrime, 790 seq64::seqevent, 912 seq64::perfrime, 790 seq64::seqdeta, 866 seq64::seqtevent, 912 seq64::seqtevent, 912 seq64::seqtevent, 912 seq64::seqtevent, 912 seq64::seqtin, 980 on_delete_event seq64::seqtin, 980 on_delete_event seq64::seqtind, 234 seq64::seqtind, 249 seq64::seqtind, 661 seq64::seqventldit, 234 seq64::seqtind, 429 seq64::seqventldit, 234 seq64::seqtind, 989 on_enter_notify_event seq64::seqventl, 989 on_enter_notify_event seq64::seqventl, 989 on_enter_notify_event seq64::seqventl, 989 on_enter_notify_event seq64::seqventl, 969 on_expose_event seq64::seqventl, 969 on_expose_event seq64::seqventl, 969 seq64::seqventl, 969 seq64::seqventl, 969 on_expose_event seq64::seqventl, 969 seq6	·	seq64::eventslots, 254
seq64::perfrolmers, 672 seq64::perfroll, 775 seq64::perfroll, 776 seq64::seqevent, 912 seq64::seqvent, 912 seq64::seqvent, 912 seq64::seqtime, 980 on_button_release_event seq64::hruitySeqEventInput, 270 seq64::FruitySeqEventInput, 273 seq64::hruitySeqEventInput, 273 seq64::seq24Perfinput, 853 seq64::seq24SeqEventInput, 856 seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 775 seq64::seqtata, 866 seq64::seqvent, 912 seq64::seqvent, 913 seq64::perfroll, 775 seq64::perfro	· · · · · · · · · · · · · · · · · · ·	seq64::lfownd, 392
seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 775 seq64::seqdevent, 910 seq64::seqdevent, 912 seq64::seqvent, 912 seq64::seqvent, 980 on_button_release_event seq64::AbstractPerflinput, 135 seq64::FruitySeqRollInput, 270 seq64::FruitySeqRollInput, 270 seq64::FruitySeqRollInput, 273 seq64::seq24Perflnput, 853 seq64::seq24Perflnput, 856 seq64::seqventslots, 253 seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 775 seq64::seqvent, 912 seq64::seqvent, 913 seq64::seqvent, 916 seq64:	•	seq64::mainwid, 411
seq64::perffule, 790 seq64::perfgule, 980 on_button_release_event seq64::AbstractPerflipput, 135 seq64::FruityPerflipput, 266 seq64::perfgule, 273 seq64::perfgule, 273 seq64::perfgule, 253 seq64::perfgule, 775 seq64::perfgule, 790 seq64::perfgule, 790 seq64::perfgule, 790 seq64::perfgule, 790 seq64::perfgule, 790 seq64::seqevent, 912 seq64::seqvent, 966 seq64::seqventedit, 234 seq64::seqventedit, 234 seq64::seqventedit, 234 seq64::seqevent, 980 on_delet_event seq64::seqevent, 912 seq64::seqevent, 912 seq64::seqventedit, 234 s	•	seq64::perfroll, 776
seq64::seqdata, 866 seq64::seqvent, 912 seq64::seqvent, 912 on_frame_down seq64::seqvent, 912 seq64::seqventslots, 254 seq64::seqvent, 966 seq64::seqventslots, 255 seq64::seqvent, 980 on_frame_end seq64::seqventslots, 255 seq64::eventslots, 255 on_button_release_event seq64::eventslots, 255 seq64::FruitySeqEventInput, 270 seq64::eventslots, 255 seq64::FruitySeqPventInput, 273 seq64::eventslots, 254 seq64::FruitySeqPventInput, 853 seq64::eventslots, 253 seq64::seq24PerInput, 856 seq64::eventslots, 254 seq64::seq24SeqEventInput, 856 seq64::eventslots, 254 seq64::seq24PerInput, 853 seq64::eventslots, 253 seq64::seq4SeqEventl, 910 seq64::performcallback, 763 seq64::perfroll, 775 seq64::perfedit, 234 seq64::perfroll, 775 seq64::perfedit, 660 seq64::seqvent, 912 seq64::seqvent, 913 seq64::seqvent, 912 seq64::seqvent, 913 seq64::seqvent, 912 seq64::seqvent, 913 seq64::seqvent, 960 seq64::seqvent, 913 seq64::seqvent, 961 seq64::seqvent, 914	• •	seq64::seqedit, 889
seq64::seqdata, 866 seq64::seqdown seq64::seqevent, 912 seq64::seqtown seq64::seqtowps, 925 seq64::seventslots, 254 seq64::seqtime, 980 on_frame_end on_button_release_event seq64::eventslots, 255 seq64::FruityPerfInput, 135 seq64::eventslots, 255 seq64::FruitySeqEventInput, 270 seq64::fruitySeqEventInput, 273 seq64::Seq24PerfInput, 853 seq64::seqtentslots, 254 seq64::seqtentslots, 253 seq64::mainwid, 430 seq64::perframes, 672 seq64::eventslots, 253 seq64::perframes, 672 seq64::eventedit, 234 seq64::seqvent, 912 seq64::seqtent, 96 seq64::seqvent, 912 seq64::seqvent, 912 seq64::seqvent, 912 seq64::seqvent, 913 seq64::seqvent, 912 seq64::seqvent, 913 seq64::seqvent, 966 seq64::seqvent, 913 seq64::seqventedit, 234 seq64::mainwnd, 430 seq64::perfedit, 661 seq64::mainwnd, 430 seq64::perfedit, 661 seq64::seqvent, 913 seq64::seqvent, 912 seq64::seqvent, 913 seq64::seqvent, 913 seq64::seqvent, 913 seq64::s	• •	seq64::seqevent, 913
seq64::seqvent, 912 seq64::seqveys, 925 seq64::seqvill, 966 seq64::seqvill, 966 on_button_release_event seq64::FruityPerfInput, 135 seq64::FruitySeqEventInput, 270 seq64::FruitySeqRollInput, 273 seq64::Seq24SeqEventInput, 853 seq64::Seq24SeqEventInput, 856 seq64::Seq24SeqEventInput, 856 seq64::perfroil, 775 seq64::perfroil, 775 seq64::perfroil, 775 seq64::seqvent, 912 seq64::seqvent, 910 seq64::seqvent, 911 seq64::seqvent, 910 seq64::seqvent,		seq64::seqroll, 967
seq64::seqroll, 966 seq64::seqroll, 966 seq64::seqtime, 980 on_button_release_event seq64::AbstractPerfInput, 135 seq64::FruitySeqRellInput, 270 seq64::FruitySeqRellInput, 273 seq64::Seq24PerfInput, 853 seq64::seqroll, 968 seq64::perfroll, 775 seq64::perfroll, 775 seq64::perfroll, 966 seq64::seqvent, 912 seq64::seqvent, 912 seq64::seqvent, 980 on_delete_event seq64::seqroll, 966 seq64::seqtime, 980 on_delete_event seq64::seqroll, 966 seq64::seqroll, 969 on_expose_event seq64::rruityPerfInput, 135 seq64::rruityPerfInput, 266 seq64::rruitySeqRollInput, 273 seq64::perfroll, 775 seq64::seqfvoll, 273 seq64::perfroll, 274 seq64::rruitySeqRollInput, 273 seq64::perfroll, 775	• •	on_frame_down
seq64::seqroil, 966 seq64::seqtime, 980 on_button_release_event seq64::AbstractPerfInput, 135 seq64::FruitySeqEventInput, 270 seq64::FruitySeqFventInput, 273 seq64::Seq24PerfInput, 853 seq64::seq24PerfInput, 856 seq64::mainwid, 410 seq64::perfroil, 775 seq64::perfroil, 775 seq64::seqvent, 912 seq64::seqvent, 912 seq64::seqventedit, 234 seq64::seqtime, 980 on_delete_event seq64::seqtedit, 889 on_enter_notify_event seq64::seqfoil, 969 on_expose_event seq64::seqroil, 969 on_expose_event seq64::seqroil, 969 on_expose_event seq64::mainwid, 409 seq64::perfroiln, 775 seq64::perfoiln, 767 seq64::perfoiln, 769 seq64::seqroil, 969 on_expose_event seq64::seqroil, 969 seq64::seqroil, 969 seq64::seqroil, 969 seq64::seqroil, 969 seq64::seqroil, 969 seq64::perfinput, 135 seq64::perfinput, 266 seq64::perfinput, 271 seq64::perfinput, 273	·	seq64::eventslots, 254
seq64::seqtime, 980 on_button_release_event seq64::AbstractPerfliput, 135 seq64::FruityPerfliput, 266 seq64::FruitySeqEventInput, 270 seq64::Seq24SeqEventInput, 273 seq64::Seq24SeqEventInput, 853 seq64::Seq24SeqEventInput, 856 seq64::eventslots, 253 seq64::perfnames, 672 seq64::perfroll, 775 seq64::perfrime, 790 seq64::seq24seqEvent, 912 seq64::seq64::seq6dat, 866 seq64::seq64::seqtent, 912 seq64::seq64::seqtent, 912 seq64::seqtime, 980 on_delete_event seq64::seqtime, 980 on_delete_event seq64::seqtime, 980 on_elete_event seq64::seqtime, 980 on_elete_event seq64::seqdeti, 889 on_enter_notify_event seq64::seqedit, 889 on_enter_notify_event seq64::seqfoll, 969 on_expose_event seq64::seqtinue, 398 seq64::mainwind, 409 seq64::perfroll, 775 seq64::perfroll, 755 seq64::perfroll, 775	· · · · · ·	on_frame_end
on_button_release_event		seq64::eventslots, 255
seq64::AbstractPerfInput, 135 seq64::eventslots, 255 seq64::FruityPerfInput, 266 seq64::eventslots, 254 seq64::FruitySeqRollInput, 273 seq64::eventslots, 254 seq64::Seq24PerfInput, 853 seq64::mainwnd, 430 seq64::eventslots, 253 seq64::mainwid, 410 seq64::perfnames, 672 seq64::eventedit, 234 seq64::perfroll, 775 seq64::eventedit, 234 seq64::perfroll, 775 seq64::perfroll, 777 seq64::seqdata, 866 seq64::perfroll, 777 seq64::seqevent, 912 seq64::seqedit, 890 seq64::seqvent, 912 seq64::seqroll, 968 seq64::seqroll, 966 seq64::seqevent, 913 seq64::seqroll, 966 seq64::seqroll, 968 on_delete_event seq64::mainwnd, 430 seq64::seqroll, 968 seq64::mainwnd, 430 seq64::seqedit, 889 on_ky_release_event seq64::seqdit, 661 seq64::seqdel, 861 seq64::seqroll, 969 seq64::seqroll, 969 on_expose_event seq64::seqroll, 969 on_expose_event seq64::seqroll, 969 on_expose_event seq64::seqroll, 969 on_expose_event	·	on_frame_home
seq64::FruityPerfInput, 266 seq64::FruitySeqEventInput, 270 seq64::FruitySeqEventInput, 273 seq64::Seq24PerfInput, 853 seq64::Seq24SeqEventInput, 856 seq64::eventslots, 253 seq64::mainwid, 410 seq64::perfnames, 672 seq64::perfroll, 775 seq64::seqdata, 866 seq64::seqedata, 866 seq64::seqevent, 912 seq64::seqteventedit, 234 seq64::seqevent, 912 seq64::seqtime, 980 on_delete_event seq64::eventedit, 234 seq64::seqroll, 968 on_key_press_event seq64::perfroll, 777 seq64::perfroll, 777 seq64::seqevent, 912 seq64::seqvent, 912 seq64::seqtime, 980 on_delete_event seq64::perfroll, 661 seq64::perfroll, 661 seq64::perfedit, 661 seq64::perfedit, 661 seq64::seqedit, 889 on_enter_notify_event seq64::seqkeys, 926 seq64::seqroll, 969 on_expose_event seq64::mainwid, 409 seq64::mainwid, 409 seq64::perfnames, 671 seq64::perfnames, 671 seq64::perfroll, 775 seq64::perfloput, 853		seq64::eventslots, 255
seq64::FruitySeqEventInput, 270 seq64::FruitySeqRollInput, 273 seq64::Seq24PerfInput, 853 seq64::Seq24SeqEventInput, 856 seq64::eventslots, 253 seq64::performcallback, 763 on_key_press_event seq64::perfedit, 660 seq64::perfedit, 660 seq64::perfedit, 660 seq64::perfedit, 660 seq64::perfedit, 660 seq64::perfedit, 660 seq64::perfedit, 661 seq64::p	·	on_frame_up
seq64::FruitySeqRollInput, 273 seq64::Seq24PerfInput, 853 seq64::Seq24SeqEventInput, 856 seq64::seq64::performcallback, 763 on_key_press_event seq64::perfriames, 672 seq64::perfroll, 775 seq64::seqdata, 866 seq64::seqvent, 912 seq64::seqtorul, 966 seq64::seqtime, 980 on_delete_event seq64::seqtiti, 234 seq64::seqtiti, 234 seq64::seqtime, 980 on_delete_event seq64::seqtiti, 234 seq64::seqtime, 980 on_delete_event seq64::seqtiti, 234 seq64::perffoil, 777 seq64::seqtime, 980 on_lete_event seq64::seqtime, 980 on_lete_event seq64::seqtiti, 234 seq64::seqtiti, 234 seq64::seqtiti, 661 seq64::seqtiti, 234 seq64::seqtiti, 234 seq64::seqtiti, 661 seq64::seqoil, 989 on_enter_notify_event seq64::seqdata, 867 seq64::seqdata, 867 seq64::seqoil, 969 on_expose_event seq64::seqroll, 969 on_expose_event seq64::mainwid, 409 seq64::perfroint, 275 seq64::seq64::perfliput, 266 seq64::perfinetiti, 234 seq64::perfinetiti, 234 seq64::perfinetiti, 234 seq64::perfinetiti, 234 seq64::seqroll, 969 on_expose_event seq64::seqroll, 969 on_expose_event seq64::mainwid, 409 seq64::perfinetiti, 236 seq64::perfinetiti, 236 seq64::perfinetiti, 236 seq64::perfinetiti, 266 seq64::perfinetiti, 266 seq64::perfinetiti, 266 seq64::perfinetiti, 273 seq64::perfinetiti, 273 seq64::perfroil, 775	•	seq64::eventslots, 254
seq64::Seq24PerfInput, 853 seq64::Seq24SeqEventInput, 856 seq64::seqe4:seqEventInput, 856 seq64::mainwid, 410 seq64::perfnames, 672 seq64::perfnames, 672 seq64::perfnames, 672 seq64::perfnames, 672 seq64::perfnames, 672 seq64::perfnames, 672 seq64::perfoll, 775 seq64::perfinpit, 790 seq64::seqdata, 866 seq64::seqevent, 912 seq64::seqevent, 912 seq64::seqevent, 912 seq64::seqroll, 966 seq64::seqroll, 966 seq64::seqtime, 980 on_delete_event seq64::perfedit, 661 seq64::perfedit, 661 seq64::seqedit, 889 on_enter_notify_event seq64::seqroll, 969 on_expose_event seq64::seqroll, 969 on_expose_event seq64::seqroll, 969 on_expose_event seq64::maintime, 398 seq64::perfnames, 671 seq64::perfnames, 671 seq64::perfnames, 671 seq64::perfnames, 671 seq64::Seq24PerfInput, 853	· · · · · · · · · · · · · · · · · · ·	·
seq64::Seq24SeqEventInput, 856 seq64::performcallback, 763 seq64::eventslots, 253 on_key_press_event seq64::mainwid, 410 seq64::eventedit, 234 seq64::perfromes, 672 seq64::eventedit, 234 seq64::perfroll, 775 seq64::perfedit, 660 seq64::perftime, 790 seq64::perfroll, 777 seq64::seqdata, 866 seq64::perfroll, 777 seq64::seqevent, 912 seq64::seqedit, 890 seq64::seqroll, 966 seq64::seqroll, 968 seq64::seqroll, 966 seq64::seqroll, 968 seq64::seqroll, 968 on_key_release_event on_delete_event seq64::mainwnd, 430 seq64::seqfiit, 661 seq64::perfedit, 661 seq64::seqedit, 889 seq64::seqdata, 867 seq64::seqedit, 889 seq64::seqkeys, 926 on_enter_notify_event seq64::seqkeys, 926 seq64::seqroll, 969 seq64::ruityPerfInput, 267 on_expose_event on_motion_notify_event seq64::ruityPerfInput, 267 seq64::FruitySeqEventInput, 271 seq64::maintime, 398 seq64::FruitySeqEventInput, 271 seq64::perfrolli, 775 seq64::FruitySeqEventInput, 273 <t< td=""><td></td><td>•</td></t<>		•
seq64::eventslots, 253 on_key_press_event seq64::mainwid, 410 seq64::eventedit, 234 seq64::perfronames, 672 seq64::keybindentry, 343 seq64::perfroll, 775 seq64::mainwnd, 430 seq64::perftime, 790 seq64::perfedit, 660 seq64::seqdata, 866 seq64::perfroll, 777 seq64::seqvent, 912 seq64::seqedit, 890 seq64::seqroll, 966 seq64::seqvent, 913 seq64::seqroll, 968 seq64::seqroll, 968 on_delete_event seq64::mainwnd, 430 seq64::seqtime, 980 on_key_release_event on_delete_event seq64::mainwnd, 430 seq64::seqroll, 968 on_leave_notify_event seq64::seqdata, 867 seq64::seqdata, 867 seq64::seqdata, 867 seq64::seqdata, 867 seq64::seqdat, 889 seq64::seqdata, 867 seq64::seqroll, 969 seq64::seqkeys, 926 on_enter_notify_event seq64::seqvent, 919 seq64::seqroll, 969 seq64::seqvent, 910 on_expose_event seq64::seqfall, 961 seq64::maintime, 398 seq64::herruityPerfInput, 266 seq64::perfnames, 671 seq64::seq6		
seq64::names, 672 seq64::keybindentry, 343 seq64::perfroll, 775 seq64::mainwnd, 430 seq64::perftime, 790 seq64::perfedit, 660 seq64::seqdata, 866 seq64::perfroll, 777 seq64::seqevent, 912 seq64::seqedit, 890 seq64::seqroll, 966 seq64::seqroll, 968 seq64::seqtime, 980 on_key_release_event on_delete_event seq64::mainwnd, 430 seq64::wentedit, 234 seq64::perfedit, 661 seq64::mainwnd, 429 on_leave_notify_event seq64::seqdata, 867 seq64::seqdata, 867 seq64::seqdit, 889 seq64::seqkeys, 926 on_enter_notify_event seq64::seqroll, 969 seq64::seqroll, 969 seq64::FruityPerfInput, 267 on_expose_event on_motion_notify_event seq64::reventslots, 253 seq64::AbstractPerfInput, 135 seq64::maintime, 398 seq64::FruityPerfInput, 266 seq64::mainwid, 409 seq64::FruitySeqEventInput, 271 seq64::perfnames, 671 seq64::FruitySeqRollInput, 273 seq64::perfroll, 775 seq64::Seq24PerfInput, 853		
seq64::perfroll, 775 seq64::perfedit, 660 seq64::perftime, 790 seq64::perfedit, 660 seq64::seqdata, 866 seq64::perfroll, 777 seq64::seqevent, 912 seq64::seqedit, 890 seq64::seqroll, 966 seq64::seqroll, 968 seq64::seqtime, 980 on_key_release_event on_delete_event seq64::mainwnd, 430 seq64::wentedit, 234 seq64::perfedit, 661 seq64::mainwnd, 429 on_leave_notify_event seq64::seqdata, 867 seq64::seqdata, 867 seq64::seqedit, 889 seq64::seqkeys, 926 on_enter_notify_event seq64::seqvel, 969 seq64::seqroll, 969 seq64::FruityPerfInput, 267 on_expose_event on_motion_notify_event seq64::wentslots, 253 seq64::AbstractPerfInput, 135 seq64::maintime, 398 seq64::FruityPerfInput, 266 seq64::mainwid, 409 seq64::FruitySeqEventInput, 271 seq64::perfnames, 671 seq64::FruitySeqRollInput, 273 seq64::perfroll, 775 seq64::Seq24PerfInput, 853	seq64::mainwid, 410	•
seq64::perftime, 790 seq64::perfedit, 660 seq64::seqdata, 866 seq64::perfroll, 777 seq64::seqevent, 912 seq64::seqedit, 890 seq64::seqtoll, 966 seq64::seqroll, 968 seq64::seqtime, 980 on_key_release_event on_delete_event seq64::mainwnd, 430 seq64::eventedit, 234 seq64::perfedit, 661 seq64::perfedit, 661 seq64::perfedit, 661 seq64::perfedit, 661 <td>seq64::perfnames, 672</td> <td>seq64::keybindentry, 343</td>	seq64::perfnames, 672	seq64::keybindentry, 343
seq64::seqdata, 866 seq64::perfroll, 777 seq64::seqevent, 912 seq64::seqedit, 890 seq64::seqkeys, 925 seq64::seqevent, 913 seq64::seqroll, 966 seq64::seqroll, 968 seq64::seqtime, 980 on_key_release_event on_delete_event seq64::mainwnd, 430 seq64::meinwnd, 429 seq64::perfedit, 661 seq64::meinwnd, 429 seq64::seqdata, 867 seq64::seqedit, 889 seq64::seqdata, 867 seq64::seqedit, 889 seq64::seqkeys, 926 on_enter_notify_event seq64::seqroll, 969 seq64::seqroll, 969 seq64::FruityPerfInput, 267 on_expose_event on_motion_notify_event seq64::maintime, 398 seq64::AbstractPerfInput, 135 seq64::mainwid, 409 seq64::FruitySeqEventInput, 266 seq64::mainwid, 409 seq64::FruitySeqEventInput, 271 seq64::perfroll, 775 seq64::Seq24PerfInput, 853	seq64::perfroll, 775	•
seq64::seqevent, 912 seq64::seqedit, 890 seq64::seqroll, 966 seq64::seqroll, 968 seq64::seqtime, 980 on_key_release_event on_delete_event seq64::mainwnd, 430 seq64::eventedit, 234 seq64::perfedit, 661 seq64::perfedit, 661 seq64::seqdata, 867 seq64::seqedit, 889 seq64::seqkeys, 926 on_enter_notify_event seq64::seqroll, 969 seq64::seqroll, 969 on_left_button_pressed seq64::seqroll, 969 seq64::FruityPerfInput, 267 on_expose_event on_motion_notify_event seq64::maintime, 398 seq64::FruityPerfInput, 266 seq64::mainwid, 409 seq64::FruitySeqEventInput, 271 seq64::perfroll, 775 seq64::FruitySeqRollInput, 273 seq64::perfroll, 775 seq64::Seq24PerfInput, 853	seq64::perftime, 790	• •
seq64::seqkeys, 925 seq64::seqroll, 966 seq64::seqtime, 980 on_key_release_event on_delete_event seq64::seqtime, 234 seq64::seqtime, 429 seq64::seqdetit, 661 seq64::seqdetit, 661 seq64::seqdetit, 889 on_leave_notify_event seq64::seqkeys, 926 seq64::seqkeys, 926 seq64::seqroll, 969 on_expose_event seq64::seqroll, 969 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::maintime, 398 seq64::perfnames, 671 seq64::perfroll, 775 seq64::Seq24PerfInput, 853	seq64::seqdata, 866	• •
seq64::seqroll, 966 seq64::seqtime, 980 on_delete_event seq64::eventedit, 234 seq64::mainwnd, 429 seq64::seqdit, 889 on_enter_notify_event seq64::seqroll, 969 seq64::seqroll, 969 on_expose_event seq64::rruityPerfInput, 267 on_expose_event seq64::maintime, 398 seq64::mainwid, 409 seq64::perfedit, 661 seq64::perfedit, 661 seq64::seqroll, 969 on_lett_button_pressed seq64::FruityPerfInput, 135 seq64::rruityPerfInput, 267 seq64::rruityPerfInput, 266 seq64::rruityPerfInput, 266 seq64::perfnames, 671 seq64::perfnames, 671 seq64::perfroll, 775	seq64::seqevent, 912	
seq64::seqtime, 980 on_delete_event seq64::wentedit, 234 seq64::mainwnd, 429 seq64::perfedit, 661 seq64::seqdata, 867 seq64::seqedit, 889 on_enter_notify_event seq64::seqkeys, 926 seq64::seqroll, 969 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::maintime, 398 seq64::perfedit, 661 seq64::perfedit, 661 seq64::perfedit, 661 seq64::seqkeys, 926 on_letr_button_pressed seq64::FruityPerfInput, 267 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::FruityPerfInput, 266 seq64::perfnames, 671 seq64::perfnames, 671 seq64::perfroll, 775 seq64::Seq24PerfInput, 853	seq64::seqkeys, 925	·
on_delete_event seq64::mainwnd, 430 seq64::mainwnd, 429 seq64::perfedit, 661 seq64::perfedit, 661 seq64::seqdata, 867 seq64::seqedit, 889 seq64::seqkeys, 926 on_enter_notify_event seq64::seqkeys, 926 seq64::seqroll, 969 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::perfoll, 775 seq64::perfoll, 775 seq64::perfoll, 775 seq64::perfoll, 775 seq64::perfoll, 775 seq64::perfedit, 661 seq64::perfedit, 661 seq64::seqdata, 867 seq64::seqkeys, 926 seq64::seqroll, 969 on_left_button_pressed seq64::pruityPerflnput, 267 on_motion_notify_event seq64::perflnput, 135 seq64::perflnput, 266 seq64::perflnput, 266 seq64::perflnput, 271 seq64::perflnput, 273 seq64::perflnput, 853	seq64::seqroll, 966	• • •
seq64::eventedit, 234 seq64::mainwnd, 429 seq64::perfedit, 661 seq64::seqdata, 867 seq64::seqedit, 889 seq64::seqkeys, 926 on_enter_notify_event seq64::seqkeys, 926 seq64::seqroll, 969 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::perfedit, 661 seq64::perfedit, 661 seq64::seqroll, 967 on_expose_event seq64::perfedit, 661 seq64::seqroll, 969 on_leave_notify_event seq64::seqroll, 969 on_left_button_pressed seq64::FruityPerfInput, 267 on_expose_event seq64::eventslots, 253 seq64::ruityPerfInput, 135 seq64::FruityPerfInput, 266 seq64::FruitySeqEventInput, 271 seq64::perfnames, 671 seq64::perfoll, 775 seq64::Seq24PerfInput, 853	·	— <i>i</i> — —
seq64::mainwnd, 429 seq64::perfedit, 661 seq64::seqdata, 867 seq64::seqedit, 889 seq64::seqkeys, 926 on_enter_notify_event seq64::seqkeys, 926 seq64::seqroll, 969 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::perfnames, 671 seq64::perfroll, 775 on_leave_notify_event seq64::seqdata, 867 seq64::seqkeys, 926 on_left_button_pressed seq64::FruityPerfInput, 267 on_motion_notify_event seq64::AbstractPerfInput, 135 seq64::FruityPerfInput, 266 seq64::FruitySeqEventInput, 271 seq64::perfnames, 671 seq64::Seq24PerfInput, 853		•
seq64::seqfati, 661 seq64::seqdata, 867 seq64::seqdati, 889 seq64::seqkeys, 926 on_enter_notify_event seq64::seqkeys, 926 seq64::seqroll, 969 on_left_button_pressed seq64::seqroll, 969 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::pruityPerflnput, 266 seq64::pruityPerflnput, 266 seq64::pruityPerflnput, 266 seq64::pruityPerflnput, 271 seq64::pruitySeqEventlnput, 271 seq64::pruitySeqRollInput, 273 seq64::pruitySeqRollInput, 273 seq64::pruitySeqRollInput, 853	•	• •
seq64::seqedit, 889 on_enter_notify_event seq64::seqkeys, 926 seq64::seqkeys, 926 seq64::seqkeys, 926 seq64::seqroll, 969 on_left_button_pressed seq64::ruityPerfInput, 267 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::pruityPerfInput, 266 seq64::pruityPerfInput, 266 seq64::pruityPerfInput, 271 seq64::pruitySeqEventInput, 271 seq64::pruitySeqRollInput, 273 seq64::pruitySeqRollInput, 273 seq64::pruitySeqRollInput, 853	•	
on_enter_notify_event seq64::seqroll, 969 seq64::seqkeys, 926 seq64::seqroll, 969 on_left_button_pressed seq64::ruityPerfInput, 267 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::pruityPerfInput, 266 seq64::pruityPerfInput, 266 seq64::pruityPerfInput, 271 seq64::perfnames, 671 seq64::pruitySeqRollInput, 273 seq64::pruitySeqRollInput, 273 seq64::pruitySeqRollInput, 853	• •	•
seq64::seqkeys, 926 seq64::seqroll, 969 on_left_button_pressed seq64::FruityPerfInput, 267 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::pruityPerfInput, 266 seq64::pruityPerfInput, 266 seq64::pruityPerfInput, 271 seq64::perfnames, 671 seq64::perfnames, 671 seq64::perfroll, 775 seq64::Seq24PerfInput, 853	·	
seq64::seqroll, 969 on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::perfnames, 671 seq64::perfroll, 775 seq64::perfoll, 775 seq64::Seq24PerfInput, 853		•
on_expose_event seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::perfnames, 671 seq64::perfroll, 775 seq64::perfroll, 775 on_motion_notify_event seq64::AbstractPerfInput, 135 seq64::FruityPerfInput, 266 seq64::FruitySeqEventInput, 271 seq64::FruitySeqRollInput, 273 seq64::Seq24PerfInput, 853	· · · · · ·	
seq64::eventslots, 253 seq64::maintime, 398 seq64::mainwid, 409 seq64::perfnames, 671 seq64::perfnames, 671 seq64::perfroll, 775 seq64::Seq24Perflnput, 853	·	· · · · · · · · · · · · · · · · · · ·
seq64::maintime, 398seq64::FruityPerfInput, 266seq64::mainwid, 409seq64::FruitySeqEventInput, 271seq64::perfnames, 671seq64::FruitySeqRollInput, 273seq64::perfroll, 775seq64::Seq24PerfInput, 853	_ · _	
seq64::mainwid, 409seq64::FruitySeqEventInput, 271seq64::perfnames, 671seq64::FruitySeqRollInput, 273seq64::perfroll, 775seq64::Seq24PerfInput, 853	•	
seq64::perfnames, 671seq64::FruitySeqRollInput, 273seq64::perfroll, 775seq64::Seq24PerfInput, 853	•	
seq64::perfroll, 775 seq64::Seq24PerfInput, 853	•	
	·	
seqo4::perrtime, 789 seq64::Seq24SeqEventInput, 856	·	· · · · · · · · · · · · · · · · · · ·
	seqo4permine, 769	seqo4seqz4seqEventinput, 856

seq64::mainwid, 411	seq64::midi_jack, 529
seq64::perfroll, 776	seq64::midi_out_jack, 560
seq64::seqdata, 866	open_client_impl
seq64::seqevent, 913	seq64::midi_jack, 527
seq64::seqkeys, 926	open_file
seq64::seqroll, 967	seq64::mainwnd, 421
on_move_down	open_performance_edit
seq64::eventslots, 254	seq64::mainwnd, 424
on_move_up	open_performance_edit_2
seq64::eventslots, 254	seq64::mainwnd, 424
on_queued	openmidi_api
seq64::sequence, 1004	seq64::rtmidi_in, 832
on_realize	seq64::rtmidi_info, 847
seq64::eventedit, 233	seq64::rtmidi_out, 851
seq64::eventslots, 253	operator<
seq64::gui_drawingarea_gtk2, 296	seq64::event, 194
seq64::gui_window_gtk2, 313	seq64::event_list::event_key, 210
seq64::maintime, 398	seq64::trigger, 1053
seq64::mainwid, 409	operator=
seq64::mainwnd, 430	seq64::automutex, 138
seq64::perfedit, 660 seq64::perfnames, 671	seq64::click, 156 seq64::editable_event, 172
seq64::perfroll, 774	-
seq64::perftime, 789	seq64::editable_events, 183
seq64::seqdata, 865	seq64::event, 193 seq64::event_list, 214
seq64::seqedit, 888	seq64::gui_drawingarea_gtk2, 284
seq64::seqevent, 911	seq64::keystroke, 383
seq64::seqkeys, 924	seq64::maintime, 397
seq64::seqmenu, 940	seq64::rc_settings, 800
seq64::seqroll, 966	seq64::sequence, 993
seq64::seqtime, 980	seq64::triggers, 1060
on right button pressed	seq64::user_instrument, 1074
seq64::FruityPerfInput, 267	seq64::user_midi_bus, 1080
on scroll event	seq64::user_settings, 1093
seq64::eventslots, 254	operator[]
seq64::perfnames, 673	seq64::midi_message, 553
seq64::perfroll, 776	options
seq64::seqdata, 867	seq64::keybindentry, 343
seq64::seqedit, 889	seq64::keys_perform, 369
seq64::seqkeys, 926	seq64::options, 640
seq64::seqroll, 968	seg64::perform, 747
on_set_focus	seq64::rc_settings, 811
seq64::eventedit, 233	options dialog
seq64::seqedit, 889	seq64::mainwnd, 426
on size allocate	optionsfile
seq64::eventslots, 254	seq64::keys perform, 369
seq64::perfnames, 673	seq64::optionsfile, 646
seq64::perfroll, 776	seq64::perform, 746
seq64::perftime, 790	seq64::rc_settings, 810
seq64::seqdata, 867	orange
seq64::seqevent, 914	seq64::gui_palette_gtk2, 304
seq64::seqkeys, 927	output
seq64::seqroll, 969	seq64::jack_assistant, 324
seq64::seqtime, 980	output_func
on_size_request	seq64::perform, 709
seq64::perfroll, 777	output_ports
open_client	seq64::midi_info, 514
seq64::midi_in_jack, 510	output_thread_func

seq64, 99	seq64::perfedit, 660
seq64::perform, 747	seq64::perform, 715
	perf
padded_height	seq64::gui_drawingarea_gtk2, 285
seq64::font, 261	seq64::gui_window_gtk2, 312
page_decrement_beats_per_minute seq64::perform, 717	seq64::options, 640
page_increment_beats_per_minute	perf_h_page_increment
seq64::perform, 717	seq64::user_settings, 1101, 1106
page_movement	perf_modify
seq64::eventslots, 251	seq64::eventedit, 231
page_topper	perf_v_page_increment
seq64::eventslots, 251	seq64::user_settings, 1101, 1106 perfedit
pager_index	seq64::perfedit, 653
seq64::eventslots, 245	seq64::perfnames, 673
paint	seq64::perform, 747
seq64::event, 203	seq64::perfroll, 778
parent	seq64::perftime, 791
seq64::editable_event, 172	perfnames
seq64::jack_assistant, 319	seq64::perfnames, 669
parent_bus	perform
seq64::midi_api, 488	seq64::keys_perform, 369
parse	seq64::mastermidibase, 453
seq64::configfile, 164 seq64::midifile, 619	seq64::perform, 690
seq64::optionsfile, 646	seq64::sequence, 1044
seq64::userfile, 1119	perfroll
parse_command_line_options	seq64::AbstractPerfInput, 136
seq64, 87	seq64::FruityPerfInput, 268
seq64::rc_settings, 811	seq64::Seq24PerfInput, 854
parse_options_files	seq64::perform, 747
seq64, 86	seq64::perfroll, 768
parse_prop_header	perfroll_key_event
seq64::midifile, 622	seq64::perform, 721
parse_proprietary_track	perftime
seq64::midifile, 623	seq64::perftime, 785
parse_smf_0	pixel_to_tick
seq64::midifile, 621	seq64::perftime, 788 play
parse_smf_1	seq64::busarray, 143
seq64::midifile, 622	seq64::mastermidibase, 443
partial_assign seq64::sequence, 993	seq64::midibase, 595
pass sysex	seg64::perform, 737
seq64::rc settings, 802, 806	seq64::sequence, 1008
paste	seq64::triggers, 1062
seq64::triggers, 1065	play_change_callback
paste_selected	seq64::seqedit, 884
seq64::sequence, 1025	play_note_off
paste_trigger	seq64::sequence, 1034
seq64::sequence, 1016	play_note_on
pattern_edit	seq64::sequence, 1034
seq64::keys_perform, 358	play_queue
pause	seq64::sequence, 1008
seq64::keys_perform, 358	playback_key_event
seq64::sequence, 1035	seq64::perform, 721
pause_key	pointer_position
seq64::perform, 716	seq64::keys_perform, 360, 361
pause_playing	poll_for_midi
seq64::mainwnd, 423	seq64::busarray, 146

seq64::mastermidibase, 446	position_increment
seq64::midibase, 593	seq64::midi_container, 495
рор	position_jack
seq64::midi_queue, 565	seq64::perform, 735
pop_front	position_reset
seq64::midi_queue, 566	seq64::midi_container, 495
pop_redo	pow2
seq64::sequence, 997	seq64::midifile, 624
seq64::triggers, 1061	ppqn
pop_trigger_redo	seq64::mainwnd, 422
seq64::perform, 723	seq64::midi_info, 515
seq64::sequence, 997	seq64::midi_splitter, 571
pop_trigger_undo	seq64::midi_timing, 576
seq64::perform, 723	seq64::midibase, 589
seq64::sequence, 997	seq64::midifile, 621
pop_undo	seq64::perform, 691
seq64::sequence, 997	seq64::rtmidi_info, 844
seq64::triggers, 1061	ppqn_is_valid
populate_menu_edit	seq64, 92
seq64::mainwnd, 429	print
populate_menu_file	seq64::busarray, 144
seq64::mainwnd, 429	seq64::event, 207
populate_menu_help	seq64::event_list, 221
seq64::mainwnd, 429	seq64::mastermidibase, 444
populate_menu_view	seq64::midibase, 596
seq64::mainwnd, 429	seq64::sequence, 1008
popup_event_menu	seq64::triggers, 1061
seq64::seqedit, 886	print_keys
popup_menu	seq64::rc_settings, 803, 807
seq64::perfedit, 659	print_message
seq64::seqedit, 886	seq64::rterror, 820
seq64::seqmenu, 936	print_triggers
popup_midibus_menu	seq64::perform, 702
seq64::seqedit, 886	seq64::sequence, 1008
popup_midich_menu	priority
seq64::seqedit, 887	seq64::rc_settings, 801, 806
popup_sequence_menu	private bus
seq64::seqedit, 887	seq64::user_settings, 1108
popup_tool_menu	private instrument
seq64::seqedit, 887	seq64::user_settings, 1108
port_exit	process_events
seq64::busarray, 145	seq64::lash, 388
seq64::mastermidibase, 442	progress_bar_colored
port handle	seq64::user_settings, 1101, 1106
seq64::midi_jack, 527	progress bar thick
port_list	seq64::user_settings, 1101, 1106
seq64::mastermidibus, 463	progress_color
seq64::midi_info, 518	seq64::gui_palette_gtk2, 303
seq64::rtmidi_info, 846	prop_item_size
port_name	seq64::midifile, 631
seq64::midibase, 588, 597	pulse_length_us
port_settings	seq64, 79
seq64::mastermidibase, 450	pulses_per_measure
port_start	seq64, 82
seq64::mastermidibase, 442	pulses_to_measurestring
position	seq64, 72
seq64::jack_assistant, 323	pulses_to_midi_measures
seq64::midi_container, 495	seq64, 73
	= = 1 = - / - =

pulses_to_string	read_varinum
seq64, 72	seq64::midifile, 626
pulses_to_timestring	record_change_callback
seq64, 73, 74	seq64::seqedit, 884
push	red
seq64::midi_message, 554	seq64::gui_palette_gtk2, 304
push_back	redo
seq64::event_list, 216	seq64::perfedit, 659
push_quantize	redo_callback
seq64::sequence, 1039	seq64::seqedit, 885
push_trigger_undo	redraw
seq64::perform, 722	seq64::mainwid, 406
seq64::sequence, 997 push_undo	seq64::perfnames, 671
seq64::sequence, 996	seq64::seqdata, 861
seq64::triggers, 1061	seq64::seqevent, 907
put	seq64::seqmenu, 938
seq64::midi_container, 494	seq64::seqroll, 954
seq64::midi_list, 548	seq64::seqtime, 978
seq64::midi_vector, 580	redraw_dirty_sequences
put_event_on_bus	seq64::perfnames, 670
seq64::sequence, 1042	seq64::perfroll, 770
3040 1304101100, 10 12	redraw_events
q_rec_change_callback	seq64::seqroll, 954
seq64::seqedit, 884	redraw_period_ms
quantize_events	seq64::gui_window_gtk2, 312
seq64::sequence, 1038	redraw_progress
query_save_changes	seq64::perfroll, 770
seq64::mainwnd, 427	ref_midi_port_info
queue	seq64::midi_info, 520
seq64::keys_perform, 353, 354	register_port
seq64::rtmidi_in_data, 834	seq64::midi_jack, 529
queue_number	remote_port_name
seq64::midi_info, 518	seq64::midi_jack, 526
seq64::midibase, 591	remove seq64::businfo, 150
seq64::rtmidi_info, 844	seq64::editable_events, 186
quit	seq64::event list, 217
seq64::gui_assistant, 276	seq64::sequence, 1043, 1044
seq64::gui_assistant_gtk2, 279	seq64::triggers, 1064
seq64::gui_window_gtk2, 312	remove_all
	seq64::sequence, 1044
rc	remove_marked
seq64, 99	seq64::event_list, 220
rc_error_dialog seq64::mainwnd, 421	seq64::sequence, 1033
rc_settings	remove selected
seg64::rc settings, 799	seq64::sequence, 1033
read_byte	seq64::triggers, 1065
seq64::midifile, 625	remove_seqedit
read_byte_array	seq64::seqmenu, 937
seq64::midifile, 627	render number
read_long	seq64::seqdata, 865
seq64::midifile, 625	render_string
read_seq_number	seq64::gui_drawingarea_gtk2, 289
seq64::midifile, 629	render_string_on_drawable
read short	seq64::font, 260
seq64::midifile, 625	render_string_on_pixmap
read_track_name	seq64::gui_drawingarea_gtk2, 289
seq64::midifile, 628	replace
	- I

04 19 11 100	24.405
seq64::editable_events, 186	seq64, 125
seq64::keys_perform, 353	s_build_chord_generator
replacement_port	seq64, 126
seq64::busarray, 147 reposition	s_build_edit_highlight
seq64::perform, 699	seq64, 126
reset	s_build_follow_progress
seq64::mainwid, 405	seq64, 127
seq64::perftime, 786	s_build_highlight_empty seq64, 125
seg64::segdata, 861	s_build_jack_session
seq64::seqevent, 907	seq64, 126
seq64::seqkeys, 924	s_build_jack_support
seq64::seqroll, 953	seq64, 126
seq64::seqtime, 978	s_build_lash_support
reset_draw_marker	seq64, 126
seq64::sequence, 1035	s_build_midi_vector
reset_draw_trigger_marker	seq64, 127
seq64::sequence, 1035	s_build_pause_support
seq64::triggers, 1069	seq64, 126
reset_sequences	s_build_portmidi_support
seq64::perform, 736	seq64, 125
restart_sysex	s_build_rtmidi_support
seq64::event, 202	seq64, 125
restore_playing_state	s_build_solid_grid
seq64::perform, 711	seq64, 127
RevSlotMap	s_build_timesig_tempo
seq64::keys_perform, 351 reveal_alsa_ports	seq64, 127
seq64::rc_settings, 803, 807	s_build_use_event_map
rewind	seq64, 126
seq64::keys_perform, 360	s_character_mapping
seq64::perfedit, 655	seq64, 128
seq64::perform, 698	s_debug_mode
rterror	seq64, 128
seq64::rterror, 819	s_event_editor
rterror_callback	seq64, 126 s_global_lash_driver
seq64, 65	_ -
rtmidi	seq64, 128 s handlesize
seq64::rtmidi, 823	seq64, 130, 131
rtmidi_api	s help 1a
seq64, 70	seq64, 124
rtmidi_callback_t	s_help_1b
seq64, 65	seq64, 125
rtmidi_in	s help 2
seq64::rtmidi_in, 831	seq64, 125
seq64::rtmidi_info, 848 rtmidi in data	s_help_3
seq64::rtmidi_in_data, 834	seq64, 125
rtmidi info	s_help_4
seq64::midi_info, 520	seq64, 125
seq64::rc_settings, 811	s_jitter_amount
seq64::rtmidi_info, 841	seq64, 130
rtmidi_out	s_seq32_jack_support
seq64::rtmidi_info, 848	seq64, 127
seq64::rtmidi_out, 850	s_seq32_lfo_support
	seq64, 128
s_arg_list	s_seq32_menu_buttons
seq64, 124	seq64, 128
s_build_alsamidi_support	s_seq32_transport

seq64, 127	select_even_or_odd_notes
s_seq32_transpose	seq64::sequence, 1023
seq64, 128	select_event
s_statistics_support	seq64::eventslots, 248
seq64, 127	select_event_handle
s_status_pairs	seq64::sequence, 1022
seq64, 128	select_events
s_strip_empty_mutes	seq64::sequence, 1020, 1021
seq64, 127	select_fg_bg_colors
save_clock	seq64::mainwid, 409
seq64::mastermidibase, 452	select_group_mute
save_events	seq64::perform, 733
seq64::editable_events, 183	select_linked
seq64::eventslots, 248	seq64::sequence, 1022
save_file	select_note_events
seq64::mainwnd, 428	seq64::sequence, 1020
save_input	select_trigger
seq64::mastermidibase, 453	seq64::sequence, 1014
save_playing_state	selected
seq64::perform, 711	seq64::trigger, 1055
save_user_config	selected_api
seq64::user_settings, 1101, 1102	seq64::rtmidi_info, 846
scale_lfo_change	selected_trigger_end
seq64::lfownd, 392	seq64::sequence, 1017
screenset_dn	selected_trigger_start
seq64::keys_perform, 355	seq64::sequence, 1017
screenset_up	selecting
seq64::keys_perform, 355	seq64::seqroll, 965
ecrall hadillet	send_message
scroll_hadjust	
seq64::gui_drawingarea_gtk2, 294	seq64::midi_out_jack, 559
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312	seq64::midi_out_jack, 559 seq
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 313	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 313 scroll_vset	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_drawingarea_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_drawingarea_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 select	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_drawingarea_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 select seq64::event, 204	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 select seq64::event, 204 seq64::triggers, 1064	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 select seq64::event, 204 seq64::triggers, 1064 select_action	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_size, 123
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_drawingarea_gtk2, 295 seq64::gui_drawingarea_gtk2, 313 select seq64::event, 204 seq64::triggers, 1064 select_action seq64::seqroll, 965	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_table, 123
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_drawingarea_gtk2, 295 seq64::gui_drawingarea_gtk2, 313 select seq64::event, 204 seq64::riggers, 1064 select_action seq64::seqroll, 965 select_action_e	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_table, 123 c_chord_table, 123 c_chord_table_text, 123
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_drawingarea_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_drawingarea_gtk2, 313 select seq64::event, 204 seq64::event, 204 seq64::riggers, 1064 select_action seq64::seqroll, 965 select_action_e seq64::sequence, 992	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_size, 123 c_chord_table, 123 c_chord_table_text, 123 c_chord_text, 123
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_drawingarea_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 select seq64::event, 204 seq64::triggers, 1064 select_action seq64::seqroll, 965 select_action_e seq64::sequence, 992 select_all	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_table, 123 c_chord_table_text, 123 c_chord_text, 123 c_controller_names, 109
seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 select seq64::event, 204 seq64::triggers, 1064 select_action seq64::seqroll, 965 select_action_e seq64::sequence, 992 select_all seq64::event_list, 221	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_table, 123 c_chord_table, 123 c_chord_text, 123 c_controller_names, 109 c_interval_text, 123
seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_window_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 select seq64::event, 204 seq64::event, 204 seq64::seqroll, 965 select_action seq64::seqroll, 965 select_action_e seq64::sequence, 992 select_all seq64::event_list, 221 seq64::sequence, 1024	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_table, 123 c_chord_table, 123 c_chord_table_text, 123 c_controller_names, 109 c_interval_text, 123 c_key_text, 123
seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::gui_drawingarea_gtk2, 294 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_drawingarea_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_drawingarea_gtk2, 313 select seq64::event, 204 seq64::event, 204 seq64::riggers, 1064 select_action seq64::seqroll, 965 select_action_e seq64::sequence, 992 select_all seq64::sequence, 1024 select_all_notes	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_table, 123 c_chord_table, 123 c_chord_text, 123 c_controller_names, 109 c_interval_text, 123 c_mainwid_x, 130
seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::gui_drawingarea_gtk2, 294 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_drawingarea_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_window_gtk2, 313 select seq64::event, 204 seq64::event, 204 seq64::seqroll, 965 select_action seq64::seqroll, 965 select_action_e seq64::sequence, 992 select_all seq64::sequence, 1024 select_all_notes seq64::sequence, 1023	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_table, 123 c_chord_table, 123 c_chord_text, 123 c_chord_text, 123 c_controller_names, 109 c_interval_text, 123 c_key_text, 123 c_mainwid_x, 130 c_mainwid_y, 130
seq64::gui_window_gtk2, 312 scroll_hset seq64::gui_window_gtk2, 313 scroll_offset_x seq64::seqroll, 962 scroll_offset_y seq64::gui_drawingarea_gtk2, 294 scroll_offset_y seq64::seqroll, 964 scroll_vadjust seq64::gui_drawingarea_gtk2, 294 seq64::gui_drawingarea_gtk2, 313 scroll_vset seq64::gui_drawingarea_gtk2, 295 seq64::gui_drawingarea_gtk2, 313 select seq64::event, 204 seq64::event, 204 seq64::riggers, 1064 select_action seq64::seqroll, 965 select_action_e seq64::sequence, 992 select_all seq64::sequence, 1024 select_all_notes	seq64::midi_out_jack, 559 seq seq64::midi_alsa_info, 477 Seq24PerfInput seq64::Seq24PerfInput, 852 seq64::perfroll, 777 seq64::triggers, 1071 Seq24SeqEventInput seq64::Seq24SeqEventInput, 855 seq64::seqevent, 914 seq64, 49 adjustment_dummy, 102 bpm_from_tempo_us, 79 build_details, 87 bussbyte, 64 c_backsequence, 117 c_bpmtag, 116 c_chord_number, 123 c_chord_table, 123 c_chord_table, 123 c_chord_text, 123 c_controller_names, 109 c_interval_text, 123 c_mainwid_x, 130

c_midi_control_16, 120	draw_type_t, 69
c_midi_control_17, 120	EVENT_AFTERTOUCH, 110
c_midi_control_18, 120	EVENT_ANY, 109
c_midi_control_19, 120	EVENT_CHANNEL_PRESSURE, 110
c_midi_control_bpm_dn, 118	EVENT_CLEAR_CHAN_MASK, 113
c_midi_control_bpm_page_dn, 120	EVENT_CONTROL_CHANGE, 110
c_midi_control_bpm_page_up, 120	EVENT_GET_CHAN_MASK, 113
c_midi_control_bpm_up, 118	EVENT_MIDI_ACTIVE_SENS, 113
c_midi_control_mod_glearn, 119	EVENT_MIDI_CLOCK, 112
c_midi_control_mod_gmute, 119	EVENT_MIDI_CONTINUE, 112
c_midi_control_mod_queue, 119	EVENT_MIDI_META, 113
c_midi_control_mod_replace, 118	EVENT_MIDI_QUARTER_FRAME, 111
c_midi_control_mod_snapshot, 118	EVENT_MIDI_RESET, 113
c_midi_control_play_ss, 119	EVENT_MIDI_SONG_F4, 111
c_midi_control_playback, 119	EVENT_MIDI_SONG_F5, 111
c_midi_control_record, 119	EVENT_MIDI_SONG_F9, 112
c_midi_control_solo, 119	EVENT_MIDI_SONG_FD, 113
c_midi_control_ss_dn, 118	EVENT_MIDI_SONG_POS, 111
c_midi_control_ss_up, 118	EVENT_MIDI_SONG_SELECT, 111
c_midi_control_thru, 120	EVENT_MIDI_START, 112
c_midi_controls, 119	EVENT_MIDI_STOP, 112
c_midi_controls_extended, 120	EVENT_MIDI_SYSEX_CONTINUE, 112
c_midi_track_ctrl, 117	EVENT_MIDI_SYSEX_END, 112
c_midibus, 114	EVENT_MIDI_SYSEX, 111
c_midibus_input_size, 114	EVENT_MIDI_TUNE_SELECT, 112
c_midibus_output_size, 114	EVENT_NOTE_OFF, 110
c midibus sysex chunk, 114	EVENT NOTE ON, 110
c_midich, 115	EVENT NULL CHANNEL, 113
c_midiclocks, 116	EVENT_PITCH_WHEEL, 110
c_midictrl, 116	EVENT_PROGRAM_CHANGE, 110
c_music_scales, 68	EVENT STATUS BIT, 109
c_musickey, 117	EVENTS ALL, 114
c_musicscale, 117	EVENTS_UNSELECTED, 114
c_mutegroups, 116	edit_action_t, 69
c notes, 116	error_message, 88
c_perf_bp_mes, 117	extract_bus_name, 84
c_perf_bw, 117	extract_port_name, 85
c_scales_policy, 121	extract_port_names, 84
c scales text, 122	extract_timing_numbers, 71
c_scales_transpose_dn, 122	FF_RW_timeout, 105
c_scales_transpose_up, 121	file access, 89
c_status_queue, 129	file accessible, 90
c_status_replace, 129	file_executable, 91
c_status_snapshot, 129	file exists, 89
c_timesig, 116	file_is_directory, 91
c_transpose, 117	file_readable, 90
c_triggers, 116	file_writable, 90
c_triggers_new, 116	font_render, 102
choose_ppqn, 100	g_midi_control_limit, 121
clamp, 105, 106	g_rc_settings, 130
clock_e, 68	g_user_settings, 130
clock_tick_duration_bogus, 81	get_current_jack_position, 96
clock_ticks_from_ppqn, 82	gs_mainwid_pointer, 130
create_jack_client, 94	gs_perfedit_pointer_0, 131
create_lash_driver, 97	gs_perfedit_pointer_1, 131
delete_lash_driver, 98	help_check, 85
delta_time_us_to_ticks, 80	info_message, 88
double_ticks_from_ppqn, 82	input_thread_func, 99
asasis_tions_nom_ppqii, or	

i	nteraction_method_t, 68	s_build_edit_highlight, 126
i	nvalid_key, 97	s_build_follow_progress, 127
į	s_ctrl_key, 102–104	s_build_highlight_empty, 125
į	s_ctrl_shift_key, 104	s_build_jack_session, 126
į	s_no_modifier, 103	s_build_jack_support, 126
i	s_null_midipulse, 98	s_build_lash_support, 126
į	s_shift_key, 103, 104	s_build_midi_vector, 127
į	s_super_key, 104	s_build_pause_support, 126
j	ack_dummy_callback, 100	s_build_portmidi_support, 125
j	ack_message_bit_bucket, 109	s_build_rtmidi_support, 125
j	ack_process_io, 109	s_build_solid_grid, 127
j	ack_process_rtmidi_input, 107	s_build_timesig_tempo, 127
j	ack_process_rtmidi_output, 108	s_build_use_event_map, 126
	ack_session_callback, 96	s_character_mapping, 128
-	ack_shutdown_callback, 93	s_debug_mode, 128
-	ack_sync_callback, 92	s event editor, 126
-	ack_timebase_callback, 93	s global lash driver, 128
į	ack_transport_callback, 94	s handlesize, 130, 131
-	keyval name, 97	s_help_1a, 124
k	keyval_normalize, 97	s_help_1b, 125
I	ash_driver, 98	s_help_2, 125
I	og2_time_sig_value, 78	s_help_3, 125
	ong_options, 124	s_help_4, 125
r	make_directory, 91	s_jitter_amount, 130
	make_section_name, 101	s_seq32_jack_support, 127
r	measures_to_ticks, 83	s_seq32_lfo_support, 128
r	measurestring_to_pulses, 74	s_seq32_menu_buttons, 128
r	message_concatenate, 88	s_seq32_transport, 127
r	midi_api_name, 106	s_seq32_transpose, 128
r	midi_input_callback, 109	s_statistics_support, 127
r	midi_input_test, 107	s_status_pairs, 128
r	midi_measures_to_pulses, 74	s_strip_empty_mutes, 127
r	midi_probe, 106	seq_app_name, 101
r	midibpm, 65	seq_client_name, 101
r	midibyte, 64	seq_event_type_t, 67
r	midilong, 65	seq_modifier_t, 66
r	midipulse, 65	seq_scroll_direction_t, 67
	midishort, 64	seq_version, 101
r	millisleep, 98	shorten_file_spec, 76
r	nin, 107	show_jack_statuses, 96
	mouse_action_e, 69	silence_jack_errors, 106
	output_thread_func, 99	silence_jack_info, 106
	parse_command_line_options, 87	string_is_void, 77
	parse_options_files, 86	string_not_void, 76
	ppqn_is_valid, 92	string_to_midibyte, 76
	pulse_length_us, 79	string_to_pulses, 75
	pulses_per_measure, 82	strings_match, 77
	oulses_to_measurestring, 72	swap, 70
	oulses_to_midi_measures, 73	tempo_us_from_bpm, 79
	oulses_to_string, 72	tempo_us_to_bytes, 78
	pulses_to_timestring, 73, 74	test_widget_click, 105
	rc, 99	ticks_to_delta_time_us, 81
	rterror_callback, 65	timestring_to_pulses, 75, 100
	rtmidi_api, 70	to_string, 89
	rtmidi_callback_t, 65	update_mainwid_sequences, 105
	s_arg_list, 124	update_perfedit_sequences, 105
	s_build_alsamidi_support, 125	usr, 99
5	s_build_chord_generator, 126	versiontext, 124

wave_func, 83	Seq24PerfInput, 852
wave_type_name, 71	seq64::Seq24SeqEventInput, 855
wave_type_t, 66	m_adding, 857
write_options_files, 87	on_button_press_event, 856
zoom_power_of_2, 78	on_button_release_event, 856
seq64::AbstractPerfInput, 133	on_motion_notify_event, 856
\sim AbstractPerfInput, 134	Seq24SeqEventInput, 855
AbstractPerfInput, 134	set_adding, 855
activate_adding, 135	seq64::automutex, 137
handle motion key, 135	\sim automutex, 138
is_adding, 135	automutex, 137
is_adding_pressed, 136	m_safety_mutex, 138
m_adding, 136	operator=, 138
m_adding_pressed, 136	seq64::busarray, 138
on_button_press_event, 135	~busarray, 140
on_button_release_event, 135	add, 140, 141
on_motion_notify_event, 135	bus, 141
perfroll, 136	busarray, 140
set_adding, 136	clock, 142
set_adding_pressed, 136	continue_from, 142
seq64::FruityPerfInput, 263	count, 141
activate adding, 268	
	get_clock, 144
FruityPerfInput, 265	get_input, 146
handle_motion_key, 268	get_midi_bus_name, 144
m_current_x, 268	get_midi_event, 147
m_current_y, 268	init_clock, 142
on_button_press_event, 265	initialize, 141
on_button_release_event, 266	is_system_port, 146
on_left_button_pressed, 267	m_container, 148
on_motion_notify_event, 266	play, 143
on_right_button_pressed, 267	poll_for_midi, 146
perfroll, 268	port_exit, 145
update_mouse_pointer, 266	print, 144
seq64::FruitySeqEventInput, 269	replacement_port, 147
FruitySeqEventInput, 269	set_all_clocks, 143
m_is_drag_pasting, 271	set_all_inputs, 146
m_is_drag_pasting_start, 271	set_clock, 143
m_justselected_one, 271	set_input, 145
on_button_press_event, 270	start, 142
on_button_release_event, 270	stop, 142
on_motion_notify_event, 271	sysex, 143
update_mouse_pointer, 269	seq64::businfo, 148
seq64::FruitySeqRollInput, 272	\sim businfo, 150
FruitySeqRollInput, 272	activate, 152
m_drag_paste_start_pos, 274	active, 150
m_erase_painting, 274	bus, 150, 152
on_button_press_event, 273	busarray, 153
on_button_release_event, 273	businfo, 149, 150
on_motion_notify_event, 273	clock, 153
update_mouse_pointer, 272	continue_from, 153
seq64::Seq24PerfInput, 851	deactivate, 152
activate_adding, 853	init_clock, 151-153
handle_motion_key, 854	init_input, 151, 152
m_effective_tick, 855	initialize, 151
on_button_press_event, 852	initialized, 151
on_button_release_event, 853	m_active, 154
on motion notify event, 853	m bus, 153
perfroll, 854	m init clock, 154
1 ,	, - , - , - , - , - , - , - , - ,

m_init_input, 154	m_name_data, 179
m_initialized, 154	m_name_meta, 178
remove, 150	m_name_seqspec, 179
start, 152	m_name_status, 178
stop, 152	m_name_timestamp, 178
sysex, 153	m_parent, 178
seq64::click, 154	meta_string, 176
button, 158	name_to_value, 172
click, 156	operator=, 172
is_left, 157	parent, 172
is_middle, 157	seqspec_string, 176
is_press, 157	set_status_from_string, 175
is_right, 157	sm_category_arrays, 177
m_button, 159	sm_category_names, 177
m_is_press, 158	sm_channel_event_names, 177
m_modifier, 159	sm_meta_event_names, 177
m_x, 158	sm_prop_event_names, 177
m_y, 159	sm_system_event_names, 177
mod_control, 158	status_string, 175
mod_control_shift, 158	stock_event_string, 175
mod_super, 158	time as measures, 174
modifier, 158	time_as_minutes, 174
operator=, 156	time_as_pulses, 174
x, 157	timestamp, 174
y, 157	timestamp_format_t, 170
seq64::condition_var, 159	timestamp_string, 173
condition_var, 160	value_to_name, 172
m_cond, 161	seq64::editable_event::name_value_t, 637
signal, 160	event_name, 638
sm_cond, 161	event_value, 638
wait, 160	seq64::editable_events, 179
seq64::configfile, 161	\sim editable_events, 182
~configfile, 163	add, 185
configfile, 163	begin, 184
get_error_message, 164	clear, 186
line after, 164	const iterator, 181
m_d, 165	count, 185
m error message, 165	current event, 186
m line, 165	dreft, 184, 185
m_name, 165	editable events, 182
next data line, 163	end, 184
parse, 164	Events, 181
set_error_message, 165	events, 184
write, 164	EventsPair, 181
seq64::editable_event, 166	eventslots, 187
~editable_event, 171	iterator, 181
analyze, 176	Key, 181
category, 173	load_events, 183
category_string, 173	m_current_event, 187
category_t, 169	m_events, 187
channel_string, 176	m_midi_parameters, 187
data_string, 176	m_sequence, 187
editable_event, 170, 171	operator=, 183
format_timestamp, 175	remove, 186
m_category, 178	replace, 186
m_format_timestamp, 178	save_events, 183
m_name_category, 178	string_to_pulses, 183
m_name_channel, 179	timing, 183

seq64::event, 188	set_channel, 199
\sim event, 193	set_data, 200
append_sysex, 202	set_note, 205
cc_match, 200	set_note_velocity, 206
check_channel, 195	set_status, 198, 199
clear_link, 203	set_status_keep_channel, 199
data, 205	set_sysex_size, 203
decrement_data1, 201	set_timestamp, 194
decrement_data2, 201	SysexContainer, 192
event, 192	transpose_note, 205
get_channel, 195	unmark, 204
get_data, 201	unpaint, 204
get_linked, 203	unselect, 204
get_note, 205	seq64::event_list, 211
get_note_velocity, 206	∼event_list, 214
get_rank, 207	add, 215
get_status, 200	any_selected_notes, 220
get_sysex, 202	append, 216
get_sysex_size, 203	begin, 215
get_timestamp, 195	clear, 217
increment_data1, 201	clear_links, 219
increment_data2, 201	const_iterator, 214
is_channel_msg, 195	count, 215
is_desired_cc_or_not_cc, 197	count_selected_events, 221
is_linked, 203	count_selected_notes, 220
is_marked, 204	dref, 218
is_note, 206 is_note_msg, 196	editable_events, 222
is_note_off, 206	empty, 215 end, 215
is_note_off_recorded, 207	event list, 214
is_note_on, 206	Events, 213
is_note_on, 200 is_one_byte_msg, 196	events, 221
is_painted, 204	EventsPair, 213
is selected, 204	is_modified, 216
is_strict_note_msg, 197	iterator, 214
is_two_byte_msg, 196	link_new, 218
link, 203	m events, 222
m channel, 208	m is modified, 222
m data, 208	mark all, 220
m_has_link, 208	mark_out_of_range, 219
m linked, 208	mark selected, 219
m marked, 209	merge, 217
m_painted, 209	midi container, 222
m selected, 209	midi splitter, 222
m_status, 208	midifile, 222
m_sysex, 208	operator=, 214
m_sysex_size, 208	print, 221
m_timestamp, 207	push_back, 216
make_clock, 205	remove, 217
mark, 204	remove_marked, 220
mod_timestamp, 198	select_all, 221
non_cc_match, 200	sequence, 222
operator<, 194	sort, 218
operator=, 193	unmark_all, 220
paint, 203	unmodify, 217
print, 207	unpaint_all, 220
restart_sysex, 202	unselect_all, 221
select, 204	verify_and_link, 219

seq64::event_list::event_key, 209	set_event_data_0, 230
event_key, 210	set_event_data_1, 230
m_rank, 211	set_event_name, 230
m_timestamp, 211	set_event_timestamp, 230
operator<, 210	set_seq_count, 229
seq64::eventedit, 223	set_seq_ppqn, 229
\sim eventedit, 228	set_seq_time_sig, 229
change_focus, 232	set_seq_title, 229
close_out, 232	v_adjustment, 231
enqueue_draw, 229	seq64::eventslots, 239
eventedit, 227	∼eventslots, 244
eventslots, 235	change_vert, 250
handle_cancel, 233	convert_y, 249
handle close, 232	current index, 244
handle delete, 232	decrement bottom, 252
handle_insert, 232	decrement_current, 252
handle_modify, 233	decrement_top, 251
handle_save, 233	delete_current_event, 247
m bottbox, 236	draw_event, 249
m_button_cancel, 237	draw_events, 250
m_button_del, 236	enqueue_draw, 249
m_button_ins, 236	event_count, 244
m_button_modify, 236	eventedit, 255
m_button_save, 237	eventslots, 243
m_editbox, 236	increment_bottom, 253
m_entry_ev_data_0, 238	increment_current, 252
m_entry_ev_data_1, 238	increment_top, 252
m_entry_ev_name, 238	insert_event, 245, 246
m_entry_ev_timestamp, 238	line_count, 244
m_eventslots, 235	line_increment, 244
m_have_focus, 239	line_maximum, 244
m_htopbox, 235	load_events, 245
m_label_category, 238	m_bottom_iterator, 257
m_label_channel, 237	m_char_w, 256
m_label_ev_count, 237	m_current_index, 257
m_label_modified, 238	m_current_iterator, 257
m_label_ppqn, 237	m_event_container, 255
m_label_right, 238	m_event_count, 256
m_label_seq_name, 237	m_line_count, 256
m_label_spacer, 237	m_line_maximum, 256
m_label_time_fmt, 238	m_line_overlap, 257
m_label_time_sig, 237	m_pager_index, 257
m_optsbox, 236	m_parent, 255
m_rightbox, 236	m_seq, 255
m_seq, 239	m_setbox_w, 256
m_showbox, 236	m_slots_chars, 255
m_table, 235	m_slots_x, 256
m_vadjust, 235	m_slots_y, 256
m_vscroll, 235	m_top_index, 257
on_delete_event, 234	m_top_iterator, 257
on_focus_in_event, 234	modify_current_event, 247
on_focus_out_event, 234	on_button_press_event, 253
on_key_press_event, 234	on_button_release_event, 253
on_realize, 233	on_expose_event, 253
on_set_focus, 233	on_focus_in_event, 253
perf_modify, 231	on_focus_out_event, 254
set_dirty, 231	on_frame_down, 254
set_event_category, 230	on_frame_end, 255

on_frame_home, 255	draw_drawable, 294
on_frame_up, 254	draw_line, 286-288
on_move_down, 254	draw_line_on_pixmap, 286, 287
on_move_up, 254	draw_normal_rectangle_on_pixmap, 293
on_realize, 253	draw_rectangle, 290-292
on_scroll_event, 254	draw_rectangle_on_pixmap, 292, 293
on_size_allocate, 254	drop_x, 285
page_movement, 251	drop_y, 285
page_topper, 251	force_draw, 285
pager_index, 245	gtk_drawarea_init, 295
save_events, 248	gui drawingarea gtk2, 283
select_event, 248	m_background, 296
set_current_event, 245	m_current_x, 297
set text, 249	m_current_y, 297
top_index, 244	m_drop_x, 297
seq64::font, 258	m_drop_y, 298
char height, 261	m_foreground, 297
char width, 261	m_gc, 296
Color, 259	m_hadjust, 296
font, 259	m_mainperf, 297
	m pixmap, 296
init, 260	– ··
m_b_on_c_pixmap, 263	m_vadjust, 296
m_b_on_y_pixmap, 262	m_window, 296
m_black_pixmap, 262	m_window_x, 297
m_c_on_b_pixmap, 263	m_window_y, 297
m_cell_h, 261	on_realize, 296
m_cell_w, 261	operator=, 284
m_clip_mask, 263	perf, 285
m_font_h, 261	render_string, 289
m_font_w, 261	render_string_on_pixmap, 289
m_offset, 262	scroll_hadjust, 294
m_padded_h, 262	scroll_hset, 295
m_pixmap, 262	scroll_vadjust, 294
m_use_new_font, 261	scroll_vset, 295
m_white_pixmap, 262	set_current_drop_x, 295
m_y_on_b_pixmap, 262	set_current_drop_y, 295
padded_height, 261	set_line, 285
render_string_on_drawable, 260	window_x, 284
seq64::gui_assistant, 274	window_y, 284
\sim gui_assistant, 276	seq64::gui_drawingarea_gtk2::rect, 817
gui_assistant, 276	height, 817
jack_idle_connect, 276	width, 818
keys, 277	x, 817
lash_timeout_connect, 276	y, 817
m_keys_perform, 277	seq64::gui_palette_gtk2, 298
quit, 276	\sim gui_palette_gtk2, 302
seq64::gui_assistant_gtk2, 277	bg_color, 305, 306
~gui_assistant_gtk2, 279	black, 303
gui_assistant_gtk2, 278	black_key, 305
jack_idle_connect, 279	black_paint, 305
lash_timeout_connect, 279	blue, 305
quit, 279	Color, 301
sm_internal_keys, 279	dark_blue, 303
seq64::gui_drawingarea_gtk2, 280	dark_cyan, 304
∼gui_drawingarea_gtk2, 284	dark_green, 303
clear_window, 285	dark_grey, 304
current_x, 284	dark_magenta, 304
current_y, 284	dark_orange, 303
33.70Ti_y, 20 T	dan_orango, ooo

dark_red, 303	seq64::jack_assistant, 314
fg_color, 306	\sim jack_assistant, 318
green, 305	activate, 322
grey, 304	client, 327
gui_palette_gtk2, 302	client_name, 327
is_inverse, 303	client_open, 327
light_grey, 304	client_uuid, 327
line color, 303	deinit, 321
load_inverse_palette, 302	error message, 330
m_bg_color, 309	get_beat_width, 319
m_black, 306	get_beats_per_measure, 320
m_blk_key, 308	get_beats_per_minute, 320
m_blk_paint, 308	get_current_jack_position, 334
m blue, 308	get_follow_transport, 326
m_dk_blue, 307	get_jack_client_info, 328
m_dk_cyan, 307	get_jack_mode, 325
m_dk_green, 306	get_jack_pos, 325
m_dk_grey, 308	get jack stop tick, 325
m_dk_magenta, 307	get_jack_stop_ttck, 325
-	
m_dk_orange, 307	get_ppqn, 319
m_dk_red, 306	info_message, 330
m_fg_color, 309	init, 321
m_green, 308	is_master, 319
m_grey, 308	is_running, 319
m_is_inverse, 306	jack_assistant, 318
m_line_color, 309	jack_frame_rate, 326
m_lt_grey, 308	jack_session_callback, 334
m_orange, 307	jack_shutdown_callback, 332
m_progress_color, 309	jack_sync_callback, 332
m_red, 307	jack_timebase_callback, 333
m_white, 307	jack_transport_callback, 332
m_wht_key, 309	m_beat_width, 337
m_wht_paint, 308	m_beats_per_measure, 337
m_yellow, 307	m_beats_per_minute, 338
orange, 304	m_follow_transport, 337
progress_color, 303	m_jack_client, 335
red, 304	m_jack_client_name, 335
white, 304	m_jack_client_uuid, 335
white_key, 305	m_jack_frame_current, 335
white_paint, 305	m_jack_frame_last, 335
yellow, 305	m_jack_frame_rate, 337
seq64::gui_window_gtk2, 310	m_jack_master, 336
\sim gui_window_gtk2, 312	m_jack_parent, 335
gui_window_gtk2, 311	m_jack_pos, 336
is_realized, 312	m_jack_running, 336
m_is_realized, 314	m_jack_stop_tick, 337
m_mainperf, 314	m_jack_tick, 336
m_redraw_period_ms, 314	m_jack_transport_state, 336
m_window_x, 314	m_jack_transport_state_last, 336
m_window_y, 314	m_jsession_ev, 336
on_realize, 313	m_ppqn, 337
perf, 312	m_toggle_jack, 337
quit, 312	output, 324
redraw_period_ms, 312	parent, 319
scroll_hadjust, 312	position, 323
scroll_hset, 313	session_event, 322
scroll_vadjust, 313	set_beat_width, 319
scroll_vset, 313	set_beats_per_measure, 320
55.555., 510	5555d.to_po

set_beats_per_minute, 320	at_pattern_edit, 368
set_follow_transport, 326	at_pause, 367
set_jack_mode, 325	at_pointer_position, 368
set_jack_running, 327	at_queue, 366
set_jack_stop_tick, 325	at_replace, 366
set_position, 330	at_rewind, 368
set_ppqn, 324	at_screenset_dn, 366
set_start_from_perfedit, 326	at_screenset_up, 366
show_position, 328	at_set_playing_screenset, 366
sm_status_pairs, 335	at_show_ui_sequence_key, 369
song_start_mode, 326	at_show_ui_sequence_number, 369
start, 322	at_snapshot_1, 366
stop, 322	at_snapshot_2, 366
sync, 329	at_song_mode, 367
tick_multiplier, 327	at_start, 367
toggle_follow_transport, 326	at_stop, 369
toggle_jack_mode, 325	at_tap_bpm, 368
toggle_song_start_mode, 326	at_toggle_jack, 367
transport_not_starting, 320	at_toggle_mutes, 368
transport_state, 320	bpm_dn, 353
seq64::jack_scratchpad, 338	bpm_up, 352
js_clock_tick, 339	event_edit, 358
js_current_tick, 339	fast_forward, 360
js_delta_tick_frac, 340	follow_transport, 360
js_dumping, 339	get_key_events, 362
js_init_clock, 339	get_key_events_rev, 363
js_jack_stopped, 339	get_key_groups, 362
js_looping, 339	get_key_groups_rev, 363
js_playback_mode, 339	get_keys, 352
js_ticks_converted, 340	group_learn, 357
js_ticks_converted_last, 340	group_off, 356, 357
js_ticks_delta, 340	group_on, 356
js total tick, 339	keep_queue, 354
seg64::jack status pair t, 340	key_name, 364
jf_bit, 340	keys perform, 351
jf_meaning, 341	lookup_keyevent_key, 363
seq64::keybindentry, 341	lookup_keyevent_seq, 363
keybindentry, 342	lookup_keygroup_group, 364
m_key, 343	lookup_keygroup_key, 363
m_perf, 344	m_key_bpm_dn, 370
m slot, 344	m key bpm up, 370
m_type, 343	m_key_event_edit, 373
on_key_press_event, 343	m_key_events, 370
options, 343	m key events rev, 370
set, 343	m_key_fast_forward, 373
type, 342	m_key_follow_transport, 373
seq64::keys_perform, 344	m_key_group_learn, 372
~keys_perform, 352	m_key_group_off, 372
at_bpm_dn, 365	m_key_group_on, 372
at_bpm_up, 365	m_key_groups, 370
at_event_edit, 368	m_key_groups_rev, 370
at_fast_forward, 368	m_key_keep_queue, 371
at_follow_transport, 368	m_key_menu_mode, 372
at_group_learn, 367	m_key_pattern_edit, 373
at_group_off, 367	m_key_pause, 372
at_group_on, 367	m_key_pause, 372 m_key_pointer_position, 373
at_keep_queue, 366	m_key_queue, 371
at_menu_mode, 367	m_key_queue, 371 m_key_replace, 371
at_menu_mode, 507	III_Ney_Iepiace, 3/ I

m_key_rewind, 373	kpt_group_on, 378
m_key_screenset_dn, 371	kpt_keep_queue, 379
m_key_screenset_up, 371	kpt_menu_mode, 380
m_key_set_playing_screenset, 371	kpt_pattern_edit, 380
m_key_show_ui_sequence_key, 370	kpt_pause, 380
m_key_show_ui_sequence_number, 370	kpt_pointer_position, 381
m_key_snapshot_1, 371	kpt_queue, 379
m_key_snapshot_2, 371	kpt_replace, 379
m_key_song_mode, 372	kpt_rewind, 381
m_key_start, 372	kpt_screenset_dn, 378
m_key_stop, 374	kpt_screenset_up, 378
m_key_tap_bpm, 373	kpt_set_playing_screenset, 378
m_key_toggle_jack, 372	kpt_show_ui_sequence_key, 379
m_key_toggle_mutes, 373	kpt_show_ui_sequence_number, 380
menu_mode, 359	kpt_snapshot_1, 379
options, 369	kpt snapshot 2, 379
optionsfile, 369	kpt_song_mode, 380
pattern_edit, 358	kpt_start, 379
pause, 358	kpt_stop, 379
perform, 369	kpt_tap_bpm, 380
pointer_position, 360, 361	kpt_toggle_jack, 380
queue, 353, 354	kpt_toggle_mutes, 381
replace, 353	seq64::keystroke, 381
RevSlotMap, 351	is, 384
rewind, 360	is_delete, 384
screenset_dn, 355	is_letter, 384
screenset_up, 355	is_press, 384
set_all_key_events, 364	key, 384
set_all_key_groups, 364	keystroke, 382, 383
set_key_event, 365	m_is_press, 386
set_key_group, 365	m_key, 386
set_keys, 352	m_modifier, 386
set_playing_screenset, 356	mod control, 385
show ui sequence key, 362	mod_control_shift, 385
show_ui_sequence_number, 362	mod_super, 385
SlotMap, 351	modifier, 385
•	operator=, 383
snapshot_1, 354 snapshot_2, 355	shift_lock, 385
• – •	seq64::lash, 386
song_mode, 359	handle config, 388
start, 357	_
stop, 359	handle_event, 388 init, 388
tap_bpm, 361	
toggle_jack, 361	lash, 387
toggle_mutes, 361	m_client, 389
seq64::keys_perform_gtk2, 374	m_is_lash_supported, 389
~keys_perform_gtk2, 376	m_lash_args, 389
key_name, 376	m_perform, 389
keys_perform_gtk2, 376	process_events, 388
set_all_key_events, 376	set_alsa_client_id, 387
set_all_key_groups, 377	start, 387
seq64::keys_perform_transfer, 377	seq64::lfownd, 390
kpt_bpm_dn, 378	~Ifownd, 392
kpt_bpm_up, 378	Ifownd, 392
kpt_event_edit, 380	m_hbox, 393
kpt_fast_forward, 381	m_phase, 394
kpt_follow_transport, 381	m_range, 394
kpt_group_learn, 378	m_scale_phase, 393
kpt_group_off, 378	m_scale_range, 393

m_scale_speed, 393	m_seqarea_seq_x, 413
m_scale_value, 393	m_seqarea_seq_y, 414
m_scale_wave, 393	m_seqarea_x, 413
m_seq, 393	m_seqarea_y, 413
m_seqdata, 393	m_text_size_x, 414
m_speed, 394	m_text_size_y, 414
m_value, 394	mainwid, 404
m_wave, 394	mainwnd, 412
m_wave_name, 394	on_button_press_event, 410
on_focus_out_event, 392	on_button_release_event, 410
scale_lfo_change, 392	on_expose_event, 409
toggle_visible, 392	on_focus_in_event, 411
seq64::maintime, 395	on_focus_out_event, 411
∼maintime, 397	on_motion_notify_event, 411
idle_progress, 397	on_realize, 409
m_bar_width, 398	redraw, 406
m_beat_width, 398	reset, 405
m_box_height, 399	select_fg_bg_colors, 409
m_box_less_pill, 399	seq_from_xy, 408
m_box_width, 399	seq_set_and_edit, 406
m_flash_height, 399	seq_set_and_eventedit, 406
m_flash_width, 399 m_flash_x, 399	set_screenset, 405 timeout, 408
m_pill_width, 399	update_mainwid_sequences, 412
m_ppqn, 400	update_markers, 407
m_tick, 400	update_sequences_on_window, 405
maintime, 397	valid_sequence, 407
mainwnd, 398	seq64::mainwnd, 415
on_expose_event, 398	~mainwnd, 421
on_realize, 398	about_dialog, 426
operator=, 397	adj_callback_bpm, 422
seq64::mainwid, 400	adj_callback_ss, 422
∼mainwid, 404	apply_song_transpose, 425
calculate_base_sizes, 409	build_info_dialog, 426
draw_marker_on_sequence, 406	choose file, 428
draw_pixmap_on_window, 405	edit_callback_notepad, 422
draw_sequence_on_pixmap, 407	edit_field_has_focus, 429
draw_sequence_pixmap_on_window, 408	enregister_perfedits, 424
draw_sequences_on_pixmap, 408	file_exit, 428
fill_background_window, 405	file_import_dialog, 426
m_armed_progress_color, 412	file_new, 425
m_button_down, 412	file_open, 425
m_last_playing, 413	file_save, 425
m_last_tick_x, 413	file_save_as, 427
m_mainwid_border, 414	handle_signal, 422
m_mainwid_spacing, 414	install_signal_handlers, 428
m_mainwid_x, 414	is_save, 428
m_mainwid_y, 414	jack_dialog, 426
m_mainwnd_cols, 413	learn_toggle, 424
m_mainwnd_rows, 413	m_adjust_bpm, 433
m_max_sets, 414	m_adjust_load_offset, 434
m_moving, 412	m_adjust_ss, 433
m_moving_seq, 412	m_button_jack, 433
m_old_seq, 412	m_button_learn, 433
m_progress_height, 415	m_button_perfedit, 433
m_screenset, 413	m_button_play, 433
m_screenset_offset, 415	m_button_stop, 433
m_screenset_slots, 415	m_call_seq_edit, 435

m_call_seq_eventedit, 435	api_clock, 452
m_entry_notes, 434	api_continue_from, 451
m_image_play, 432	api_flush, 451
m_is_running, 434	api_get_midi_event, 452
m_main_cursor, 432	api_init, 450
m_main_time, 432	api_init_clock, 451
m_main_wid, 432	api_is_more_input, 452
m_menu_edit, 431	api_poll_for_midi, 452
m_menu_file, 431	api_port_start, 452
m_menu_help, 431	api_set_beats_per_minute, 451
m_menu_mode, 434	api_set_ppqn, 451
m_menu_view, 431	api_start, 450
m_menubar, 431	api_stop, 451
m_options, 432	clock, 450
m_perf_edit, 432	continue_from, 443
m_perf_edit_2, 432	dump_midi_input, 445
m_ppqn, 431	filter_by_channel, 440, 441
m_sigpipe, 431	flush, 444
m_spinbutton_bpm, 433	get_beats_per_minute, 441
m_spinbutton_load_offset, 434	get_bpm, 441
m_spinbutton_ss, 434	get_clock, 449
m_timeout_connect, 434	get_input, 448
m_tooltips, 431	get_midi_event, 447
mainwnd, 420	get_midi_in_bus_name, 446
new_file, 428	get_midi_out_bus_name, 445
new_open_error_dialog, 427	get_num_in_buses, 440
on_delete_event, 429	get_num_out_buses, 440
on_grouplearnchange, 430	get_ppqn, 441
on_key_press_event, 430	get_sequence, 441
on_key_release_event, 430	init, 440
on_realize, 430	init_clock, 443
open_file, 421 open_performance_edit, 424	initialize_buses, 445 input, 450
open_performance_edit_2, 424	is dumping, 441
options dialog, 426	is_input_system_port, 448
pause_playing, 423	is_more_input, 447
populate_menu_edit, 429	m_beats_per_minute, 455
populate_menu_file, 429	m bus announce, 454
populate_menu_help, 429	m_dumping_input, 455
populate menu view, 429	m filter by channel, 455
ppqn, 422	m_inbus_array, 454
query_save_changes, 427	m_master_clocks, 454
rc_error_dialog, 421	m_master_inputs, 454
save file, 428	m max busses, 453
sequence_key, 425	m_mutex, 455
set play image, 423	m_outbus_array, 454
set_song_mute, 426	m_ppqn, 454
set_songlive_image, 423	m_queue, 454
signal_action, 428	m_seq, 455
start_playing, 423	m_vector_sequence, 455
stop_playing, 424	mastermidibase, 439
timer_callback, 422	midi_alsa_info, 453
toLower, 425	perform, 453
toggle_playing, 424	periorin, 453 play, 443
update_window_title, 425	poll_for_midi, 446
64::mastermidibase, 435	port_exit, 442
~mastermidibase, 440	port_settings, 450
activate, 450	port_start, 442
400, 100	ροιτ_οιαιτ, ττε

print, 444	m_local_addr_port, 474
save_clock, 452	m_seq, 474
save_input, 453	midi_alsa, 468
set_beats_per_minute, 449	set_virtual_name, 473
set_clock, 444, 447	seq64::midi_alsa_info, 475
set_input, 448	\sim midi_alsa_info, 477
set_ppqn, 449	api_flush, 479
set_sequence_input, 445	api_get_midi_event, 477
start, 441	api_poll_for_midi, 478
stop, 442	api_port_start, 478
sysex, 444	api_set_beats_per_minute, 478
seq64::mastermidibus, 456	api_set_ppqn, 478
~mastermidibus, 458	get_all_port_info, 479
activate, 462	m_alsa_seq, 480
api_continue_from, 462	m_num_poll_descriptors, 480
api_flush, 461, 463	m_poll_descriptors, 480
api get midi event, 459, 462	midi_alsa_info, 476
api_init, 459, 463	seq, 477
api_is_more_input, 459, 462	sm_input_caps, 480
api_poll_for_midi, 459, 462	sm_output_caps, 480
api_port_start, 462, 463	seq64::midi_api, 481
api_set_beats_per_minute, 461, 463	\sim midi_api, 483
api_set_ppqn, 461, 463	api_clock, 487
api_start, 461	api connect, 484
api_stop, 461	api_continue_from, 486
m_alsa_seq, 464	api_deinit_in, 485
m_midi_master, 464	api_flush, 486
m_num_poll_descriptors, 464	api_ndsh, 400 api_get_bus_name, 487
m_poll_descriptors, 464	api_get_bds_name, 407 api_get_midi_event, 485
m_use_jack_polling, 465	api_get_port_name, 487
mastermidibus, 458	api_init_in, 484
midi_alsa_info, 464	api_init_in_sub, 485
midi_jack_info, 464	api_init_out, 484
port_list, 463	api_init_out_sub, 484
seq64::midi_alsa, 465	api_play, 485
~midi_alsa, 468	api_poll_for_midi, 485
api_clock, 472	api_set_beats_per_minute, 487
api continue from, 472	api_set_ppqn, 487
api_deinit_in, 470	api_start, 486
api flush, 472	api_stop, 486
api_get_midi_event, 470	api_sysex, 486
api init in, 469	cancel_callback, 489
api_init_in_sub, 470	error, 488
api init out, 469	input_data, 489
api_init_out_sub, 469	is_input_port, 484
api_play, 471	is port open, 487
api poll for midi, 470	is_system_port, 484
api_set_beats_per_minute, 473	is_virtual_port, 484
api_set_ppqn, 473	m_connected, 490
api_start, 472	m_error_callback, 490
api_stop, 472	m_error_callback_user_data, 490
api_stop, 472 api_sysex, 471	m_error_string, 490
get_client, 468	m_first_error_occurred, 490
get_port, 469	m_input_data, 490
m_dest_addr_client, 474	m_master_info, 489
m_dest_addr_port, 474	m_parent_bus, 489
m_input_port_name, 474	master_info, 488
m_local_addr_client, 474	master midi mode, 488
	mastor_mar_mode, 400

midi_api, 483	\sim midi_info, 514
parent_bus, 488	add_bus, 519
set_port_open, 489	api_connect, 516
user_callback, 489	api_flush, 516
seq64::midi_container, 491	api_get_midi_event, 516
\sim midi_container, 493	api_poll_for_midi, 516
add_event, 496	api_port_start, 516
add_long, 496	api_set_beats_per_minute, 515
add_short, 496	api_set_ppqn, 515
add_variable, 495	app_name, 515
clear, 495	bpm, 515
done, 494	bus_container, 519
fill, 493	clear, 515
fill_meta_track_end, 497	connect_name, 518
fill_proprietary, 497	error, 518
fill_seq_name, 497	full_port_count, 515
fill_seq_number, 496	get_all_port_info, 519
fill_time_sig_and_tempo, 497	get bus id, 517
get, 495	get bus name, 517
-	- - ·
m_position_for_get, 500	get_input, 517
m_sequence, 500	get_port_count, 516
midi_container, 493	get_port_id, 517
midifile, 499	get_port_name, 517
position, 495	get_system, 517
position_increment, 495	get_virtual, 517
position_reset, 495	global_queue, 518, 519
put, 494	input_ports, 514
size, 494	m_app_name, 521
song_fill_seq_event, 499	m_bpm, 521
song_fill_seq_trigger, 499	m_bus_container, 520
seq64::midi_control, 500	m_error_string, 521
action, 501	m_global_queue, 521
active, 502	m_input, 520
data, 502	m_midi_handle, 521
in_range, 504	m_midi_mode_input, 520
inverse_active, 502	m_output, 520
m_active, 504	m_ppqn, 521
m_data, 504	midi_handle, 514, 519
m inverse active, 504	midi_info, 514
m_max_value, 505	midi_mode, 514
m min value, 504	nc_midi_port_info, 519
m status, 504	output_ports, 514
match, 503	port_list, 518
max_value, 503	ppqn, 515
midi control, 502	queue_number, 518
min value, 503	ref_midi_port_info, 520
set, 503	rtmidi_info, 520
status, 502	seq64::midi_jack, 522
seq64::midi_in_alsa, 505	~midi_jack, 526
midi in alsa, 507	api_clock, 533
seq64::midi_in_jack, 507	api_clock, 535
~midi_in_jack, 509	api_continue_from, 532
_	• — —
api_get_midi_event, 510	api_deinit_in, 531
api_poll_for_midi, 509	api_flush, 532
m_client_name, 510	api_get_midi_event, 531
midi_in_jack, 509	api_get_port_name, 534
open_client, 510	api_init_in, 530
seq64::midi_info, 511	api_init_in_sub, 531

api_init_out, 529	midi_jack_info, 540
api_init_out_sub, 530	multi_client, 540
api_play, 532	seq64::midi_list, 545
api_poll_for_midi, 531	\sim midi_list, 548
api_set_beats_per_minute, 533	CharList, 547
api_set_ppqn, 533	clear, 548
api_start, 533	done, 548
api_stop, 533	get, 548
api_sysex, 532	m_char_list, 549
client_handle, 526, 527	midi_list, 547
close client, 528	put, 548
close_port, 528	size, 548
connect_port, 528	seq64::midi_measures, 549
create_ringbuffer, 528	beats, 551
jack_data, 526	divisions, 551
m_jack_data, 535	m beats, 552
m_jack_info, 535	m_divisions, 552
m_multi_client, 535	m_measures, 551
m_remote_port_name, 535	measures, 550
midi_jack, 524	midi_measures, 550
midi_jack_info, 534	seq64::midi_message, 552
multi_client, 526	array, 554
open_client, 529	at, 553
open_client_impl, 527	container, 553
port_handle, 527	count, 554
register_port, 529	empty, 554
remote_port_name, 526	m_bytes, 554
set_virtual_name, 534	m_timestamp, 555
seq64::midi_jack_data, 535	midi_message, 553
\sim midi_jack_data, 536	operator[], 553
m_jack_buffmessage, 537	push, 554
m_jack_buffsize, 537	timestamp, 554
m_jack_client, 537	seq64::midi_out_alsa, 555
m_jack_lasttime, 537	midi_out_alsa, 557
m_jack_port, 537	seq64::midi_out_jack, 557
m_jack_rtmidiin, 537	\sim midi_out_jack, 559
midi_jack_data, 536	midi_out_jack, 559
valid_buffer, 536	open_client, 560
seq64::midi_jack_info, 538	send_message, 559
\sim midi_jack_info, 540	seq64::midi_port_info, 560
add, 544	add, 561
api_connect, 541	clear, 562
api_flush, 542	connect_name, 563
api_get_midi_event, 541	get_bus_id, 562
api_poll_for_midi, 541	get_bus_name, 562
api_port_start, 542	get_input, 562
api_set_beats_per_minute, 542	get_port_count, 562
api_set_ppqn, 541	get_port_id, 562
client_handle, 540, 543	get_port_name, 562
connect, 544	get_queue_number, 563
disconnect, 544	get_system, 563
extract_names, 544	get_virtual, 563
get_all_port_info, 543	m_port_container, 563
jack process io, 544	m port count, 563
m_jack_client, 545	midi_port_info, 561
m_jack_ports, 545	seq64::midi_port_info::port_info_t, 792
m multi client, 545	m client name, 793
midi_jack, 544	m_client_number, 793
····	5

m_is_input, 794	size, 580
m_is_system, 794	seq64::midibase, 581
m_is_virtual, 794	\sim midibase, 587
m_port_name, 793	api_clock, 600
m_port_number, 793	api_continue_from, 599
m_queue_number, 794	api_deinit_in, 598
seq64::midi_queue, 564	api_flush, 599
\sim midi_queue, 565	api_get_midi_event, 598
add, 565	api_init_in, 599
allocate, 566	api_init_in_sub, 598
count, 565	api_init_out, 599
deallocate, 566	api_init_out_sub, 598
empty, 565	api_play, 598
front, 566	api_poll_for_midi, 597
full, 565	api_start, 600
m_back, 567	api_stop, 600
m_front, 567	api_sysex, 599
m_ring, 567	bpm, 589
m_ring_size, 567	bus_name, 588, 597
m_size, 567	clock, 596
midi_queue, 565	connect_name, 588
pop, 565	continue_from, 596
pop_front, 566	deinit_in, 594
seq64::midi_splitter, 567 ~midi_splitter, 570	display_name, 588, 597 flush, 595
count, 571	get_bus_id, 588
increment, 570	get_bus_index, 588
initialize, 570	get_clock, 590
log_main_sequence, 570	get_clock_mod, 593
m_ppqn, 572	get_input, 591
m_smf0_channels, 573	get_midi_event, 593
m_smf0_channels_count, 572	get_port_id, 589
m_smf0_main_sequence, 573	init_clock, 596
m_smf0_seq_number, 573	init_in, 594
m_use_default_ppqn, 572	init_in_sub, 594
midi_splitter, 569	init_out, 594
ppqn, 571	init_out_sub, 594
split, 571	is_input_port, 589, 590
split_channel, 571	is_output_port, 590
seq64::midi_timing, 573	is_system_port, 590
beat_width, 575	is_virtual_port, 589
beats_per_measure, 575	m_bpm, 601
beats_per_minute, 574, 575	m_bus_id, 601
m_beat_width, 576	m_bus_index, 601
m_beats_per_measure, 576	m_bus_name, 602
m_beats_per_minute, 576	m_clock_mod, 600
m_ppqn, 576	m_clock_type, 601
midi_timing, 574	m_display_name, 602
ppqn, 576	m_inputing, 601
seq64::midi_vector, 577	m_is_input_port, 602
~midi_vector, 579	m_is_system_port, 602
CharVector, 579	m_is_virtual_port, 602
clear, 581	m_lasttick, 602
done, 580	m_mutex, 603
get, 580	m_port_id, 601
m_char_vector, 581	m_port_name, 602
midi_vector, 579	m_ppqn, 601
put, 580	m_queue, 601

mastermidibus, 600	errdump, 632
match, 589	error_is_fatal, 621
midibase, 586	error_message, 621
play, 595	is_sysex_special_id, 633
poll_for_midi, 593	m_char_list, 635
port_name, 588, 597	m_data, 634
ppqn, 589	m_disable_reported, 634
print, 596	m_error_is_fatal, 634
queue_number, 591	m_error_message, 634
set_alt_name, 592	m file size, 634
set_bus_id, 591	m_global_bgsequence, 635
set_clock, 590	m_mutex, 634
set_clock_mod, 593	m_name, 634
set_clock_status, 590	m_new_format, 635
set_input, 597	m_pos, 634
set_input_status, 591	m_ppqn, 635
set_multi_name, 592	m_smf0_splitter, 635
set_name, 591	m use default ppqn, 635
set port id, 597	midifile, 618
set_system_port_flag, 590	parse, 619
show_bus_values, 588	parse_prop_header, 622
start, 595	parse_proprietary_track, 623
stop, 595	parse_smf_0, 621
•	parse_smf_1, 622
sysex, 595 seq64::midibus, 603	pow2, 624
~midibus, 607	•
api clock, 611, 613	ppqn, 621
api_clock, 611, 613	prop_item_size, 631 read_byte, 625
api_continue_from, 610, 613	read_byte, 623
api_deinit_in, 609, 612	read_long, 625
api_defini_in, 609, 612 api_flush, 610	read_seq_number, 629
api_nusii, 610 api_get_midi_event, 612	read_short, 625
api_get_midi_event, 612 api_init_in, 608, 611	- ·
• — —	read_track_name, 628
api_init_in_sub, 609, 612	read_varinum, 626
api_init_out, 608, 612	seq_number_size, 633
api_init_out_sub, 608, 612	track_end_size, 633
api_play, 609, 614	track_name_size, 632
api_poll_for_midi, 613	varinum_size, 631
api_start, 610, 613	write, 620
api_stop, 611, 613	write_byte, 627
api_sysex, 610	write_header, 629
get_client, 608	write_long, 626
get_port, 608	write_prop_header, 629
m_dest_addr_client, 614	write_proprietary_track, 630
m_dest_addr_port, 614	write_seq_number, 628
m_input_port_name, 615	write_short, 626
m_local_addr_client, 615	write_song, 621
m_local_addr_port, 615	write_track, 633
m_master_info, 615	write_track_end, 629
m_rt_midi, 615	write_track_name, 628
m_seq, 614	write_varinum, 627
mastermidibus, 614	seq64::mutex, 636
midibus, 606, 607	lock, 637
set_virtual_name, 611	m_mutex_lock, 637
seq64::midifile, 615	mutex, 637
\sim midifile, 619	sm_recursive_mutex, 637
add_trigger, 625	unlock, 637
checklen, 624	seq64::options, 638

add_extended_keys_page, 643	m_button_loop, 664
add_jack_sync_page, 643	m_button_play, 663
add_keyboard_page, 643	m_button_redo, 664
add_midi_clock_page, 642	m_button_snap, 663
add_midi_input_page, 643	m_button_stop, 663
add_mouse_page, 643	m_button_undo, 664
button, 639	m_button_xpose, 663
clock_callback_mod, 641	m_bw, 666
clock_callback_off, 640	m_entry_bpm, 665
clock_callback_on, 641	m_entry_bw, 665
clock_mod_callback, 641	m_entry_snap, 663
filter_callback, 641	m_entry_xpose, 663
input_callback, 641	m_hadjust, 662
lash_support_callback, 642	m hbox, 665
m_button_jack_connect, 644	m_hlbox, 665
m_button_jack_disconnect, 644	m_hscroll, 662
m_button_jack_master, 644	m_image_play, 663
m_button_jack_master_cond, 644	m is running, 666
m_button_jack_transport, 644	m menu bpm, 666
m button ok, 644	m_menu_bw, 666
m mainperf, 643	m_menu_snap, 662
m_notebook, 644	m_menu_xpose, 663
m_tooltips, 643	m_peer_perfedit, 661
mouse_click_edit_callback, 642	m perfnames, 662
mouse_fruity_callback, 642	m_perfroll, 662
mouse_mod4_callback, 642	m_perftime, 662
mouse_seq24_callback, 642	m_ppqn, 666
mouse_snap_split_callback, 642	m_snap, 666
options, 640	m_standard_bpm, 666
perf, 640	m_table, 661
transport_callback, 641	m_tooltips, 665
seq64::optionsfile, 645	m_vadjust, 662
\sim optionsfile, 646	m_vscroll, 662
error_message, 648	on_delete_event, 661
optionsfile, 646	on_key_press_event, 660
parse, 646	on_key_release_event, 661
write, 648	on_realize, 660
seq64::perfedit, 649	pause_playing, 660
\sim perfedit, 654	perfedit, 653
collapse, 658	popup_menu, 659
copy, 658	redo, 659
draw_sequences, 659	rewind, 655
enqueue_draw, 654	set_beat_width, 657
enregister_peer, 655	set_beats_per_bar, 657
expand, 658	set_follow_transport, 656
fast_forward, 656	set_guides, 658
get_toggle_jack, 655	set_image, 659
grow, 658	set_jack_mode, 656
init_before_show, 654	set_looped, 658
m_bpm, 666	set_snap, 657
m_button_bpm, 665	set_transpose, 656
m_button_bw, 665	set_zoom, 655
m_button_collapse, 664	start_playing, 660
m_button_copy, 664	stop_playing, 660
m_button_expand, 664	timeout, 659
m_button_follow, 665	toggle_follow_transport, 656
m_button_grow, 664 m_button_jack, 664	toggle_jack, 655 toggle_playing, 660
iii_buttoii_jaon, bu+	toggie_playing, ood

transpose_button_callback, 657	expand, 727
undo, 659	FF_RW_timeout, 697
update_perfedit_sequences, 661	FF_rewind, 697
zoom_check, 654	fast_forward, 699
seq64::perfnames, 667	ff_rw_button_t, 690
~perfnames, 670	ff_rw_type, 698
change_vert, 671	filter_by_channel, 694
convert y, 670	finish, 702
draw_sequence, 670	get_32nds_per_quarter, 693
draw sequences, 670	get_beat_width, 693
enqueue_draw, 670	get_beats_per_bar, 692
m_char_w, 674	get_beats_per_minute, 705
m_namebox_w, 674	get_current_jack_position, 750
m_names_chars, 674	get_follow_transport, 698
m_names_x, 674	get_group_mute_state, 710
m_names_y, 674	get_input, 746
m_parent, 673	get_input, 740 get_jack_tick, 702
	
m_seqs_in_set, 675	get_key_events, 712
m_sequence_active, 675	get_key_events_rev, 712
m_sequence_max, 675	get_key_groups, 712
m_sequence_offset, 675	get_key_groups_rev, 712
m_setbox_w, 674	get_left_tick, 703
m_xy_offset, 674	get_max_trigger, 727
on_button_press_event, 672	get_offset, 711
on_button_release_event, 672	get_playing_screenset, 725
on_expose_event, 671	get_right_tick, 704
on_realize, 671	get_screen_set_notepad, 730
on_scroll_event, 673	get_screenset, 725
on_size_allocate, 673	get_sequence, 718, 719
perfedit, 673	get_start_tick, 703
perfnames, 669	get_tick, 702
redraw, 671	get_toggle_jack, 696
redraw_dirty_sequences, 670	get_transpose, 705
seq64::perform, 675	gui, 694
∼perform, 690	handle_midi_control, 729
activate, 734	handle_midi_control_ex, 730
add clock, 745	have redo, 726
add_input, 745	have_undo, 726
add_sequence, 700	highlight, 718
all_notes_off, 736	increment_beats_per_minute, 717
any_group_unmutes, 731	increment screenset, 718
apply_song_transpose, 704	init_jack_transport, 738
armed_saved, 708	inner_start, 743
clamp track, 743	inner_stop, 743
clear_all, 699	input_func, 709
	input thread func, 747
clear_sequence_triggers, 702	• – – •
clocks_per_metronome, 693	install_sequence, 742
collapse, 727	is_active, 704
combine_bytes, 696	is_control_status, 691
copy, 727	is_dirty_edit, 723
copy_triggers, 722	is_dirty_main, 723
create_master_bus, 744	is_dirty_names, 724
current_screen_set_notepad, 730	is_dirty_perf, 724
decrement_beats_per_minute, 716	is_edit_sequence, 692
decrement_screenset, 717	is_exportable, 724
deinit_jack_transport, 739	is_group_learning, 733
delete_sequence, 701	is_input_system_port, 746
enregister, 695	is_jack_master, 695

is_jack_running, 695	m_midiclockpos, 758
is_modified, 691, 739	m_midiclockrunning, 757
is_mseq_valid, 742	m_midiclocktick, 757
is_pattern_playing, 695, 741	m_mode_group, 752
is_running, 695	m_mode_group_learn, 752
is_screenset_valid, 740	m_mute_group, 751
is_seq_valid, 741	m_mute_group_selected, 752
is_sequence_in_edit, 701	m_notify, 761
is_smf_0, 718	m_offset, 758
jack_assistant, 746	m_one_measure, 756
jack_shutdown, 748	m_out_thread, 753
jack_sync_callback, 748	m_out_thread_launched, 754
jack_timebase_callback, 749	m_outputing, 754
jack_transport_callback, 748	m_playback_mode, 755
key_name, 711	m_playing_screen, 752
keybindentry, 746	m_playscreen_offset, 752
keys, 694	m_ppqn, 755
launch, 699	m_redo_vect, 760
launch_input_thread, 738	m_reposition, 751
launch_output_thread, 738	m_right_tick, 757
learn_toggle, 716	m_running, 754
left_right_size, 704	m_screen_set_notepad, 758
lookup_keyevent_key, 713	m_screenset, 759
lookup_keyevent_seq, 713	m_seqs, 752
lookup_keygroup_group, 714	m_seqs_active, 752
lookup_keygroup_key, 714	m_seqs_in_set, 759
m_32nds_per_quarter, 756	m_sequence_count, 759
m_FF_RW_button_type, 751	m_sequence_high, 759
m_armed_saved, 751	m_sequence_max, 759
m_armed_statuses, 751	m_sequence_state, 753
111_d11110d_0tata000, 701	m_sequence_state, 750
m beat width, 755	
m_beat_width, 755	m_song_start_mode, 750
m_beat_width, 755 m_beats_per_bar, 755	m_song_start_mode, 750 m_start_from_perfedit, 750
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_names, 753
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_names, 753 m_was_active_perf, 753
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_names, 753 m_was_active_perf, 753 mainwnd, 746
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_names, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_names, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760 m_is_pattern_playing, 754	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_names, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_in_puting, 754 m_is_pattern_playing, 754 m_jack_asst, 760	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_names, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_is_modified, 760 m_is_pattern_playing, 754 m_jack_asst, 760 m_jack_tick, 757	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_perf, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729 midi_control_off, 728
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760 m_is_pattern_playing, 754 m_jack_asst, 760 m_jack_tick, 757 m_left_tick, 756	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_perf, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729 midi_control_off, 728 midi_control_on, 728
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760 m_is_pattern_playing, 754 m_jack_asst, 760 m_jack_tick, 757 m_left_tick, 756 m_looping, 755	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_perf, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729 midi_control_off, 728 midi_control_on, 728 midi_control_toggle, 727
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760 m_is_pattern_playing, 754 m_jack_asst, 760 m_jack_tick, 757 m_left_tick, 756 m_looping, 755 m_master_bus, 756	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_perf, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729 midi_control_off, 728 midi_control_off, 728 midi_control_on, 728 midi_control_toggle, 727 midifile, 746
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760 m_is_pattern_playing, 754 m_jack_asst, 760 m_jack_tick, 757 m_left_tick, 756 m_looping, 755 m_master_bus, 756 m_master_clocks, 756	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_perf, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729 midi_control_off, 728 midi_control_off, 728 midi_control_toggle, 727 midifile, 746 modify, 691
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760 m_is_pattern_playing, 754 m_jack_asst, 760 m_jack_tick, 757 m_left_tick, 756 m_looping, 755 m_master_bus, 756 m_master_inputs, 756	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_perf, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729 midi_control_off, 728 midi_control_off, 728 midi_control_toggle, 727 midifile, 746 modify, 691 move_triggers, 722
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760 m_is_pattern_playing, 754 m_jack_asst, 760 m_jack_tick, 757 m_left_tick, 756 m_looping, 755 m_master_bus, 756 m_master_inputs, 756 m_max_sets, 759	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_perf, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729 midi_control_off, 728 midi_control_off, 728 midi_control_toggle, 727 midifile, 746 modify, 691 move_triggers, 722 mute_all_tracks, 708
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_have_undo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760 m_is_pattern_playing, 754 m_jack_asst, 760 m_jack_tick, 757 m_left_tick, 756 m_looping, 755 m_master_bus, 756 m_master_inputs, 756 m_max_sets, 759 m_midi_cc_off, 758	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_perf, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729 midi_control_off, 728 midi_control_off, 728 midi_control_toggle, 727 midifile, 746 modify, 691 move_triggers, 722 mute_all_tracks, 708 mute_group_offset, 741
m_beat_width, 755 m_beats_per_bar, 755 m_bpm, 755 m_clocks_per_metronome, 755 m_condition_var, 760 m_control_status, 758 m_dont_reset_ticks, 758 m_edit_sequence, 759 m_excell_FF_RW, 751 m_gui_support, 761 m_have_redo, 760 m_in_thread, 754 m_in_thread_launched, 754 m_inputing, 754 m_is_modified, 760 m_is_pattern_playing, 754 m_jack_asst, 760 m_jack_tick, 757 m_left_tick, 756 m_looping, 755 m_master_bus, 756 m_master_inputs, 756 m_max_sets, 759	m_song_start_mode, 750 m_start_from_perfedit, 750 m_starting_tick, 757 m_tick, 757 m_tick, 757 m_tracks_mute_state, 751 m_transpose, 753 m_undo_vect, 760 m_us_per_quarter_note, 756 m_usemidiclock, 757 m_was_active_edit, 753 m_was_active_main, 753 m_was_active_perf, 753 m_was_active_perf, 753 mainwnd, 746 mainwnd_key_event, 721 master_bus, 694 max_active_set, 738 midi_control_event, 729 midi_control_off, 728 midi_control_off, 728 midi_control_toggle, 727 midifile, 746 modify, 691 move_triggers, 722 mute_all_tracks, 708

mute_screenset, 709	set_key_group, 744
new_sequence, 700	set_left_tick, 703
off_sequences, 735	set_looping, 738
options, 747	set_mode_group_learn, 733
optionsfile, 746	set_mode_group_mute, 732
output_func, 709	set_offset, 711
output_thread_func, 747	set_orig_ticks, 737
page_decrement_beats_per_minute, 717	set_playback_mode, 741
page_increment_beats_per_minute, 717	set_playing_screenset, 731
pause_key, 716	set_reposition, 698
pause_playing, 715	set_right_tick, 703
perfedit, 747	set_running, 740
perform, 690	set_screen_set_notepad, 730, 731
perfroll, 747	set_screenset, 725
perfroll_key_event, 721	set_sequence_control_status, 705
play, 737	set_song_mute, 732
playback_key_event, 721	set_start_tick, 703
pop_trigger_redo, 723	set_tick, 702
pop trigger undo, 723	set_transpose, 705
position_jack, 735	set was active, 736
ppqn, 691	show_ui_sequence_key, 712
print_triggers, 702	show_ui_sequence_number, 712, 713
push_trigger_undo, 722	·
	sm_mc_dummy, 750
reposition, 699	song_start_mode, 695
reset_sequences, 736	split_trigger, 727
restore_playing_state, 711	start, 734
rewind, 698	start_from_perfedit, 697
save_playing_state, 711	start_jack, 735
select_and_mute_group, 732	start_key, 716
select_group_mute, 733	start_playing, 714
seq_in_playing_screen, 739	stop, 734
sequence_count, 691	stop_jack, 735
sequence_key, 719	stop_key, 716
sequence_label, 719	stop_playing, 716
sequence_max, 691	toggle_all_tracks, 708
sequence_playing_change, 707	toggle_follow_transport, 698
sequence_playing_off, 708	toggle_jack_mode, 696
sequence_playing_on, 707	toggle_other_names, 726
sequence_playing_toggle, 707	toggle_other_seqs, 725
set_32nds_per_quarter, 693	toggle_playing_tracks, 708
set_active, 736	toggle_song_start_mode, 695
set_and_copy_mute_group, 733	unset_edit_sequence, 692
set_beat_width, 693	unset_mode_group_learn, 733
set_beats_per_bar, 692	unset_mode_group_mute, 732
set_beats_per_minute, 737	unset_sequence_control_status, 705
set_clock, 745	us_per_quarter_note, 693, 694
set_clock_bus, 720	valid_midi_control_seq, 740
set_edit_sequence, 691	seq64::performcallback, 761
set_follow_transport, 698	on_grouplearnchange, 763
set_group_mute_state, 710	seq64::perfroll, 763
set_have_redo, 726	\sim perfroll, 768
set_have_undo, 726	change_horz, 772
set_input, 745	change_vert, 772
set_input_bus, 720	convert_drop_xy, 773
set_jack_mode, 696	convert_x, 771
set_jack_stop_tick, 696	convert_xy, 771
set_jack_tick, 702	draw all, 770
set_key_event, 744	draw_background_on, 772
_ ·-	_ 5

draw_drawable_row, 772	redraw_progress, 770
draw_progress, 770	Seq24PerfInput, 777
draw_sequence_on, 772	set_guides, 768
enqueue_draw, 773	set_ppqn, 770
fill_background_pixmap, 769	set_zoom, 773
follow_progress, 770	snap_x, 772
FruityPerfInput, 777	split_trigger, 773
horizontal_adjust, 773	update_sizes, 769
horizontal_set, 774	vertical_adjust, 774
increment_size, 769	vertical_set, 774
init_before_show, 769	seq64::perftime, 783
m_4bar_offset, 780	\sim perftime, 786
m_background_x, 779	change_horz, 787
m_beat_length, 780	draw_background, 787
m_divs_per_beat, 779	draw_pixmap_on_window, 789
m_drop_sequence, 781	draw_progress_on_window, 787
m_drop_tick, 781	enqueue_draw, 787
m_drop_tick_trigger_offset, 781	idle_progress, 789
m_fruity_interaction, 782	increment_size, 787
m_grow_direction, 782	key_press_event, 790
m_growing, 782	m_4bar_offset, 791
m_h_page_increment, 778	m_left_marker_tick, 792
m_have_button_press, 780	m_measure_length, 792
m_interaction, 782	m_parent, 791
m_measure_length, 780	m_perf_scale_x, 792
m_moving, 782	m_ppqn, 791
m_names_y, 779	m_right_marker_tick, 792
m_old_progress_ticks, 780	m_snap, 792
m_page_factor, 778	m_tick_offset, 791
m_parent, 778	m_timearea_y, 792
m_perf_scale_x, 779	on_button_press_event, 790
m_ppqn, 778	on_button_release_event, 790
m_roll_length_ticks, 781	on_expose_event, 789
m_seq24_interaction, 782	on_realize, 789
m_sequence_active, 781	on_size_allocate, 790
m_sequence_max, 781	perfedit, 791
m_sequence_offset, 781	perftime, 785
m_size_box_w, 779	pixel_to_tick, 788
m_snap, 778	reset, 786
m_ticks_per_bar, 779	set_guides, 786
m_trans_button_press, 780	set_ppqn, 787
m_transport_follow, 780	set_scale, 786
m_v_page_increment, 778	set_zoom, 787
m_zoom, 779	tick_offset, 788
on_button_press_event, 775	tick_to_pixel, 788
on_button_release_event, 775	update_pixmap, 789
on_expose_event, 775	update_sizes, 789
on_focus_in_event, 776	seq64::rc_settings, 794
on_focus_out_event, 776	allow_click_edit, 801, 805
on_key_press_event, 777	allow_mod4_mode, 801, 805
on_motion_notify_event, 776	allow_snap_split, 801, 805
on_realize, 774	app_client_name, 804
on_scroll_event, 776	application_name, 804
on_size_allocate, 776	auto_option_save, 801, 805
on_size_request, 777	config_directory, 804, 809
perfedit, 778	config_filename, 804, 809
perfroll, 768	config_filename_alt, 804, 810
redraw_dirty_sequences, 770	config_filespec, 800

device_ignore, 803, 807	user_filename, 804, 809
device_ignore_num, 803, 807	user_filename_alt, 804, 810
filename, 803, 808	user_filespec, 800
filter_by_channel, 802, 807	with_jack, 802
help_check, 811	with_jack_master, 802, 806
home_config_directory, 810	with_jack_master_cond, 802, 806
interaction_method, 803, 808	with_jack_midi, 802, 806
jack_session_uuid, 803, 808	with_jack_transport, 802, 806
lash_support, 801, 805	seq64::rect, 816
last_used_dir, 804, 808	height, 817
legacy format, 801, 805	width, 817
m_allow_click_edit, 813	x, 816
m_allow_mod4_mode, 812	y, 816
m_allow_snap_split, 812	seq64::rterror, 818
m_app_client_name, 816	∼rterror, 819
m_application_name, 816	get_message, 820
m_auto_option_save, 812	getType, 820
m_config_directory, 815	m_message, 820
m_config_filename, 815	m_type, 820
m_config_filename_alt, 815	print_message, 820
m_device_ignore, 814	rterror, 819
m_device_ignore_num, 814	Type, 819
m_filename, 815	what, 820
m_filter_by_channel, 814	seq64::rtmidi, 821
m_interaction_method, 814	∼rtmidi, 823
m_jack_session_uuid, 815	api_clock, 824
m_lash_support, 812	api_connect, 823
m_last_used_dir, 815	api_continue_from, 824
m_legacy_format, 812	api_deinit_in, 825
m_manual_alsa_ports, 814	api_flush, 826
m_pass_sysex, 813	api_get_midi_event, 825
m_print_keys, 814	api_init_in, 825
m_priority, 813	api_init_in_sub, 825
m_reveal_alsa_ports, 814	api_init_out, 825
m_show_midi, 813	api_init_out_sub, 825
m_stats, 813	api_play, 823
m_user_filename, 815	api_poll_for_midi, 826
m_user_filename_alt, 815	api_set_beats_per_minute, 824
m_with_jack_master, 813	api_set_ppqn, 824
m_with_jack_master_cond, 813	api_start, 824
m_with_jack_midi, 814	api_stop, 824
m_with_jack_transport, 813	api_sysex, 826
mainwnd, 811	delete_api, 828
manual_alsa_ports, 803, 807	full_port_count, 827
operator=, 800	get_api, 827, 828
options, 811	get_bus_id, 826
optionsfile, 810	get_bus_name, 826
parse_command_line_options, 811	get_port_count, 827
pass_sysex, 802, 806	get_port_id, 827
print_keys, 803, 807	get_port_name, 827
priority, 801, 806	is_port_open, 826
rc_settings, 799	m_midi_api, 828
reveal_alsa_ports, 803, 807	m_midi_info, 828
rtmidi_info, 811	midibus, 828
set_config_files, 809	rtmidi, 823
set_defaults, 800	set_api, 828
show_midi, 801, 805	seq64::rtmidi_in, 829
stats, 802, 806	\sim rtmidi_in, 831

cancel_callback, 832	mastermidibus, 847
openmidi_api, 832	midi_mode, 841, 842
rtmidi_in, 831	midibus, 847
user_callback, 832	openmidi_api, 847
seq64::rtmidi_in_data, 833	port_list, 846
api_data, 836	ppqn, <mark>844</mark>
continue_sysex, 835	queue_number, 844
do_input, 835	rtmidi_in, 848
first_message, 835	rtmidi_info, 841
ignore_flags, 834, 835	rtmidi_out, 848
m_api_data, 838	selected_api, 846
m_continue_sysex, 838	set_api_info, 846
m_do_input, 837	sm_selected_api, 848
m_first_message, 838	seq64::rtmidi_out, 849
m_ignore_flags, 837	∼rtmidi_out, 850
m message, 837	openmidi_api, 851
m queue, 837	rtmidi out, 850
m_user_callback, 838	seq64::seqdata, 857
m_user_data, 838	∼segdata, 861
m_using_callback, 838	change_horz, 864
message, 834	convert_x, 864
queue, 834	draw events on, 864
rtmidi_in_data, 834	draw_events_on_pixmap, 865
test_ignore_flags, 834	draw line on window, 862
user_callback, 837	draw_pixmap_on_window, 865
user_data, 836, 837	idle_redraw, 862
using callback, 836	Ifownd, 868
seq64::rtmidi_info, 838	m_cc, 869
~rtmidi_info, 841	m_drag_handle, 870
add_bus, 842	m_dragging, 870
add_input, 842	m number h, 869
add_output, 842	m_number_offset_y, 869
api_connect, 846	m_number_w, 869
api_flush, 845	m numbers, 869
api get midi event, 845	m old, 869
api_poll_for_midi, 845	m_scroll_offset_ticks, 868
api_port_start, 845	m_scroll_offset_x, 869
api_set_beats_per_minute, 845	m_seq, 868
api_set_ppqn, 845	m_status, 869
app_name, 844	m_zoom, 868
bpm, 845	on_button_press_event, 866
clear, 842	on button release event, 866
delete_api, 847	on_expose_event, 865
full_port_count, 843	on leave notify event, 867
get_all_port_info, 844	on_motion_notify_event, 866
get_an_port_mio, 644 get_api_info, 846	on_realize, 865
get bus id, 842	on_scroll_event, 867
get_bus_name, 843	on_size_allocate, 867
get_bus_name, 643 get_compiled_api, 841	redraw, 861
get_input, 843 get_port_count, 843	render_number, 865 reset, 861
get_port_id, 843	seqdata, 860
get_port_name, 843	seqevent, 868 segroll, 868
get_system, 844 get_version, 841	• •
get_version, 841 get_virtual, 844	set_data_type, 862
get_virtual, 844 global_queue, 844	set_zoom, 861 update_pixmap, 862
m_info_api, 848	update_pixmap, 662 update_sizes, 862
π_πιο_αρι, στο	υρυαισ_οι ∠σ ο, <u>ου</u> Ζ

xy to rect, 862	m_initial_snap, 891
seq64::seqedit, 870	m_initial_zoom, 891
~seqedit, 878	m_key, 892
apply_length, 882	m_lfo_wnd, 897
change_focus, 888	m_measures, 892
create_menu_image, 887	m_menu_bpm, 895
create_menus, 886	m_menu_bw, 895
do_action, 887	m_menu_chords, 895
fill_top_bar, 885	m_menu_data, 895
get measures, 882	m menu key, 895
handle_close, 888	m_menu_length, 894
horizontal_adjust, 881	m_menu_midibus, 894
horizontal_set, 881	m_menu_midich, 894
m_bgsequence, 892	m_menu_note_length, 894
m_button_bpm, 901	m_menu_rec_vol, 895
m_button_bus, 898	m_menu_scale, 895
m_button_bw, 901	m_menu_sequences, 895
m button channel, 898	m_menu_snap, 894
m_button_chord, 900	m_menu_tools, 893
m_button_data, 901	m_menu_zoom, 894
m_button_key, 900	m menubar, 893
m_button_length, 899	m note length, 892
m_button_lfo, 897	m_pp_eighth, 893
m_button_note_length, 899	m_pp_sixteenth, 893
m_button_quantize, 898	m_pp_whole, 893
m_button_rec_vol, 901	m_ppqn, 893
m_button_redo, 898	m_scale, 892
m button scale, 900	m_seq, 893
m_button_sequence, 898	m_seqdata_wid, 896
m_button_snap, 899	m_seqevent_wid, 896
m_button_tools, 898	m_seqkeys_wid, 896
m button undo, 897	m_seqroll_wid, 897
m_button_zoom, 899	m_seqtime_wid, 896
m_chord, 892	m_snap, <mark>892</mark>
m_editing_cc, 902	m_table, 897
m_editing_status, 902	m_toggle_play, 901
m_entry_bpm, 901	m_toggle_q_rec, 902
m_entry_bus, 898	m_toggle_record, 902
m_entry_bw, 901	m_toggle_thru, 902
m_entry_channel, 899	m_toggle_transpose, 894
m_entry_chord, 900	m_tooltips, 900
m_entry_data, 901	m_vadjust, 896
m_entry_key, 900	m_vbox, 897
m_entry_length, 900	m_vscroll_new, 896
m_entry_name, 902	m_zoom, 892
m_entry_note_length, 899	mouse_action, 888
m_entry_scale, 900	name_change_callback, 884
m_entry_sequence, 898	on_delete_event, 889
m_entry_snap, 899	on_focus_in_event, 889
m_entry_zoom, 899	on_focus_out_event, 889
m_hadjust, 896	on_key_press_event, 890
m_have_focus, 902	on_realize, 888
m_hbox, 897	on_scroll_event, 889
m_hbox2, 897	on_set_focus, 889
m_hscroll_new, 896	play_change_callback, 884
m_image_transpose, 894	popup_event_menu, 886
m_initial_chord, 891	popup_menu, 886
m_initial_note_length, 891	popup_midibus_menu, 886

popup_midich_menu, 887	m_selected, 915
popup_sequence_menu, 887	m_selecting, 916
popup_tool_menu, 887	m_seq, 915
q_rec_change_callback, 884	m_seq24_interaction, 915
record_change_callback, 884	m_seqdata_wid, 916
redo_callback, 885	m_snap, 915
seqedit, 878	m_status, 917
seqmenu, 891	m_zoom, 915
set_background_sequence, 883	on_button_press_event, 912
set_beat_width, 880	on_button_release_event, 912
set_beats_per_bar, 879	on expose event, 912
set_chord, 883	on_focus_in_event, 913
set_data_type, 885	on_focus_out_event, 913
set_key, 883	on_key_press_event, 913
set_measures, 882	on_motion_notify_event, 913
set_midi_bus, 883	on_realize, 911
set_midi_channel, 882	on size allocate, 914
set_note_length, 879	redraw, 907
set rec vol, 880	reset, 907
set_scale, 883	Seq24SeqEventInput, 914
set snap, 879	segevent, 906
set transpose image, 880	set_data_type, 907
set zoom, 878	set snap, 907
start_playing, 888	set_zoom, 907
start_playing, 666 stop_playing, 888	
thru_change_callback, 884	snap_x, 911
	snap_y, 911
timeout, 887	start_paste, 910
transpose_change_callback, 884	update_pixmap, 908
undo_callback, 885	update_sizes, 908
update_all_windows, 885	x_to_w, 909
update_all_windows, 885 vertical_adjust, 881	x_to_w, 909 seq64::seqkeys, 917
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928 m_keying, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928 m_keying, 928 m_keying_note, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928 m_keying, 928 m_keying_note, 928 m_scale, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928 m_keying, 928 m_keying, 928 m_scale, 928 m_scroll_offset_key, 927
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928 m_keying, 928 m_keying_note, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917 m_fruity_interaction, 915	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928 m_keying, 928 m_keying, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928 m_seq, 927
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917 m_fruity_interaction, 915 m_growing, 916	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928 m_keying, 928 m_keying, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928 m_seq, 927 m_show_octave_letters, 928
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917 m_fruity_interaction, 915 m_growing, 916 m_move_snap_offset_x, 917	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928 m_keying, 928 m_keying, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928 m_seq, 927 m_show_octave_letters, 928 on_button_press_event, 925
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917 m_fruity_interaction, 915 m_growing, 916 m_move_snap_offset_x, 917 m_moving, 916	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_key, 928 m_keying, 928 m_keying, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928 m_seq, 927 m_show_octave_letters, 928 on_button_release_event, 925
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917 m_fruity_interaction, 915 m_growing, 916 m_move_snap_offset_x, 917 m_moving, 916 m_moving_init, 916	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_keying, 928 m_keying, 928 m_keying_note, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928 m_seq, 927 m_show_octave_letters, 928 on_button_press_event, 925 on_enter_notify_event, 926
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917 m_fruity_interaction, 915 m_growing, 916 m_move_snap_offset_x, 917 m_moving, 916 m_moving_init, 916 m_old, 915	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_keying, 928 m_keying, 928 m_keying_note, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928 m_seq, 927 m_show_octave_letters, 928 on_button_press_event, 925 on_enter_notify_event, 926 on_expose_event, 925
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917 m_fruity_interaction, 915 m_growing, 916 m_move_snap_offset_x, 917 m_moving, 916 m_moving_init, 916 m_old, 915 m_painting, 916 m_paste, 917	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_keying, 928 m_keying, 928 m_keying_note, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928 m_seq, 927 m_show_octave_letters, 928 on_button_press_event, 925 on_enter_notify_event, 926 on_expose_event, 925 on_leave_notify_event, 926 on_motion_notify_event, 926
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917 m_fruity_interaction, 915 m_growing, 916 m_move_snap_offset_x, 917 m_moving, 916 m_moving_init, 916 m_old, 915 m_painting, 916	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_keying, 928 m_keying, 928 m_keying_note, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928 m_seq, 927 m_show_octave_letters, 928 on_button_press_event, 925 on_enter_notify_event, 926 on_expose_event, 925 on_leave_notify_event, 926
update_all_windows, 885 vertical_adjust, 881 vertical_set, 881 seq64::seqevent, 903 ~seqevent, 906 change_horz, 910 convert_t, 911 convert_x, 910 draw_background, 908 draw_events_on, 910 draw_events_on_pixmap, 908 draw_pixmap_on_window, 908 draw_selection_on_window, 908 drop_event, 909 force_draw, 909 FruitySeqEventInput, 914 idle_redraw, 909 m_cc, 917 m_fruity_interaction, 915 m_growing, 916 m_move_snap_offset_x, 917 m_moving, 916 m_moving_init, 916 m_painting, 916 m_paste, 917 m_ppqn, 915	x_to_w, 909 seq64::seqkeys, 917 ~seqkeys, 921 change_vert, 924 convert_y, 923 draw_area, 923 draw_key, 923 force_draw, 922 FruitySeqRollInput, 927 is_black_key, 924 m_hint_key, 928 m_hint_state, 928 m_keying, 928 m_keying, 928 m_keying_note, 928 m_scale, 928 m_scroll_offset_key, 927 m_scroll_offset_y, 928 m_seq, 927 m_show_octave_letters, 928 on_button_press_event, 925 on_enter_notify_event, 926 on_expose_event, 925 on_leave_notify_event, 926 on_realize, 924

reset, 924	toggle_current_sequence, 936
seqkeys, 920	unmute_all_tracks, 940
seqroll, 927	unset_edit_sequence, 934
set_hint_key, 921	seq64::seqroll, 942
set_hint_state, 922	\sim seqroll, 949
set_key, 921	add_chord, 950
set_listen_button_press, 922	add_note, 950
set_listen_button_release, 922	adding, 964
set_listen_motion_notify, 922	align_selection, 961
set_scale, 921	button_press, 961
update_pixmap, 923	button_press_initial, 960
update_sizes, 924	button_release, 961
seq64::seqmenu, 929	change_horz, 959
∼seqmenu, 934	change_vert, 959
const_iterator, 933	clear_flags, 962
create_seqedit, 937	clear_old, 962
current_seq, 934	clear_selected, 962
delete_current_sequence, 936	complete_paste, 954
get_current_sequence, 935	convert_sel_box_to_rect, 957
get_sequence, 935	convert_tn, 956
is_current_seq_active, 935	convert_tn_box_to_rect, 957
is_current_seq_in_edit, 935	convert_xy, 956
is_edit_sequence, 935	draw_background_on_pixmap, 952
is_modified, 934, 935	draw_events_on, 958
iterator, 933	draw_events_on_pixmap, 952
m_clipboard, 941	draw_progress_on_window, 953
m_current_seq, 941 m_eventedit, 941	draw_selection_on_window, 953 drop_action, 965
m_mainperf, 941	follow_progress, 954
m_menu, 941	force_draw, 954
m modified, 942	FruitySeqRollInput, 970
m_seqedit, 941	get_selected_box, 958
mainwnd, 940	grow_selected_box, 950 grow_selected_notes, 960
mute_all_tracks, 940	growing, 965
new current sequence, 935	horizontal adjust, 954
new sequence, 936	idle_progress, 958
on realize, 940	idle redraw, 958
popup_menu, 936	m_adding, 972
redraw, 938	m_background_sequence, 974
remove_seqedit, 937	m_cc, 975
seq_clear_perf, 939	m chord, 972
seq copy, 938	m_drawing_background_seq, 974
seq_cut, 938	m fruity interaction, 971
seq edit, 936	m growing, 972
seq_event_edit, 936	m_horizontal_adjust, 970
seq_new, 938	m_is_drag_pasting, 973
seq_paste, 939	m_is_drag_pasting_start, 973
seq_set_and_edit, 937	m_justselected_one, 973
seq_set_and_eventedit, 938	m_key, 972
seqedit, 940	m_move_delta_x, 973
SeqeditMap, 933	m_move_delta_y, 973
SeqeditPair, 933	m_move_snap_offset_x, 973
seqmenu, 933	m_moving, 972
set_bus_and_midi_channel, 939	m_moving_init, 972
set_edit_sequence, 934	m_note_length, 971
set_transposable, 939	m_old, 970
sm_seqedit_list, 941	m_painting, 972
toggle_all_tracks, 940	m_paste, 973

m_pos, 971	update_background, 952
m_ppqn, 971	update_mouse_pointer, 960
m_progress_x, 973	update_pixmap, 952
m_scale, 971	update_sizes, 952
m_scroll_offset_key, 974	vertical_adjust, 955
m_scroll_offset_ticks, 974	xy_to_rect, 956
m_scroll_offset_x, 974	seq64::seqtime, 975
m_scroll_offset_y, 974	\sim seqtime, 978
m_selected, 970	change_horz, 979
m_selecting, 972	draw_pixmap_on_window, 978
m_seq, 970	draw_progress_on_window, 978
m_seqkeys_wid, 971	idle_progress, 980
m_snap, 971	m_ppqn, 981
m_status, 975	m_scroll_offset_ticks, 981
m_trans_button_press, 974	m_scroll_offset_x, 981
m_transport_follow, 974	m_seq, 981
m_vertical_adjust, 970	m_zoom, 981
m_zoom, 971	on_button_press_event, 980
motion_notify, 962	on_button_release_event, 980
move_selected_notes, 959	on_expose_event, 980
move_selection_box, 959	on_realize, 980
moving, 965	on_size_allocate, 980
normal_action, 965	redraw, 978
note_off_length, 950	reset, 978
on_button_press_event, 966	seqtime, 977
on_button_release_event, 966	set_zoom, 978
on_enter_notify_event, 969	update_pixmap, 979
on_expose_event, 966	update_sizes, 979
on_focus_in_event, 967	seq64::sequence, 981
on_focus_out_event, 967	\sim sequence, 993
on_key_press_event, 968	add_chord, 1010
on_leave_notify_event, 969	add_event, 1010, 1011
on_motion_notify_event, 967	add_note, 1009
on_realize, 966	add_trigger, 1012
on_scroll_event, 968	adjust_offset, 1043
on_size_allocate, 969	adjust_timestamp, 1026
redraw, 954	adjust_trigger_offsets_to_length, 1043
redraw_events, 954	any_selected_notes, 994
reset, 953	append_event, 1011
scroll_offset_x, 962	apply_song_transpose, 1001
scroll_offset_y, 964	background_sequence, 1040, 1041
select_action, 965	change_event_data_lfo, 1030
selecting, 965	change_event_data_range, 1029
seqroll, 948	channel_match, 1044
set_adding, 960	check_queued_tick, 1005
set_background_sequence, 951	clear_triggers, 1019
set_chord, 951	clip_timestamp, 1027
set_current_offset_x_y, 964	clocks_per_metronome, 999
set_data_type, 951	copy_events, 1041
set_key, 951	copy_selected, 1024
set_note_length, 950	copy_selected_trigger, 1016
set_scale, 951	copy_triggers, 1018
set_snap, 949	cut_selected, 1024
set_zoom, 949	cut_selected_trigger, 1016
snap_x, 955	decrement_selected, 1031
snap_y, 955	del_selected_trigger, 1016
start_paste, 954	del_trigger, 1013
update_and_draw, 953	event_count, 995
. – – ,	_ ,

event_in_range, 1041	m_dirty_main, 1048
EventStack, 992	m_dirty_names, 1048
events, 994	m_dirty_perf, 1048
get_32nds_per_quarter, 999	m_editing, 1048
get_beat_width, 999	m_events, 1045
get_beats_per_bar, 998	m_events_clipboard, 1045
get_clipboard_box, 1026	m_events_redo, 1046
get_editing, 1001	m_events_undo, 1046
get_hold_undo, 996	m_events_undo_hold, 1045
get_last_tick, 1003	m_have_redo, 1046
get_length, 1003	m_have_undo, 1045
get_max_trigger, 1018	m_iterator_draw, 1046
get_measures, 998	m_last_tick, 1049
get_midi_bus, 1019	m length, 1050
get_midi_channel, 1007	m masterbus, 1047
get_minmax_note_events, 1037	m_maxbeats, 1049
get_name, 1001	m_midi_channel, 1046
get_next_event, 1037, 1038	m_musical_key, 1051
get_next_note_event, 1035	m_musical_scale, 1051
get next trigger, 1038	m_mutex, 1051
get_num_selected_events, 1023	m name, 1049
	_
get_num_selected_notes, 1023	m_note_off_margin, 1052
get_playing, 1004	m_note_off_velocity, 1051
get_ppqn, 998	m_note_on_velocity, 1051
get_quantized_rec, 1005	m_notes_on, 1047
get_queued, 1004	m_parent, 1045
get_queued_tick, 1004	m_playing, 1047
get_raise, 1002	m_playing_notes, 1047
get_recording, 1005	m_ppqn, 1049
get_selected_box, 1025	m_quantized_rec, 1048
get_song_mute, 1000	m_queued, 1048
get_thru, 1006	m_queued_tick, 1049
get_transposable, 1001	m_raise, 1049
get_trigger_count, 994	m_rec_vol, 1051
get_trigger_offset, 1019	m_recording, 1047
get_trigger_paste_tick, 994	m_seq_number, 1049
get_trigger_state, 1013	m_snap_tick, 1050
get_triggers, 1014	m_song_mute, 1047
grow_selected, 1032	m_thru, 1048
grow_trigger, 1013	m_time_beat_width, 1050
have redo, 996	m_time_beats_per_measure, 1050
have undo, 996	m_transposable, 1047
increment_selected, 1031	m_trigger_offset, 1049
intersect events, 1016	m_triggers, 1045
intersect notes, 1015	m_us_per_quarter_note, 1050
intersect_triggers, 1014	m_was_playing, 1047
is_dirty_edit, 1006	mark selected, 1033
is_dirty_main, 1006	measures_to_ticks, 999
is_dirty_names, 1006	mod_last_tick, 1003
is_dirty_perf, 1006	modify, 995
is smf 0, 1007	move selected notes, 1028
:	
link_new, 1034	move_selected_triggers_to, 1017
m_32nds_per_quarter, 1050	move_triggers, 1018
m_background_sequence, 1051	multiply_pattern, 1040
m_bus, 1046	musical_key, 1040
m_channel_match, 1046	musical_scale, 1040
m_clocks_per_metronome, 1050	name, 1001
m_dirty_edit, 1048	note_off_margin, 1041

number, 995	set_name, 997
off_playing_notes, 1034	set_parent, 1042
off_queued, 1004	set_playing, 1003
on_queued, 1004	set_quantized_rec, 1005
operator=, 993	set_raise, 1002
partial_assign, 993	set_rec_vol, 1000
paste_selected, 1025	set_recording, 1005
paste_trigger, 1016	set_snap_tick, 1005
pause, 1035	set_song_mute, 1000
perform, 1044	set_thru, 1005
play, 1008	set_transposable, 1001
play_note_off, 1034	set_trigger_offset, 1043
play_note_on, 1034	set_trigger_paste_tick, 994
play_queue, 1008	shift_notes, 1040
pop_redo, 997	show_events, 1041
pop_trigger_redo, 997	sort_events, 1012
pop_trigger_undo, 997	split_trigger, 1012
pop_undo, 997	stop, 1035
print, 1008	stream_event, 1028
print_triggers, 1008	stretch_selected, 1032
push_quantize, 1039	toggle_playing, 1004
push_trigger_undo, 997	toggle_queued, 1004
push_undo, 996	toggle_song_mute, 1000
put_event_on_bus, 1042	transpose_notes, 1039
quantize_events, 1038	triggerlist, 994
remove, 1043, 1044	triggers, 1045
remove_all, 1044	trim_timestamp, 1027
remove_marked, 1033	unpaint_all, 1033
remove_selected, 1033	unselect, 1033
reset_draw_marker, 1035	unselect_triggers, 1014
reset_draw_trigger_marker, 1035	us_per_quarter_note, 1000
select_action_e, 992	verify_and_link, 1033
select_all, 1024	zero_markers, 1034
select_all_notes, 1023	seq64::trigger, 1052
select_even_or_odd_notes, 1023	decrement_offset, 1055
select_event_handle, 1022	decrement_tick_end, 1055
select_events, 1020, 1021	decrement_tick_start, 1054
select_linked, 1022	increment_offset, 1055
select_note_events, 1020	increment_tick_end, 1054
select_trigger, 1014	increment_tick_start, 1054
selected_trigger_end, 1017	length, 1053
selected_trigger_start, 1017	m_offset, 1056
sequence, 993	m_selected, 1056
set_32nds_per_quarter, 999	m_tick_end, 1056
set_beat_width, 998	m_tick_start, 1056
set_beats_per_bar, 998	offset, 1055
set_dirty, 1007	operator<, 1053
set_dirty_mp, 1007	selected, 1055
set_editing, 1001	tick_end, 1054
set_have_redo, 996	tick_start, 1054
set_have_undo, 996	trigger, 1053
set_hold_undo, 995	seq64::triggers, 1056
set_last_tick, 1003	∼triggers, 1060
set_length, 1002	add, 1062
set_master_midi_bus, 1019	adjust_offset, 1070
set_measures, 998	adjust_offsets_to_length, 1063
set_midi_bus, 1019	clear, 1068
set_midi_channel, 1007	copy, 1067

copy_selected, 1065	m_is_valid, 1077
FruityPerfInput, 1071	name, 1075
get_maximum, 1067	operator=, 1074
get_selected_end, 1067	set_controller, 1076
get_selected_start, 1066	set_defaults, 1075
get_state, 1064	set_name, 1076
get_trigger_paste_tick, 1070	user_instrument, 1074
grow, 1063	seg64::user instrument t, 1077
grow edit t, 1059	controllers, 1078
intersect, 1065	controllers_active, 1078
List, 1059	instrument, 1078
m_clipboard, 1071	seq64::user_midi_bus, 1078
m_iterator_draw_trigger, 1072	channel_count, 1081
m_iterator_play_trigger, 1072	channel max, 1081
m_length, 1072	copy_definitions, 1082
m_parent, 1071	instrument, 1081
m_paste_tick, 1072	is_valid, 1080
m_ppqn, 1072	m_channel_count, 1082
m_redo_stack, 1072	m_is_valid, 1082
	m midi bus def, 1082
m_trigger_copied, 1072 m_triggers, 1071	:
_ ***	name, 1080 operator=, 1080
m_undo_stack, 1072	•
midi_container, 1070	set_defaults, 1080
midifile, 1071	set_instrument, 1081
move, 1067	set_name, 1082
move_selected, 1066	user_midi_bus, 1079, 1080
next, 1069	seq64::user_midi_bus_t, 1083
next_trigger, 1069	alias, 1083
operator=, 1060	instrument, 1083
paste, 1065	seq64::user_settings, 1083
play, 1062	add_bus, 1093
pop_redo, 1061	add_instrument, 1093
pop_undo, 1061	allow_two_perfedits, 1100, 1106
print, 1061	baseline_ppqn, 1106
push_undo, 1061	bpm_page_increment, 1105, 1108
remove, 1064	bpm_precision, 1105, 1107
remove_selected, 1065	bpm_step_increment, 1105, 1108
reset_draw_trigger_marker, 1069	bus, 1094
select, 1064	bus_count, 1094
Seq24PerfInput, 1071	bus_instrument, 1094
sequence, 1071	bus_name, 1094
set_length, 1061	BussConstIterator, 1091
set_ppqn, 1061	BussIterator, 1091
set_trigger_paste_tick, 1069	Busses, 1091
split, 1063, 1070	control height, 1099, 1104
Stack, 1059	controller_active, 1095
Stack, 1059 triggerlist, 1061	_ •
triggerlist, 1061	controller_active, 1095
	controller_active, 1095 controller_name, 1096
triggerlist, 1061 triggers, 1060	controller_active, 1095 controller_name, 1096 dump_summary, 1104
triggerlist, 1061 triggers, 1060 unselect, 1065 seq64::user_instrument, 1073	controller_active, 1095 controller_name, 1096 dump_summary, 1104 global_seq_feature, 1099 gmute_tracks, 1097
triggerlist, 1061 triggers, 1060 unselect, 1065 seq64::user_instrument, 1073 controller_active, 1076	controller_active, 1095 controller_name, 1096 dump_summary, 1104 global_seq_feature, 1099 gmute_tracks, 1097 grid_brackets, 1096, 1102
triggerlist, 1061 triggers, 1060 unselect, 1065 seq64::user_instrument, 1073 controller_active, 1076 controller_count, 1075	controller_active, 1095 controller_name, 1096 dump_summary, 1104 global_seq_feature, 1099 gmute_tracks, 1097 grid_brackets, 1096, 1102 grid_is_black, 1096
triggerlist, 1061 triggers, 1060 unselect, 1065 seq64::user_instrument, 1073 controller_active, 1076 controller_count, 1075 controller_max, 1075	controller_active, 1095 controller_name, 1096 dump_summary, 1104 global_seq_feature, 1099 gmute_tracks, 1097 grid_brackets, 1096, 1102 grid_is_black, 1096 grid_is_normal, 1096
triggerlist, 1061 triggers, 1060 unselect, 1065 seq64::user_instrument, 1073 controller_active, 1076 controller_count, 1075 controller_max, 1075 controller_name, 1075	controller_active, 1095 controller_name, 1096 dump_summary, 1104 global_seq_feature, 1099 gmute_tracks, 1097 grid_brackets, 1096, 1102 grid_is_black, 1096 grid_is_normal, 1096 grid_is_white, 1096
triggerlist, 1061 triggers, 1060 unselect, 1065 seq64::user_instrument, 1073 controller_active, 1076 controller_count, 1075 controller_max, 1075 controller_name, 1075 copy_definitions, 1077	controller_active, 1095 controller_name, 1096 dump_summary, 1104 global_seq_feature, 1099 gmute_tracks, 1097 grid_brackets, 1096, 1102 grid_is_black, 1096 grid_is_white, 1096 grid_style, 1096, 1102
triggerlist, 1061 triggers, 1060 unselect, 1065 seq64::user_instrument, 1073 controller_active, 1076 controller_count, 1075 controller_max, 1075 controller_name, 1075 copy_definitions, 1077 is_valid, 1075	controller_active, 1095 controller_name, 1096 dump_summary, 1104 global_seq_feature, 1099 gmute_tracks, 1097 grid_brackets, 1096, 1102 grid_is_black, 1096 grid_is_white, 1096 grid_style, 1096, 1102 instrument, 1094
triggerlist, 1061 triggers, 1060 unselect, 1065 seq64::user_instrument, 1073 controller_active, 1076 controller_count, 1075 controller_max, 1075 controller_name, 1075 copy_definitions, 1077	controller_active, 1095 controller_name, 1096 dump_summary, 1104 global_seq_feature, 1099 gmute_tracks, 1097 grid_brackets, 1096, 1102 grid_is_black, 1096 grid_is_white, 1096 grid_style, 1096, 1102

instrument_count, 1094	mainwid_y, 1099
instrument_name, 1095	mainwnd_cols, 1097, 1102
InstrumentConstIterator, 1092	mainwnd_rows, 1096, 1102
InstrumentIterator, 1092	max_sequence, 1097
Instruments, 1092	max_sets, 1097, 1102
inverse_colors, 1101, 1107	max_zoom, 1105
m_allow_two_perfedits, 1112	mc_baseline_ppqn, 1117
m_bpm_page_increment, 1115	mc max zoom, 1117
m bpm precision, 1115	mc_min_zoom, 1117
m_bpm_step_increment, 1115	midi_beat_width, 1105, 1108
m control height, 1110	midi_beats_per_bar, 1104, 1108
m_current_zoom, 1110	midi_beats_per_minute, 1104, 1108
m_global_seq_feature_save, 1111	midi_buss_override, 1105, 1107
m_gmute_tracks, 1116	midi_ppqn, 1104, 1107
m_grid_brackets, 1109	min_zoom, 1105
m_grid_style, 1109	normalize, 1093
m_h_perf_page_increment, 1112	operator=, 1093
m instruments, 1109	perf_h_page_increment, 1101, 1106
m inverse colors, 1112	perf_v_page_increment, 1101, 1106
m_mainwid_border, 1110	private_bus, 1108
	• —
m_mainwid_spacing, 1110	private_instrument, 1108
m_mainwid_x, 1117	progress_bar_colored, 1101, 1106
m_mainwid_y, 1117	progress_bar_thick, 1101, 1106
m_mainwnd_cols, 1110	save_user_config, 1101, 1102
m_mainwnd_rows, 1110	seqarea_seq_x, 1098, 1103
m_max_sequence, 1116	seqarea_seq_y, 1098, 1103
m_max_sets, 1110	seqarea_x, 1098, 1103
m_midi_beat_width, 1114	seqarea_y, 1098, 1103
m_midi_beats_per_measure, 1114	seqchars_x, 1097, 1103
m_midi_beats_per_minute, 1114	seqchars_y, 1098, 1103
m_midi_buses, 1109	seqedit_bgsequence, 1100
m_midi_buss_override, 1114	seqedit_key, 1100
m_midi_ppqn, 1114	seqedit_scale, 1099, 1100
m_progress_bar_colored, 1112	seqs_in_set, 1097
m_progress_bar_thick, 1112	set_bus_instrument, 1094
m_save_user_config, 1117	set_defaults, 1093
m_seqarea_seq_x, 1116	set_instrument_controllers, 1095
m_seqarea_seq_y, 1116	text_x, 1097, 1102
m_seqarea_x, 1116	text_y, 1097, 1103
m_seqarea_y, 1116	use_more_icons, 1101, 1107
m_seqchars_x, 1113	use_new_font, 1100, 1106
m_seqchars_y, 1113	user_settings, 1092, 1093
m_seqedit_bgsequence, 1111	userfile, 1109
m_seqedit_key, 1111	velocity_override, 1105, 1107
m_seqedit_scale, 1111	window_redraw_rate, 1101, 1107
m_seqs_in_set, 1115	zoom, 1099
m_text_x, 1113	seq64::userfile, 1118
m_text_y, 1113	\sim userfile, 1119
m_total_seqs, 1115	dump_setting_summary, 1120
m_use_more_icons, 1113	parse, 1119
m_use_new_font, 1111	userfile, 1119
m_v_perf_page_increment, 1112	write, 1120
m_velocity_override, 1114	seq_app_name
m_window_redraw_rate_ms, 1112	seq64, 101
mainwid_border, 1098, 1104	seq_clear_perf
mainwid_grid_style_t, 1092	seq64::seqmenu, 939
mainwid_spacing, 1098, 1104	seq_client_name
mainwid_x, 1098	seq64, 101

seq_copy	seq64::seqmenu, 933
seq64::seqmenu, 938	seqevent
seq_cut	seq64::seqdata, 868
seq64::seqmenu, 938	seq64::seqevent, 906
seq_edit	seqkeys
seq64::seqmenu, 936	seq64::seqkeys, 920
seq_event_edit	seqmenu
seq64::seqmenu, 936	seq64::seqedit, 891
seq_event_type_t	seq64::seqmenu, 933
seq64, 67	seqroll
seq_from_xy	seq64::seqdata, 868
seq64::mainwid, 408	seq64::seqkeys, 927
seq_in_playing_screen	seq64::seqroll, 948
seq64::perform, 739	seqs_in_set
seq_modifier_t	seq64::user_settings, 1097
seq64, 66	seqspec_string
seq_new	seq64::editable_event, 176
seq64::seqmenu, 938	seqtime
seq_number_size	seq64::seqtime, 977
seq64::midifile, 633	sequence
seq_paste	seq64::event_list, 222
seq64::seqmenu, 939	seq64::sequence, 993
seq_scroll_direction_t	seq64::triggers, 1071
seq64, 67	sequence_count
seq_set_and_edit	seq64::perform, 691
seq64::mainwid, 406	sequence_key
seq64::seqmenu, 937	seq64::mainwnd, 425
seq_set_and_eventedit	seq64::perform, 719
seq64::mainwid, 406	sequence_label
seq64::seqmenu, 938	seq64::perform, 719
seq_version	sequence_max
seq64, 101	seq64::perform, 691
seqarea_seq_x	sequence_playing_change
seq64::user_settings, 1098, 1103	seq64::perform, 707
seqarea_seq_y	sequence_playing_off
seq64::user_settings, 1098, 1103	seq64::perform, 708
seqarea_x	sequence_playing_on
seq64::user_settings, 1098, 1103	seq64::perform, 707
seqarea_y	sequence_playing_toggle
seq64::user_settings, 1098, 1103	seq64::perform, 707
seqchars_x	session_event
seq64::user_settings, 1097, 1103	seq64::jack_assistant, 322
seqchars_y	set
seq64::user_settings, 1098, 1103	seq64::keybindentry, 343
seqdata	seq64::midi_control, 503
seq64::seqdata, 860	set_32nds_per_quarter
seqedit	seq64::perform, 693
seq64::seqedit, 878	seq64::sequence, 999
seq64::seqmenu, 940	set_active
seqedit_bgsequence	seq64::perform, 736
seq64::user_settings, 1100	set_adding
seqedit_key	seq64::AbstractPerfInput, 136
seq64::user_settings, 1100	seq64::Seq24SeqEventInput, 855
seqedit_scale	seq64::seqroll, 960
seq64::user_settings, 1099, 1100	set_adding_pressed
SeqeditMap	seq64::AbstractPerfInput, 136
seq64::seqmenu, 933	set_all_clocks
SeqeditPair	seq64::busarray, 143

set_all_inputs	set_clock_status
seq64::busarray, 146	seq64::midibase, 590
set_all_key_events	set_config_files
seq64::keys_perform, 364	seq64::rc_settings, 809
seq64::keys_perform_gtk2, 376	set_controller
set_all_key_groups	seq64::user_instrument, 1076
seq64::keys_perform, 364	set_current_drop_x
seq64::keys_perform_gtk2, 377	seq64::gui_drawingarea_gtk2, 295
set_alsa_client_id	set_current_drop_y
seq64::lash, 387	seq64::gui_drawingarea_gtk2, 295
set_alt_name	set_current_event
seq64::midibase, 592	seq64::eventslots, 245
set_and_copy_mute_group	set_current_offset_x_y
seq64::perform, 733	seq64::seqroll, 964
set_api	set_data
seq64::rtmidi, 828	seq64::event, 200
set_api_info	set_data_type
seq64::rtmidi_info, 846	seq64::seqdata, 862
set_background_sequence	seq64::seqedit, 885
seq64::seqedit, 883	seq64::seqevent, 907
seq64::seqroll, 951	seq64::seqroll, 951
set beat width	set_defaults
seq64::jack_assistant, 319	seq64::rc_settings, 800
seq64::perfedit, 657	seq64::user_instrument, 1075
seq64::perform, 693	seq64::user_midi_bus, 1080
seq64::seqedit, 880	seq64::user_settings, 1093
seq64::sequence, 998	set_dirty
set_beats_per_bar	seq64::eventedit, 231
seq64::perfedit, 657	seq64::sequence, 1007
seq64::perform, 692	set_dirty_mp
seq64::seqedit, 879	seq64::sequence, 1007
·	
seq64::sequence, 998	set_edit_sequence
set_beats_per_measure	seq64::perform, 691
seq64::jack_assistant, 320	seq64::seqmenu, 934
set_beats_per_minute	set_editing
seq64::jack_assistant, 320	seq64::sequence, 1001
seq64::mastermidibase, 449	set_error_message
seq64::perform, 737	seq64::configfile, 165
set_bus_and_midi_channel	set_event_category
seq64::seqmenu, 939	seq64::eventedit, 230
set_bus_id	set_event_data_0
seq64::midibase, 591	seq64::eventedit, 230
set_bus_instrument	set_event_data_1
seq64::user_settings, 1094	seq64::eventedit, 230
set_channel	set_event_name
seq64::event, 199	seq64::eventedit, 230
set_chord	set_event_timestamp
seq64::seqedit, 883	seq64::eventedit, 230
seq64::seqroll, 951	set_follow_transport
set_clock	seq64::jack_assistant, 326
seq64::busarray, 143	seq64::perfedit, 656
seq64::mastermidibase, 444, 447	seq64::perform, 698
seq64::midibase, 590	set_group_mute_state
seq64::perform, 745	seq64::perform, 710
set clock bus	set_guides
seq64::perform, 720	seq64::perfedit, 658
set clock mod	seq64::perfroll, 768
seq64::midibase, 593	seq64::perftime, 786

set_have_redo	seq64::gui_drawingarea_gtk2, 285
seq64::perform, 726	set_listen_button_press
seq64::sequence, 996	seq64::seqkeys, 922
set_have_undo	set_listen_button_release
seq64::perform, 726	seq64::seqkeys, 922
seq64::sequence, 996	set_listen_motion_notify
set_hint_key	seq64::seqkeys, 922
seq64::seqkeys, 921	set_looped
set_hint_state	seq64::perfedit, 658
seq64::seqkeys, 922	set_looping
set_hold_undo	seq64::perform, 738
seq64::sequence, 995	set_master_midi_bus
set_image seq64::perfedit, 659	seq64::sequence, 1019
set input	set_measures seq64::seqedit, 882
seq64::busarray, 145	seq64::sequence, 998
seq64::mastermidibase, 448	set_midi_bus
seq64::midibase, 597	seq64::seqedit, 883
seq64::perform, 745	seq64::sequence, 1019
set_input_bus	set_midi_channel
seq64::perform, 720	seq64::seqedit, 882
set input status	seq64::sequence, 1007
seq64::midibase, 591	set_mode_group_learn
set instrument	seq64::perform, 733
seq64::user_midi_bus, 1081	set_mode_group_mute
set_instrument_controllers	seq64::perform, 732
seq64::user_settings, 1095	set_multi_name
set_jack_mode	seq64::midibase, 592
seq64::jack_assistant, 325	set_name
seq64::perfedit, 656	seq64::midibase, 591
seq64::perform, 696	seq64::sequence, 997
set_jack_running	seq64::user_instrument, 1076
seq64::jack_assistant, 327	seq64::user_midi_bus, 1082
set_jack_stop_tick	set_note
seq64::jack_assistant, 325	seq64::event, 205
seq64::perform, 696	set_note_length
set_jack_tick	seq64::seqedit, 879
seq64::perform, 702	seq64::seqroll, 950
set_key	set_note_velocity
seq64::seqedit, 883	seq64::event, 206
seq64::seqkeys, 921	set_offset
seq64::seqroll, 951	seq64::perform, 711
set_key_event	set_orig_ticks
seq64::keys_perform, 365	seq64::perform, 737
seq64::perform, 744	set_parent
set_key_group	seq64::sequence, 1042
seq64::keys_perform, 365 seq64::perform, 744	set_play_image seq64::mainwnd, 423
set_keys	set_playback_mode
seq64::keys_perform, 352	seq64::perform, 741
set_last_tick	set_playing
seq64::sequence, 1003	seq64::sequence, 1003
set_left_tick	set_playing_screenset
seq64::perform, 703	seq64::keys_perform, 356
set_length	seq64::perform, 731
seq64::sequence, 1002	set_port_id
seq64::triggers, 1061	seq64::midibase, 597
set line	set_port_open
	55poiopoii

seq64::midi_api, 489	seq64::mainwnd, 423
set_position	set_start_from_perfedit
seq64::jack_assistant, 330	seq64::jack_assistant, 326
set_ppqn	set_start_tick
seq64::jack_assistant, 324	seq64::perform, 703
seq64::mastermidibase, 449	set_status
seq64::perfroll, 770	seq64::event, 198, 199
seq64::perftime, 787	set_status_from_string
seq64::triggers, 1061	seq64::editable_event, 175
set_quantized_rec	set_status_keep_channel
seq64::sequence, 1005	seq64::event, 199
set_raise	set_sysex_size
seq64::sequence, 1002	seq64::event, 203
set_rec_vol	set_system_port_flag
seq64::seqedit, 880	seq64::midibase, 590
seq64::sequence, 1000	set_text
set_recording	seq64::eventslots, 249
seq64::sequence, 1005	set_thru
set_reposition	seq64::sequence, 1005
seq64::perform, 698	set_tick
set_right_tick	seq64::perform, 702
seq64::perform, 703	set_timestamp
set_running	seq64::event, 194
seq64::perform, 740	set_transposable
set_scale	seq64::seqmenu, 939
seq64::perftime, 786	seq64::sequence, 1001
seq64::seqedit, 883	set_transpose
seq64::seqkeys, 921	seq64::perfedit, 656
seq64::seqroll, 951	seq64::perform, 705
set_screen_set_notepad	set_transpose_image
seq64::perform, 730, 731	seq64::seqedit, 880
set_screenset	set_trigger_offset
seq64::mainwid, 405	seq64::sequence, 1043
seq64::perform, 725	set_trigger_paste_tick
set_seq_count	seq64::sequence, 994
seq64::eventedit, 229	seq64::triggers, 1069
set_seq_ppqn	set_virtual_name
seq64::eventedit, 229	seq64::midi_alsa, 473
set_seq_time_sig	seq64::midi_jack, 534
seq64::eventedit, 229	seq64::midibus, 611
set_seq_title	set_was_active
seq64::eventedit, 229	seq64::perform, 736
set_sequence_control_status	set_zoom
seq64::perform, 705	seq64::perfedit, 655
set_sequence_input	seq64::perfroll, 773
seq64::mastermidibase, 445	seq64::perftime, 787
set_snap	seq64::seqdata, 861
seq64::perfedit, 657	seq64::seqedit, 878
seq64::seqedit, 879	seq64::seqevent, 907
seq64::seqevent, 907	seq64::seqroll, 949
seq64::seqroll, 949	seq64::seqtime, 978
set_snap_tick	shift_lock
seq64::sequence, 1005	seq64::keystroke, 385
set_song_mute	shift_notes
seq64::mainwnd, 426	seq64::sequence, 1040
seq64::perform, 732	shorten_file_spec
seq64::sequence, 1000	seq64, 76
set_songlive_image	show_bus_values

seq64::midibase, 588	seq64::editable_event, 177
show_events	snap_x
seq64::sequence, 1041	seq64::perfroll, 772
show_jack_statuses	seq64::seqevent, 911
seq64, 96	seq64::seqroll, 955
show_midi	snap_y
seq64::rc_settings, 801, 805	seq64::seqevent, 911
show_position	seq64::seqroll, 955
seq64::jack_assistant, 328	snapshot_1
show_ui_sequence_key	seq64::keys_perform, 354
seq64::keys_perform, 362	snapshot_2
seq64::perform, 712	seq64::keys_perform, 355
show_ui_sequence_number	song_fill_seq_event
seq64::keys_perform, 362	seq64::midi_container, 499
seq64::perform, 712, 713	song_fill_seq_trigger
signal	seq64::midi_container, 499
seq64::condition_var, 160	song_mode
signal_action	seq64::keys_perform, 359
seq64::mainwnd, 428	song_start_mode
silence_jack_errors	seq64::jack_assistant, 326
seq64, 106	seq64::perform, 695
silence_jack_info	sort
seq64, 106	seq64::event_list, 218
size	sort_events
seq64::midi_container, 494	seq64::sequence, 1012
seq64::midi_list, 548	split
seq64::midi_vector, 580	seq64::midi_splitter, 571
SlotMap	seq64::triggers, 1063, 1070
seq64::keys_perform, 351	split_channel
sm_category_arrays	seq64::midi_splitter, 571
seq64::editable_event, 177	split_trigger
sm_category_names	seq64::perform, 727
seq64::editable_event, 177	seq64::perfroll, 773
sm_channel_event_names	seq64::sequence, 1012
seq64::editable_event, 177	Stack
sm_cond	seq64::triggers, 1059
seq64::condition_var, 161	start
sm_input_caps	seq64::busarray, 142
seq64::midi_alsa_info, 480	seq64::businfo, 152
sm_internal_keys	seq64::jack_assistant, 322
seq64::gui_assistant_gtk2, 279	seq64::keys_perform, 357
sm_mc_dummy	seq64::lash, 387
seq64::perform, 750	seq64::mastermidibase, 441
sm_meta_event_names	seq64::midibase, 595
seq64::editable_event, 177	seq64::perform, 734
sm_output_caps	start_from_perfedit
seq64::midi_alsa_info, 480	seq64::perform, 697
sm_prop_event_names	start_jack
seq64::editable_event, 177	seq64::perform, 735
sm_recursive_mutex	start_key
seq64::mutex, 637	seq64::perform, 716
sm_selected_api	start_paste
seq64::rtmidi_info, 848	seq64::seqevent, 910
sm_seqedit_list	seq64::seqroll, 954
seq64::seqmenu, 941	start_playing
sm_status_pairs	seq64::mainwnd, 423
seq64::jack_assistant, 335	seq64::perfedit, 660
sm_system_event_names	seq64::perform, 714

seq64::seqedit, 888	seq64, 78
stats	test_ignore_flags
seq64::rc_settings, 802, 806	seq64::rtmidi_in_data, 834
status	test_widget_click
seq64::midi_control, 502	seq64, 105
status_string	text_x
seq64::editable_event, 175	seq64::user_settings, 1097, 1102
stock_event_string	text_y
seq64::editable_event, 175	seq64::user_settings, 1097, 1103
stop	thru_change_callback
seq64::busarray, 142	seq64::seqedit, 884
seq64::businfo, 152	tick_end
seq64::jack_assistant, 322	seq64::trigger, 1054
seq64::keys_perform, 359	tick_multiplier
seq64::mastermidibase, 442	seq64::jack_assistant, 327
seq64::midibase, 595	tick_offset
seq64::perform, 734	seq64::perftime, 788
seq64::sequence, 1035	tick_start
stop_jack	seq64::trigger, 1054
seq64::perform, 735	tick_to_pixel
stop_key	seq64::perftime, 788
seq64::perform, 716	ticks_to_delta_time_us
stop_playing	 seq64, <mark>81</mark>
seq64::mainwnd, 424	time_as_measures
seq64::perfedit, 660	seq64::editable_event, 174
seq64::perform, 716	time_as_minutes
seq64::seqedit, 888	seq64::editable_event, 174
stream_event	time_as_pulses
seq64::sequence, 1028	seq64::editable_event, 174
stretch_selected	timeout
seq64::sequence, 1032	seq64::mainwid, 408
string_is_void	seq64::perfedit, 659
seq64, 77	seq64::seqedit, 887
string_not_void	timer_callback
seq64, 76	
string_to_midibyte	seq64::mainwnd, 422
seq64, 76	timestamp
string_to_pulses	seq64::editable_event, 174
seq64, 75	seq64::midi_message, 554
seq64::editable_events, 183	timestamp_format_t
strings_match	seq64::editable_event, 170
seq64, 77	timestamp_string
swap	seq64::editable_event, 173
seq64, 70	timestring_to_pulses
sync	seq64, 75, 100
seq64::jack_assistant, 329	timing
sysex	seq64::editable_events, 183
seq64::busarray, 143	to_string
seq64::businfo, 153	seq64, <mark>89</mark>
seq64::mastermidibase, 444	toLower
seq64::midibase, 595	seq64::mainwnd, 425
SysexContainer	toggle_all_tracks
seq64::event, 192	seq64::perform, 708
5540 moroni, 102	seq64::seqmenu, 940
tap_bpm	toggle_current_sequence
seq64::keys_perform, 361	seq64::seqmenu, 936
tempo_us_from_bpm	toggle_follow_transport
seq64, 79	seq64::jack_assistant, 326
tempo_us_to_bytes	seq64::perfedit, 656
	· · · · · · · · · · · · · · · · · · ·

seq64::perform, 698	Туре
toggle_jack	seq64::rterror, 819
seq64::keys_perform, 361	type
seq64::perfedit, 655	seq64::keybindentry, 342
toggle_jack_mode	undo
seq64::jack_assistant, 325	seq64::perfedit, 659
seq64::perform, 696	undo_callback
toggle_mutes	seq64::seqedit, 885
seq64::keys_perform, 361	unlock
toggle_other_names	seq64::mutex, 637
seq64::perform, 726	unmark
toggle_other_seqs	seq64::event, 204
seq64::perform, 725	unmark_all
toggle_playing	seq64::event_list, 220
seq64::mainwnd, 424	unmodify
seq64::perfedit, 660	-
seq64::sequence, 1004	seq64::event_list, 217
toggle_playing_tracks	unmute_all_tracks
seq64::perform, 708	seq64::seqmenu, 940
toggle_queued	unpaint
seq64::sequence, 1004	seq64::event, 204
toggle_song_mute	unpaint_all
seq64::sequence, 1000	seq64::event_list, 220
toggle_song_start_mode	seq64::sequence, 1033
seq64::jack_assistant, 326	unselect
seq64::perform, 695	seq64::event, 204
toggle_visible	seq64::sequence, 1033
seq64::lfownd, 392	seq64::triggers, 1065
top_index	unselect_all
seq64::eventslots, 244	seq64::event_list, 221
track_end_size	unselect_triggers
seq64::midifile, 633	seq64::sequence, 1014
·	unset_edit_sequence
track_name_size	seq64::perform, 692
seq64::midifile, 632	seq64::seqmenu, 934
transport_callback	unset_mode_group_learn
seq64::options, 641	seq64::perform, 733
transport_not_starting	unset_mode_group_mute
seq64::jack_assistant, 320	seq64::perform, 732
transport_state	unset_sequence_control_status
seq64::jack_assistant, 320	seq64::perform, 705
transpose_button_callback	update_all_windows
seq64::perfedit, 657	seq64::seqedit, 885
transpose_change_callback	update_and_draw
seq64::seqedit, 884	seq64::seqroll, 953
transpose_note	update_background
seq64::event, 205	seq64::seqroll, 952
transpose_notes	update_mainwid_sequences
seq64::sequence, 1039	seq64, 105
trigger	seq64::mainwid, 412
seq64::trigger, 1053	update_markers
triggerlist	seq64::mainwid, 407
seq64::sequence, 994	update_mouse_pointer
seq64::triggers, 1061	seq64::FruityPerfInput, 266
triggers	seq64::FruitySeqEventInput, 269
seq64::sequence, 1045	seq64::FruitySeqRollInput, 272
seq64::triggers, 1060	seq64::seqroll, 960
trim_timestamp	update_perfedit_sequences
seq64::sequence, 1027	seq64, 105
1 /	1 /

nog64::norfodit 661	valid aggregation
seq64::perfedit, 661	valid_sequence
update_pixmap	seq64::mainwid, 407
seq64::perftime, 789	value_to_name
seq64::seqdata, 862	seq64::editable_event, 172
seq64::seqevent, 908	varinum_size
seq64::seqkeys, 923	seq64::midifile, 631
seq64::seqroll, 952	velocity_override
seq64::seqtime, 979	seq64::user_settings, 1105, 1107
update_sequences_on_window	verify_and_link
seq64::mainwid, 405	seq64::event_list, 219
update_sizes	seq64::sequence, 1033
seq64::perfroll, 769	versiontext
seq64::perftime, 789	seq64, 124
seq64::seqdata, 862	vertical_adjust
seq64::seqevent, 908	seq64::perfroll, 774
seq64::seqkeys, 924	seq64::seqedit, 881
seq64::seqroll, 952	seq64::seqroll, 955
seq64::seqtime, 979	vertical_set
update_window_title	seq64::perfroll, 774
seq64::mainwnd, 425	seq64::seqedit, 881
us_per_quarter_note	
seq64::perform, 693, 694	wait
seq64::sequence, 1000	seq64::condition_var, 160
use more icons	wave_func
seq64::user_settings, 1101, 1107	seq64, 83
use_new_font	wave_type_name
seq64::user_settings, 1100, 1106	seq64, 71
user_callback	wave_type_t
seq64::midi_api, 489	seq64, 66
· — ·	what
seq64::rtmidi_in, 832	seq64::rterror, 820
seq64::rtmidi_in_data, 837	white
user_data	seq64::gui_palette_gtk2, 304
seq64::rtmidi_in_data, 836, 837	white key
user_filename	seq64::gui_palette_gtk2, 305
seq64::rc_settings, 804, 809	white_paint
user_filename_alt	seq64::gui_palette_gtk2, 305
seq64::rc_settings, 804, 810	width
user_filespec	seq64::gui_drawingarea_gtk2::rect, 818
seq64::rc_settings, 800	seq64::rect, 817
user_instrument	window redraw rate
seq64::user_instrument, 1074	
user_midi_bus	seq64::user_settings, 1101, 1107
seq64::user_midi_bus, 1079, 1080	window_x
user_settings	seq64::gui_drawingarea_gtk2, 284
seq64::user_settings, 1092, 1093	window_y
userfile	seq64::gui_drawingarea_gtk2, 284
seq64::user_settings, 1109	with_jack
seq64::userfile, 1119	seq64::rc_settings, 802
using_callback	with_jack_master
seq64::rtmidi_in_data, 836	seq64::rc_settings, 802, 806
usr	with_jack_master_cond
seq64, 99	seq64::rc_settings, 802, 806
•	with_jack_midi
v_adjustment	seq64::rc_settings, 802, 806
seq64::eventedit, 231	with_jack_transport
valid_buffer	seq64::rc_settings, 802, 806
seq64::midi_jack_data, 536	write
valid_midi_control_seq	seq64::configfile, 164
seq64::perform, 740	seq64::midifile, 620

```
seq64::optionsfile, 648
     seq64::userfile, 1120
write_byte
     seq64::midifile, 627
write_header
    seq64::midifile, 629
write long
     seq64::midifile, 626
write options files
     seq64, 87
write_prop_header
     seq64::midifile, 629
write_proprietary_track
    seq64::midifile, 630
write_seq_number
     seq64::midifile, 628
write short
     seq64::midifile, 626
write_song
    seq64::midifile, 621
write track
    seq64::midifile, 633
write_track_end
     seq64::midifile, 629
write_track_name
    seq64::midifile, 628
write_varinum
    seq64::midifile, 627
Х
    seq64::click, 157
    seq64::gui_drawingarea_gtk2::rect, 817
    seq64::rect, 816
x_to_w
     seq64::seqevent, 909
xy_to_rect
    seq64::seqdata, 862
    seq64::seqroll, 956
у
    seq64::click, 157
     seq64::gui_drawingarea_gtk2::rect, 817
    seq64::rect, 816
yellow
     seq64::gui_palette_gtk2, 305
zero_markers
    seq64::sequence, 1034
zoom
     seq64::user_settings, 1099
zoom_check
    seq64::perfedit, 654
zoom_power_of_2
    seq64, 78
```