

Actor Zeta

Generated by Doxygen 1.8.13

Contents

1	API Reference Start Page	1
2	~Projects~_actor-zeta_libactor_zeta_core_test_README README	3
3	Namespace Index	5
3.1	Namespace List	5
4	Hierarchical Index	7
4.1	Class Hierarchy	7
5	Class Index	9
5.1	Class List	9
6	File Index	11
6.1	File List	11
7	Namespace Documentation	13
7.1	actor_zeta Namespace Reference	13
7.1.1	Function Documentation	14
7.1.1.1	const_pointer_cast()	14
7.1.1.2	dynamic_pointer_cast()	14
7.1.1.3	get_pointer()	14
7.1.1.4	intrusive_ptr_add_ref()	14
7.1.1.5	intrusive_ptr_release()	15
7.1.1.6	operator!==([1/3])	15
7.1.1.7	operator!==([2/3])	15

7.1.1.8	operator!=(()) [3/3]	15
7.1.1.9	operator<()	15
7.1.1.10	operator==(()) [1/3]	15
7.1.1.11	operator==(()) [2/3]	16
7.1.1.12	operator==(()) [3/3]	16
7.1.1.13	static_pointer_cast()	16
7.1.1.14	swap()	16
7.2	actor_zeta::actor Namespace Reference	16
7.3	actor_zeta::behavior Namespace Reference	17
7.3.1	Function Documentation	17
7.3.1.1	make_response()	17
7.4	actor_zeta::contacts Namespace Reference	17
7.5	actor_zeta::environment Namespace Reference	18
7.5.1	Function Documentation	18
7.5.1.1	send()	18
7.6	actor_zeta::executor Namespace Reference	18
7.7	actor_zeta::messaging Namespace Reference	19
7.7.1	Enumeration Type Documentation	19
7.7.1.1	enqueue_result	19
7.7.1.2	message_priority	20
7.7.2	Function Documentation	20
7.7.2.1	make_message() [1/4]	20
7.7.2.2	make_message() [2/4]	20
7.7.2.3	make_message() [3/4]	20
7.7.2.4	make_message() [4/4]	21
7.8	actor_zeta::network Namespace Reference	21
7.8.1	Typedef Documentation	21
7.8.1.1	shared_muxplexer_ptr	21
7.8.1.2	unique_muxplexer_ptr	21
7.8.2	Enumeration Type Documentation	21
7.8.2.1	type_connect	21

8 Class Documentation	23
8.1 actor_zeta::behavior::abstract_action Class Reference	23
8.1.1 Detailed Description	23
8.1.2 Constructor & Destructor Documentation	24
8.1.2.1 abstract_action()	24
8.1.2.2 ~abstract_action()	24
8.1.3 Member Function Documentation	24
8.1.3.1 name()	24
8.1.3.2 operator()()	24
8.2 actor_zeta::actor::abstract_actor Class Reference	25
8.2.1 Detailed Description	26
8.2.2 Constructor & Destructor Documentation	26
8.2.2.1 ~abstract_actor()	26
8.2.2.2 abstract_actor()	26
8.2.3 Member Function Documentation	26
8.2.3.1 address()	26
8.2.3.2 env()	27
8.2.3.3 send() [1/2]	27
8.2.3.4 send() [2/2]	27
8.2.3.5 type()	27
8.3 actor_zeta::executor::abstract_coordinator Class Reference	27
8.3.1 Detailed Description	28
8.3.2 Constructor & Destructor Documentation	28
8.3.2.1 ~abstract_coordinator()	28
8.3.2.2 abstract_coordinator()	28
8.3.3 Member Function Documentation	28
8.3.3.1 max_throughput()	28
8.3.3.2 num_workers()	28
8.3.3.3 start()	29
8.3.3.4 submit()	29

8.4	actor_zeta::behavior::action Class Reference	29
8.4.1	Detailed Description	29
8.4.2	Constructor & Destructor Documentation	29
8.4.2.1	action() [1/4]	29
8.4.2.2	action() [2/4]	30
8.4.2.3	action() [3/4]	30
8.4.2.4	~action()	30
8.4.2.5	action() [4/4]	30
8.4.3	Member Function Documentation	30
8.4.3.1	operator()()	30
8.4.3.2	operator=() [1/2]	30
8.4.3.3	operator=() [2/2]	31
8.5	actor_zeta::actor::actor Class Reference	31
8.5.1	Detailed Description	31
8.5.2	Constructor & Destructor Documentation	31
8.5.2.1	actor() [1/5]	32
8.5.2.2	actor() [2/5]	32
8.5.2.3	actor() [3/5]	32
8.5.2.4	actor() [4/5]	32
8.5.2.5	actor() [5/5]	32
8.5.2.6	~actor()	32
8.5.3	Member Function Documentation	32
8.5.3.1	address()	33
8.5.3.2	operator bool()	33
8.5.3.3	operator"()()	33
8.5.3.4	operator->()	33
8.5.3.5	operator=() [1/4]	33
8.5.3.6	operator=() [2/4]	33
8.5.3.7	operator=() [3/4]	33
8.5.3.8	operator=() [4/4]	34

8.5.3.9	type()	34
8.6	actor_zeta::actor::actor_address Class Reference	34
8.6.1	Detailed Description	34
8.6.2	Constructor & Destructor Documentation	34
8.6.2.1	actor_address() [1/4]	35
8.6.2.2	actor_address() [2/4]	35
8.6.2.3	actor_address() [3/4]	35
8.6.2.4	actor_address() [4/4]	35
8.6.2.5	~actor_address()	35
8.6.3	Member Function Documentation	35
8.6.3.1	operator bool()	35
8.6.3.2	operator"!()	35
8.6.3.3	operator->()	36
8.6.3.4	operator=() [1/2]	36
8.6.3.5	operator=() [2/2]	36
8.7	actor_zeta::behavior::behavior Class Reference	36
8.7.1	Detailed Description	36
8.7.2	Constructor & Destructor Documentation	37
8.7.2.1	behavior() [1/3]	37
8.7.2.2	behavior() [2/3]	37
8.7.2.3	behavior() [3/3]	37
8.7.2.4	~behavior()	37
8.7.3	Member Function Documentation	37
8.7.3.1	insert()	37
8.7.3.2	operator=() [1/2]	37
8.7.3.3	operator=() [2/2]	38
8.7.3.4	run()	38
8.8	actor_zeta::actor::blocking_actor Class Reference	38
8.8.1	Detailed Description	39
8.8.2	Constructor & Destructor Documentation	39

8.8.2.1	<code>blocking_actor()</code>	39
8.8.2.2	<code>~blocking_actor()</code>	39
8.8.3	Member Function Documentation	39
8.8.3.1	<code>act()</code>	40
8.8.3.2	<code>launch()</code>	40
8.8.3.3	<code>run()</code>	40
8.9	<code>actor_zeta::messaging::blocking_mail_queue< T ></code> Class Template Reference	40
8.9.1	Detailed Description	41
8.9.2	Member Typedef Documentation	41
8.9.2.1	<code>cache_type</code>	41
8.9.2.2	<code>const_pointer</code>	41
8.9.2.3	<code>const_reference</code>	42
8.9.2.4	<code>lock_guard</code>	42
8.9.2.5	<code>pointer</code>	42
8.9.2.6	<code>queue_base_type</code>	42
8.9.2.7	<code>reference</code>	42
8.9.2.8	<code>unique_lock</code>	42
8.9.3	Constructor & Destructor Documentation	42
8.9.3.1	<code>blocking_mail_queue()</code> [1/3]	43
8.9.3.2	<code>blocking_mail_queue()</code> [2/3]	43
8.9.3.3	<code>blocking_mail_queue()</code> [3/3]	43
8.9.3.4	<code>~blocking_mail_queue()</code>	43
8.9.4	Member Function Documentation	43
8.9.4.1	<code>get()</code>	43
8.9.4.2	<code>high_priority_cache()</code>	43
8.9.4.3	<code>low_priority_cache()</code>	44
8.9.4.4	<code>normal_priority_cache()</code>	44
8.9.4.5	<code>operator=()</code> [1/2]	44
8.9.4.6	<code>operator=()</code> [2/2]	44
8.9.4.7	<code>put()</code>	44

8.9.4.8	sync()	44
8.10	actor_zeta::contacts::book_contacts Class Reference	45
8.10.1	Detailed Description	45
8.10.2	Constructor & Destructor Documentation	45
8.10.2.1	book_contacts() [1/2]	45
8.10.2.2	book_contacts() [2/2]	45
8.10.3	Member Function Documentation	45
8.10.3.1	all_view()	45
8.10.3.2	get()	46
8.10.3.3	get_group()	46
8.10.3.4	operator=()	46
8.10.3.5	put()	46
8.10.3.6	put_in_group()	46
8.11	actor_zeta::network::broker Class Reference	47
8.11.1	Detailed Description	48
8.11.2	Constructor & Destructor Documentation	48
8.11.2.1	broker()	48
8.11.2.2	~broker()	48
8.11.3	Member Function Documentation	48
8.11.3.1	initialize()	48
8.11.4	Member Data Documentation	48
8.11.4.1	multiplexer_	49
8.12	actor_zeta::network::connection_identifying Class Reference	49
8.12.1	Detailed Description	49
8.12.2	Constructor & Destructor Documentation	49
8.12.2.1	connection_identifying() [1/4]	49
8.12.2.2	connection_identifying() [2/4]	50
8.12.2.3	connection_identifying() [3/4]	50
8.12.2.4	connection_identifying() [4/4]	50
8.12.3	Member Function Documentation	50

8.12.3.1	ip()	50
8.12.3.2	operator=() [1/2]	50
8.12.3.3	operator=() [2/2]	50
8.12.3.4	operator==()	51
8.12.3.5	port()	51
8.12.3.6	to_string()	51
8.13	actor_zeta::environment::cooperation Class Reference	51
8.13.1	Detailed Description	51
8.13.2	Constructor & Destructor Documentation	52
8.13.2.1	cooperation() [1/3]	52
8.13.2.2	cooperation() [2/3]	52
8.13.2.3	cooperation() [3/3]	52
8.13.2.4	~cooperation()	52
8.13.3	Member Function Documentation	52
8.13.3.1	add()	52
8.13.3.2	add_shared()	52
8.13.3.3	get()	53
8.13.3.4	operator=() [1/2]	53
8.13.3.5	operator=() [2/2]	53
8.13.3.6	send()	53
8.13.3.7	send_all()	53
8.13.3.8	send_current()	53
8.14	actor_zeta::executor::coordinator< Policy > Class Template Reference	54
8.14.1	Detailed Description	55
8.14.2	Member Typedef Documentation	55
8.14.2.1	job_ptr	55
8.14.2.2	policy_data	55
8.14.2.3	worker_type	55
8.14.3	Constructor & Destructor Documentation	55
8.14.3.1	coordinator()	55

8.14.4 Member Function Documentation	56
8.14.4.1 data()	56
8.14.4.2 start()	56
8.14.4.3 submit()	56
8.14.4.4 worker_by_id()	56
8.15 actor_zeta::executor::work_sharing::coordinator_data Struct Reference	56
8.15.1 Detailed Description	57
8.15.2 Constructor & Destructor Documentation	57
8.15.2.1 coordinator_data()	57
8.15.3 Member Data Documentation	57
8.15.3.1 cv	57
8.15.3.2 lock	57
8.15.3.3 queue	57
8.16 actor_zeta::environment::environment Class Reference	58
8.16.1 Detailed Description	58
8.16.2 Constructor & Destructor Documentation	58
8.16.2.1 environment() [1/4]	59
8.16.2.2 environment() [2/4]	59
8.16.2.3 environment() [3/4]	59
8.16.2.4 ~environment()	59
8.16.2.5 environment() [4/4]	59
8.16.3 Member Function Documentation	59
8.16.3.1 manager_execution_device()	59
8.16.3.2 manager_group()	59
8.16.3.3 operator=() [1/2]	60
8.16.3.4 operator=() [2/2]	60
8.16.3.5 start()	60
8.16.4 Member Data Documentation	60
8.16.4.1 coordinator_	60
8.17 actor_zeta::executor::executable Struct Reference	60

8.17.1	Member Enumeration Documentation	61
8.17.1.1	executable_result	61
8.17.2	Constructor & Destructor Documentation	62
8.17.2.1	~executable()	62
8.17.3	Member Function Documentation	62
8.17.3.1	attach_to_scheduler()	62
8.17.3.2	detach_from_scheduler()	62
8.17.3.3	run()	62
8.18	actor_zeta::executor::execution_device Struct Reference	63
8.18.1	Detailed Description	63
8.18.2	Constructor & Destructor Documentation	63
8.18.2.1	execution_device() [1/3]	63
8.18.2.2	execution_device() [2/3]	64
8.18.2.3	execution_device() [3/3]	64
8.18.2.4	~execution_device()	64
8.18.3	Member Function Documentation	64
8.18.3.1	put_execute_latest()	64
8.19	actor_zeta::environment::group Class Reference	64
8.19.1	Detailed Description	65
8.19.2	Constructor & Destructor Documentation	65
8.19.2.1	group() [1/4]	65
8.19.2.2	group() [2/4]	65
8.19.2.3	group() [3/4]	65
8.19.2.4	group() [4/4]	66
8.19.2.5	~group()	66
8.19.3	Member Function Documentation	66
8.19.3.1	add() [1/3]	66
8.19.3.2	add() [2/3]	66
8.19.3.3	add() [3/3]	66
8.19.3.4	add_shared()	66

8.19.3.5	entry_point()	67
8.19.3.6	name()	67
8.19.3.7	operator=() [1/2]	67
8.19.3.8	operator=() [2/2]	67
8.19.3.9	send()	67
8.19.3.10	send_all()	67
8.19.3.11	send_current()	67
8.20	actor_zeta::contacts::group_contacts Class Reference	68
8.20.1	Detailed Description	68
8.20.2	Constructor & Destructor Documentation	68
8.20.2.1	group_contacts() [1/2]	68
8.20.2.2	group_contacts() [2/2]	68
8.20.3	Member Function Documentation	68
8.20.3.1	get()	68
8.20.3.2	get_all()	69
8.20.3.3	operator=()	69
8.20.3.4	put()	69
8.21	actor_zeta::intrusive_ptr< T > Class Template Reference	69
8.21.1	Detailed Description	70
8.21.2	Member Typedef Documentation	70
8.21.2.1	pointer	70
8.21.2.2	reference	70
8.21.3	Constructor & Destructor Documentation	70
8.21.3.1	intrusive_ptr() [1/5]	70
8.21.3.2	intrusive_ptr() [2/5]	70
8.21.3.3	intrusive_ptr() [3/5]	71
8.21.3.4	intrusive_ptr() [4/5]	71
8.21.3.5	intrusive_ptr() [5/5]	71
8.21.3.6	~intrusive_ptr()	71
8.21.4	Member Function Documentation	71

8.21.4.1	detach()	71
8.21.4.2	downcast()	71
8.21.4.3	get()	72
8.21.4.4	operator bool()	72
8.21.4.5	operator"!()	72
8.21.4.6	operator*()	72
8.21.4.7	operator->()	72
8.21.4.8	operator=() [1/2]	72
8.21.4.9	operator=() [2/2]	72
8.21.4.10	release()	73
8.21.4.11	reset()	73
8.21.4.12	swap()	73
8.21.4.13	upcast()	73
8.22	actor_zeta::actor::local_actor Class Reference	73
8.22.1	Detailed Description	75
8.22.2	Member Typedef Documentation	75
8.22.2.1	mailbox_type	75
8.22.3	Constructor & Destructor Documentation	75
8.22.3.1	~local_actor()	75
8.22.3.2	local_actor()	76
8.22.4	Member Function Documentation	76
8.22.4.1	attach()	76
8.22.4.2	device() [1/2]	76
8.22.4.3	device() [2/2]	76
8.22.4.4	has_next_message()	76
8.22.4.5	initialize()	76
8.22.4.6	launch()	77
8.22.4.7	mailbox()	77
8.22.4.8	next_message()	77
8.22.4.9	pop_to_cache()	77

8.22.4.10 <code>push_to_cache()</code>	77
8.22.5 Member Data Documentation	77
8.22.5.1 <code>contacts</code>	77
8.22.5.2 <code>life</code>	78
8.23 <code>actor_zeta::messaging::message</code> Class Reference	78
8.23.1 Detailed Description	78
8.23.2 Constructor & Destructor Documentation	79
8.23.2.1 <code>message()</code> [1/11]	79
8.23.2.2 <code>message()</code> [2/11]	79
8.23.2.3 <code>message()</code> [3/11]	79
8.23.2.4 <code>~message()</code>	79
8.23.2.5 <code>message()</code> [4/11]	79
8.23.2.6 <code>message()</code> [5/11]	79
8.23.2.7 <code>message()</code> [6/11]	80
8.23.2.8 <code>message()</code> [7/11]	80
8.23.2.9 <code>message()</code> [8/11]	80
8.23.2.10 <code>message()</code> [9/11]	80
8.23.2.11 <code>message()</code> [10/11]	80
8.23.2.12 <code>message()</code> [11/11]	81
8.23.3 Member Function Documentation	81
8.23.3.1 <code>clone()</code>	81
8.23.3.2 <code>get()</code>	81
8.23.3.3 <code>is_callback()</code>	81
8.23.3.4 <code>operator=()</code> [1/2]	81
8.23.3.5 <code>operator=()</code> [2/2]	81
8.23.3.6 <code>priority()</code>	82
8.23.3.7 <code>return_address()</code>	82
8.23.3.8 <code>type()</code>	82
8.24 <code>actor_zeta::messaging::message_body</code> Class Reference	82
8.24.1 Detailed Description	83

8.24.2	Constructor & Destructor Documentation	83
8.24.2.1	message_body() [1/5]	83
8.24.2.2	message_body() [2/5]	83
8.24.2.3	message_body() [3/5]	83
8.24.2.4	message_body() [4/5]	83
8.24.2.5	message_body() [5/5]	83
8.24.2.6	~message_body()	84
8.24.3	Member Function Documentation	84
8.24.3.1	clear()	84
8.24.3.2	empty()	84
8.24.3.3	get()	84
8.24.3.4	operator=() [1/3]	84
8.24.3.5	operator=() [2/3]	84
8.24.3.6	operator=() [3/3]	84
8.24.3.7	swap()	85
8.25	actor_zeta::messaging::message_header Class Reference	85
8.25.1	Detailed Description	85
8.25.2	Constructor & Destructor Documentation	85
8.25.2.1	message_header() [1/11]	86
8.25.2.2	message_header() [2/11]	86
8.25.2.3	message_header() [3/11]	86
8.25.2.4	~message_header()	86
8.25.2.5	message_header() [4/11]	86
8.25.2.6	message_header() [5/11]	86
8.25.2.7	message_header() [6/11]	87
8.25.2.8	message_header() [7/11]	87
8.25.2.9	message_header() [8/11]	87
8.25.2.10	message_header() [9/11]	87
8.25.2.11	message_header() [10/11]	87
8.25.2.12	message_header() [11/11]	87

8.25.3	Member Function Documentation	88
8.25.3.1	is_callback()	88
8.25.3.2	operator=() [1/2]	88
8.25.3.3	operator=() [2/2]	88
8.25.3.4	priorities()	88
8.25.3.5	return_address()	88
8.25.3.6	type()	88
8.26	actor_zeta::network::multiplexer Struct Reference	89
8.26.1	Detailed Description	89
8.26.2	Constructor & Destructor Documentation	89
8.26.2.1	~multiplexer()	89
8.26.3	Member Function Documentation	89
8.26.3.1	close()	89
8.26.3.2	new_tcp_connection()	89
8.26.3.3	new_tcp_listener()	90
8.26.3.4	registration_broker_address()	90
8.26.3.5	run()	90
8.26.3.6	write()	90
8.27	actor_zeta::ref_counted Class Reference	90
8.27.1	Detailed Description	91
8.27.2	Constructor & Destructor Documentation	92
8.27.2.1	~ref_counted()	92
8.27.2.2	ref_counted() [1/2]	92
8.27.2.3	ref_counted() [2/2]	92
8.27.3	Member Function Documentation	92
8.27.3.1	deref()	92
8.27.3.2	get_reference_count()	92
8.27.3.3	operator=()	92
8.27.3.4	ref()	93
8.27.3.5	unique()	93

8.27.4 Member Data Documentation	93
8.27.4.1 rc_	93
8.28 actor_zeta::behavior::request Class Reference	93
8.28.1 Detailed Description	93
8.28.2 Constructor & Destructor Documentation	94
8.28.2.1 request() [1/4]	94
8.28.2.2 request() [2/4]	94
8.28.2.3 request() [3/4]	94
8.28.2.4 ~request()	94
8.28.2.5 request() [4/4]	94
8.28.3 Member Function Documentation	94
8.28.3.1 contacts()	94
8.28.3.2 message()	95
8.28.3.3 operator=() [1/2]	95
8.28.3.4 operator=() [2/2]	95
8.29 actor_zeta::behavior::response Class Reference	95
8.29.1 Detailed Description	95
8.29.2 Constructor & Destructor Documentation	96
8.29.2.1 response() [1/4]	96
8.29.2.2 response() [2/4]	96
8.29.2.3 response() [3/4]	96
8.29.2.4 ~response()	96
8.29.2.5 response() [4/4]	96
8.29.3 Member Function Documentation	96
8.29.3.1 message()	96
8.29.3.2 operator=() [1/2]	97
8.29.3.3 operator=() [2/2]	97
8.29.3.4 receiver()	97
8.30 actor_zeta::actor::scheduled_actor Class Reference	97
8.30.1 Detailed Description	99

8.30.2	Constructor & Destructor Documentation	99
8.30.2.1	scheduled_actor()	99
8.30.3	Member Function Documentation	99
8.30.3.1	attach_to_scheduler()	99
8.30.3.2	detach_from_scheduler()	99
8.30.3.3	launch()	100
8.30.3.4	run()	100
8.30.3.5	send() [1/2]	100
8.30.3.6	send() [2/2]	100
8.31	actor_zeta::environment::shared_group Class Reference	100
8.31.1	Detailed Description	101
8.31.2	Constructor & Destructor Documentation	101
8.31.2.1	shared_group() [1/3]	101
8.31.2.2	shared_group() [2/3]	101
8.31.2.3	shared_group() [3/3]	101
8.31.2.4	~shared_group()	101
8.31.3	Member Function Documentation	102
8.31.3.1	add()	102
8.31.3.2	operator=() [1/2]	102
8.31.3.3	operator=() [2/2]	102
8.31.3.4	send_all()	102
8.31.3.5	send_current()	102
8.32	actor_zeta::skip Class Reference	103
8.32.1	Detailed Description	103
8.32.2	Constructor & Destructor Documentation	104
8.32.2.1	skip()	104
8.32.3	Member Function Documentation	104
8.32.3.1	operator()()	104
8.33	actor_zeta::sync_contacts Class Reference	104
8.33.1	Detailed Description	105

8.33.2	Constructor & Destructor Documentation	105
8.33.2.1	sync_contacts()	105
8.33.3	Member Function Documentation	105
8.33.3.1	operator()()	105
8.34	actor_zeta::behavior::type_action Class Reference	106
8.34.1	Detailed Description	106
8.34.2	Constructor & Destructor Documentation	106
8.34.2.1	type_action() [1/6]	106
8.34.2.2	type_action() [2/6]	106
8.34.2.3	type_action() [3/6]	106
8.34.2.4	~type_action()	107
8.34.2.5	type_action() [4/6]	107
8.34.2.6	type_action() [5/6]	107
8.34.2.7	type_action() [6/6]	107
8.34.3	Member Function Documentation	107
8.34.3.1	hash()	107
8.34.3.2	operator=() [1/2]	107
8.34.3.3	operator=() [2/2]	108
8.34.3.4	operator==()	108
8.34.3.5	to_string()	108
8.35	actor_zeta::executor::work_sharing Class Reference	108
8.35.1	Detailed Description	109
8.35.2	Member Typedef Documentation	109
8.35.2.1	queue_type	109
8.35.3	Constructor & Destructor Documentation	109
8.35.3.1	~work_sharing()	109
8.35.4	Member Function Documentation	109
8.35.4.1	central_enqueue()	110
8.35.4.2	dequeue()	110
8.35.4.3	enqueue()	110

8.35.4.4	external_enqueue()	110
8.35.4.5	foreach_central_resumable()	110
8.35.4.6	internal_enqueue()	110
8.35.4.7	resume_job_later()	111
8.36	actor_zeta::executor::worker< Policy > Class Template Reference	111
8.36.1	Detailed Description	112
8.36.2	Member Typedef Documentation	112
8.36.2.1	coordinator_ptr	112
8.36.2.2	job_ptr	112
8.36.2.3	policy_data	113
8.36.3	Constructor & Destructor Documentation	113
8.36.3.1	worker() [1/2]	113
8.36.3.2	worker() [2/2]	113
8.36.4	Member Function Documentation	113
8.36.4.1	data()	113
8.36.4.2	external_enqueue()	113
8.36.4.3	id()	114
8.36.4.4	max_throughput()	114
8.36.4.5	operator=()	114
8.36.4.6	parent()	114
8.36.4.7	put_execute_latest()	114
8.36.4.8	start()	114
8.37	actor_zeta::executor::work_sharing::worker_data Struct Reference	115
8.37.1	Detailed Description	115
8.37.2	Constructor & Destructor Documentation	115
8.37.2.1	worker_data()	115

9 File Documentation	117
9.1 abstract_action.hpp File Reference	117
9.2 abstract_actor.cpp File Reference	118
9.3 abstract_actor.hpp File Reference	119
9.4 abstract_coordinator.hpp File Reference	120
9.5 action.cpp File Reference	121
9.6 action.hpp File Reference	122
9.7 actor.cpp File Reference	123
9.8 actor.hpp File Reference	123
9.9 actor_address.cpp File Reference	125
9.10 actor_address.hpp File Reference	126
9.11 behavior.cpp File Reference	127
9.12 behavior.hpp File Reference	127
9.13 blocking_actor.cpp File Reference	129
9.14 blocking_actor.hpp File Reference	129
9.15 blocking_mail_queue.hpp File Reference	130
9.16 book_contacts.cpp File Reference	131
9.17 book_contacts.hpp File Reference	132
9.18 broker.cpp File Reference	133
9.19 broker.hpp File Reference	134
9.20 connection_identifying.cpp File Reference	135
9.21 connection_identifying.hpp File Reference	136
9.22 cooperation.cpp File Reference	137
9.23 cooperation.hpp File Reference	137
9.24 coordinator.hpp File Reference	139
9.25 description.h File Reference	139
9.26 environment.cpp File Reference	139
9.27 environment.hpp File Reference	140
9.28 executable.hpp File Reference	141
9.29 execution_device.hpp File Reference	142

9.30 forwards.hpp File Reference	143
9.31 forwards.hpp File Reference	144
9.32 group.cpp File Reference	144
9.33 group.hpp File Reference	145
9.34 group_contacts.cpp File Reference	146
9.35 group_contacts.hpp File Reference	147
9.36 intrusive_ptr.hpp File Reference	148
9.37 local_actor.cpp File Reference	150
9.38 local_actor.hpp File Reference	150
9.39 message.cpp File Reference	151
9.40 message.hpp File Reference	152
9.41 message_body.hpp File Reference	153
9.41.1 Function Documentation	154
9.41.1.1 swap()	154
9.42 message_header.cpp File Reference	154
9.43 message_header.hpp File Reference	155
9.44 message_priority.hpp File Reference	156
9.45 multiplexer.hpp File Reference	156
9.46 README.md File Reference	158
9.47 README.md File Reference	158
9.48 README.md File Reference	158
9.49 ref_counted.cpp File Reference	158
9.50 ref_counted.hpp File Reference	159
9.51 request.hpp File Reference	160
9.52 response.hpp File Reference	160
9.53 scheduled_actor.cpp File Reference	162
9.54 scheduled_actor.hpp File Reference	162
9.55 shared_group.cpp File Reference	163
9.56 shared_group.hpp File Reference	164
9.57 skip.cpp File Reference	165
9.58 skip.hpp File Reference	165
9.59 sync_contacts.cpp File Reference	167
9.60 sync_contacts.hpp File Reference	167
9.61 time_unit.hpp File Reference	168
9.61.1 Enumeration Type Documentation	169
9.61.1.1 time_unit	169
9.62 type_action.cpp File Reference	169
9.63 type_action.hpp File Reference	170
9.64 work_sharing.hpp File Reference	171
9.65 worker.hpp File Reference	171

Chapter 1

API Reference Start Page

Introduction

What is this?

Actor-zeta is an open source C++11 actor model implementation featuring lightweight & fast and more.

The library uses a standard practice for its versioning: *major.minor.patchlevel*. example : **0.0.1**

This project is in a very early / experimental stage.

Dependencies

CMake

Supported Compilers

GCC >= 4.8 Clang >= 3.3

Supported Operating Systems

Linux Mac OS X

Chapter 2

~Projects~_actor-zeta_libactor_zeta_core_test_R↵ EADME README

Test coverage is almost non-existent, but it's a start.

actor-zeta is an open source C++11 actor model implementation featuring lightweight & fast and more.

The library uses a standard practice for its versioning: major.minor.patchlevel. example : 0.0.1

This project is in a very early / experimental stage.

Dependencies

- CMake

Supported Compilers

- GCC >= 4.8
- Clang >= 3.3

Supported Operating Systems

- Linux
- Mac OS X

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

actor_zeta	13
actor_zeta::actor	16
actor_zeta::behavior	17
actor_zeta::contacts	17
actor_zeta::environment	18
actor_zeta::executor	18
actor_zeta::messaging	19
actor_zeta::network	21

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

actor_zeta::behavior::abstract_action	23
actor_zeta::skip	103
actor_zeta::sync_contacts	104
actor_zeta::executor::abstract_coordinator	27
actor_zeta::executor::coordinator< Policy >	54
actor_zeta::behavior::action	29
actor_zeta::actor::actor	31
actor_zeta::actor::actor_address	34
actor_zeta::behavior::behavior	36
actor_zeta::messaging::blocking_mail_queue< T >	40
actor_zeta::messaging::blocking_mail_queue< messaging::message >	40
actor_zeta::contacts::book_contacts	45
actor_zeta::network::connection_identifying	49
actor_zeta::environment::cooperation	51
actor_zeta::executor::work_sharing::coordinator_data	56
actor_zeta::environment::environment	58
actor_zeta::executor::executable	60
actor_zeta::actor::blocking_actor	38
actor_zeta::actor::scheduled_actor	97
actor_zeta::network::broker	47
actor_zeta::executor::execution_device	63
actor_zeta::executor::worker< Policy >	111
actor_zeta::environment::group	64
actor_zeta::contacts::group_contacts	68
actor_zeta::intrusive_ptr< T >	69
actor_zeta::intrusive_ptr< actor_zeta::actor::abstract_actor >	69
actor_zeta::messaging::message	78
actor_zeta::messaging::message_body	82
actor_zeta::messaging::message_header	85
actor_zeta::network::multiplexer	89
actor_zeta::ref_counted	90
actor_zeta::actor::abstract_actor	25
actor_zeta::actor::local_actor	73
actor_zeta::actor::blocking_actor	38

actor_zeta::actor::scheduled_actor	97
actor_zeta::behavior::request	93
actor_zeta::behavior::response	95
actor_zeta::environment::shared_group	100
actor_zeta::behavior::type_action	106
actor_zeta::executor::work_sharing	108
actor_zeta::executor::work_sharing::worker_data	115

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

actor_zeta::behavior::abstract_action	23
Abstract concept of an action	
actor_zeta::actor::abstract_actor	25
Abstract concept of an actor	
actor_zeta::executor::abstract_coordinator	27
Abstract concept of an coordination approach	
actor_zeta::behavior::action	29
Basic action implementation	
actor_zeta::actor::actor	31
Basic actor implementation	
actor_zeta::actor::actor_address	34
This represents an actor's address container	
actor_zeta::behavior::behavior	36
Class for lyfecycle determination	
actor_zeta::actor::blocking_actor	38
Represents actor type with blocking mode	
actor_zeta::messaging::blocking_mail_queue< T >	40
A mailbox class for message queue	
actor_zeta::contacts::book_contacts	45
Navigation map for actor & groups	
actor_zeta::network::broker	47
A broker for messaging	
actor_zeta::network::connection_identifying	49
Implementation of connection functionality	
actor_zeta::environment::cooperation	51
A logic combiner for groups	
actor_zeta::executor::coordinator< Policy >	54
Provides ruler for environment	
actor_zeta::executor::work_sharing::coordinator_data	56
This structure is used as coordinator data keeper	
actor_zeta::environment::environment	58
An actors workplace platform	
actor_zeta::executor::executable	60
actor_zeta::executor::execution_device	63
Execution_device	

actor_zeta::environment::group	
A group combinator for actors	64
actor_zeta::contacts::group_contacts	
Address container for groups	68
actor_zeta::intrusive_ptr< T >	
This class represents smart pointers	69
actor_zeta::actor::local_actor	
Class for location dependant type actor	73
actor_zeta::messaging::message	
Class to represent messages	78
actor_zeta::messaging::message_body	
A message value container	82
actor_zeta::messaging::message_header	
A message description	85
actor_zeta::network::multiplexer	
Multiplexing utility class	89
actor_zeta::ref_counted	
This class represents reference counter	90
actor_zeta::behavior::request	
This is a request container for messaging	93
actor_zeta::behavior::response	
This is a response container for messaging	95
actor_zeta::actor::scheduled_actor	
Represents scheduling type of actor	97
actor_zeta::environment::shared_group	
Group realisation with common resource	100
actor_zeta::skip	
A class used for skipping action	103
actor_zeta::sync_contacts	
A class used for contact operations synchronization	104
actor_zeta::behavior::type_action	
Type of commands used for lyfecycle	106
actor_zeta::executor::work_sharing	
Stands for work sharing approach	108
actor_zeta::executor::worker< Policy >	
A simple worker module class	111
actor_zeta::executor::work_sharing::worker_data	
This structure is used as worker data keeper	115

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

abstract_action.hpp	117
abstract_actor.cpp	118
abstract_actor.hpp	119
abstract_coordinator.hpp	120
action.cpp	121
action.hpp	122
actor.cpp	123
actor.hpp	123
actor_address.cpp	125
actor_address.hpp	126
behavior.cpp	127
behavior.hpp	127
blocking_actor.cpp	129
blocking_actor.hpp	129
blocking_mail_queue.hpp	130
book_contacts.cpp	131
book_contacts.hpp	132
broker.cpp	133
broker.hpp	134
connection_identifying.cpp	135
connection_identifying.hpp	136
cooperation.cpp	137
cooperation.hpp	137
coordinator.hpp	139
description.h	139
environment.cpp	139
environment.hpp	140
executable.hpp	141
execution_device.hpp	142
libactor_zeta_core/actor-zeta/forwards.hpp	143
libactor_zeta_io/actor-zeta/forwards.hpp	144
group.cpp	144
group.hpp	145
group_contacts.cpp	146
group_contacts.hpp	147

intrusive_ptr.hpp	148
local_actor.cpp	150
local_actor.hpp	150
message.cpp	151
message.hpp	152
message_body.hpp	153
message_header.cpp	154
message_header.hpp	155
message_priority.hpp	156
multiplexer.hpp	156
ref_counted.cpp	158
ref_counted.hpp	159
request.hpp	160
response.hpp	160
scheduled_actor.cpp	162
scheduled_actor.hpp	162
shared_group.cpp	163
shared_group.hpp	164
skip.cpp	165
skip.hpp	165
sync_contacts.cpp	167
sync_contacts.hpp	167
time_unit.hpp	168
type_action.cpp	169
type_action.hpp	170
work_sharing.hpp	171
worker.hpp	171

Chapter 7

Namespace Documentation

7.1 actor_zeta Namespace Reference

Namespaces

- [actor](#)
- [behavior](#)
- [contacts](#)
- [environment](#)
- [executor](#)
- [messaging](#)
- [network](#)

Classes

- class [intrusive_ptr](#)
This class represents smart pointers.
- class [ref_counted](#)
This class represents reference counter.
- class [skip](#)
A class used for skipping action.
- class [sync_contacts](#)
A class used for contact operations synchronization.

Functions

- `template<class T, class U >`
`bool operator== (intrusive_ptr< T > const &a, intrusive_ptr< U > const &b) noexcept`
- `template<class T, class U >`
`bool operator!= (intrusive_ptr< T > const &a, intrusive_ptr< U > const &b) noexcept`
- `template<class T >`
`bool operator== (intrusive_ptr< T > const &a, T *b) noexcept`
- `template<class T >`
`bool operator!= (intrusive_ptr< T > const &a, T *b) noexcept`
- `template<class T >`
`bool operator== (T *a, intrusive_ptr< T > const &b) noexcept`

- `template<class T >`
`bool operator!= (T *a, intrusive_ptr< T > const &b) noexcept`
- `template<class T, class U >`
`bool operator< (intrusive_ptr< T > const &a, intrusive_ptr< U > const &b) noexcept`
- `template<class T >`
`void swap (intrusive_ptr< T > &a, intrusive_ptr< T > &b) noexcept`
- `template<class T >`
`T * get_pointer (intrusive_ptr< T > const &p) noexcept`
- `template<class T, class U >`
`intrusive_ptr< T > static_pointer_cast (intrusive_ptr< U > const &r) noexcept`
- `template<class T, class U >`
`intrusive_ptr< T > const_pointer_cast (intrusive_ptr< U > const &r) noexcept`
- `template<class T, class U >`
`intrusive_ptr< T > dynamic_pointer_cast (intrusive_ptr< U > const &r) noexcept`
- `void intrusive_ptr_add_ref (ref_counted *p)`
- `void intrusive_ptr_release (ref_counted *p)`

7.1.1 Function Documentation

7.1.1.1 const_pointer_cast()

```
template<class T, class U >
intrusive_ptr<T> actor_zeta::const_pointer_cast (
    intrusive_ptr< U > const & r ) [noexcept]
```

7.1.1.2 dynamic_pointer_cast()

```
template<class T, class U >
intrusive_ptr<T> actor_zeta::dynamic_pointer_cast (
    intrusive_ptr< U > const & r ) [noexcept]
```

7.1.1.3 get_pointer()

```
template<class T >
T* actor_zeta::get_pointer (
    intrusive_ptr< T > const & p ) [noexcept]
```

7.1.1.4 intrusive_ptr_add_ref()

```
void actor_zeta::intrusive_ptr_add_ref (
    ref_counted * p ) [inline]
```

7.1.1.5 intrusive_ptr_release()

```
void actor_zeta::intrusive_ptr_release (
    ref_counted * p ) [inline]
```

7.1.1.6 operator!=() [1/3]

```
template<class T , class U >
bool actor_zeta::operator!= (
    intrusive_ptr< T > const & a,
    intrusive_ptr< U > const & b ) [noexcept]
```

7.1.1.7 operator!=() [2/3]

```
template<class T >
bool actor_zeta::operator!= (
    intrusive_ptr< T > const & a,
    T * b ) [noexcept]
```

7.1.1.8 operator!=() [3/3]

```
template<class T >
bool actor_zeta::operator!= (
    T * a,
    intrusive_ptr< T > const & b ) [noexcept]
```

7.1.1.9 operator<()

```
template<class T , class U >
bool actor_zeta::operator< (
    intrusive_ptr< T > const & a,
    intrusive_ptr< U > const & b ) [noexcept]
```

7.1.1.10 operator==(1/3)

```
template<class T , class U >
bool actor_zeta::operator==(
    intrusive_ptr< T > const & a,
    intrusive_ptr< U > const & b ) [noexcept]
```

7.1.1.11 operator==([2/3]

```
template<class T >
bool actor_zeta::operator== (
    intrusive_ptr< T > const & a,
    T * b ) [noexcept]
```

7.1.1.12 operator==([3/3]

```
template<class T >
bool actor_zeta::operator== (
    T * a,
    intrusive_ptr< T > const & b ) [noexcept]
```

7.1.1.13 static_pointer_cast()

```
template<class T , class U >
intrusive_ptr<T> actor_zeta::static_pointer_cast (
    intrusive_ptr< U > const & r ) [noexcept]
```

7.1.1.14 swap()

```
template<class T >
void actor_zeta::swap (
    intrusive_ptr< T > & a,
    intrusive_ptr< T > & b ) [inline], [noexcept]
```

7.2 actor_zeta::actor Namespace Reference**Classes**

- class [abstract_actor](#)
abstract concept of an actor
- class [actor](#)
Basic actor implementation.
- class [actor_address](#)
This represents an actor's address container.
- class [blocking_actor](#)
Represents actor type with blocking mode.
- class [local_actor](#)
Class for location dependant type actor.
- class [scheduled_actor](#)
Represents scheduling type of actor.

7.3 actor_zeta::behavior Namespace Reference

Classes

- class [abstract_action](#)
abstract concept of an action
- class [action](#)
Basic action implementation.
- class [behavior](#)
Class for lifecycle determination.
- class [request](#)
This is a request container for messaging.
- class [response](#)
This is a response container for messaging.
- class [type_action](#)
Type of commands used for lifecycle.

Functions

- [response](#) * [make_response](#) ([actor::actor_address](#) receiver_, [messaging::message](#) *msg)

7.3.1 Function Documentation

7.3.1.1 make_response()

```
response* actor_zeta::behavior::make_response (  
    actor::actor\_address receiver_,  
    messaging::message * msg ) [inline]
```

7.4 actor_zeta::contacts Namespace Reference

Classes

- class [book_contacts](#)
Navigation map for actor & groups.
- class [group_contacts](#)
Address container for groups.

7.5 actor_zeta::environment Namespace Reference

Classes

- class [cooperation](#)
A logic combiner for groups.
- class [environment](#)
An actors workplace platform.
- class [group](#)
A group combinator for actors.
- class [shared_group](#)
Group realisation with common resource.

Functions

- `template<class V >`
`void send (actor_zeta::environment::cooperation &c, std::string commanda, V value)`

7.5.1 Function Documentation

7.5.1.1 `send()`

```
template<class V >
void actor_zeta::environment::send (
    actor\_zeta::environment::cooperation & c,
    std::string commanda,
    V value ) [inline]
```

7.6 actor_zeta::executor Namespace Reference

Classes

- class [abstract_coordinator](#)
abstract concept of an coordination approach
- class [coordinator](#)
Provides ruler for environment.
- struct [executable](#)
- struct [execution_device](#)
execution_device
- class [work_sharing](#)
Stands for work sharing approach.
- class [worker](#)
A simple worker module class.

7.7 actor_zeta::messaging Namespace Reference

Classes

- class [blocking_mail_queue](#)
A mailbox class for message queue.
- class [message](#)
Class to represent messages.
- class [message_body](#)
A message value container.
- class [message_header](#)
A message description.

Enumerations

- enum [enqueue_result](#) { [enqueue_result::success](#), [enqueue_result::unblocked_reader](#), [enqueue_result::queue_closed](#) }
A strongly typed enum class representing enqueue result.
- enum [message_priority](#) : int { [message_priority::low](#) = 0, [message_priority::normal](#), [message_priority::high](#) }

Functions

- template<std::size_t N, typename T >
[message](#) * [make_message](#) (const char(&aStr)[N], T data)
- template<typename T >
[message](#) * [make_message](#) (const std::string &type, T data)
- template<typename T >
[message](#) * [make_message](#) (const std::string &type, T data, [actor::actor_address](#) address)
- template<std::size_t N, typename T >
[message](#) * [make_message](#) (const char(&aStr)[N], T data, [actor::actor_address](#) address)

7.7.1 Enumeration Type Documentation

7.7.1.1 enqueue_result

```
enum actor_zeta::messaging::enqueue_result [strong]
```

A strongly typed enum class representing enqueue result.

Enumerator

success	is coded as std::int of value 0
unblocked_reader	is coded as std::int of value 1
queue_closed	is coded as std::int of value 2

7.7.1.2 message_priority

```
enum actor_zeta::messaging::message_priority : int [strong]
```

Enumerator

low	is coded as std::int of value 0
normal	is coded as std::int of value 1
high	is coded as std::int of value 2

7.7.2 Function Documentation

7.7.2.1 make_message() [1/4]

```
template<std::size_t N, typename T >
message* actor_zeta::messaging::make_message (
    const char(&) aStr[N],
    T data ) [inline]
```

7.7.2.2 make_message() [2/4]

```
template<typename T >
message* actor_zeta::messaging::make_message (
    const std::string & type,
    T data ) [inline]
```

7.7.2.3 make_message() [3/4]

```
template<typename T >
message* actor_zeta::messaging::make_message (
    const std::string & type,
    T data,
    actor::actor_address address ) [inline]
```

7.7.2.4 make_message() [4/4]

```
template<std::size_t N, typename T >
message* actor_zeta::messaging::make_message (
    const char(&) aStr[N],
    T data,
    actor::actor_address address ) [inline]
```

7.8 actor_zeta::network Namespace Reference

Classes

- class [broker](#)
A broker for messaging.
- class [connection_identifying](#)
Implementation of connection functionality.
- struct [multiplexer](#)
Multiplexing utility class.

Typedefs

- using [unique_multiplexer_ptr](#) = std::unique_ptr< [multiplexer](#) >
- using [shared_multiplexer_ptr](#) = std::shared_ptr< [multiplexer](#) >

Enumerations

- enum [type_connect](#) : int { [type_connect::tcp](#), [type_connect::udp](#) }
A strongly typed enum class representing connection type.

7.8.1 Typedef Documentation

7.8.1.1 shared_multiplexer_ptr

```
using actor_zeta::network::shared_multiplexer_ptr = typedef std::shared_ptr<multiplexer>
```

7.8.1.2 unique_multiplexer_ptr

```
using actor_zeta::network::unique_multiplexer_ptr = typedef std::unique_ptr<multiplexer>
```

7.8.2 Enumeration Type Documentation

7.8.2.1 type_connect

```
enum actor_zeta::network::type_connect : int [strong]
```

A strongly typed enum class representing connection type.

Enumerator

tcp	is coded as std::int of value 0
udp	is coded as std::int of value 1

Chapter 8

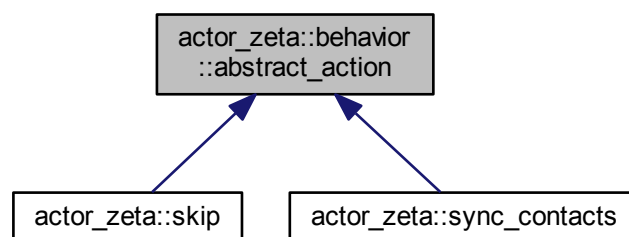
Class Documentation

8.1 actor_zeta::behavior::abstract_action Class Reference

abstract concept of an action

```
#include <abstract_action.hpp>
```

Inheritance diagram for actor_zeta::behavior::abstract_action:



Public Member Functions

- template<std::size_t N>
 abstract_action (const char(&aStr)[N])
- virtual response * operator() (request *)=0
- auto name () const -> const type_action &
- virtual ~abstract_action ()=default

8.1.1 Detailed Description

abstract concept of an action

8.1.2 Constructor & Destructor Documentation

8.1.2.1 `abstract_action()`

```
template<std::size_t N>
actor_zeta::behavior::abstract_action::abstract_action (
    const char(&) aStr[N] ) [inline]
```

8.1.2.2 `~abstract_action()`

```
virtual actor_zeta::behavior::abstract_action::~~abstract_action ( ) [virtual], [default]
```

8.1.3 Member Function Documentation

8.1.3.1 `name()`

```
auto actor_zeta::behavior::abstract_action::name ( ) const -> const type\_action & [inline]
```

8.1.3.2 `operator>()`

```
virtual response* actor_zeta::behavior::abstract_action::operator() (
    request * ) [pure virtual]
```

Implemented in [actor_zeta::sync_contacts](#), and [actor_zeta::skip](#).

The documentation for this class was generated from the following file:

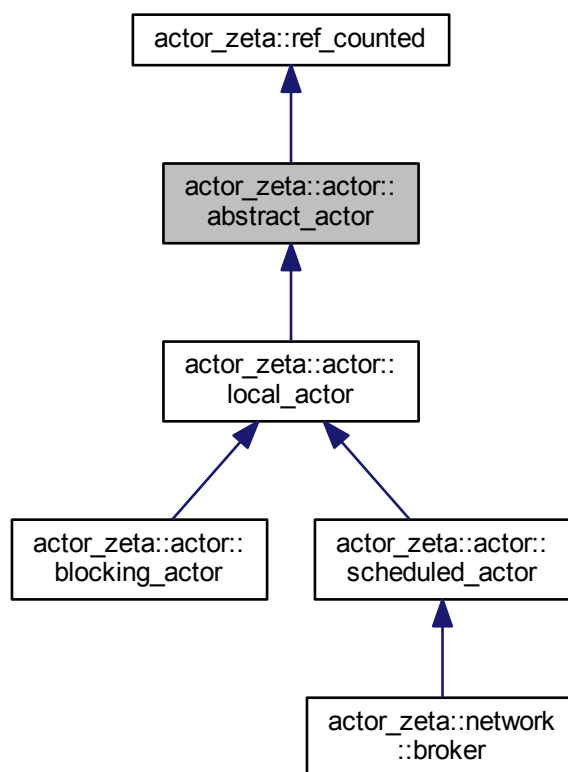
- [abstract_action.hpp](#)

8.2 actor_zeta::actor::abstract_actor Class Reference

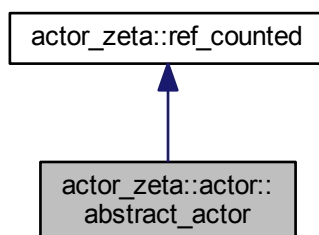
abstract concept of an actor

```
#include <abstract_actor.hpp>
```

Inheritance diagram for actor_zeta::actor::abstract_actor:



Collaboration diagram for actor_zeta::actor::abstract_actor:



Public Member Functions

- `const std::string & type () const`
- `virtual bool send (messaging::message *)=0`
- `virtual bool send (messaging::message *, executor::execution_device *)=0`
- `virtual ~abstract_actor ()`
- `actor_address address () const noexcept`
- `environment::environment * env () const`

Protected Member Functions

- `abstract_actor (environment::environment *, const std::string &)`

Additional Inherited Members

8.2.1 Detailed Description

abstract concept of an actor

8.2.2 Constructor & Destructor Documentation

8.2.2.1 ~abstract_actor()

```
virtual actor_zeta::actor::abstract_actor::~~abstract_actor ( ) [inline], [virtual]
```

8.2.2.2 abstract_actor()

```
actor_zeta::actor::abstract_actor::abstract_actor (
    environment::environment * env,
    const std::string & type ) [protected]
```

8.2.3 Member Function Documentation

8.2.3.1 address()

```
actor_address actor_zeta::actor::abstract_actor::address ( ) const [noexcept]
```

8.2.3.2 env()

```
environment::environment * actor_zeta::actor::abstract_actor::env ( ) const
```

8.2.3.3 send() [1/2]

```
virtual bool actor_zeta::actor::abstract_actor::send (
    messaging::message * ) [pure virtual]
```

Implemented in [actor_zeta::actor::scheduled_actor](#).

8.2.3.4 send() [2/2]

```
virtual bool actor_zeta::actor::abstract_actor::send (
    messaging::message * ,
    executor::execution_device * ) [pure virtual]
```

Implemented in [actor_zeta::actor::scheduled_actor](#).

8.2.3.5 type()

```
const std::string & actor_zeta::actor::abstract_actor::type ( ) const
```

The documentation for this class was generated from the following files:

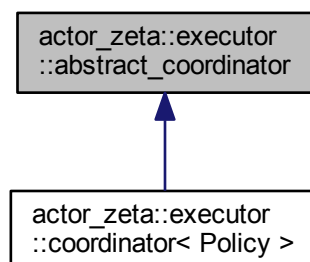
- [abstract_actor.hpp](#)
- [abstract_actor.cpp](#)

8.3 actor_zeta::executor::abstract_coordinator Class Reference

abstract concept of an coordination approach

```
#include <abstract_coordinator.hpp>
```

Inheritance diagram for actor_zeta::executor::abstract_coordinator:



Public Member Functions

- virtual void [submit](#) ([executable](#) *)=0
- virtual void [start](#) ()=0
- virtual [~abstract_coordinator](#) ()=default
- [abstract_coordinator](#) (size_t num_worker_threads, size_t max_throughput_param)
- size_t [max_throughput](#) () const
- size_t [num_workers](#) () const

8.3.1 Detailed Description

abstract concept of an coordination approach

8.3.2 Constructor & Destructor Documentation

8.3.2.1 [~abstract_coordinator\(\)](#)

```
virtual actor_zeta::executor::abstract_coordinator::~abstract_coordinator ( ) [virtual],
[default]
```

8.3.2.2 [abstract_coordinator\(\)](#)

```
actor_zeta::executor::abstract_coordinator::abstract_coordinator (
    size_t num_worker_threads,
    size_t max_throughput_param ) [inline], [explicit]
```

8.3.3 Member Function Documentation

8.3.3.1 [max_throughput\(\)](#)

```
size_t actor_zeta::executor::abstract_coordinator::max_throughput ( ) const [inline]
```

8.3.3.2 [num_workers\(\)](#)

```
size_t actor_zeta::executor::abstract_coordinator::num_workers ( ) const [inline]
```

8.3.3.3 start()

```
virtual void actor_zeta::executor::abstract_coordinator::start ( ) [pure virtual]
```

Implemented in [actor_zeta::executor::coordinator< Policy >](#).

8.3.3.4 submit()

```
virtual void actor_zeta::executor::abstract_coordinator::submit (
    executable * ) [pure virtual]
```

The documentation for this class was generated from the following file:

- [abstract_coordinator.hpp](#)

8.4 actor_zeta::behavior::action Class Reference

Basic action implementation.

```
#include <action.hpp>
```

Public Member Functions

- [action](#) ()=default
- [action](#) (const [action](#) &)=delete
- [action](#) & [operator=](#) (const [action](#) &)=delete
- [action](#) ([action](#) &&)=default
- [action](#) & [operator=](#) ([action](#) &&)=default
- [~action](#) ()=default
- [action](#) ([abstract_action](#) *)
- [response](#) * [operator\(\)](#) ([request](#) *)

8.4.1 Detailed Description

Basic action implementation.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 action() [1/4]

```
actor_zeta::behavior::action::action ( ) [default]
```

8.4.2.2 action() [2/4]

```
actor_zeta::behavior::action::action (
    const action & ) [delete]
```

8.4.2.3 action() [3/4]

```
actor_zeta::behavior::action::action (
    action && ) [default]
```

8.4.2.4 ~action()

```
actor_zeta::behavior::action::~~action ( ) [default]
```

8.4.2.5 action() [4/4]

```
actor_zeta::behavior::action::action (
    abstract_action * aa ) [explicit]
```

8.4.3 Member Function Documentation**8.4.3.1 operator()**

```
response * actor_zeta::behavior::action::operator() (
    request * request_ )
```

8.4.3.2 operator=() [1/2]

```
action& actor_zeta::behavior::action::operator= (
    const action & ) [delete]
```

8.4.3.3 operator=() [2/2]

```
action& actor_zeta::behavior::action::operator= (
    action && ) [default]
```

The documentation for this class was generated from the following files:

- [action.hpp](#)
- [action.cpp](#)

8.5 actor_zeta::actor::actor Class Reference

Basic actor implementation.

```
#include <actor.hpp>
```

Public Member Functions

- [actor](#) ()=default
- [actor](#) (const [actor](#) &a)=delete
- [actor](#) ([actor](#) &&a)=default
- [actor](#) & [operator](#)= (const [actor](#) &a)=delete
- [actor](#) & [operator](#)= ([actor](#) &&a)=default
- template<class T >
[actor](#) ([intrusive_ptr](#)< T > ptr)
- template<class T >
[actor](#) (T *ptr)
- template<class T >
[actor](#) & [operator](#)= ([intrusive_ptr](#)< T > ptr)
- template<class T >
[actor](#) & [operator](#)= (T *ptr)
- [actor_address](#) [address](#) () const noexcept
- [~actor](#) ()
- [abstract_actor](#) * [operator->](#) () const noexcept
- [operator bool](#) () const noexcept
- const std::string & [type](#) () const
- bool [operator!](#) () const noexcept

8.5.1 Detailed Description

Basic actor implementation.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 actor() [1/5]

```
actor_zeta::actor::actor::actor ( ) [default]
```

8.5.2.2 actor() [2/5]

```
actor_zeta::actor::actor::actor (
    const actor & a ) [delete]
```

8.5.2.3 actor() [3/5]

```
actor_zeta::actor::actor::actor (
    actor && a ) [default]
```

8.5.2.4 actor() [4/5]

```
template<class T >
actor_zeta::actor::actor::actor (
    intrusive_ptr< T > ptr ) [inline], [explicit]
```

8.5.2.5 actor() [5/5]

```
template<class T >
actor_zeta::actor::actor::actor (
    T * ptr ) [inline], [explicit]
```

8.5.2.6 ~actor()

```
actor_zeta::actor::actor::~~actor ( )
```

8.5.3 Member Function Documentation

8.5.3.1 address()

```
actor_address actor_zeta::actor::actor::address ( ) const [noexcept]
```

8.5.3.2 operator bool()

```
actor_zeta::actor::actor::operator bool ( ) const [inline], [explicit], [noexcept]
```

8.5.3.3 operator"!"()

```
bool actor_zeta::actor::actor::operator! ( ) const [inline], [noexcept]
```

8.5.3.4 operator->()

```
abstract_actor* actor_zeta::actor::actor::operator-> ( ) const [inline], [noexcept]
```

8.5.3.5 operator=() [1/4]

```
actor& actor_zeta::actor::actor::operator= (
    const actor & a ) [delete]
```

8.5.3.6 operator=() [2/4]

```
actor& actor_zeta::actor::actor::operator= (
    actor && a ) [default]
```

8.5.3.7 operator=() [3/4]

```
template<class T >
actor& actor_zeta::actor::actor::operator= (
    intrusive_ptr< T > ptr ) [inline]
```

8.5.3.8 `operator=()` [4 / 4]

```
template<class T >
actor& actor_zeta::actor::actor::operator= (
    T * ptr ) [inline]
```

8.5.3.9 `type()`

```
const std::string & actor_zeta::actor::actor::type ( ) const
```

The documentation for this class was generated from the following files:

- [actor.hpp](#)
- [actor.cpp](#)

8.6 `actor_zeta::actor::actor_address` Class Reference

This represents an actor's address container.

```
#include <actor_address.hpp>
```

Public Member Functions

- `actor_address()`=default
- `actor_address(actor_address &&)=default`
- `actor_address(const actor_address &)=default`
- `actor_address & operator= (actor_address &&)=default`
- `actor_address & operator= (const actor_address &)=default`
- `actor_address(abstract_actor *aa)`
- `~actor_address()`
- `abstract_actor * operator-> () const noexcept`
- `operator bool () const noexcept`
- `bool operator! () const noexcept`

8.6.1 Detailed Description

This represents an actor's address container.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 actor_address() [1/4]

```
actor_zeta::actor::actor_address::actor_address ( ) [default]
```

8.6.2.2 actor_address() [2/4]

```
actor_zeta::actor::actor_address::actor_address (
    actor_address && ) [default]
```

8.6.2.3 actor_address() [3/4]

```
actor_zeta::actor::actor_address::actor_address (
    const actor_address & ) [default]
```

8.6.2.4 actor_address() [4/4]

```
actor_zeta::actor::actor_address::actor_address (
    abstract_actor * aa ) [inline], [explicit]
```

8.6.2.5 ~actor_address()

```
actor_zeta::actor::actor_address::~~actor_address ( )
```

8.6.3 Member Function Documentation**8.6.3.1 operator bool()**

```
actor_zeta::actor::actor_address::operator bool ( ) const [inline], [explicit], [noexcept]
```

8.6.3.2 operator"!()

```
bool actor_zeta::actor::actor_address::operator! ( ) const [inline], [noexcept]
```

8.6.3.3 operator->()

```
abstract_actor* actor_zeta::actor::actor_address::operator-> ( ) const [inline], [noexcept]
```

8.6.3.4 operator=() [1/2]

```
actor_address& actor_zeta::actor::actor_address::operator= (
    actor_address && ) [default]
```

8.6.3.5 operator=() [2/2]

```
actor_address& actor_zeta::actor::actor_address::operator= (
    const actor_address & ) [default]
```

The documentation for this class was generated from the following files:

- [actor_address.hpp](#)
- [actor_address.cpp](#)

8.7 actor_zeta::behavior::behavior Class Reference

Class for lyfecycle determination.

```
#include <behavior.hpp>
```

Public Member Functions

- [behavior](#) ()=default
- [behavior](#) (const [behavior](#) &)=delete
- [behavior](#) & [operator](#)= (const [behavior](#) &)=delete
- [behavior](#) ([behavior](#) &&)=default
- [behavior](#) & [operator](#)= ([behavior](#) &&)=default
- [~behavior](#) ()=default
- void [insert](#) ([abstract_action](#) *aa)
- [response](#) * [run](#) ([request](#) *)

8.7.1 Detailed Description

Class for lyfecycle determination.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 behavior() [1/3]

```
actor_zeta::behavior::behavior::behavior ( ) [default]
```

8.7.2.2 behavior() [2/3]

```
actor_zeta::behavior::behavior::behavior (
    const behavior & ) [delete]
```

8.7.2.3 behavior() [3/3]

```
actor_zeta::behavior::behavior::behavior (
    behavior && ) [default]
```

8.7.2.4 ~behavior()

```
actor_zeta::behavior::behavior::~~behavior ( ) [default]
```

8.7.3 Member Function Documentation

8.7.3.1 insert()

```
void actor_zeta::behavior::behavior::insert (
    abstract_action * aa )
```

8.7.3.2 operator=() [1/2]

```
behavior& actor_zeta::behavior::behavior::operator= (
    const behavior & ) [delete]
```

8.7.3.3 operator=() [2/2]

```
behavior& actor_zeta::behavior::behavior::operator= (
    behavior && ) [default]
```

8.7.3.4 run()

```
response * actor_zeta::behavior::behavior::run (
    request * d )
```

The documentation for this class was generated from the following files:

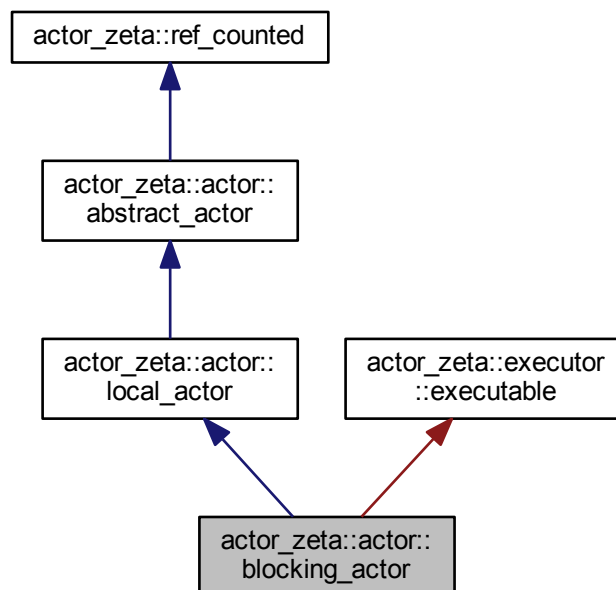
- [behavior.hpp](#)
- [behavior.cpp](#)

8.8 actor_zeta::actor::blocking_actor Class Reference

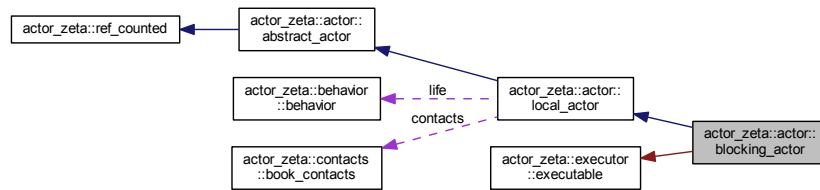
Represents actor type with blocking mode.

```
#include <blocking_actor.hpp>
```

Inheritance diagram for actor_zeta::actor::blocking_actor:



Collaboration diagram for actor_zeta::actor::blocking_actor:



Public Member Functions

- `blocking_actor` (`environment::environment *`, `const std::string &`)
- `virtual void act ()`
- `executor::executable::executable_result run (executor::execution_device *, size_t)` override final
- `void launch (executor::execution_device *, bool)` override final
- `virtual ~blocking_actor ()`

Additional Inherited Members

8.8.1 Detailed Description

Represents actor type with blocking mode.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 `blocking_actor()`

```

actor_zeta::actor::blocking_actor::blocking_actor (
    environment::environment * env,
    const std::string & type )

```

8.8.2.2 `~blocking_actor()`

```

virtual actor_zeta::actor::blocking_actor::~~blocking_actor ( ) [inline], [virtual]

```

8.8.3 Member Function Documentation

8.8.3.1 act()

```
void actor_zeta::actor::blocking_actor::act ( ) [virtual]
```

8.8.3.2 launch()

```
void actor_zeta::actor::blocking_actor::launch (
    executor::execution_device * e,
    bool hide ) [final], [override], [virtual]
```

Implements [actor_zeta::actor::local_actor](#).

8.8.3.3 run()

```
executor::executable::executable_result actor_zeta::actor::blocking_actor::run (
    executor::execution_device * e,
    size_t max_throughput ) [final], [override], [virtual]
```

Implements [actor_zeta::executor::executable](#).

The documentation for this class was generated from the following files:

- [blocking_actor.hpp](#)
- [blocking_actor.cpp](#)

8.9 actor_zeta::messaging::blocking_mail_queue< T > Class Template Reference

A mailbox class for message queue.

```
#include <blocking_mail_queue.hpp>
```

Public Types

- using [pointer](#) = T *
- using [const_pointer](#) = const T *
- using [reference](#) = T &
- using [const_reference](#) = const T &
- using [cache_type](#) = std::list< [pointer](#) >
- using [queue_base_type](#) = std::list< [pointer](#) >
- using [unique_lock](#) = std::unique_lock< std::mutex >
- using [lock_guard](#) = std::lock_guard< std::mutex >

Public Member Functions

- [blocking_mail_queue](#) (const [blocking_mail_queue](#) &)=delete
- [blocking_mail_queue](#) & [operator=](#) (const [blocking_mail_queue](#) &)=delete
- [blocking_mail_queue](#) ([blocking_mail_queue](#) &&)=default
- [blocking_mail_queue](#) & [operator=](#) ([blocking_mail_queue](#) &&)=default
- [blocking_mail_queue](#) ()=default
- [~blocking_mail_queue](#) ()=default
- [enqueue_result](#) put ([pointer](#) m)
- void [sync](#) ()
- [pointer](#) get ()
- [cache_type](#) & [low_priority_cache](#) ()
- [cache_type](#) & [normal_priority_cache](#) ()
- [cache_type](#) & [high_priority_cache](#) ()

8.9.1 Detailed Description

```
template<typename T>
class actor_zeta::messaging::blocking_mail_queue< T >
```

A mailbox class for message queue.

Template Parameters

<i>T</i>	
----------	--

8.9.2 Member Typedef Documentation

8.9.2.1 cache_type

```
template<typename T>
using actor\_zeta::messaging::blocking\_mail\_queue< T >::cache\_type = std::list<pointer>
```

8.9.2.2 const_pointer

```
template<typename T>
using actor\_zeta::messaging::blocking\_mail\_queue< T >::const\_pointer = const T *
```

8.9.2.3 const_reference

```
template<typename T>
using actor_zeta::messaging::blocking_mail_queue< T >::const_reference = const T &
```

8.9.2.4 lock_guard

```
template<typename T>
using actor_zeta::messaging::blocking_mail_queue< T >::lock_guard = std::lock_guard<std::
::mutex>
```

8.9.2.5 pointer

```
template<typename T>
using actor_zeta::messaging::blocking_mail_queue< T >::pointer = T *
```

8.9.2.6 queue_base_type

```
template<typename T>
using actor_zeta::messaging::blocking_mail_queue< T >::queue_base_type = std::list<pointer>
```

8.9.2.7 reference

```
template<typename T>
using actor_zeta::messaging::blocking_mail_queue< T >::reference = T &
```

8.9.2.8 unique_lock

```
template<typename T>
using actor_zeta::messaging::blocking_mail_queue< T >::unique_lock = std::unique_lock<std::
::mutex>
```

8.9.3 Constructor & Destructor Documentation

8.9.3.1 blocking_mail_queue() [1/3]

```
template<typename T>
actor_zeta::messaging::blocking_mail_queue< T >::blocking_mail_queue (
    const blocking_mail_queue< T > & ) [delete]
```

8.9.3.2 blocking_mail_queue() [2/3]

```
template<typename T>
actor_zeta::messaging::blocking_mail_queue< T >::blocking_mail_queue (
    blocking_mail_queue< T > && ) [default]
```

8.9.3.3 blocking_mail_queue() [3/3]

```
template<typename T>
actor_zeta::messaging::blocking_mail_queue< T >::blocking_mail_queue ( ) [default]
```

8.9.3.4 ~blocking_mail_queue()

```
template<typename T>
actor_zeta::messaging::blocking_mail_queue< T >::~~blocking_mail_queue ( ) [default]
```

8.9.4 Member Function Documentation

8.9.4.1 get()

```
template<typename T>
pointer actor_zeta::messaging::blocking_mail_queue< T >::get ( ) [inline]
```

8.9.4.2 high_priority_cache()

```
template<typename T>
cache_type& actor_zeta::messaging::blocking_mail_queue< T >::high_priority_cache ( ) [inline]
```

8.9.4.3 low_priority_cache()

```
template<typename T>
cache_type& actor_zeta::messaging::blocking_mail_queue< T >::low_priority_cache ( ) [inline]
```

8.9.4.4 normal_priority_cache()

```
template<typename T>
cache_type& actor_zeta::messaging::blocking_mail_queue< T >::normal_priority_cache ( ) [inline]
```

8.9.4.5 operator=() [1/2]

```
template<typename T>
blocking_mail_queue& actor_zeta::messaging::blocking_mail_queue< T >::operator= (
    const blocking_mail_queue< T > & ) [delete]
```

8.9.4.6 operator=() [2/2]

```
template<typename T>
blocking_mail_queue& actor_zeta::messaging::blocking_mail_queue< T >::operator= (
    blocking_mail_queue< T > && ) [default]
```

8.9.4.7 put()

```
template<typename T>
enqueue_result actor_zeta::messaging::blocking_mail_queue< T >::put (
    pointer m ) [inline]
```

8.9.4.8 sync()

```
template<typename T>
void actor_zeta::messaging::blocking_mail_queue< T >::sync ( ) [inline]
```

The documentation for this class was generated from the following file:

- [blocking_mail_queue.hpp](#)

8.10 actor_zeta::contacts::book_contacts Class Reference

Navigation map for actor & groups.

```
#include <book_contacts.hpp>
```

Public Member Functions

- [book_contacts](#) ()=default
- [book_contacts](#) ([book_contacts](#) &&)=default
- [book_contacts](#) & [operator=](#) ([book_contacts](#) &&)=default
- void [put](#) (const [actor::actor_address](#) &)
- const [actor::actor_address](#) & [get](#) (const std::string &)
- void [put_in_group](#) (const std::string &, const [actor::actor_address](#) &)
- const [actor::actor_address](#) & [get_group](#) (const std::string &)
- void [all_view](#) ()

8.10.1 Detailed Description

Navigation map for actor & groups.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 [book_contacts\(\)](#) [1/2]

```
actor_zeta::contacts::book_contacts::book_contacts ( ) [default]
```

8.10.2.2 [book_contacts\(\)](#) [2/2]

```
actor_zeta::contacts::book_contacts::book_contacts (
    book\_contacts && ) [default]
```

8.10.3 Member Function Documentation

8.10.3.1 [all_view\(\)](#)

```
void actor_zeta::contacts::book_contacts::all_view ( )
```

8.10.3.2 get()

```
const actor::actor_address & actor_zeta::contacts::book_contacts::get (
    const std::string & name )
```

8.10.3.3 get_group()

```
const actor::actor_address & actor_zeta::contacts::book_contacts::get_group (
    const std::string & name )
```

8.10.3.4 operator=()

```
book_contacts& actor_zeta::contacts::book_contacts::operator= (
    book_contacts && ) [default]
```

8.10.3.5 put()

```
void actor_zeta::contacts::book_contacts::put (
    const actor::actor_address & vc )
```

8.10.3.6 put_in_group()

```
void actor_zeta::contacts::book_contacts::put_in_group (
    const std::string & name,
    const actor::actor_address & address )
```

The documentation for this class was generated from the following files:

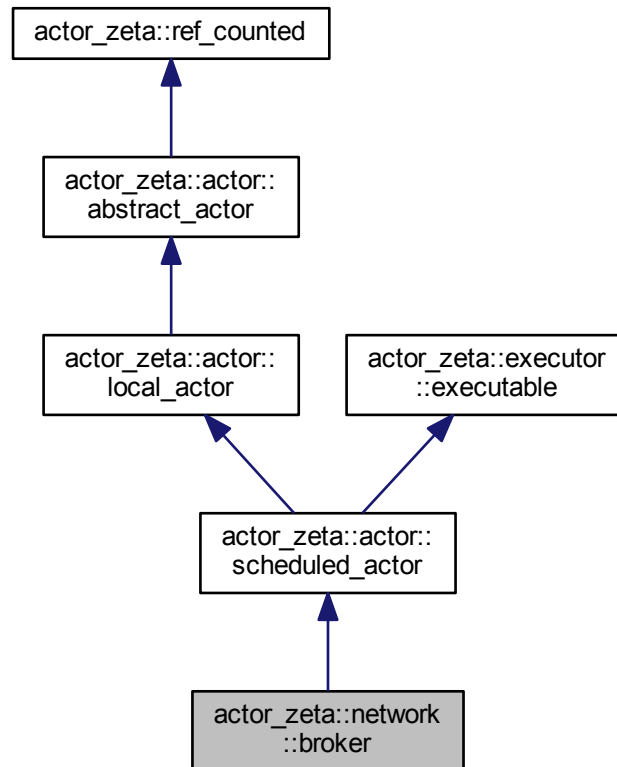
- [book_contacts.hpp](#)
- [book_contacts.cpp](#)

8.11 actor_zeta::network::broker Class Reference

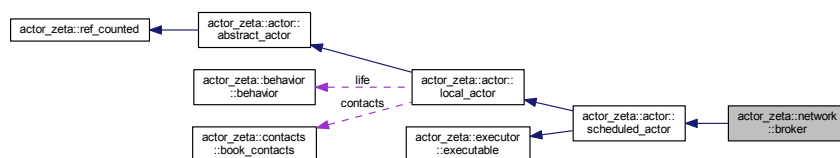
A broker for messaging.

```
#include <broker.hpp>
```

Inheritance diagram for actor_zeta::network::broker:



Collaboration diagram for actor_zeta::network::broker:



Public Member Functions

- **broker** (**environment::environment** *, const std::string &, **shared_muxplexer_ptr**)
- virtual **~broker** ()=default

Protected Member Functions

- void [initialize](#) () override

Protected Attributes

- std::shared_ptr< [multiplexer](#) > [multiplexer_](#)

Additional Inherited Members

8.11.1 Detailed Description

A broker for messaging.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 [broker\(\)](#)

```
actor_zeta::network::broker::broker (
    environment::environment * env,
    const std::string & name,
    shared_multiplexer_ptr m_ptr )
```

8.11.2.2 [~broker\(\)](#)

```
virtual actor_zeta::network::broker::~~broker ( ) [virtual], [default]
```

8.11.3 Member Function Documentation

8.11.3.1 [initialize\(\)](#)

```
void actor_zeta::network::broker::initialize ( ) [override], [protected], [virtual]
```

Reimplemented from [actor_zeta::actor::local_actor](#).

8.11.4 Member Data Documentation

8.11.4.1 multiplexer_

```
std::shared_ptr<multiplexer> actor_zeta::network::broker::multiplexer_ [protected]
```

The documentation for this class was generated from the following files:

- [broker.hpp](#)
- [broker.cpp](#)

8.12 actor_zeta::network::connection_identifying Class Reference

Implementation of connection functionality.

```
#include <connection_identifying.hpp>
```

Public Member Functions

- [connection_identifying](#) ()=delete
- [connection_identifying](#) (const [connection_identifying](#) &)=default
- [connection_identifying](#) & operator= (const [connection_identifying](#) &)=default
- [connection_identifying](#) ([connection_identifying](#) &&)=default
- [connection_identifying](#) & operator= ([connection_identifying](#) &&)=default
- [connection_identifying](#) (const [type_connect](#) &tc, const std::string &ip_, const int &port_)
- std::string [to_string](#) () const
- const std::string & [ip](#) () const
- const int & [port](#) () const
- bool [operator==](#) (const [connection_identifying](#) &ci) const

8.12.1 Detailed Description

Implementation of connection functionality.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 [connection_identifying\(\)](#) [1/4]

```
actor_zeta::network::connection_identifying::connection_identifying ( ) [delete]
```

8.12.2.2 connection_identifying() [2/4]

```
actor_zeta::network::connection_identifying::connection_identifying (
    const connection_identifying & ) [default]
```

8.12.2.3 connection_identifying() [3/4]

```
actor_zeta::network::connection_identifying::connection_identifying (
    connection_identifying && ) [default]
```

8.12.2.4 connection_identifying() [4/4]

```
actor_zeta::network::connection_identifying::connection_identifying (
    const type_connect & tc,
    const std::string & ip_,
    const int & port_ )
```

8.12.3 Member Function Documentation**8.12.3.1 ip()**

```
const std::string & actor_zeta::network::connection_identifying::ip ( ) const
```

8.12.3.2 operator=() [1/2]

```
connection_identifying& actor_zeta::network::connection_identifying::operator= (
    const connection_identifying & ) [default]
```

8.12.3.3 operator=() [2/2]

```
connection_identifying& actor_zeta::network::connection_identifying::operator= (
    connection_identifying && ) [default]
```

8.12.3.4 operator==()

```
bool actor_zeta::network::connection_identifying::operator== (
    const connection_identifying & ci ) const
```

8.12.3.5 port()

```
const int & actor_zeta::network::connection_identifying::port ( ) const
```

8.12.3.6 to_string()

```
std::string actor_zeta::network::connection_identifying::to_string ( ) const
```

The documentation for this class was generated from the following files:

- [connection_identifying.hpp](#)
- [connection_identifying.cpp](#)

8.13 actor_zeta::environment::cooperation Class Reference

A logic combiner for groups.

```
#include <cooperation.hpp>
```

Public Member Functions

- [cooperation](#) ()=default
- [cooperation](#) (const [cooperation](#) &a)=delete
- [cooperation](#) ([cooperation](#) &&)=default
- [cooperation](#) & [operator=](#) (const [cooperation](#) &a)=delete
- [cooperation](#) & [operator=](#) ([cooperation](#) &&)=default
- [~cooperation](#) ()=default
- void [add](#) ([group](#) &&)
- [actor_zeta::actor::actor_address](#) [get](#) (const std::string &) const
- void [send](#) ([messaging::message](#) *)
- void [send_current](#) (const std::string &, [messaging::message](#) *)
- void [send_all](#) ([messaging::message](#) *)
- void [add_shared](#) ([actor::abstract_actor](#) *)

8.13.1 Detailed Description

A logic combiner for groups.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 cooperation() [1/3]

```
actor_zeta::environment::cooperation::cooperation ( ) [default]
```

8.13.2.2 cooperation() [2/3]

```
actor_zeta::environment::cooperation::cooperation (
    const cooperation & a ) [delete]
```

8.13.2.3 cooperation() [3/3]

```
actor_zeta::environment::cooperation::cooperation (
    cooperation && ) [default]
```

8.13.2.4 ~cooperation()

```
actor_zeta::environment::cooperation::~~cooperation ( ) [default]
```

8.13.3 Member Function Documentation

8.13.3.1 add()

```
void actor_zeta::environment::cooperation::add (
    group && g )
```

8.13.3.2 add_shared()

```
void actor_zeta::environment::cooperation::add_shared (
    actor::abstract_actor * actor )
```

8.13.3.3 get()

```
actor_zeta::actor::actor_address actor_zeta::environment::cooperation::get (
    const std::string & name ) const
```

8.13.3.4 operator=() [1/2]

```
cooperation& actor_zeta::environment::cooperation::operator= (
    const cooperation & a ) [delete]
```

8.13.3.5 operator=() [2/2]

```
cooperation& actor_zeta::environment::cooperation::operator= (
    cooperation && ) [default]
```

8.13.3.6 send()

```
void actor_zeta::environment::cooperation::send (
    messaging::message * msg )
```

8.13.3.7 send_all()

```
void actor_zeta::environment::cooperation::send_all (
    messaging::message * msg )
```

8.13.3.8 send_current()

```
void actor_zeta::environment::cooperation::send_current (
    const std::string & name,
    messaging::message * message )
```

The documentation for this class was generated from the following files:

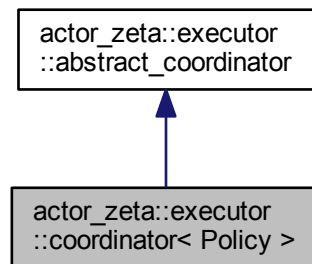
- [cooperation.hpp](#)
- [cooperation.cpp](#)

8.14 actor_zeta::executor::coordinator< Policy > Class Template Reference

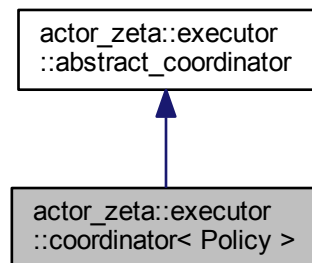
Provides ruler for environment.

```
#include <coordinator.hpp>
```

Inheritance diagram for actor_zeta::executor::coordinator< Policy >:



Collaboration diagram for actor_zeta::executor::coordinator< Policy >:



Public Types

- using [job_ptr](#) = [executable](#) *
- using [policy_data](#) = typename Policy::coordinator_data
- using [worker_type](#) = [worker](#)< Policy >

Public Member Functions

- [policy_data](#) & [data](#) ()
- [worker_type](#) * [worker_by_id](#) (size_t id)
- void [submit](#) ([job_ptr](#) job) override
- [coordinator](#) (size_t num_worker_threads, size_t max_throughput_param)
- void [start](#) ()

8.14.1 Detailed Description

```
template<class Policy>
class actor_zeta::executor::coordinator< Policy >
```

Provides ruler for environment.

Template Parameters

<i>Policy</i>	
---------------	--

8.14.2 Member Typedef Documentation

8.14.2.1 job_ptr

```
template<class Policy >
using actor_zeta::executor::coordinator< Policy >::job_ptr = executable *
```

8.14.2.2 policy_data

```
template<class Policy >
using actor_zeta::executor::coordinator< Policy >::policy_data = typename Policy::coordinator↵
_data
```

8.14.2.3 worker_type

```
template<class Policy >
using actor_zeta::executor::coordinator< Policy >::worker_type = worker<Policy>
```

8.14.3 Constructor & Destructor Documentation

8.14.3.1 coordinator()

```
template<class Policy >
actor_zeta::executor::coordinator< Policy >::coordinator (
    size_t num_worker_threads,
    size_t max_throughput_param ) [inline]
```

8.14.4 Member Function Documentation

8.14.4.1 data()

```
template<class Policy >
policy_data& actor_zeta::executor::coordinator< Policy >::data ( ) [inline]
```

8.14.4.2 start()

```
template<class Policy >
void actor_zeta::executor::coordinator< Policy >::start ( ) [inline], [virtual]
```

Implements [actor_zeta::executor::abstract_coordinator](#).

8.14.4.3 submit()

```
template<class Policy >
void actor_zeta::executor::coordinator< Policy >::submit (
    job_ptr job ) [inline], [override]
```

8.14.4.4 worker_by_id()

```
template<class Policy >
worker_type* actor_zeta::executor::coordinator< Policy >::worker_by_id (
    size_t id ) [inline]
```

The documentation for this class was generated from the following file:

- [coordinator.hpp](#)

8.15 actor_zeta::executor::work_sharing::coordinator_data Struct Reference

This structure is used as coordinator data keeper.

```
#include <work_sharing.hpp>
```

Public Member Functions

- [coordinator_data](#) ([executor::abstract_coordinator](#) *)

Public Attributes

- [queue_type](#) `queue`
- `std::mutex` `lock`
- `std::condition_variable` `cv`

8.15.1 Detailed Description

This structure is used as coordinator data keeper.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 coordinator_data()

```
actor_zeta::executor::work_sharing::coordinator_data::coordinator_data (  
    executor::abstract_coordinator * ) [inline], [explicit]
```

8.15.3 Member Data Documentation

8.15.3.1 cv

```
std::condition_variable actor_zeta::executor::work_sharing::coordinator_data::cv
```

8.15.3.2 lock

```
std::mutex actor_zeta::executor::work_sharing::coordinator_data::lock
```

8.15.3.3 queue

```
queue_type actor_zeta::executor::work_sharing::coordinator_data::queue
```

The documentation for this struct was generated from the following file:

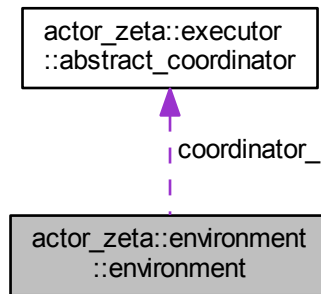
- [work_sharing.hpp](#)

8.16 actor_zeta::environment::environment Class Reference

An actors workplace platform.

```
#include <environment.hpp>
```

Collaboration diagram for actor_zeta::environment::environment:



Public Member Functions

- [environment](#) ()=delete
- [environment](#) (const [environment](#) &)=delete
- [environment](#) & [operator=](#) (const [environment](#) &)=delete
- [environment](#) ([environment](#) &&)=default
- [environment](#) & [operator=](#) ([environment](#) &&)=default
- virtual [~environment](#) ()=default
- virtual int [start](#) ()
- [environment](#) ([executor::abstract_coordinator](#) *)
- [executor::abstract_coordinator](#) & [manager_execution_device](#) ()
- [cooperation](#) & [manager_group](#) ()

Protected Attributes

- [executor::abstract_coordinator](#) * [coordinator_](#)

8.16.1 Detailed Description

An actors workplace platform.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 environment() [1/4]

```
actor_zeta::environment::environment::environment ( ) [delete]
```

8.16.2.2 environment() [2/4]

```
actor_zeta::environment::environment::environment (
    const environment & ) [delete]
```

8.16.2.3 environment() [3/4]

```
actor_zeta::environment::environment::environment (
    environment && ) [default]
```

8.16.2.4 ~environment()

```
virtual actor_zeta::environment::environment::~~environment ( ) [virtual], [default]
```

8.16.2.5 environment() [4/4]

```
actor_zeta::environment::environment::environment (
    executor::abstract_coordinator * ac )
```

8.16.3 Member Function Documentation**8.16.3.1 manager_execution_device()**

```
executor::abstract_coordinator & actor_zeta::environment::environment::manager_execution_↵
device ( )
```

8.16.3.2 manager_group()

```
cooperation & actor_zeta::environment::environment::manager_group ( )
```

8.16.3.3 operator=() [1/2]

```
environment& actor_zeta::environment::environment::operator= (
    const environment & ) [delete]
```

8.16.3.4 operator=() [2/2]

```
environment& actor_zeta::environment::environment::operator= (
    environment && ) [default]
```

8.16.3.5 start()

```
int actor_zeta::environment::environment::start ( ) [virtual]
```

8.16.4 Member Data Documentation

8.16.4.1 coordinator_

```
executor::abstract_coordinator* actor_zeta::environment::environment::coordinator_ [protected]
```

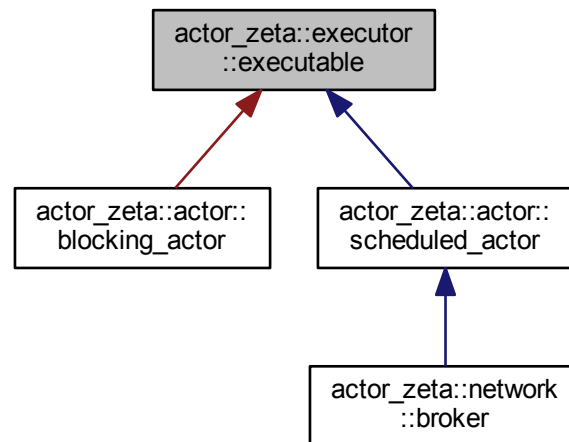
The documentation for this class was generated from the following files:

- [environment.hpp](#)
- [environment.cpp](#)

8.17 actor_zeta::executor::executable Struct Reference

```
#include <executable.hpp>
```

Inheritance diagram for actor_zeta::executor::executable:



Public Types

- enum [executable_result](#) { [executable_result::resume](#), [executable_result::awaiting](#), [executable_result::done](#), [executable_result::shutdown](#) }

Public Member Functions

- virtual [~executable](#) ()=default
- virtual void [attach_to_scheduler](#) ()=0
- virtual void [detach_from_scheduler](#) ()=0
- virtual [executable_result](#) run ([executor::execution_device](#) *, size_t max_throughput)=0

8.17.1 Member Enumeration Documentation

8.17.1.1 executable_result

enum [actor_zeta::executor::executable::executable_result](#) [strong]

Enumerator

resume	is coded as std::int of value 0
awaiting	is coded as std::int of value 1
done	is coded as std::int of value 2
shutdown	is coded as std::int of value 3

8.17.2 Constructor & Destructor Documentation

8.17.2.1 ~executable()

```
virtual actor_zeta::executor::executable::~executable ( ) [virtual], [default]
```

8.17.3 Member Function Documentation

8.17.3.1 attach_to_scheduler()

```
virtual void actor_zeta::executor::executable::attach_to_scheduler ( ) [pure virtual]
```

Implemented in [actor_zeta::actor::scheduled_actor](#).

8.17.3.2 detach_from_scheduler()

```
virtual void actor_zeta::executor::executable::detach_from_scheduler ( ) [pure virtual]
```

Implemented in [actor_zeta::actor::scheduled_actor](#).

8.17.3.3 run()

```
virtual executable_result actor_zeta::executor::executable::run (
    executor::execution_device * ,
    size_t max_throughput ) [pure virtual]
```

Implemented in [actor_zeta::actor::scheduled_actor](#), and [actor_zeta::actor::blocking_actor](#).

The documentation for this struct was generated from the following file:

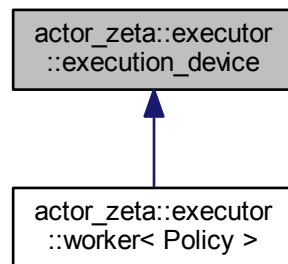
- [executable.hpp](#)

8.18 actor_zeta::executor::execution_device Struct Reference

[execution_device](#)

```
#include <execution_device.hpp>
```

Inheritance diagram for actor_zeta::executor::execution_device:



Public Member Functions

- [execution_device](#) ()=default
- [execution_device](#) ([execution_device](#) &&)=delete
- [execution_device](#) (const [execution_device](#) &)=delete
- virtual [~execution_device](#) ()=default
- virtual void [put_execute_latest](#) ([executable](#) *)=0

8.18.1 Detailed Description

[execution_device](#)

8.18.2 Constructor & Destructor Documentation

8.18.2.1 execution_device() [1/3]

```
actor_zeta::executor::execution_device::execution_device ( ) [default]
```

8.18.2.2 `execution_device()` [2/3]

```
actor_zeta::executor::execution_device::execution_device (
    execution_device && ) [delete]
```

8.18.2.3 `execution_device()` [3/3]

```
actor_zeta::executor::execution_device::execution_device (
    const execution_device & ) [delete]
```

8.18.2.4 `~execution_device()`

```
virtual actor_zeta::executor::execution_device::~~execution_device ( ) [virtual], [default]
```

8.18.3 Member Function Documentation**8.18.3.1** `put_execute_latest()`

```
virtual void actor_zeta::executor::execution_device::put_execute_latest (
    executable * ) [pure virtual]
```

The documentation for this struct was generated from the following file:

- [execution_device.hpp](#)

8.19 `actor_zeta::environment::group` Class Reference

A group combinator for actors.

```
#include <group.hpp>
```


Public Member Functions

- [group](#) ()=default
- [group](#) (const [group](#) &)=delete
- [group](#) ([group](#) &&)=default
- [group](#) & [operator=](#) (const [group](#) &)=delete
- [group](#) & [operator=](#) ([group](#) &&)=default
- [group](#) (const std::string &name, [actor::abstract_actor](#) *actor)
- [~group](#) ()=default
- void [add](#) ([actor::actor](#) &&)
- void [add](#) ([actor::abstract_actor](#) *)
- void [add](#) (const std::string &, [actor::abstract_actor](#) *)
- void [add_shared](#) ([actor::abstract_actor](#) *)
- [actor::actor_address](#) [entry_point](#) () const
- void [send](#) ([messaging::message](#) *)
- void [send_current](#) (const std::string &, [messaging::message](#) *)
- void [send_all](#) ([messaging::message](#) *)
- const std::string & [name](#) () const

8.19.1 Detailed Description

A group combinator for actors.

8.19.2 Constructor & Destructor Documentation

8.19.2.1 [group\(\)](#) [1/4]

```
actor_zeta::environment::group::group ( ) [default]
```

8.19.2.2 [group\(\)](#) [2/4]

```
actor_zeta::environment::group::group (
    const group & ) [delete]
```

8.19.2.3 [group\(\)](#) [3/4]

```
actor_zeta::environment::group::group (
    group && ) [default]
```

8.19.2.4 group() [4/4]

```
actor_zeta::environment::group::group (
    const std::string & name,
    actor::abstract_actor * actor )
```

8.19.2.5 ~group()

```
actor_zeta::environment::group::~~group ( ) [default]
```

8.19.3 Member Function Documentation

8.19.3.1 add() [1/3]

```
void actor_zeta::environment::group::add (
    actor::actor && a )
```

8.19.3.2 add() [2/3]

```
void actor_zeta::environment::group::add (
    actor::abstract_actor * actor )
```

8.19.3.3 add() [3/3]

```
void actor_zeta::environment::group::add (
    const std::string & root_name,
    actor::abstract_actor * actor )
```

8.19.3.4 add_shared()

```
void actor_zeta::environment::group::add_shared (
    actor::abstract_actor * actor )
```

8.19.3.5 entry_point()

```
actor::actor_address actor_zeta::environment::group::entry_point ( ) const
```

8.19.3.6 name()

```
const std::string & actor_zeta::environment::group::name ( ) const
```

8.19.3.7 operator=() [1/2]

```
group& actor_zeta::environment::group::operator= (
    const group & ) [delete]
```

8.19.3.8 operator=() [2/2]

```
group& actor_zeta::environment::group::operator= (
    group && ) [default]
```

8.19.3.9 send()

```
void actor_zeta::environment::group::send (
    messaging::message * msg )
```

8.19.3.10 send_all()

```
void actor_zeta::environment::group::send_all (
    messaging::message * msg )
```

8.19.3.11 send_current()

```
void actor_zeta::environment::group::send_current (
    const std::string & name,
    messaging::message * msg )
```

The documentation for this class was generated from the following files:

- [group.hpp](#)
- [group.cpp](#)

8.20 actor_zeta::contacts::group_contacts Class Reference

Address container for groups.

```
#include <group_contacts.hpp>
```

Public Member Functions

- [group_contacts](#) ()
- [group_contacts](#) ([group_contacts](#) &&)=default
- [group_contacts](#) & [operator=](#) ([group_contacts](#) &&)=default
- void [put](#) (const [actor_zeta::actor::actor_address](#) &vc)
- const [actor_zeta::actor::actor_address](#) & [get](#) ()
- const std::vector< [actor_zeta::actor::actor_address](#) > & [get_all](#) () const

8.20.1 Detailed Description

Address container for groups.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 group_contacts() [1/2]

```
actor_zeta::contacts::group_contacts::group_contacts ( )
```

8.20.2.2 group_contacts() [2/2]

```
actor_zeta::contacts::group_contacts::group_contacts (
    group\_contacts && ) [default]
```

8.20.3 Member Function Documentation

8.20.3.1 get()

```
const actor\_zeta::actor::actor\_address & actor_zeta::contacts::group_contacts::get ( )
```

8.20.3.2 get_all()

```
const std::vector< actor_zeta::actor::actor_address > & actor_zeta::contacts::group_contacts←
::get_all ( ) const
```

8.20.3.3 operator=()

```
group_contacts& actor_zeta::contacts::group_contacts::operator= (
    group_contacts && ) [default]
```

8.20.3.4 put()

```
void actor_zeta::contacts::group_contacts::put (
    const actor_zeta::actor::actor_address & vc )
```

The documentation for this class was generated from the following files:

- [group_contacts.hpp](#)
- [group_contacts.cpp](#)

8.21 actor_zeta::intrusive_ptr< T > Class Template Reference

This class represents smart pointers.

```
#include <intrusive_ptr.hpp>
```

Public Types

- using [pointer](#) = T *
- using [reference](#) = T &

Public Member Functions

- constexpr [intrusive_ptr](#) ()
- [intrusive_ptr](#) ([pointer](#) raw_ptr, bool add_ref=true)
- [intrusive_ptr](#) ([intrusive_ptr](#) &&other)
- [intrusive_ptr](#) (const [intrusive_ptr](#) &other)
- template<class Y >
 [intrusive_ptr](#) ([intrusive_ptr](#)< Y > other)
- [intrusive_ptr](#) & [operator=](#) ([pointer](#) ptr)
- [intrusive_ptr](#) & [operator=](#) ([intrusive_ptr](#) other)
- [~intrusive_ptr](#) ()
- [pointer detach](#) () noexcept
- [pointer release](#) () noexcept
- void [reset](#) ([pointer](#) new_value=nullptr, bool add_ref=true)
- [pointer get](#) () const
- [pointer operator->](#) () const
- [reference operator*](#) () const
- bool [operator!](#) () const
- [operator bool](#) () const
- void [swap](#) ([intrusive_ptr](#) &other) noexcept
- template<class C >
 [intrusive_ptr](#)< C > [downcast](#) () const
- template<class C >
 [intrusive_ptr](#)< C > [upcast](#) () const

8.21.1 Detailed Description

```
template<class T>
class actor_zeta::intrusive_ptr< T >
```

This class represents smart pointers.

Template Parameters

<i>T</i>	
----------	--

8.21.2 Member Typedef Documentation

8.21.2.1 pointer

```
template<class T>
using actor_zeta::intrusive_ptr< T >::pointer = T *
```

8.21.2.2 reference

```
template<class T>
using actor_zeta::intrusive_ptr< T >::reference = T &
```

8.21.3 Constructor & Destructor Documentation

8.21.3.1 intrusive_ptr() [1/5]

```
template<class T>
constexpr actor_zeta::intrusive_ptr< T >::intrusive_ptr ( ) [inline]
```

8.21.3.2 intrusive_ptr() [2/5]

```
template<class T>
actor_zeta::intrusive_ptr< T >::intrusive_ptr (
    pointer raw_ptr,
    bool add_ref = true ) [inline]
```

8.21.3.3 intrusive_ptr() [3/5]

```
template<class T>
actor_zeta::intrusive_ptr< T >::intrusive_ptr (
    intrusive_ptr< T > && other ) [inline]
```

8.21.3.4 intrusive_ptr() [4/5]

```
template<class T>
actor_zeta::intrusive_ptr< T >::intrusive_ptr (
    const intrusive_ptr< T > & other ) [inline]
```

8.21.3.5 intrusive_ptr() [5/5]

```
template<class T>
template<class Y >
actor_zeta::intrusive_ptr< T >::intrusive_ptr (
    intrusive_ptr< Y > other ) [inline]
```

8.21.3.6 ~intrusive_ptr()

```
template<class T>
actor_zeta::intrusive_ptr< T >::~~intrusive_ptr ( ) [inline]
```

8.21.4 Member Function Documentation**8.21.4.1 detach()**

```
template<class T>
pointer actor_zeta::intrusive_ptr< T >::detach ( ) [inline], [noexcept]
```

8.21.4.2 downcast()

```
template<class T>
template<class C >
intrusive_ptr<C> actor_zeta::intrusive_ptr< T >::downcast ( ) const [inline]
```

8.21.4.3 get()

```
template<class T>
pointer actor_zeta::intrusive_ptr< T >::get ( ) const [inline]
```

8.21.4.4 operator bool()

```
template<class T>
actor_zeta::intrusive_ptr< T >::operator bool ( ) const [inline], [explicit]
```

8.21.4.5 operator"!()

```
template<class T>
bool actor_zeta::intrusive_ptr< T >::operator! ( ) const [inline]
```

8.21.4.6 operator*()

```
template<class T>
reference actor_zeta::intrusive_ptr< T >::operator* ( ) const [inline]
```

8.21.4.7 operator->()

```
template<class T>
pointer actor_zeta::intrusive_ptr< T >::operator-> ( ) const [inline]
```

8.21.4.8 operator=() [1/2]

```
template<class T>
intrusive_ptr& actor_zeta::intrusive_ptr< T >::operator= (
    pointer ptr ) [inline]
```

8.21.4.9 operator=() [2/2]

```
template<class T>
intrusive_ptr& actor_zeta::intrusive_ptr< T >::operator= (
    intrusive_ptr< T > other ) [inline]
```


8.21.4.10 release()

```
template<class T>
pointer actor_zeta::intrusive_ptr< T >::release ( ) [inline], [noexcept]
```

8.21.4.11 reset()

```
template<class T>
void actor_zeta::intrusive_ptr< T >::reset (
    pointer new_value = nullptr,
    bool add_ref = true ) [inline]
```

8.21.4.12 swap()

```
template<class T>
void actor_zeta::intrusive_ptr< T >::swap (
    intrusive_ptr< T > & other ) [inline], [noexcept]
```

8.21.4.13 upcast()

```
template<class T>
template<class C >
intrusive_ptr<C> actor_zeta::intrusive_ptr< T >::upcast ( ) const [inline]
```

The documentation for this class was generated from the following file:

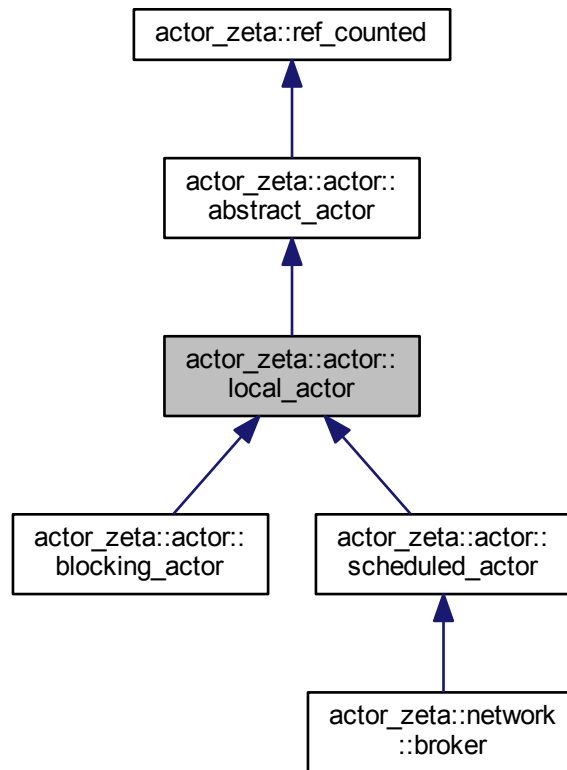
- [intrusive_ptr.hpp](#)

8.22 actor_zeta::actor::local_actor Class Reference

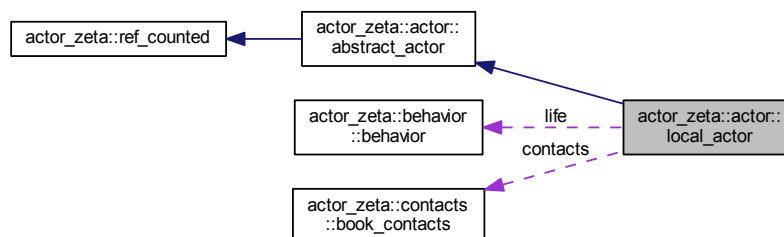
Class for location dependant type actor.

```
#include <local_actor.hpp>
```

Inheritance diagram for actor_zeta::actor::local_actor:



Collaboration diagram for actor_zeta::actor::local_actor:



Public Types

- using `mailbox_type` = `messaging::blocking_mail_queue`< `messaging::message` >

Public Member Functions

- virtual void `launch` (`executor::execution_device` *, bool)=0
- virtual `~local_actor` ()

Protected Member Functions

- void `device` (`executor::execution_device` *e)
- `executor::execution_device` * `device` () const
- void `attach` (`behavior::abstract_action` *)
- `local_actor` (`environment::environment` *, const std::string &)
- virtual void `initialize` ()
- `messaging::message` * `next_message` ()
- bool `has_next_message` ()
- `mailbox_type` & `mailbox` ()
- bool `push_to_cache` (`messaging::message` *)
- `messaging::message` * `pop_to_cache` ()

Protected Attributes

- `contacts::book_contacts` `contacts`
- `behavior::behavior` `life`

8.22.1 Detailed Description

Class for location dependant type actor.

8.22.2 Member Typedef Documentation

8.22.2.1 mailbox_type

```
using actor_zeta::actor::local_actor::mailbox_type = messaging::blocking_mail_queue<messaging::message>
```

8.22.3 Constructor & Destructor Documentation

8.22.3.1 ~local_actor()

```
actor_zeta::actor::local_actor::~~local_actor ( ) [virtual]
```

8.22.3.2 local_actor()

```
actor_zeta::actor::local_actor::local_actor (
    environment::environment * env,
    const std::string & type ) [protected]
```

8.22.4 Member Function Documentation

8.22.4.1 attach()

```
void actor_zeta::actor::local_actor::attach (
    behavior::abstract_action * ptr_aa ) [protected]
```

8.22.4.2 device() [1/2]

```
void actor_zeta::actor::local_actor::device (
    executor::execution_device * e ) [inline], [protected]
```

8.22.4.3 device() [2/2]

```
executor::execution_device* actor_zeta::actor::local_actor::device ( ) const [inline], [protected]
```

8.22.4.4 has_next_message()

```
bool actor_zeta::actor::local_actor::has_next_message ( ) [protected]
```

8.22.4.5 initialize()

```
void actor_zeta::actor::local_actor::initialize ( ) [protected], [virtual]
```

Reimplemented in [actor_zeta::network::broker](#).

8.22.4.6 launch()

```
virtual void actor_zeta::actor::local_actor::launch (
    executor::execution_device * ,
    bool ) [pure virtual]
```

Implemented in [actor_zeta::actor::scheduled_actor](#), and [actor_zeta::actor::blocking_actor](#).

8.22.4.7 mailbox()

```
mailbox_type& actor_zeta::actor::local_actor::mailbox ( ) [inline], [protected]
```

8.22.4.8 next_message()

```
messaging::message * actor_zeta::actor::local_actor::next_message ( ) [protected]
```

8.22.4.9 pop_to_cache()

```
messaging::message * actor_zeta::actor::local_actor::pop_to_cache ( ) [protected]
```

8.22.4.10 push_to_cache()

```
bool actor_zeta::actor::local_actor::push_to_cache (
    messaging::message * msg_ptr ) [protected]
```

8.22.5 Member Data Documentation

8.22.5.1 contacts

```
contacts::book_contacts actor_zeta::actor::local_actor::contacts [protected]
```

8.22.5.2 life

```
behavior::behavior actor_zeta::actor::local_actor::life [protected]
```

The documentation for this class was generated from the following files:

- [local_actor.hpp](#)
- [local_actor.cpp](#)

8.23 actor_zeta::messaging::message Class Reference

Class to represent messages.

```
#include <message.hpp>
```

Public Member Functions

- [message](#) ()=delete
- [message](#) (const [message](#) &)=delete
- [message](#) & [operator=](#) (const [message](#) &)=delete
- [message](#) ([message](#) &&)=default
- [message](#) & [operator=](#) ([message](#) &&)=default
- [~message](#) ()=default
- template<std::size_t N, typename T >
[message](#) (const char(&aStr)[N], const T &t)
- template<std::size_t N, typename T >
[message](#) (const char(&aStr)[N], const T &t, [actor::actor_address](#) aa)
- template<std::size_t N, typename T >
[message](#) (const char(&aStr)[N], const T &t, [message_priority](#) p)
- template<std::size_t N, typename T >
[message](#) (const char(&aStr)[N], const T &t, [message_priority](#) p, [actor::actor_address](#) aa)
- template<typename T >
[message](#) (const char *str, std::size_t len, const T &t)
- template<typename T >
[message](#) (const char *str, std::size_t len, const T &t, [actor::actor_address](#) aa)
- template<typename T >
[message](#) (const char *str, std::size_t len, const T &t, [message_priority](#) p)
- template<typename T >
[message](#) (const char *str, std::size_t len, const T &t, [message_priority](#) p, [actor::actor_address](#) aa)
- [message_priority priority](#) () const
- auto [type](#) () const noexcept -> const [behavior::type_action](#) &
- [actor::actor_address return_address](#) () const
- bool [is_callback](#) () const
- template<typename T >
auto [get](#) () -> T
- auto [clone](#) () const -> [message](#) *

8.23.1 Detailed Description

Class to represent messages.

8.23.2 Constructor & Destructor Documentation

8.23.2.1 message() [1/11]

```
actor_zeta::messaging::message::message ( ) [delete]
```

8.23.2.2 message() [2/11]

```
actor_zeta::messaging::message::message (
    const message & ) [delete]
```

8.23.2.3 message() [3/11]

```
actor_zeta::messaging::message::message (
    message && ) [default]
```

8.23.2.4 ~message()

```
actor_zeta::messaging::message::~~message ( ) [default]
```

8.23.2.5 message() [4/11]

```
template<std::size_t N, typename T >
actor_zeta::messaging::message::message (
    const char(&) aStr[N],
    const T & t ) [inline]
```

8.23.2.6 message() [5/11]

```
template<std::size_t N, typename T >
actor_zeta::messaging::message::message (
    const char(&) aStr[N],
    const T & t,
    actor::actor_address aa ) [inline]
```

8.23.2.7 message() [6/11]

```
template<std::size_t N, typename T >
actor_zeta::messaging::message::message (
    const char(&) aStr[N],
    const T & t,
    message_priority p ) [inline]
```

8.23.2.8 message() [7/11]

```
template<std::size_t N, typename T >
actor_zeta::messaging::message::message (
    const char(&) aStr[N],
    const T & t,
    message_priority p,
    actor::actor_address aa ) [inline]
```

8.23.2.9 message() [8/11]

```
template<typename T >
actor_zeta::messaging::message::message (
    const char * str,
    std::size_t len,
    const T & t ) [inline]
```

8.23.2.10 message() [9/11]

```
template<typename T >
actor_zeta::messaging::message::message (
    const char * str,
    std::size_t len,
    const T & t,
    actor::actor_address aa ) [inline]
```

8.23.2.11 message() [10/11]

```
template<typename T >
actor_zeta::messaging::message::message (
    const char * str,
    std::size_t len,
    const T & t,
    message_priority p ) [inline]
```


8.23.2.12 message() [11/11]

```
template<typename T >
actor_zeta::messaging::message::message (
    const char * str,
    std::size_t len,
    const T & t,
    message_priority p,
    actor::actor_address aa ) [inline]
```

8.23.3 Member Function Documentation

8.23.3.1 clone()

```
auto actor_zeta::messaging::message::clone ( ) const -> message * [inline]
```

8.23.3.2 get()

```
template<typename T >
auto actor_zeta::messaging::message::get ( ) -> T [inline]
```

8.23.3.3 is_callback()

```
bool actor_zeta::messaging::message::is_callback ( ) const
```

8.23.3.4 operator=() [1/2]

```
message& actor_zeta::messaging::message::operator= (
    const message & ) [delete]
```

8.23.3.5 operator=() [2/2]

```
message& actor_zeta::messaging::message::operator= (
    message && ) [default]
```

8.23.3.6 priority()

```
message_priority actor_zeta::messaging::message::priority ( ) const
```

8.23.3.7 return_address()

```
actor::actor_address actor_zeta::messaging::message::return_address ( ) const
```

8.23.3.8 type()

```
auto actor_zeta::messaging::message::type ( ) const -> const behavior::type_action & [noexcept]
```

The documentation for this class was generated from the following files:

- [message.hpp](#)
- [message.cpp](#)

8.24 actor_zeta::messaging::message_body Class Reference

A message value container.

```
#include <message_body.hpp>
```

Public Member Functions

- constexpr [message_body](#) () noexcept
- template<typename ValueType >
 [message_body](#) (const ValueType &value)
- [message_body](#) (const [message_body](#) &other)
- [message_body](#) ([message_body](#) &&other) noexcept
- template<typename ValueType >
 [message_body](#) (ValueType &&value)
- ~[message_body](#) () noexcept
- [message_body](#) & swap ([message_body](#) &rhs) noexcept
- [message_body](#) & operator= (const [message_body](#) &rhs)
- [message_body](#) & operator= ([message_body](#) &&rhs) noexcept
- template<class ValueType >
 [message_body](#) & operator= (ValueType &&rhs)
- bool [empty](#) () const noexcept
- void [clear](#) () noexcept
- template<typename T >
 T [get](#) ()

8.24.1 Detailed Description

A message value container.

8.24.2 Constructor & Destructor Documentation

8.24.2.1 message_body() [1/5]

```
constexpr actor_zeta::messaging::message_body::message_body ( ) [inline], [noexcept]
```

8.24.2.2 message_body() [2/5]

```
template<typename ValueType >  
actor_zeta::messaging::message_body::message_body (   
    const ValueType & value ) [inline]
```

8.24.2.3 message_body() [3/5]

```
actor_zeta::messaging::message_body::message_body (   
    const message\_body & other ) [inline]
```

8.24.2.4 message_body() [4/5]

```
actor_zeta::messaging::message_body::message_body (   
    message\_body && other ) [inline], [noexcept]
```

8.24.2.5 message_body() [5/5]

```
template<typename ValueType >  
actor_zeta::messaging::message_body::message_body (   
    ValueType && value ) [inline]
```

8.24.2.6 ~message_body()

```
actor_zeta::messaging::message_body::~~message_body ( ) [inline], [noexcept]
```

8.24.3 Member Function Documentation

8.24.3.1 clear()

```
void actor_zeta::messaging::message_body::clear ( ) [inline], [noexcept]
```

8.24.3.2 empty()

```
bool actor_zeta::messaging::message_body::empty ( ) const [inline], [noexcept]
```

8.24.3.3 get()

```
template<typename T >  
T actor_zeta::messaging::message_body::get ( ) [inline]
```

8.24.3.4 operator=() [1/3]

```
message_body& actor_zeta::messaging::message_body::operator= (   
    const message_body & rhs ) [inline]
```

8.24.3.5 operator=() [2/3]

```
message_body& actor_zeta::messaging::message_body::operator= (   
    message_body && rhs ) [inline], [noexcept]
```

8.24.3.6 operator=() [3/3]

```
template<class ValueType >  
message_body& actor_zeta::messaging::message_body::operator= (   
    ValueType && rhs ) [inline]
```

8.24.3.7 swap()

```
message_body& actor_zeta::messaging::message_body::swap (
    message_body & rhs ) [inline], [noexcept]
```

The documentation for this class was generated from the following file:

- [message_body.hpp](#)

8.25 actor_zeta::messaging::message_header Class Reference

A message description.

```
#include <message_header.hpp>
```

Public Member Functions

- [message_header](#) ()=delete
TODO: sender.
- [message_header](#) (const [message_header](#) &)=default
- [message_header](#) & operator= (const [message_header](#) &)=default
- [message_header](#) ([message_header](#) &&)=default
- [message_header](#) & operator= ([message_header](#) &&)=default
- ~[message_header](#) ()=default
- template<std::size_t N>
[message_header](#) (const char(&aStr)[N])
- template<std::size_t N>
[message_header](#) (const char(&aStr)[N], [actor::actor_address](#) aa)
- template<std::size_t N>
[message_header](#) (const char(&aStr)[N], [message_priority](#) p)
- template<std::size_t N>
[message_header](#) (const char(&aStr)[N], [message_priority](#) p, [actor::actor_address](#) aa)
- [message_header](#) (const char *str, std::size_t len)
- [message_header](#) (const char *str, std::size_t len, [actor::actor_address](#) aa)
- [message_header](#) (const char *str, std::size_t len, [message_priority](#) p)
- [message_header](#) (const char *str, std::size_t len, [message_priority](#) p, [actor::actor_address](#) aa)
- [message_priority](#) priorities () const
- auto [type](#) () const noexcept -> const [behavior::type_action](#) &
- [actor::actor_address](#) return_address () const
- bool [is_callback](#) () const

8.25.1 Detailed Description

A message description.

8.25.2 Constructor & Destructor Documentation

8.25.2.1 message_header() [1/11]

```
actor_zeta::messaging::message_header::message_header ( ) [delete]
```

TODO: sender.

8.25.2.2 message_header() [2/11]

```
actor_zeta::messaging::message_header::message_header (
    const message_header & ) [default]
```

8.25.2.3 message_header() [3/11]

```
actor_zeta::messaging::message_header::message_header (
    message_header && ) [default]
```

8.25.2.4 ~message_header()

```
actor_zeta::messaging::message_header::~~message_header ( ) [default]
```

8.25.2.5 message_header() [4/11]

```
template<std::size_t N>
actor_zeta::messaging::message_header::message_header (
    const char(&) aStr[N] ) [inline], [explicit]
```

8.25.2.6 message_header() [5/11]

```
template<std::size_t N>
actor_zeta::messaging::message_header::message_header (
    const char(&) aStr[N],
    actor::actor_address aa ) [inline]
```

8.25.2.7 message_header() [6/11]

```
template<std::size_t N>
actor_zeta::messaging::message_header::message_header (
    const char(&) aStr[N],
    message_priority p ) [inline]
```

8.25.2.8 message_header() [7/11]

```
template<std::size_t N>
actor_zeta::messaging::message_header::message_header (
    const char(&) aStr[N],
    message_priority p,
    actor::actor_address aa ) [inline]
```

8.25.2.9 message_header() [8/11]

```
actor_zeta::messaging::message_header::message_header (
    const char * str,
    std::size_t len )
```

8.25.2.10 message_header() [9/11]

```
actor_zeta::messaging::message_header::message_header (
    const char * str,
    std::size_t len,
    actor::actor_address aa )
```

8.25.2.11 message_header() [10/11]

```
actor_zeta::messaging::message_header::message_header (
    const char * str,
    std::size_t len,
    message_priority p )
```

8.25.2.12 message_header() [11/11]

```
actor_zeta::messaging::message_header::message_header (
    const char * str,
    std::size_t len,
    message_priority p,
    actor::actor_address aa )
```

8.25.3 Member Function Documentation

8.25.3.1 `is_callback()`

```
bool actor_zeta::messaging::message_header::is_callback ( ) const
```

8.25.3.2 `operator=()` [1/2]

```
message_header& actor_zeta::messaging::message_header::operator= (
    const message_header & ) [default]
```

8.25.3.3 `operator=()` [2/2]

```
message_header& actor_zeta::messaging::message_header::operator= (
    message_header && ) [default]
```

8.25.3.4 `priorities()`

```
message_priority actor_zeta::messaging::message_header::priorities ( ) const
```

8.25.3.5 `return_address()`

```
actor::actor_address actor_zeta::messaging::message_header::return_address ( ) const
```

8.25.3.6 `type()`

```
auto actor_zeta::messaging::message_header::type ( ) const -> const behavior::type_action &
[noexcept]
```

The documentation for this class was generated from the following files:

- [message_header.hpp](#)
- [message_header.cpp](#)

8.26 actor_zeta::network::multiplexer Struct Reference

Multiplexing utility class.

```
#include <multiplexer.hpp>
```

Public Member Functions

- virtual int [run](#) ()=0
- virtual [~multiplexer](#) ()=default
- virtual void [new_tcp_listener](#) (const std::string &host, uint16_t port)=0
- virtual void [new_tcp_connection](#) (const std::string &host, uint16_t port)=0
- virtual void [close](#) (const [connection_identifying](#) &)=0
- virtual void [write](#) (const [connection_identifying](#) &, const std::string &)=0
- virtual void [registration_broker_address](#) (const actor::actor_address &)=0

8.26.1 Detailed Description

Multiplexing utility class.

8.26.2 Constructor & Destructor Documentation

8.26.2.1 ~multiplexer()

```
virtual actor_zeta::network::multiplexer::~~multiplexer ( ) [virtual], [default]
```

8.26.3 Member Function Documentation

8.26.3.1 close()

```
virtual void actor_zeta::network::multiplexer::close (
    const connection\_identifying & ) [pure virtual]
```

8.26.3.2 new_tcp_connection()

```
virtual void actor_zeta::network::multiplexer::new_tcp_connection (
    const std::string & host,
    uint16_t port ) [pure virtual]
```

8.26.3.3 new_tcp_listener()

```
virtual void actor_zeta::network::multiplexer::new_tcp_listener (
    const std::string & host,
    uint16_t port ) [pure virtual]
```

8.26.3.4 registration_broker_address()

```
virtual void actor_zeta::network::multiplexer::registration_broker_address (
    const actor::actor_address & ) [pure virtual]
```

8.26.3.5 run()

```
virtual int actor_zeta::network::multiplexer::run ( ) [pure virtual]
```

8.26.3.6 write()

```
virtual void actor_zeta::network::multiplexer::write (
    const connection_identifying & ,
    const std::string & ) [pure virtual]
```

The documentation for this struct was generated from the following file:

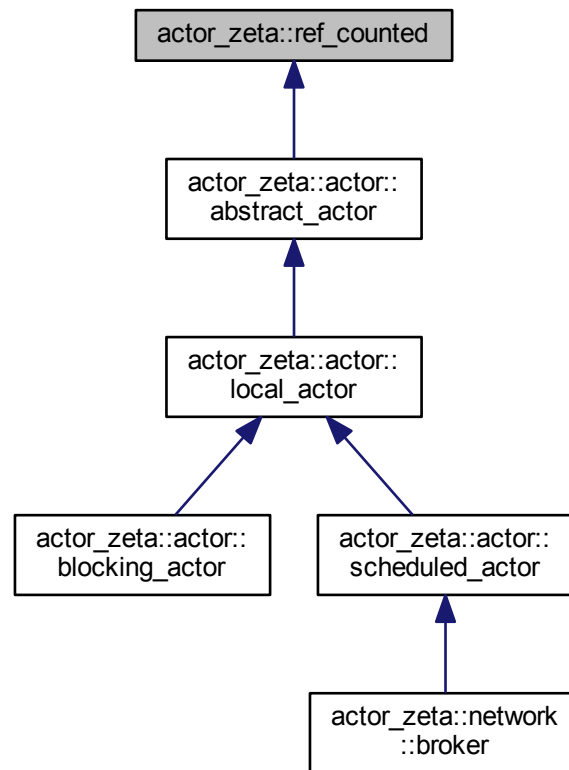
- [multiplexer.hpp](#)

8.27 actor_zeta::ref_counted Class Reference

This class represents reference counter.

```
#include <ref_counted.hpp>
```

Inheritance diagram for actor_zeta::ref_counted:



Public Member Functions

- [~ref_counted](#) ()
- [ref_counted](#) ()
- [ref_counted](#) (const [ref_counted](#) &)
- [ref_counted](#) & [operator=](#) (const [ref_counted](#) &)
- void [ref](#) () noexcept
- void [deref](#) () noexcept
- bool [unique](#) () const noexcept
- size_t [get_reference_count](#) () const noexcept

Protected Attributes

- std::atomic< size_t > [rc_](#)

8.27.1 Detailed Description

This class represents reference counter.

8.27.2 Constructor & Destructor Documentation

8.27.2.1 `~ref_counted()`

```
actor_zeta::ref_counted::~~ref_counted ( )
```

8.27.2.2 `ref_counted()` [1/2]

```
actor_zeta::ref_counted::ref_counted ( )
```

8.27.2.3 `ref_counted()` [2/2]

```
actor_zeta::ref_counted::ref_counted (
    const ref\_counted & )
```

8.27.3 Member Function Documentation

8.27.3.1 `deref()`

```
void actor_zeta::ref_counted::deref ( ) [noexcept]
```

8.27.3.2 `get_reference_count()`

```
size_t actor_zeta::ref_counted::get_reference_count ( ) const [inline], [noexcept]
```

8.27.3.3 `operator=()`

```
ref\_counted & actor_zeta::ref_counted::operator= (
    const ref\_counted & )
```

8.27.3.4 ref()

```
void actor_zeta::ref_counted::ref ( ) [inline], [noexcept]
```

8.27.3.5 unique()

```
bool actor_zeta::ref_counted::unique ( ) const [inline], [noexcept]
```

8.27.4 Member Data Documentation

8.27.4.1 rc_

```
std::atomic<size_t> actor_zeta::ref_counted::rc_ [protected]
```

The documentation for this class was generated from the following files:

- [ref_counted.hpp](#)
- [ref_counted.cpp](#)

8.28 actor_zeta::behavior::request Class Reference

This is a request container for messaging.

```
#include <request.hpp>
```

Public Member Functions

- [request](#) ()=default
- [request](#) (const [request](#) &)=default
- [request](#) & [operator=](#) (const [request](#) &)=default
- [request](#) ([request](#) &&)=default
- [request](#) & [operator=](#) ([request](#) &&)=default
- [~request](#) ()=default
- [request](#) ([contacts::book_contacts](#) &contacts_, [messaging::message](#) *msg)
- [messaging::message](#) * [message](#) ()
- [contacts::book_contacts](#) & [contacts](#) ()

8.28.1 Detailed Description

This is a request container for messaging.

8.28.2 Constructor & Destructor Documentation

8.28.2.1 request() [1/4]

```
actor_zeta::behavior::request::request ( ) [default]
```

8.28.2.2 request() [2/4]

```
actor_zeta::behavior::request::request (
    const request & ) [default]
```

8.28.2.3 request() [3/4]

```
actor_zeta::behavior::request::request (
    request && ) [default]
```

8.28.2.4 ~request()

```
actor_zeta::behavior::request::~~request ( ) [default]
```

8.28.2.5 request() [4/4]

```
actor_zeta::behavior::request::request (
    contacts::book_contacts & contacts_,
    messaging::message * msg ) [inline]
```

8.28.3 Member Function Documentation

8.28.3.1 contacts()

```
contacts::book_contacts& actor_zeta::behavior::request::contacts ( ) [inline]
```

8.28.3.2 message()

```
messaging::message* actor_zeta::behavior::request::message ( ) [inline]
```

8.28.3.3 operator=() [1/2]

```
request& actor_zeta::behavior::request::operator= (
    const request & ) [default]
```

8.28.3.4 operator=() [2/2]

```
request& actor_zeta::behavior::request::operator= (
    request && ) [default]
```

The documentation for this class was generated from the following file:

- [request.hpp](#)

8.29 actor_zeta::behavior::response Class Reference

This is a response container for messaging.

```
#include <response.hpp>
```

Public Member Functions

- [response](#) ()=default
- [response](#) (const [response](#) &)=default
- [response](#) & [operator=](#) (const [response](#) &)=default
- [response](#) ([response](#) &&)=default
- [response](#) & [operator=](#) ([response](#) &&)=default
- [~response](#) ()=default
- [response](#) ([actor::actor_address](#) receiver_, [messaging::message](#) *msg)
- [messaging::message](#) * [message](#) ()
- [actor::actor_address](#) [receiver](#) () const

8.29.1 Detailed Description

This is a response container for messaging.

8.29.2 Constructor & Destructor Documentation

8.29.2.1 response() [1/4]

```
actor_zeta::behavior::response::response ( ) [default]
```

8.29.2.2 response() [2/4]

```
actor_zeta::behavior::response::response (
    const response & ) [default]
```

8.29.2.3 response() [3/4]

```
actor_zeta::behavior::response::response (
    response && ) [default]
```

8.29.2.4 ~response()

```
actor_zeta::behavior::response::~~response ( ) [default]
```

8.29.2.5 response() [4/4]

```
actor_zeta::behavior::response::response (
    actor::actor_address receiver_,
    messaging::message * msg ) [inline]
```

8.29.3 Member Function Documentation

8.29.3.1 message()

```
messaging::message* actor_zeta::behavior::response::message ( ) [inline]
```


8.29.3.2 operator=() [1/2]

```
response& actor_zeta::behavior::response::operator= (
    const response & ) [default]
```

8.29.3.3 operator=() [2/2]

```
response& actor_zeta::behavior::response::operator= (
    response && ) [default]
```

8.29.3.4 receiver()

```
actor::actor_address actor_zeta::behavior::response::receiver ( ) const [inline]
```

The documentation for this class was generated from the following file:

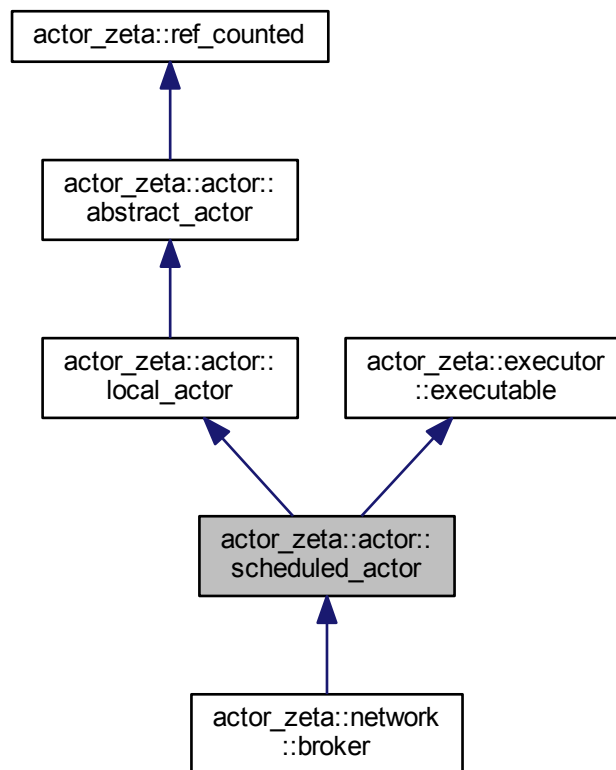
- [response.hpp](#)

8.30 actor_zeta::actor::scheduled_actor Class Reference

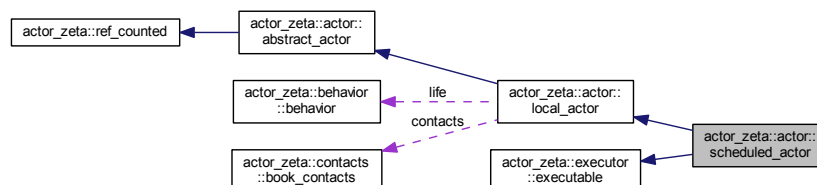
Represents scheduling type of actor.

```
#include <scheduled_actor.hpp>
```

Inheritance diagram for actor_zeta::actor::scheduled_actor:



Collaboration diagram for actor_zeta::actor::scheduled_actor:



Public Member Functions

- `bool send (messaging::message *)` override final
- `bool send (messaging::message *, executor::execution_device *)` override final
- `void launch (executor::execution_device *, bool)` override final
- `executor::executable::executable_result run (executor::execution_device *, size_t max_throughput)` override final

Protected Member Functions

- [scheduled_actor](#) ([environment::environment](#) *, const std::string &)
- void [attach_to_scheduler](#) () override final
- void [detach_from_scheduler](#) () override final

Additional Inherited Members

8.30.1 Detailed Description

Represents scheduling type of actor.

8.30.2 Constructor & Destructor Documentation

8.30.2.1 scheduled_actor()

```
actor_zeta::actor::scheduled_actor::scheduled_actor (  
    environment::environment * env,  
    const std::string & name ) [protected]
```

8.30.3 Member Function Documentation

8.30.3.1 attach_to_scheduler()

```
void actor_zeta::actor::scheduled_actor::attach_to_scheduler ( ) [final], [override], [protected],  
[virtual]
```

Implements [actor_zeta::executor::executable](#).

8.30.3.2 detach_from_scheduler()

```
void actor_zeta::actor::scheduled_actor::detach_from_scheduler ( ) [final], [override], [protected],  
[virtual]
```

Implements [actor_zeta::executor::executable](#).

8.30.3.3 launch()

```
void actor_zeta::actor::scheduled_actor::launch (
    executor::execution_device * e,
    bool hide ) [final], [override], [virtual]
```

Implements [actor_zeta::actor::local_actor](#).

8.30.3.4 run()

```
executor::executable::executable_result actor_zeta::actor::scheduled_actor::run (
    executor::execution_device * e,
    size_t max_throughput ) [final], [override], [virtual]
```

Implements [actor_zeta::executor::executable](#).

8.30.3.5 send() [1/2]

```
bool actor_zeta::actor::scheduled_actor::send (
    messaging::message * msg ) [final], [override], [virtual]
```

Implements [actor_zeta::actor::abstract_actor](#).

8.30.3.6 send() [2/2]

```
bool actor_zeta::actor::scheduled_actor::send (
    messaging::message * mep,
    executor::execution_device * e ) [final], [override], [virtual]
```

Implements [actor_zeta::actor::abstract_actor](#).

The documentation for this class was generated from the following files:

- [scheduled_actor.hpp](#)
- [scheduled_actor.cpp](#)

8.31 actor_zeta::environment::shared_group Class Reference

Group realisation with common resource.

```
#include <shared_group.hpp>
```

Public Member Functions

- [shared_group](#) ()=default
- [shared_group](#) (const [shared_group](#) &)=delete
- [shared_group](#) ([shared_group](#) &&)=default
- [shared_group](#) & operator= (const [shared_group](#) &)=delete
- [shared_group](#) & operator= ([shared_group](#) &&)=default
- [~shared_group](#) ()=default
- void [add](#) ([actor::abstract_actor](#) *)
- void [send_current](#) (const std::string &, [messaging::message](#) *)
- void [send_all](#) ([messaging::message](#) *)

8.31.1 Detailed Description

Group realisation with common resource.

8.31.2 Constructor & Destructor Documentation

8.31.2.1 [shared_group\(\)](#) [1/3]

```
actor_zeta::environment::shared_group::shared_group ( ) [default]
```

8.31.2.2 [shared_group\(\)](#) [2/3]

```
actor_zeta::environment::shared_group::shared_group (
    const shared\_group & ) [delete]
```

8.31.2.3 [shared_group\(\)](#) [3/3]

```
actor_zeta::environment::shared_group::shared_group (
    shared\_group && ) [default]
```

8.31.2.4 [~shared_group\(\)](#)

```
actor_zeta::environment::shared_group::~~shared_group ( ) [default]
```

8.31.3 Member Function Documentation

8.31.3.1 add()

```
void actor_zeta::environment::shared_group::add (
    actor::abstract_actor * a )
```

8.31.3.2 operator=() [1/2]

```
shared_group& actor_zeta::environment::shared_group::operator= (
    const shared_group & ) [delete]
```

8.31.3.3 operator=() [2/2]

```
shared_group& actor_zeta::environment::shared_group::operator= (
    shared_group && ) [default]
```

8.31.3.4 send_all()

```
void actor_zeta::environment::shared_group::send_all (
    messaging::message * msg )
```

8.31.3.5 send_current()

```
void actor_zeta::environment::shared_group::send_current (
    const std::string & name,
    messaging::message * msg )
```

The documentation for this class was generated from the following files:

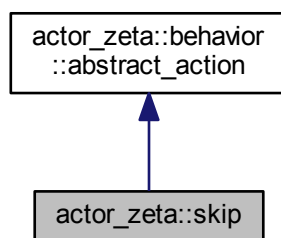
- [shared_group.hpp](#)
- [shared_group.cpp](#)

8.32 actor_zeta::skip Class Reference

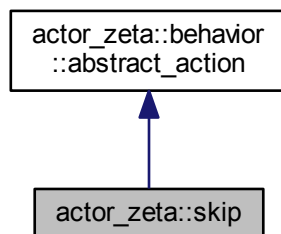
A class used for skipping action.

```
#include <skip.hpp>
```

Inheritance diagram for actor_zeta::skip:



Collaboration diagram for actor_zeta::skip:



Public Member Functions

- [skip](#) ()
- [behavior::response](#) * [operator\(\)](#) ([behavior::request](#) *) override final

8.32.1 Detailed Description

A class used for skipping action.

8.32.2 Constructor & Destructor Documentation

8.32.2.1 skip()

```
actor_zeta::skip::skip ( )
```

8.32.3 Member Function Documentation

8.32.3.1 operator>()

```
behavior::response * actor_zeta::skip::operator() (
    behavior::request * r ) [final], [override], [virtual]
```

Implements [actor_zeta::behavior::abstract_action](#).

The documentation for this class was generated from the following files:

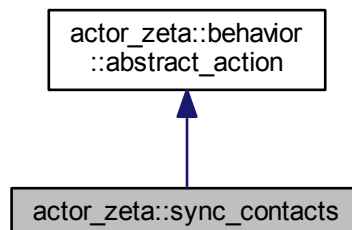
- [skip.hpp](#)
- [skip.cpp](#)

8.33 actor_zeta::sync_contacts Class Reference

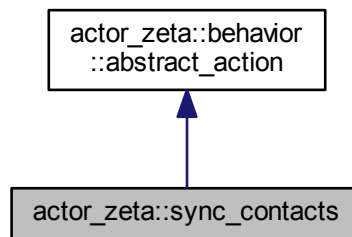
A class used for contact operations synchronization.

```
#include <sync_contacts.hpp>
```

Inheritance diagram for actor_zeta::sync_contacts:



Collaboration diagram for actor_zeta::sync_contacts:



Public Member Functions

- [sync_contacts\(\)](#)
- [behavior::response * operator\(\)](#) ([behavior::request *](#)) override final

8.33.1 Detailed Description

A class used for contact operations synchronization.

8.33.2 Constructor & Destructor Documentation

8.33.2.1 sync_contacts()

```
actor_zeta::sync_contacts::sync_contacts ( )
```

8.33.3 Member Function Documentation

8.33.3.1 operator>()

```
behavior::response * actor_zeta::sync_contacts::operator() (
    behavior::request * request ) [final], [override], [virtual]
```

Implements [actor_zeta::behavior::abstract_action](#).

The documentation for this class was generated from the following files:

- [sync_contacts.hpp](#)
- [sync_contacts.cpp](#)

8.34 actor_zeta::behavior::type_action Class Reference

Type of commands used for lyfecycle.

```
#include <type_action.hpp>
```

Public Member Functions

- [type_action](#) ()=delete
- [type_action](#) (const [type_action](#) &)=default
- [type_action](#) & [operator=](#) (const [type_action](#) &t)=default
- [type_action](#) ([type_action](#) &&)=default
- [type_action](#) & [operator=](#) ([type_action](#) &&)=default
- [~type_action](#) ()=default
- template<std::size_t N>
 [type_action](#) (const char(&aStr)[N])
- [type_action](#) (const char *, std::size_t)
- [type_action](#) (const std::string &)
- bool [operator==](#) (const [type_action](#) &t) const noexcept
- auto [hash](#) () const noexcept -> std::size_t
- auto [to_string](#) () const -> const std::string &

8.34.1 Detailed Description

Type of commands used for lyfecycle.

8.34.2 Constructor & Destructor Documentation

8.34.2.1 [type_action\(\)](#) [1/6]

```
actor_zeta::behavior::type_action::type_action ( ) [delete]
```

8.34.2.2 [type_action\(\)](#) [2/6]

```
actor_zeta::behavior::type_action::type_action (
    const type\_action & ) [default]
```

8.34.2.3 [type_action\(\)](#) [3/6]

```
actor_zeta::behavior::type_action::type_action (
    type\_action && ) [default]
```

8.34.2.4 ~type_action()

```
actor_zeta::behavior::type_action::~~type_action ( ) [default]
```

8.34.2.5 type_action() [4/6]

```
template<std::size_t N>
actor_zeta::behavior::type_action::type_action (
    const char(&) aStr[N] ) [inline]
```

8.34.2.6 type_action() [5/6]

```
actor_zeta::behavior::type_action::type_action (
    const char * str,
    std::size_t len )
```

8.34.2.7 type_action() [6/6]

```
actor_zeta::behavior::type_action::type_action (
    const std::string & token )
```

8.34.3 Member Function Documentation

8.34.3.1 hash()

```
auto actor_zeta::behavior::type_action::hash ( ) const -> std::size_t [noexcept]
```

8.34.3.2 operator=() [1/2]

```
type_action& actor_zeta::behavior::type_action::operator= (
    const type_action & t ) [default]
```

8.34.3.3 operator=() [2/2]

```
type_action& actor_zeta::behavior::type_action::operator= (
    type_action && ) [default]
```

8.34.3.4 operator==()

```
bool actor_zeta::behavior::type_action::operator== (
    const type_action & t ) const [noexcept]
```

8.34.3.5 to_string()

```
auto actor_zeta::behavior::type_action::to_string ( ) const -> const std::string&
```

The documentation for this class was generated from the following files:

- [type_action.hpp](#)
- [type_action.cpp](#)

8.35 actor_zeta::executor::work_sharing Class Reference

Stands for work sharing approach.

```
#include <work_sharing.hpp>
```

Classes

- struct [coordinator_data](#)
This structure is used as coordinator data keeper.
- struct [worker_data](#)
This structure is used as worker data keeper.

Public Types

- using [queue_type](#) = std::list< [executable](#) * >

Public Member Functions

- [~work_sharing](#) ()
- template<class Coordinator >
void [enqueue](#) (Coordinator *self, [executable](#) *job)
- template<class Coordinator >
void [central_enqueue](#) (Coordinator *self, [executable](#) *job)
- template<class Worker >
void [external_enqueue](#) (Worker *self, [executable](#) *job)
- template<class Worker >
void [internal_enqueue](#) (Worker *self, [executable](#) *job)
- template<class Worker >
void [resume_job_later](#) (Worker *self, [executable](#) *job)
- template<class Worker >
[executable](#) * [dequeue](#) (Worker *self)
- template<class Coordinator , class UnaryFunction >
void [foreach_central_resumable](#) (Coordinator *self, UnaryFunction f)

8.35.1 Detailed Description

Stands for work sharing approach.

8.35.2 Member Typedef Documentation

8.35.2.1 queue_type

```
using actor_zeta::executor::work_sharing::queue_type = std::list<executable *>
```

8.35.3 Constructor & Destructor Documentation

8.35.3.1 ~work_sharing()

```
actor_zeta::executor::work_sharing::~~work_sharing ( ) [inline]
```

8.35.4 Member Function Documentation

8.35.4.1 central_enqueue()

```
template<class Coordinator >
void actor_zeta::executor::work_sharing::central_enqueue (
    Coordinator * self,
    executable * job ) [inline]
```

8.35.4.2 dequeue()

```
template<class Worker >
executable* actor_zeta::executor::work_sharing::dequeue (
    Worker * self ) [inline]
```

8.35.4.3 enqueue()

```
template<class Coordinator >
void actor_zeta::executor::work_sharing::enqueue (
    Coordinator * self,
    executable * job ) [inline]
```

8.35.4.4 external_enqueue()

```
template<class Worker >
void actor_zeta::executor::work_sharing::external_enqueue (
    Worker * self,
    executable * job ) [inline]
```

8.35.4.5 foreach_central_resumable()

```
template<class Coordinator , class UnaryFunction >
void actor_zeta::executor::work_sharing::foreach_central_resumable (
    Coordinator * self,
    UnaryFunction f ) [inline]
```

8.35.4.6 internal_enqueue()

```
template<class Worker >
void actor_zeta::executor::work_sharing::internal_enqueue (
    Worker * self,
    executable * job ) [inline]
```

8.35.4.7 resume_job_later()

```
template<class Worker >
void actor_zeta::executor::work_sharing::resume_job_later (
    Worker * self,
    executable * job ) [inline]
```

The documentation for this class was generated from the following file:

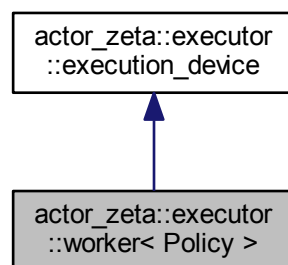
- [work_sharing.hpp](#)

8.36 actor_zeta::executor::worker< Policy > Class Template Reference

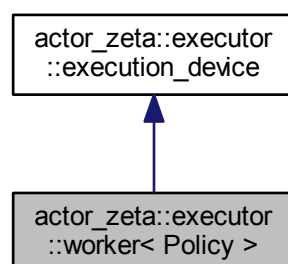
A simple worker module class.

```
#include <worker.hpp>
```

Inheritance diagram for actor_zeta::executor::worker< Policy >:



Collaboration diagram for actor_zeta::executor::worker< Policy >:



Public Types

- using `job_ptr` = `executable` *
- using `coordinator_ptr` = `coordinator`< `Policy` > *
- using `policy_data` = `typename Policy::worker_data`

Public Member Functions

- `worker` (`size_t worker_id`, `coordinator_ptr worker_parent`, `size_t throughput`)
- void `start` ()
- `worker` (`const worker &`)=delete
- `worker & operator=` (`const worker &`)=delete
- void `external_enqueue` (`job_ptr job`)
- void `put_execute_latest` (`job_ptr job`) override
- `coordinator_ptr parent` ()
- `size_t id` () const
- `policy_data & data` ()
- `size_t max_throughput` ()

8.36.1 Detailed Description

```
template<class Policy>
class actor_zeta::executor::worker< Policy >
```

A simple worker module class.

Template Parameters

<i>Policy</i>	
---------------	--

8.36.2 Member Typedef Documentation

8.36.2.1 `coordinator_ptr`

```
template<class Policy >
using actor_zeta::executor::worker< Policy >::coordinator_ptr = coordinator<Policy> *
```

8.36.2.2 `job_ptr`

```
template<class Policy >
using actor_zeta::executor::worker< Policy >::job_ptr = executable *
```


8.36.2.3 policy_data

```
template<class Policy >
using actor_zeta::executor::worker< Policy >::policy_data = typename Policy::worker_data
```

8.36.3 Constructor & Destructor Documentation

8.36.3.1 worker() [1/2]

```
template<class Policy >
actor_zeta::executor::worker< Policy >::worker (
    size_t worker_id,
    coordinator_ptr worker_parent,
    size_t throughput ) [inline]
```

8.36.3.2 worker() [2/2]

```
template<class Policy >
actor_zeta::executor::worker< Policy >::worker (
    const worker< Policy > & ) [delete]
```

8.36.4 Member Function Documentation

8.36.4.1 data()

```
template<class Policy >
policy_data& actor_zeta::executor::worker< Policy >::data ( ) [inline]
```

8.36.4.2 external_enqueue()

```
template<class Policy >
void actor_zeta::executor::worker< Policy >::external_enqueue (
    job_ptr job ) [inline]
```

8.36.4.3 id()

```
template<class Policy >
size_t actor_zeta::executor::worker< Policy >::id ( ) const [inline]
```

8.36.4.4 max_throughput()

```
template<class Policy >
size_t actor_zeta::executor::worker< Policy >::max_throughput ( ) [inline]
```

8.36.4.5 operator=()

```
template<class Policy >
worker& actor_zeta::executor::worker< Policy >::operator= (
    const worker< Policy > & ) [delete]
```

8.36.4.6 parent()

```
template<class Policy >
coordinator_ptr actor_zeta::executor::worker< Policy >::parent ( ) [inline]
```

8.36.4.7 put_execute_latest()

```
template<class Policy >
void actor_zeta::executor::worker< Policy >::put_execute_latest (
    job_ptr job ) [inline], [override]
```

8.36.4.8 start()

```
template<class Policy >
void actor_zeta::executor::worker< Policy >::start ( ) [inline]
```

The documentation for this class was generated from the following file:

- [worker.hpp](#)

8.37 actor_zeta::executor::work_sharing::worker_data Struct Reference

This structure is used as worker data keeper.

```
#include <work_sharing.hpp>
```

Public Member Functions

- [worker_data](#) ([executor::abstract_coordinator](#) *)

8.37.1 Detailed Description

This structure is used as worker data keeper.

8.37.2 Constructor & Destructor Documentation

8.37.2.1 worker_data()

```
actor_zeta::executor::work_sharing::worker_data::worker_data (  
    executor::abstract\_coordinator * ) [inline], [explicit]
```

The documentation for this struct was generated from the following file:

- [work_sharing.hpp](#)

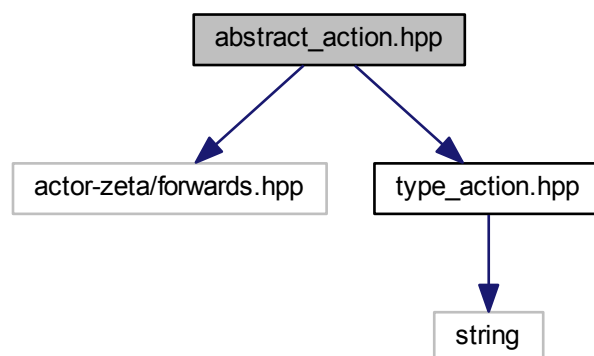
Chapter 9

File Documentation

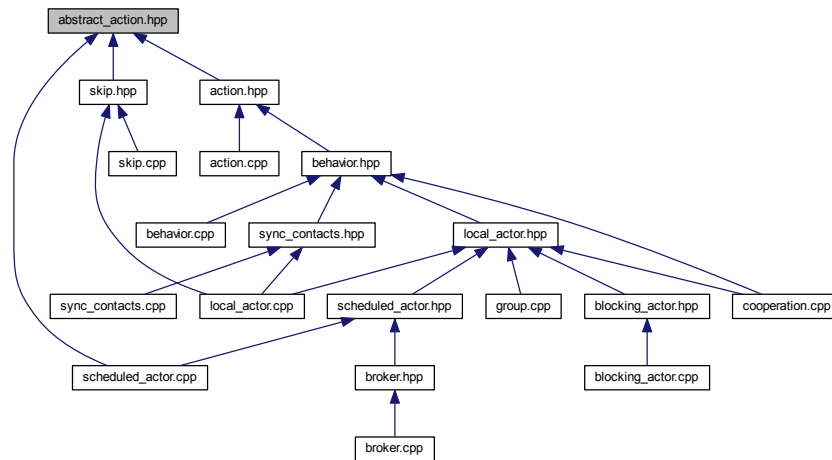
9.1 abstract_action.hpp File Reference

```
#include "actor-zeta/forwards.hpp"  
#include "type_action.hpp"
```

Include dependency graph for abstract_action.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::behavior::abstract_action](#)

abstract concept of an action

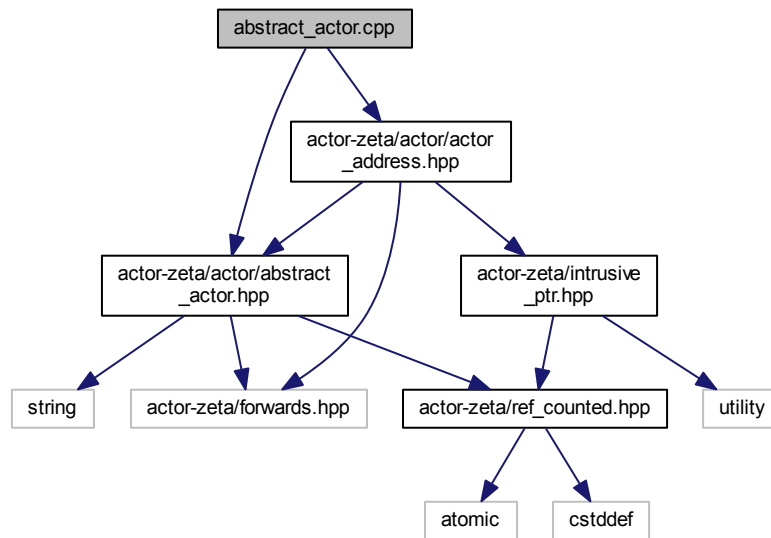
Namespaces

- [actor_zeta](#)
- [actor_zeta::behavior](#)

9.2 abstract_actor.cpp File Reference

```
#include "actor-zeta/actor/abstract_actor.hpp"
#include "actor-zeta/actor/actor_address.hpp"
```

Include dependency graph for abstract_actor.cpp:



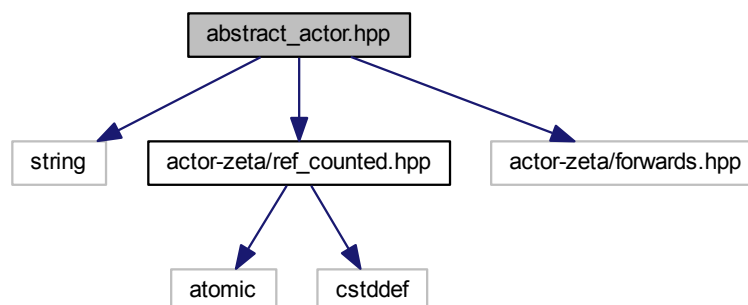
Namespaces

- [actor_zeta](#)
- [actor_zeta::actor](#)

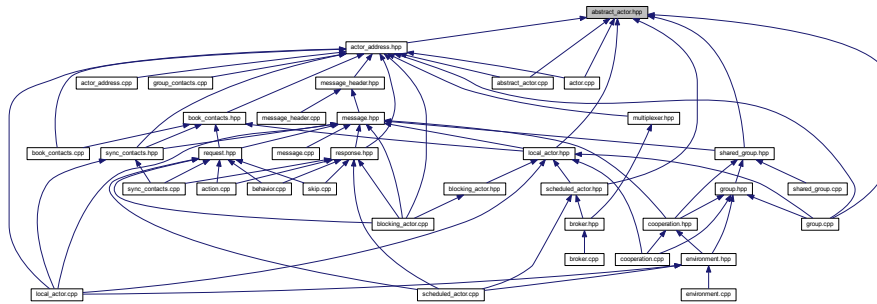
9.3 abstract_actor.hpp File Reference

```
#include <string>
#include "actor-zeta/ref_counted.hpp"
#include "actor-zeta/forwards.hpp"
```

Include dependency graph for abstract_actor.hpp:



This graph shows which files directly or indirectly include this file:



Classes

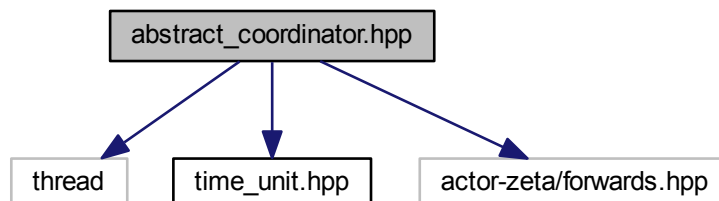
- class `actor_zeta::actor::abstract_actor`
abstract concept of an actor

Namespaces

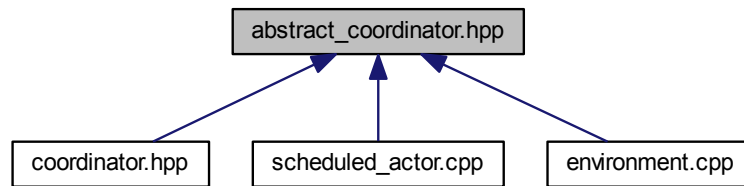
- actor_zeta
- actor_zeta::actor

9.4 abstract_coordinator.hpp File Reference

```
#include <thread>
#include "time_unit.hpp"
#include "actor-zeta/forwards.hpp"
Include dependency graph for abstract_coordinator.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `actor_zeta::executor::abstract_coordinator`
abstract concept of an coordination approach

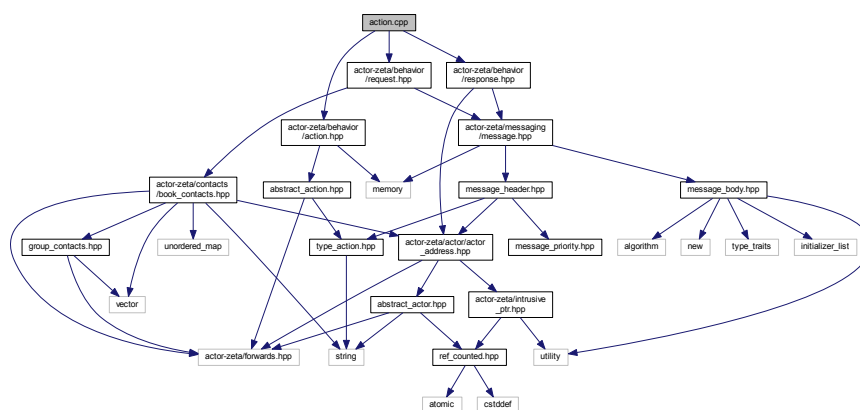
Namespaces

- `actor_zeta`
- `actor_zeta::executor`

9.5 action.cpp File Reference

```
#include "actor-zeta/behavior/action.hpp"
#include "actor-zeta/behavior/request.hpp"
#include "actor-zeta/behavior/response.hpp"
```

Include dependency graph for `action.cpp`:

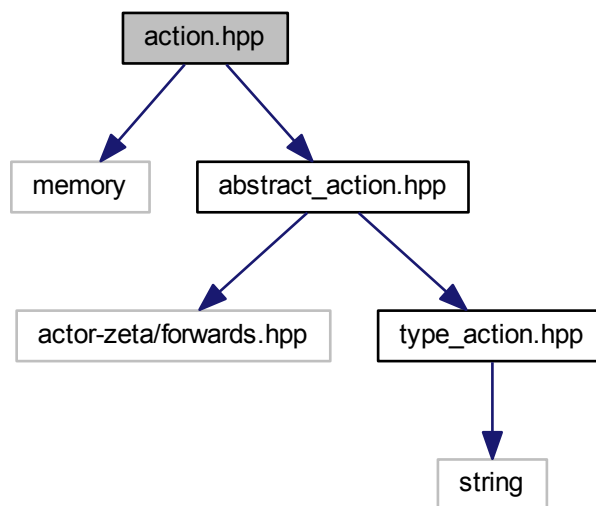


Namespaces

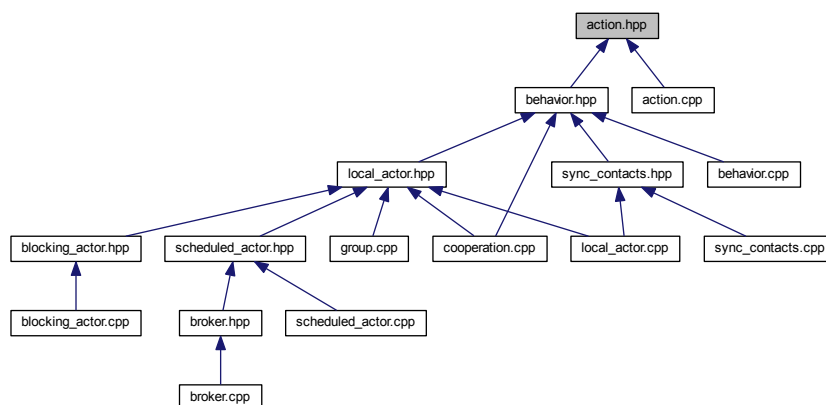
- `actor_zeta`
- `actor_zeta::behavior`

9.6 action.hpp File Reference

```
#include <memory>
#include "abstract_action.hpp"
Include dependency graph for action.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `actor_zeta::behavior::action`
Basic action implementation.

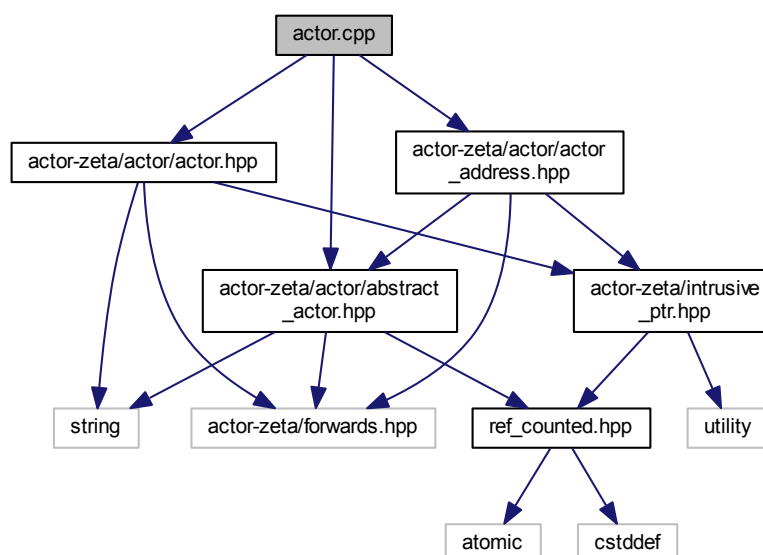
Namespaces

- [actor_zeta](#)
- [actor_zeta::behavior](#)

9.7 actor.cpp File Reference

```
#include "actor-zeta/actor/actor.hpp"
#include "actor-zeta/actor/abstract_actor.hpp"
#include "actor-zeta/actor/actor_address.hpp"
```

Include dependency graph for actor.cpp:



Namespaces

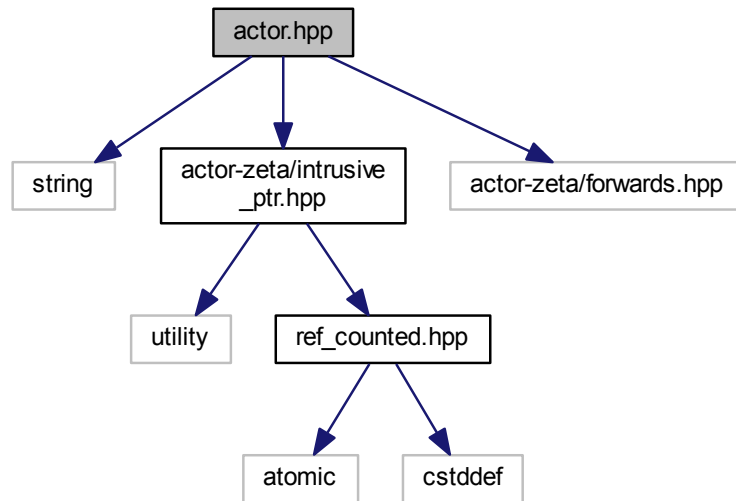
- [actor_zeta](#)
- [actor_zeta::actor](#)

9.8 actor.hpp File Reference

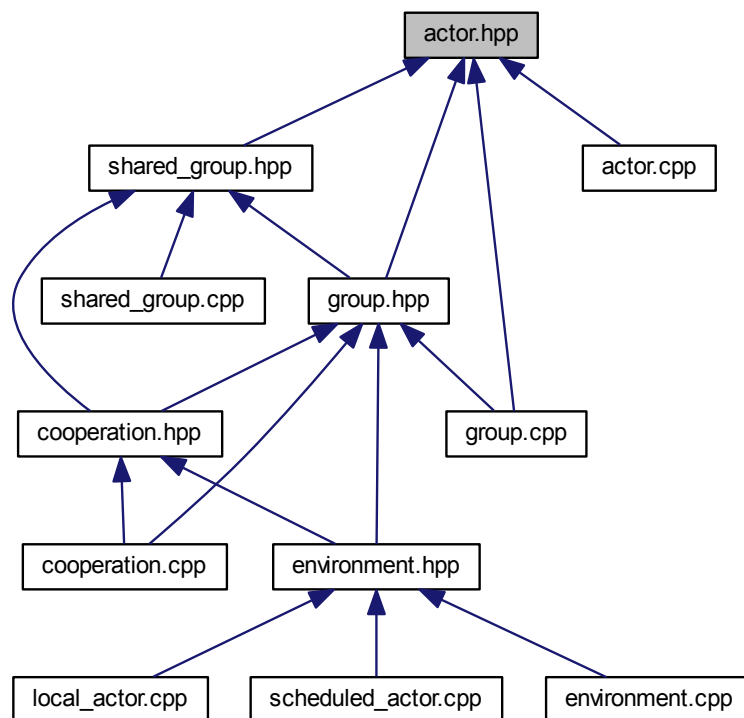
```
#include <string>
#include "actor-zeta/intrusive_ptr.hpp"
```

```
#include "actor-zeta/forwards.hpp"
```

Include dependency graph for actor.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::actor::actor](#)

Basic actor implementation.

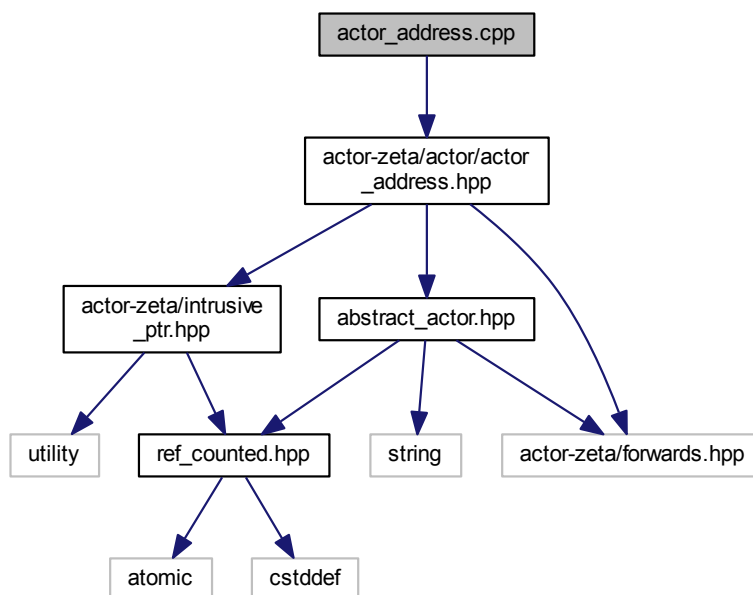
Namespaces

- [actor_zeta](#)
- [actor_zeta::actor](#)

9.9 actor_address.cpp File Reference

```
#include "actor-zeta/actor/actor_address.hpp"
```

Include dependency graph for actor_address.cpp:

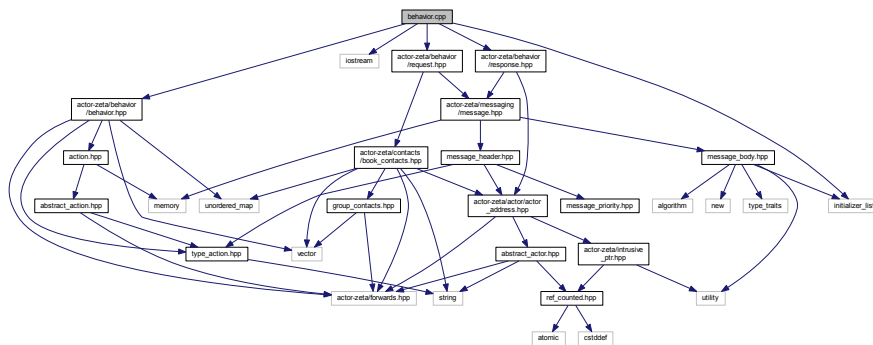


Namespaces

- [actor_zeta](#)
- [actor_zeta::actor](#)

9.11 behavior.cpp File Reference

```
#include <initializer_list>
#include "iostream"
#include "actor-zeta/behavior/behavior.hpp"
#include "actor-zeta/behavior/request.hpp"
#include "actor-zeta/behavior/response.hpp"
Include dependency graph for behavior.cpp:
```



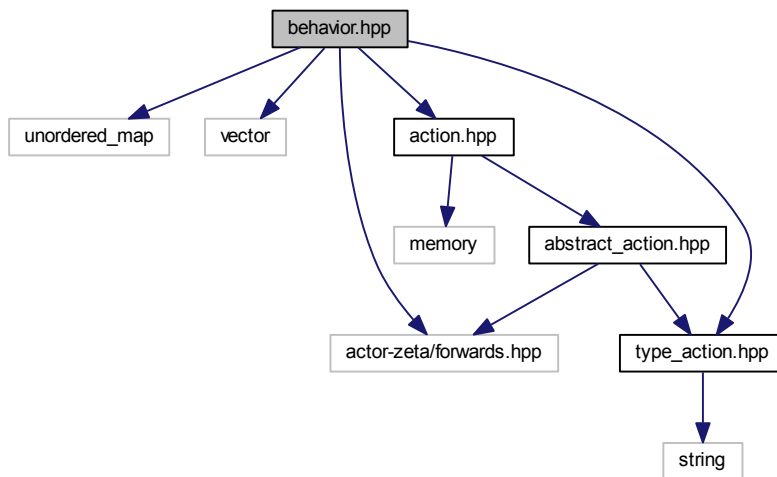
Namespaces

- [actor_zeta](#)
- [actor_zeta::behavior](#)

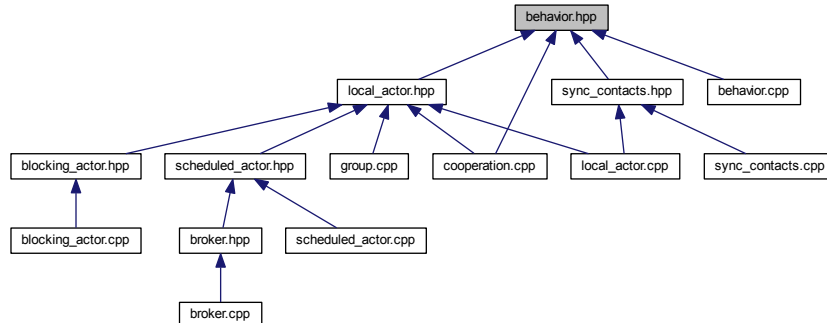
9.12 behavior.hpp File Reference

```
#include <unordered_map>
#include <vector>
#include "actor-zeta/forwards.hpp"
#include "action.hpp"
#include "type_action.hpp"
```

Include dependency graph for behavior.hpp:



This graph shows which files directly or indirectly include this file:



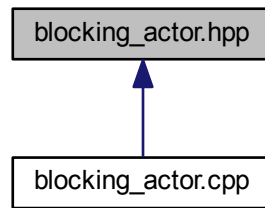
Classes

- class `actor_zeta::behavior::behavior`
Class for lifecycle determination.

Namespaces

- `actor_zeta`
- `actor_zeta::behavior`

This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::actor::blocking_actor](#)
Represents actor type with blocking mode.

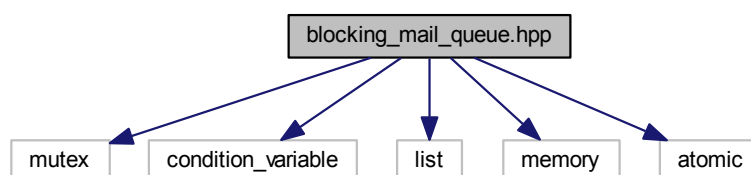
Namespaces

- [actor_zeta](#)
- [actor_zeta::actor](#)

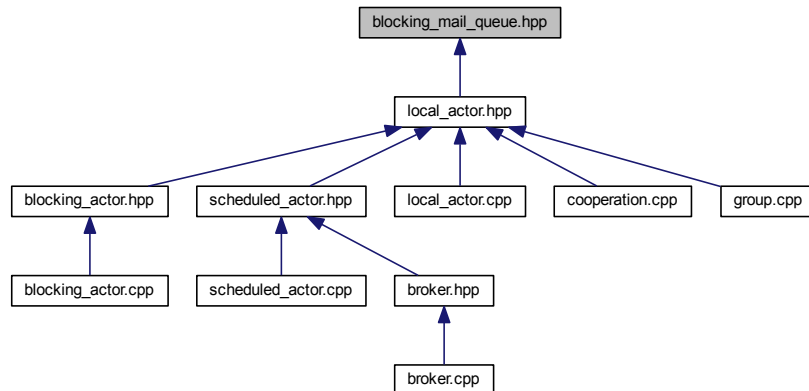
9.15 blocking_mail_queue.hpp File Reference

```
#include <mutex>
#include <condition_variable>
#include <list>
#include <memory>
#include <atomic>
```

Include dependency graph for blocking_mail_queue.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::messaging::blocking_mail_queue< T >](#)
A mailbox class for message queue.

Namespaces

- [actor_zeta](#)
- [actor_zeta::messaging](#)

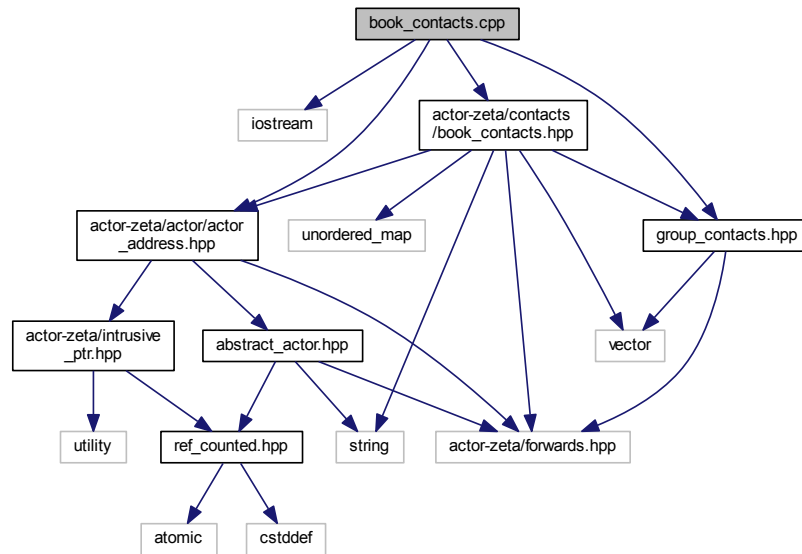
Enumerations

- enum [actor_zeta::messaging::enqueue_result](#) { [actor_zeta::messaging::enqueue_result::success](#), [actor_zeta::messaging::enqueue_result::unblocked_reader](#), [actor_zeta::messaging::enqueue_result::queue_closed](#) }
A strongly typed enum class representing enqueue result.

9.16 book_contacts.cpp File Reference

```
#include <iostream>
#include "actor-zeta/contacts/book_contacts.hpp"
#include "actor-zeta/contacts/group_contacts.hpp"
```

```
#include "actor-zeta/actor/actor_address.hpp"
Include dependency graph for book_contacts.cpp:
```



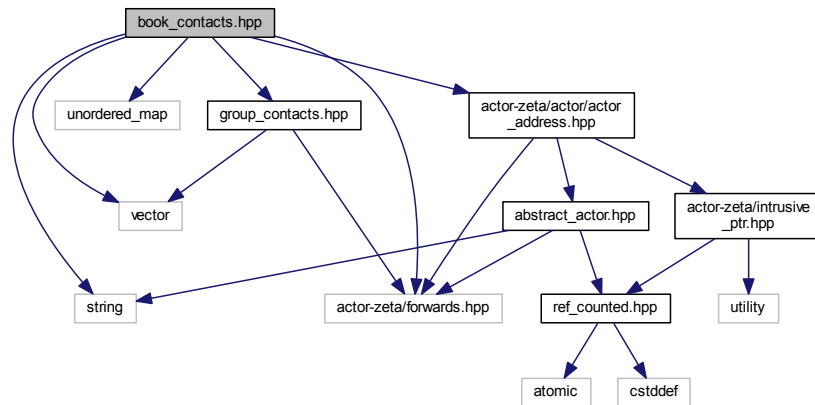
Namespaces

- [actor_zeta](#)
- [actor_zeta::contacts](#)

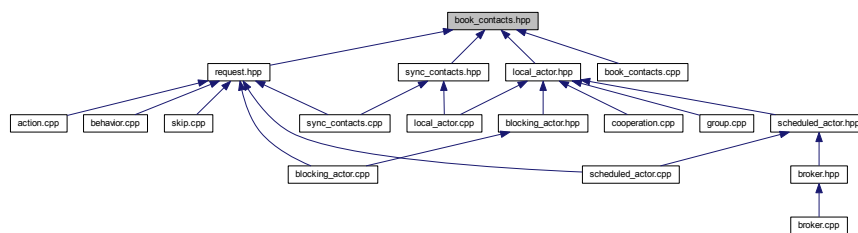
9.17 book_contacts.hpp File Reference

```
#include <string>
#include <vector>
#include <unordered_map>
#include "actor-zeta/forwards.hpp"
#include "actor-zeta/actor/actor_address.hpp"
#include "group_contacts.hpp"
```

Include dependency graph for book_contacts.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::contacts::book_contacts](#)

Navigation map for actor & groups.

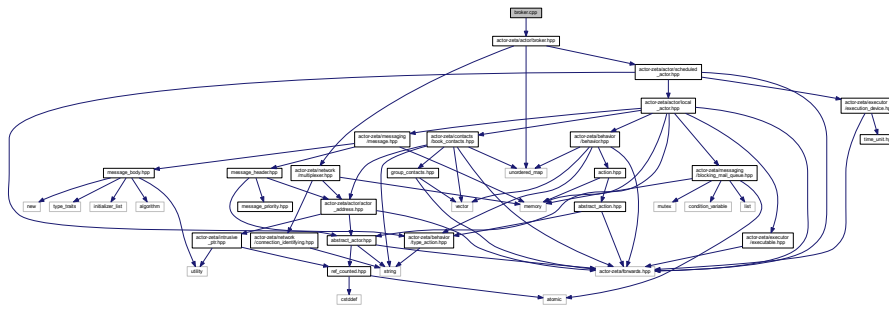
Namespaces

- [actor_zeta](#)
- [actor_zeta::contacts](#)

9.18 broker.cpp File Reference

```
#include "actor-zeta/actor/broker.hpp"
```

Include dependency graph for broker.cpp:



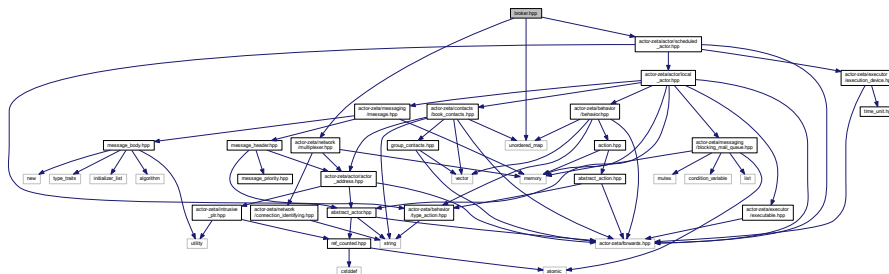
Namespaces

- [actor_zeta](#)
- [actor_zeta::network](#)

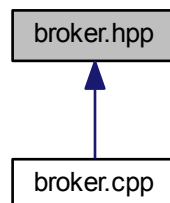
9.19 broker.hpp File Reference

```
#include <unordered_map>
#include "actor-zeta/actor/scheduled_actor.hpp"
#include "actor-zeta/network/multiplexer.hpp"
```

Include dependency graph for broker.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::network::broker](#)

A broker for messaging.

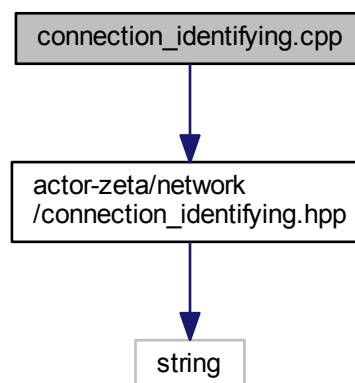
Namespaces

- [actor_zeta](#)
- [actor_zeta::network](#)

9.20 connection_identifying.cpp File Reference

```
#include "actor-zeta/network/connection_identifying.hpp"
```

Include dependency graph for connection_identifying.cpp:



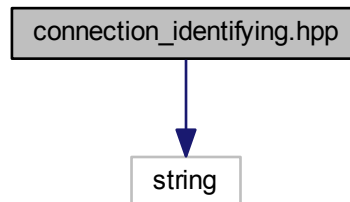
Namespaces

- [actor_zeta](#)
- [actor_zeta::network](#)

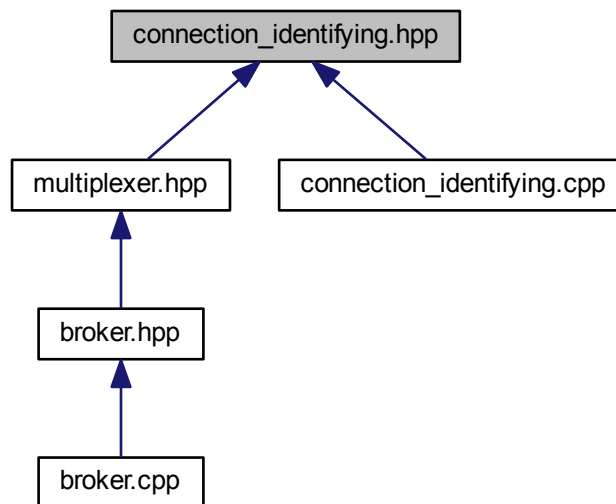
9.21 connection_identifying.hpp File Reference

```
#include <string>
```

Include dependency graph for connection_identifying.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::network::connection_identifying](#)
Implementation of connection functionality.

Namespaces

- [actor_zeta](#)
- [actor_zeta::network](#)

Enumerations

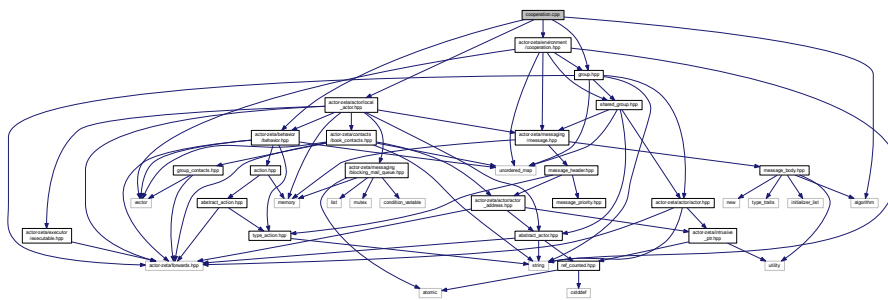
- enum [actor_zeta::network::type_connect](#) : int { [actor_zeta::network::type_connect::tcp](#), [actor_zeta::network::type_connect::udp](#) }

A strongly typed enum class representing connection type.

9.22 cooperation.cpp File Reference

```
#include "actor-zeta/behavior/behavior.hpp"
#include "actor-zeta/environment/cooperation.hpp"
#include "actor-zeta/environment/group.hpp"
#include "actor-zeta/actor/local_actor.hpp"
#include <algorithm>
```

Include dependency graph for cooperation.cpp:



Namespaces

- [actor_zeta](#)
- [actor_zeta::environment](#)

9.23 cooperation.hpp File Reference

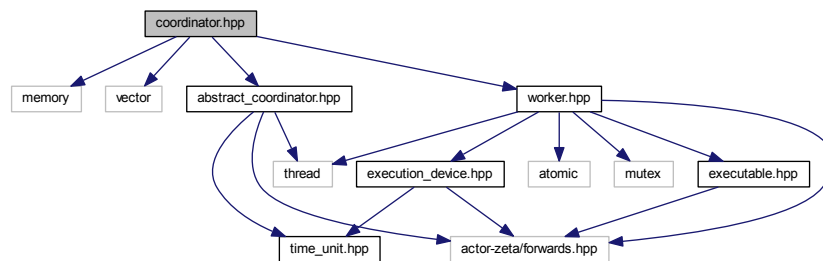
```
#include <unordered_map>
#include <string>
#include <vector>
#include "actor-zeta/messaging/message.hpp"
#include "group.hpp"
```


Functions

- `template<class V >`
`void actor_zeta::environment::send (actor_zeta::environment::cooperation &c, std::string commanda, V value)`

9.24 coordinator.hpp File Reference

```
#include <memory>
#include <vector>
#include "worker.hpp"
#include "abstract_coordinator.hpp"
Include dependency graph for coordinator.hpp:
```



Classes

- `class actor_zeta::executor::coordinator< Policy >`
Provides ruler for environment.

Namespaces

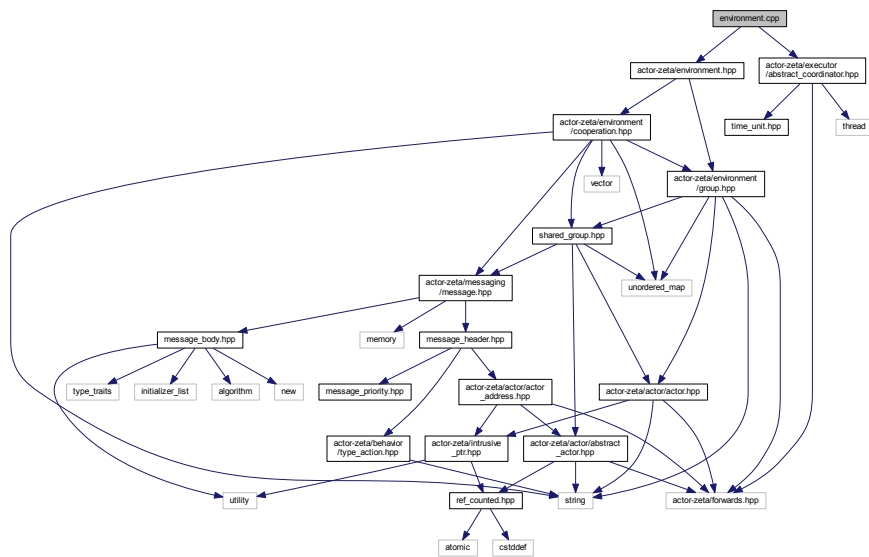
- `actor_zeta`
- `actor_zeta::executor`

9.25 description.h File Reference

9.26 environment.cpp File Reference

```
#include "actor-zeta/environment.hpp"
#include "actor-zeta/executor/abstract_coordinator.hpp"
```

Include dependency graph for environment.cpp:

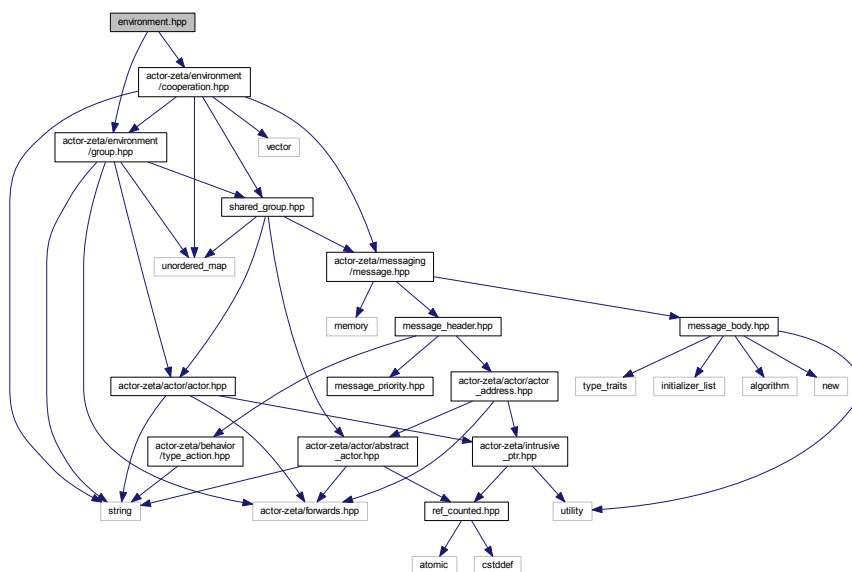


Namespaces

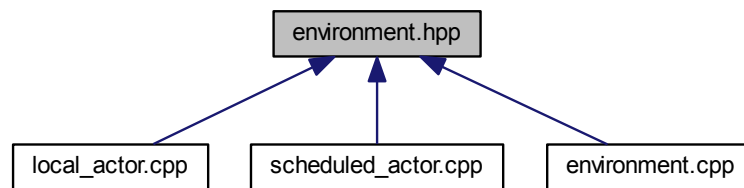
- [actor_zeta](#)
- [actor_zeta::environment](#)

9.27 environment.hpp File Reference

```
#include "actor-zeta/environment/group.hpp"
#include "actor-zeta/environment/cooperation.hpp"
Include dependency graph for environment.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `actor_zeta::environment::environment`
An actors workplace platform.

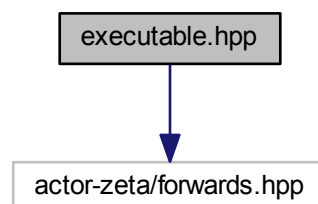
Namespaces

- `actor_zeta`
- `actor_zeta::environment`

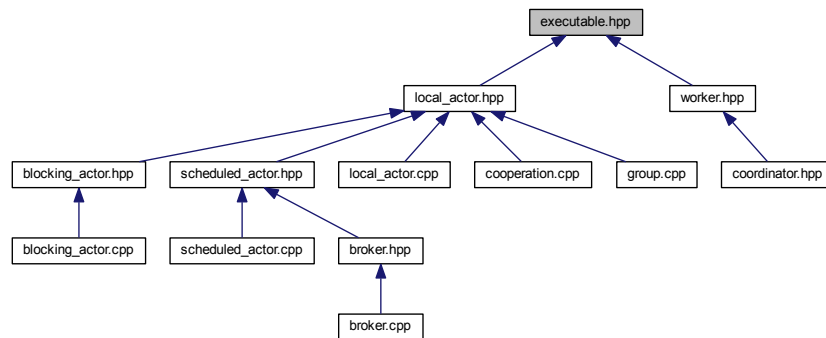
9.28 executable.hpp File Reference

```
#include "actor-zeta/forwards.hpp"
```

Include dependency graph for `executable.hpp`:



This graph shows which files directly or indirectly include this file:



Classes

- struct [actor_zeta::executor::executable](#)

Namespaces

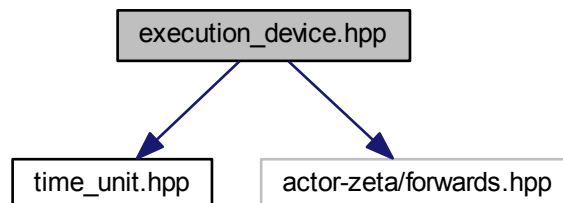
- [actor_zeta](#)
- [actor_zeta::executor](#)

9.29 execution_device.hpp File Reference

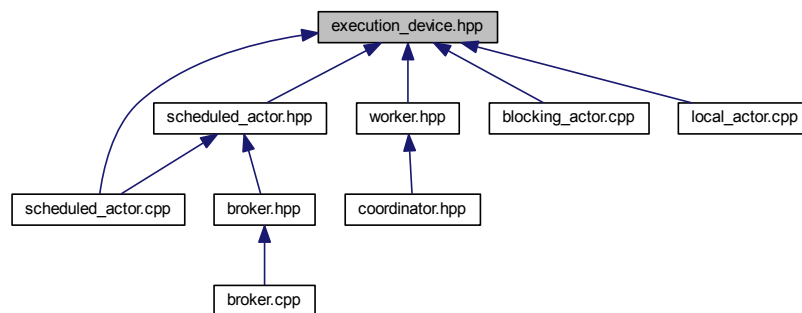
```
#include "time_unit.hpp"
```

```
#include "actor-zeta/forwards.hpp"
```

Include dependency graph for execution_device.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct [actor_zeta::executor::execution_device](#)
execution_device

Namespaces

- [actor_zeta](#)
- [actor_zeta::executor](#)

9.30 forwards.hpp File Reference

Classes

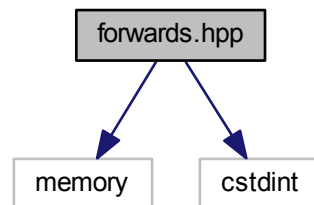
- class [actor_zeta::executor::coordinator< Policy >](#)
Provides ruler for environment.

Namespaces

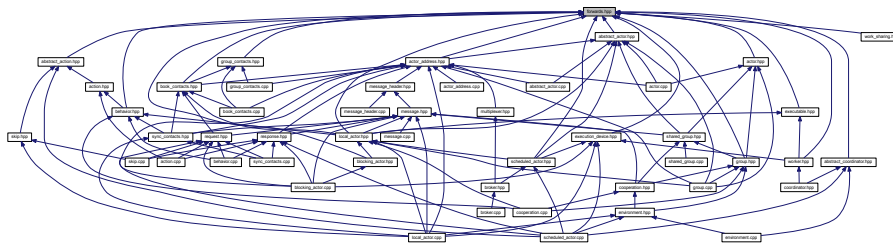
- [actor_zeta](#)
- [actor_zeta::messaging](#)
- [actor_zeta::actor](#)
- [actor_zeta::behavior](#)
- [actor_zeta::contacts](#)
- [actor_zeta::environment](#)
- [actor_zeta::executor](#)

9.31 forwards.hpp File Reference

```
#include <memory>
#include <cstdint>
Include dependency graph for libactor_zeta_io/actor-zeta/forwards.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

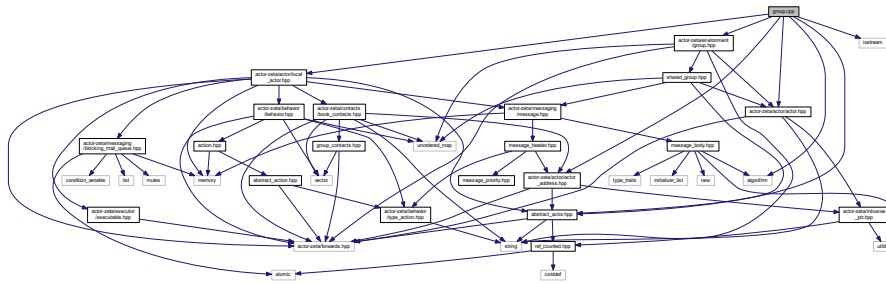
- [actor_zeta](#)
- [actor_zeta::network](#)

9.32 group.cpp File Reference

```
#include "actor-zeta/actor/local_actor.hpp"
#include "actor-zeta/environment/group.hpp"
#include <algorithm>
#include <iostream>
#include "actor-zeta/actor/actor_address.hpp"
#include "actor-zeta/actor/abstract_actor.hpp"
```



```
#include "actor-zeta/actor/actor.hpp"
Include dependency graph for group.cpp:
```

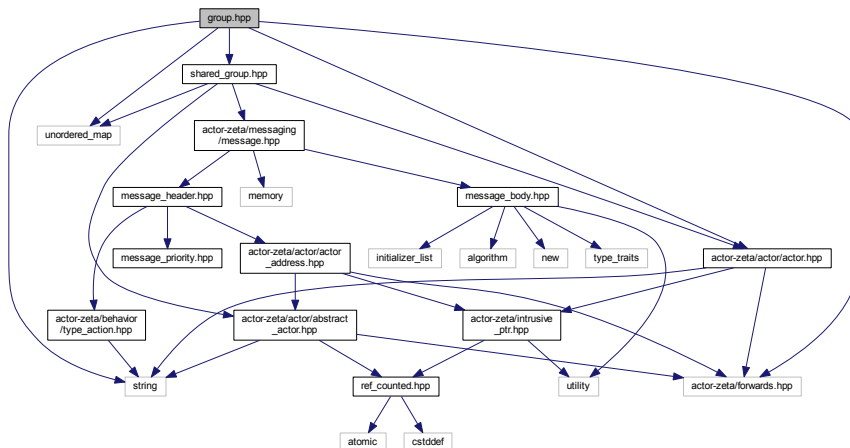


Namespaces

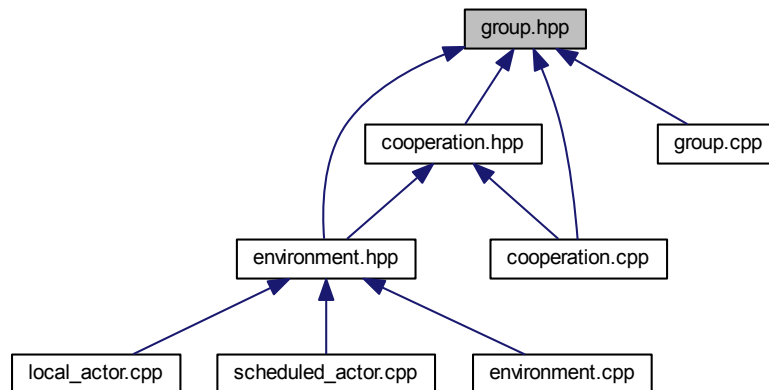
- actor_zeta
- actor_zeta::environment

9.33 group.hpp File Reference

```
#include <unordered_map>
#include <string>
#include "actor-zeta/forwards.hpp"
#include "actor-zeta/actor/actor.hpp"
#include "shared_group.hpp"
Include dependency graph for group.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::environment::group](#)

A group combinator for actors.

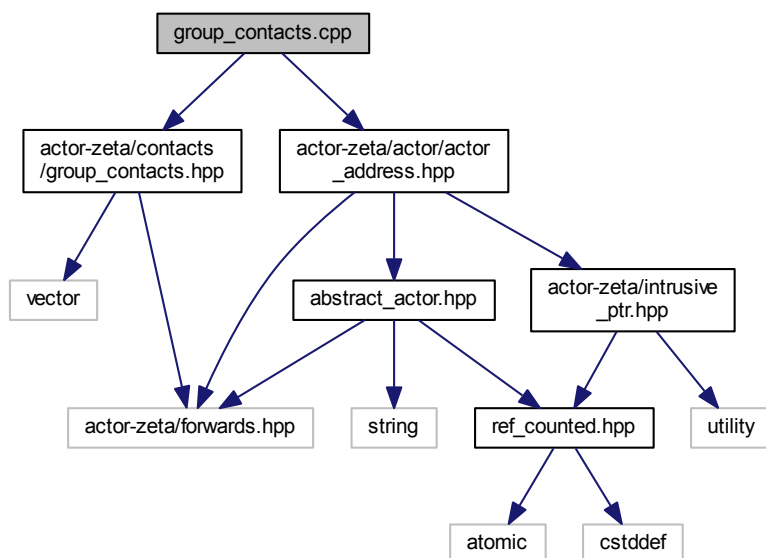
Namespaces

- [actor_zeta](#)
- [actor_zeta::environment](#)

9.34 group_contacts.cpp File Reference

```
#include "actor-zeta/contacts/group_contacts.hpp"
#include "actor-zeta/actor/actor_address.hpp"
```

Include dependency graph for group_contacts.cpp:



Namespaces

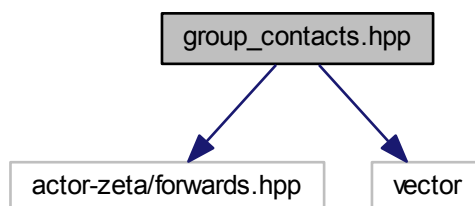
- [actor_zeta](#)
- [actor_zeta::contacts](#)

9.35 group_contacts.hpp File Reference

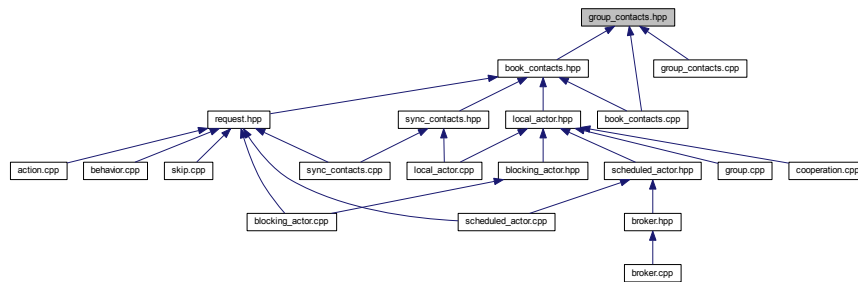
```
#include "actor-zeta/forwards.hpp"
```

```
#include <vector>
```

Include dependency graph for group_contacts.hpp:



This graph shows which files directly or indirectly include this file:



Classes

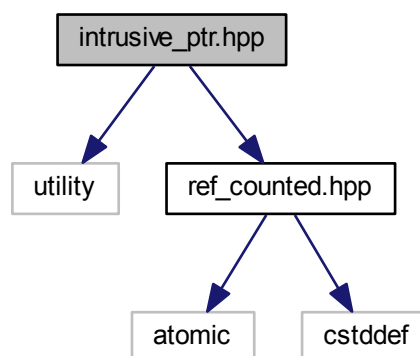
- class [actor_zeta::contacts::group_contacts](#)
Address container for groups.

Namespaces

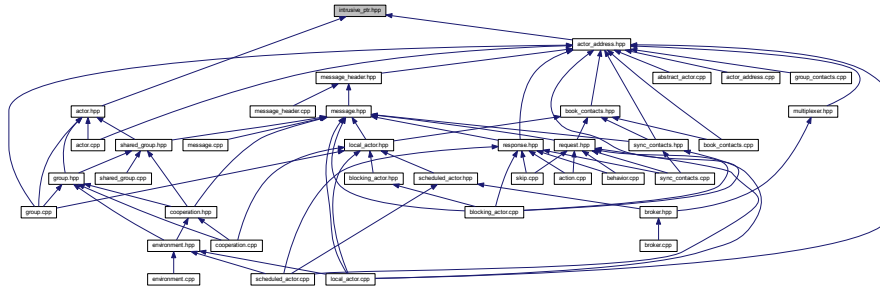
- [actor_zeta](#)
- [actor_zeta::contacts](#)

9.36 intrusive_ptr.hpp File Reference

```
#include <utility>
#include "ref_counted.hpp"
Include dependency graph for intrusive_ptr.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::intrusive_ptr< T >](#)
This class represents smart pointers.

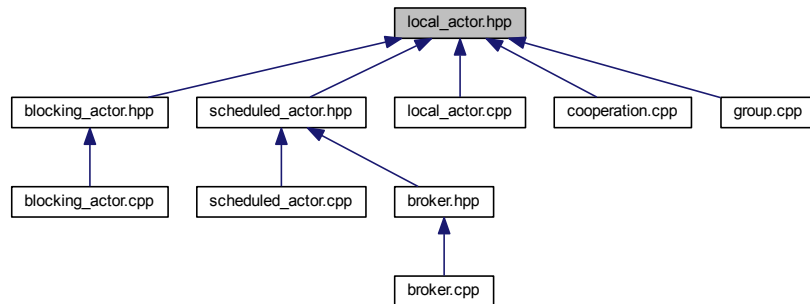
Namespaces

- [actor_zeta](#)

Functions

- `template<class T , class U >`
`bool actor_zeta::operator== (intrusive_ptr< T > const &a, intrusive_ptr< U > const &b) noexcept`
- `template<class T , class U >`
`bool actor_zeta::operator!= (intrusive_ptr< T > const &a, intrusive_ptr< U > const &b) noexcept`
- `template<class T >`
`bool actor_zeta::operator== (intrusive_ptr< T > const &a, T *b) noexcept`
- `template<class T >`
`bool actor_zeta::operator!= (intrusive_ptr< T > const &a, T *b) noexcept`
- `template<class T >`
`bool actor_zeta::operator== (T *a, intrusive_ptr< T > const &b) noexcept`
- `template<class T >`
`bool actor_zeta::operator!= (T *a, intrusive_ptr< T > const &b) noexcept`
- `template<class T , class U >`
`bool actor_zeta::operator< (intrusive_ptr< T > const &a, intrusive_ptr< U > const &b) noexcept`
- `template<class T >`
`void actor_zeta::swap (intrusive_ptr< T > &a, intrusive_ptr< T > &b) noexcept`
- `template<class T >`
`T * actor_zeta::get_pointer (intrusive_ptr< T > const &p) noexcept`
- `template<class T , class U >`
`intrusive_ptr< T > actor_zeta::static_pointer_cast (intrusive_ptr< U > const &r) noexcept`
- `template<class T , class U >`
`intrusive_ptr< T > actor_zeta::const_pointer_cast (intrusive_ptr< U > const &r) noexcept`
- `template<class T , class U >`
`intrusive_ptr< T > actor_zeta::dynamic_pointer_cast (intrusive_ptr< U > const &r) noexcept`

This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::actor::local_actor](#)
Class for location dependant type actor.

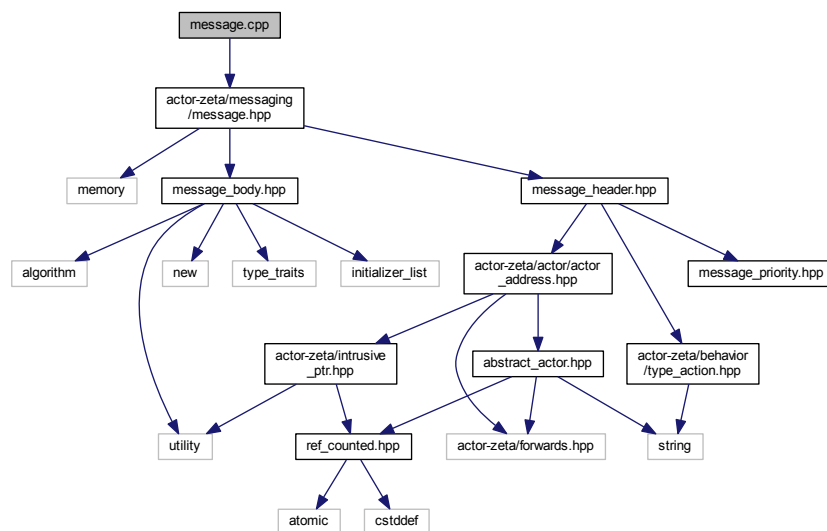
Namespaces

- [actor_zeta](#)
- [actor_zeta::actor](#)

9.39 message.cpp File Reference

```
#include "actor-zeta/messaging/message.hpp"
```

Include dependency graph for message.cpp:

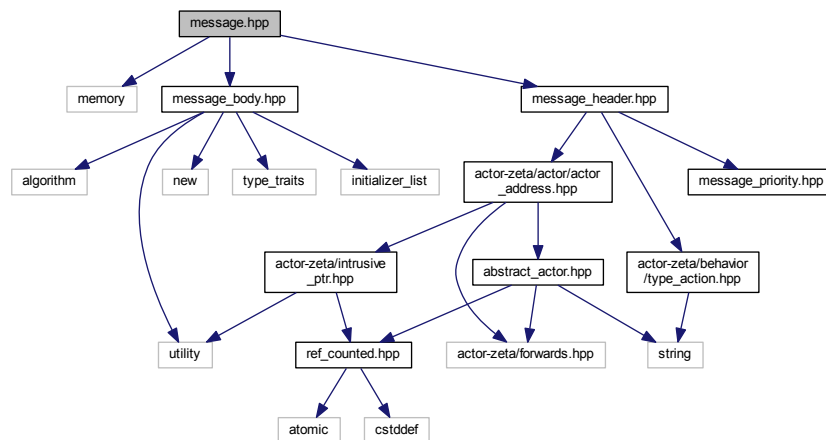


Namespaces

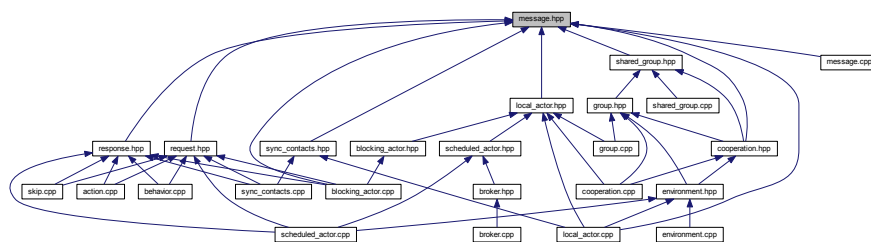
- [actor_zeta](#)
- [actor_zeta::messaging](#)

9.40 message.hpp File Reference

```
#include <memory>
#include "message_header.hpp"
#include "message_body.hpp"
Include dependency graph for message.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::messaging::message](#)
Class to represent messages.

Namespaces

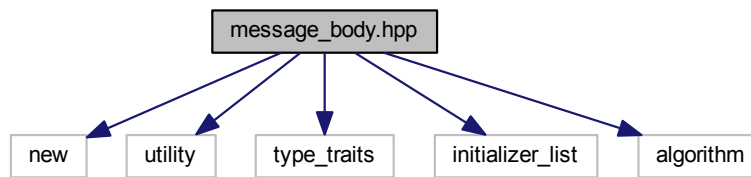
- [actor_zeta](#)
- [actor_zeta::messaging](#)

Functions

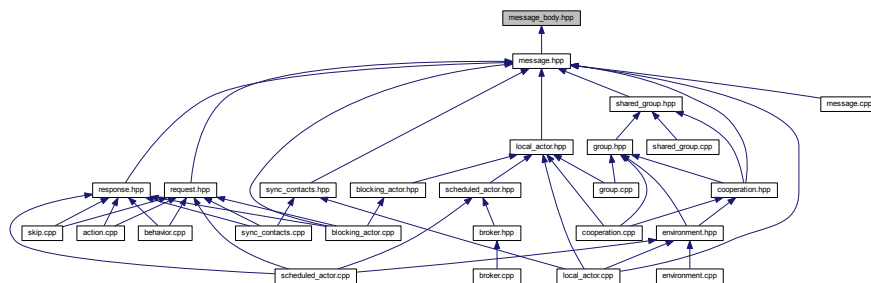
- template<std::size_t N, typename T >
message * [actor_zeta::messaging::make_message](#) (const char(&aStr)[N], T data)
- template<typename T >
message * [actor_zeta::messaging::make_message](#) (const std::string &type, T data)
- template<typename T >
message * [actor_zeta::messaging::make_message](#) (const std::string &type, T data, actor::actor_address address)
- template<std::size_t N, typename T >
message * [actor_zeta::messaging::make_message](#) (const char(&aStr)[N], T data, actor::actor_address address)

9.41 message_body.hpp File Reference

```
#include <new>
#include <utility>
#include <type_traits>
#include <initializer_list>
#include <algorithm>
Include dependency graph for message_body.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::messaging::message_body](#)
A message value container.

Namespaces

- [actor_zeta](#)
- [actor_zeta::messaging](#)

Functions

- void [swap](#) ([actor_zeta::messaging::message_body](#) &lhs, [actor_zeta::messaging::message_body](#) &rhs) noexcept

9.41.1 Function Documentation

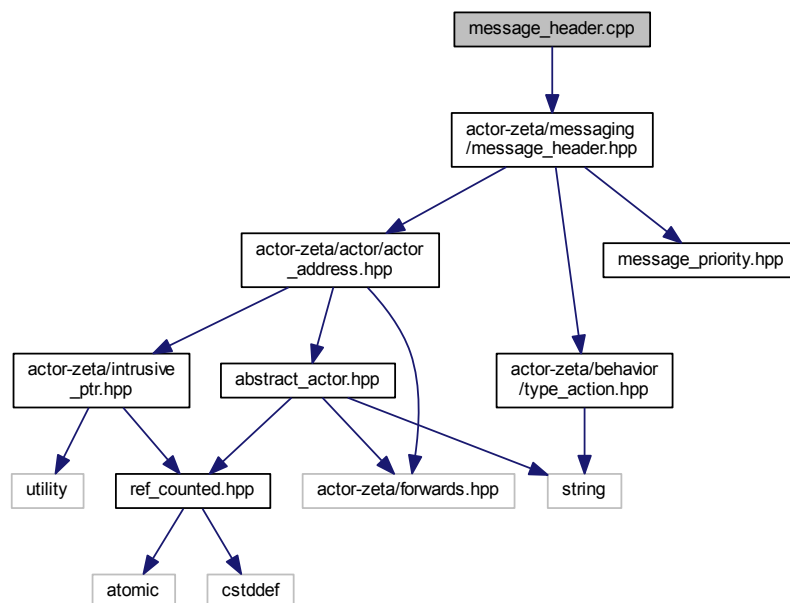
9.41.1.1 swap()

```
void swap (
    actor_zeta::messaging::message_body & lhs,
    actor_zeta::messaging::message_body & rhs ) [inline], [noexcept]
```

9.42 message_header.cpp File Reference

```
#include "actor-zeta/messaging/message_header.hpp"
```

Include dependency graph for message_header.cpp:



- actor_zeta
- actor_zeta::messaging

```
#include "actor-zeta/actor/actor_address.hpp"
#include "actor-zeta/behavior/type_action.hpp"
#include "message_priority.hpp"
```

```

graph TD
    message_header_hpp[message_header.hpp] --> actor_zeta_actor_actor_address_hpp[actor-zeta/actor/actor_address.hpp]
    message_header_hpp --> message_priority_hpp[message_priority.hpp]
    message_header_hpp --> actor_zeta_behavior_type_action_hpp[actor-zeta/behavior/type_action.hpp]
    actor_zeta_actor_actor_address_hpp --> actor_zeta_intrusive_ptr_hpp[actor-zeta/intrusive_ptr.hpp]
    actor_zeta_actor_actor_address_hpp --> abstract_actor_hpp[abstract_actor.hpp]
    actor_zeta_actor_actor_address_hpp --> actor_zeta_forwards_hpp[actor-zeta/forwards.hpp]
    actor_zeta_intrusive_ptr_hpp --> utility[utility]
    actor_zeta_intrusive_ptr_hpp --> ref_counted_hpp[ref_counted.hpp]
    abstract_actor_hpp --> ref_counted_hpp
    abstract_actor_hpp --> actor_zeta_forwards_hpp
    abstract_actor_hpp --> string[string]
    actor_zeta_behavior_type_action_hpp --> string
    ref_counted_hpp --> atomic[atomic]
    ref_counted_hpp --> cstddef[cstddef]
  
```

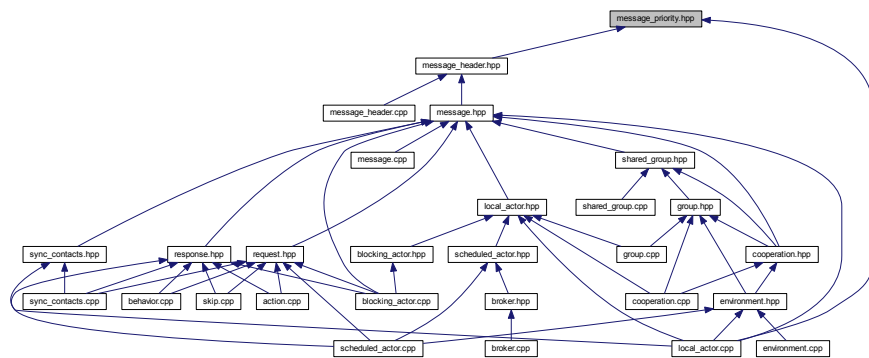
- class `actor_zeta::messaging::message_header`
A message description.

Namespaces

- [actor_zeta](#)
- [actor_zeta::messaging](#)

9.44 message_priority.hpp File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [actor_zeta](#)
- [actor_zeta::messaging](#)

Enumerations

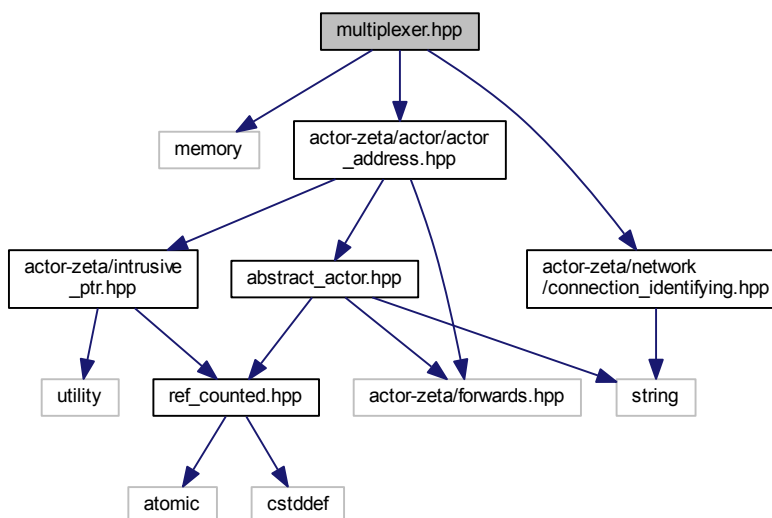
- enum [actor_zeta::messaging::message_priority](#) : int { [actor_zeta::messaging::message_priority::low](#) = 0, [actor_zeta::messaging::message_priority::normal](#), [actor_zeta::messaging::message_priority::high](#) }

9.45 multiplexer.hpp File Reference

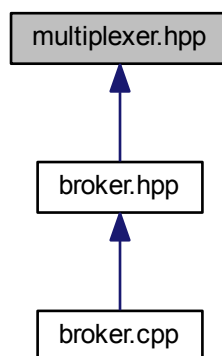
```
#include <memory>
#include "actor-zeta/actor/actor_address.hpp"
```

```
#include "actor-zeta/network/connection_identifying.hpp"
```

Include dependency graph for multiplexer.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct `actor_zeta::network::multiplexer`
Multiplexing utility class.

Namespaces

- `actor_zeta`
- `actor_zeta::network`

Typedefs

- using [actor_zeta::network::unique_muxer_ptr](#) = std::unique_ptr< muxer >
- using [actor_zeta::network::shared_muxer_ptr](#) = std::shared_ptr< muxer >

9.46 README.md File Reference

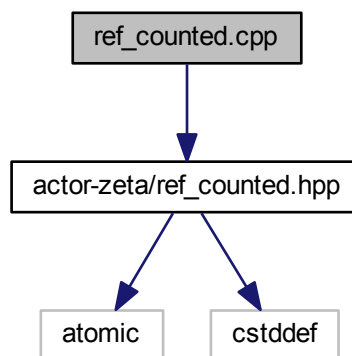
9.47 README.md File Reference

9.48 README.md File Reference

9.49 ref_counted.cpp File Reference

```
#include "actor-zeta/ref_counted.hpp"
```

Include dependency graph for ref_counted.cpp:



Namespaces

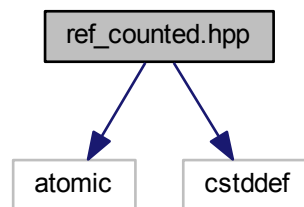
- [actor_zeta](#)

9.50 ref_counted.hpp File Reference

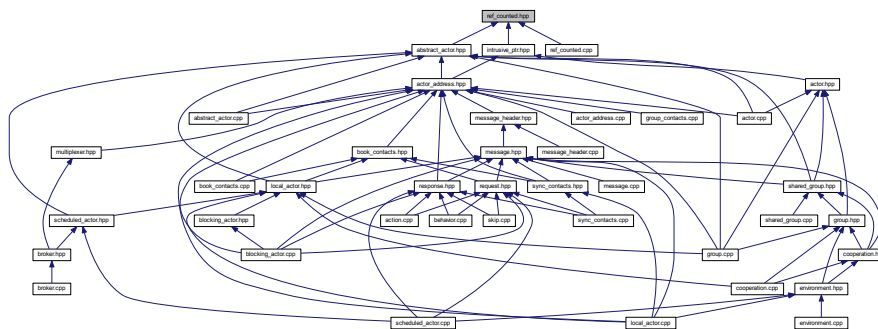
```
#include <atomic>
```

```
#include <cstdint>
```

Include dependency graph for ref_counted.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::ref_counted](#)

This class represents reference counter.

Namespaces

- [actor_zeta](#)

Functions

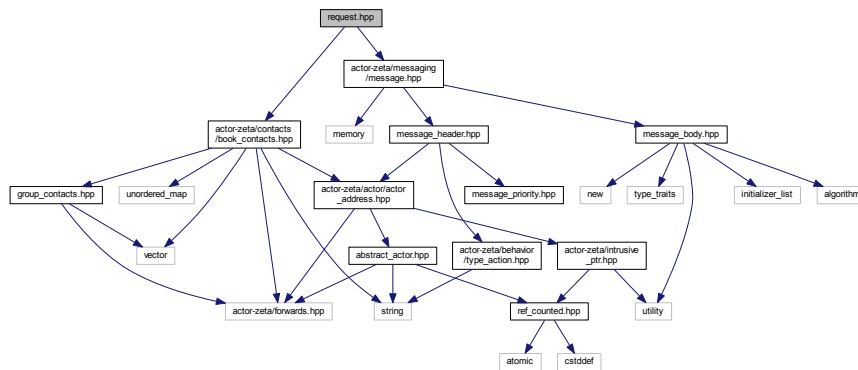
- void [actor_zeta::intrusive_ptr_add_ref](#) (ref_counted *p)
- void [actor_zeta::intrusive_ptr_release](#) (ref_counted *p)

9.51 request.hpp File Reference

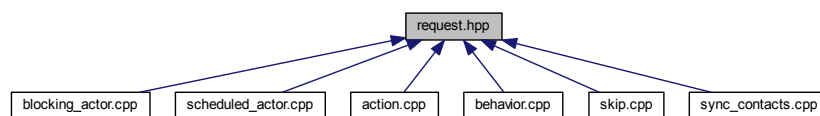
```
#include "actor-zeta/contacts/book_contacts.hpp"
```

```
#include "actor-zeta/messaging/message.hpp"
```

Include dependency graph for request.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::behavior::request](#)
This is a request container for messaging.

Namespaces

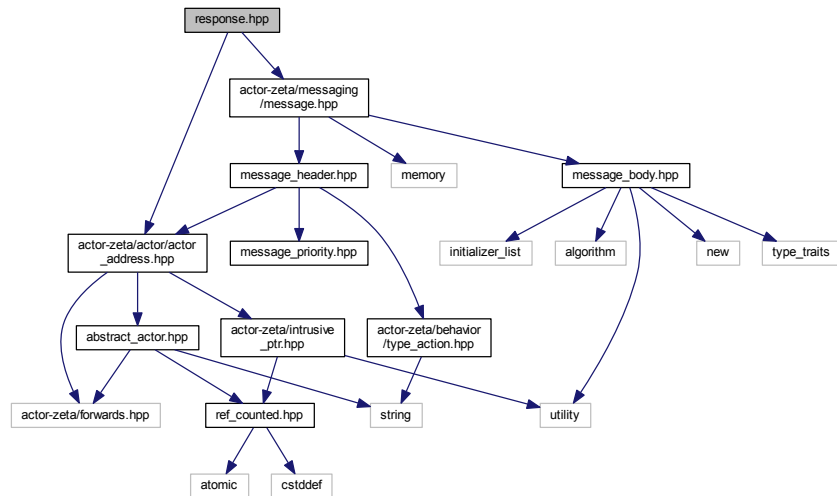
- [actor_zeta](#)
- [actor_zeta::behavior](#)

9.52 response.hpp File Reference

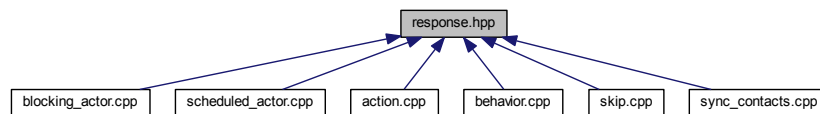
```
#include "actor-zeta/actor/actor_address.hpp"
```

```
#include "actor-zeta/messaging/message.hpp"
```


Include dependency graph for response.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::behavior::response](#)
This is a response container for messaging.

Namespaces

- [actor_zeta](#)
- [actor_zeta::behavior](#)

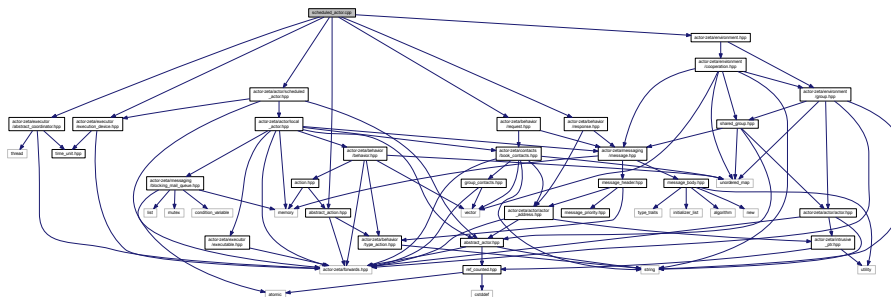
Functions

- response * [actor_zeta::behavior::make_response](#) (actor::actor_address receiver_, messaging::message *msg)

9.53 `scheduled_actor.cpp` File Reference

```
#include "actor-zeta/actor/scheduled_actor.hpp"
#include "actor-zeta/executor/abstract_coordinator.hpp"
#include "actor-zeta/executor/execution_device.hpp"
#include "actor-zeta/environment.hpp"
#include "actor-zeta/behavior/abstract_action.hpp"
#include "actor-zeta/behavior/request.hpp"
#include "actor-zeta/behavior/response.hpp"
```

Include dependency graph for `scheduled_actor.cpp`:



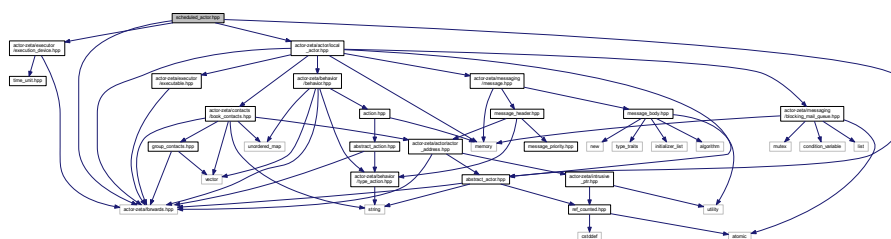
Namespaces

- [actor_zeta](#)
- [actor_zeta::actor](#)

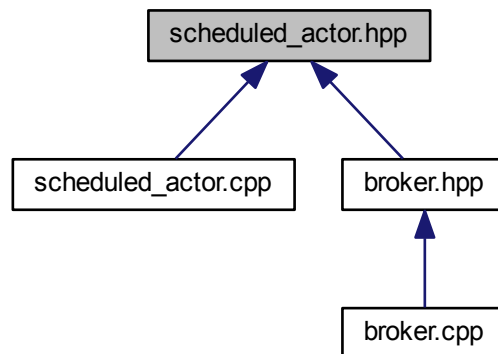
9.54 `scheduled_actor.hpp` File Reference

```
#include "actor-zeta/executor/execution_device.hpp"
#include "actor-zeta/actor/local_actor.hpp"
#include "actor-zeta/actor/abstract_actor.hpp"
#include "actor-zeta/forwards.hpp"
```

Include dependency graph for `scheduled_actor.hpp`:



This graph shows which files directly or indirectly include this file:



Classes

- class `actor_zeta::actor::scheduled_actor`
Represents scheduling type of actor.

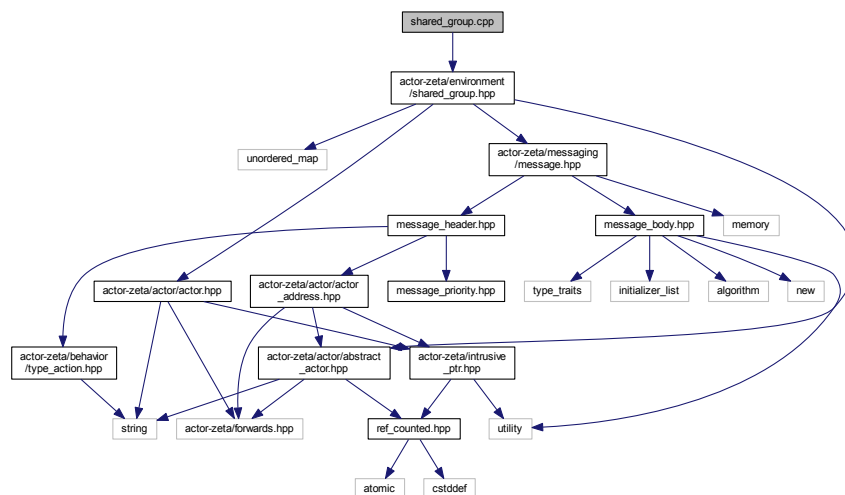
Namespaces

- `actor_zeta`
- `actor_zeta::actor`

9.55 shared_group.cpp File Reference

```
#include "actor-zeta/environment/shared_group.hpp"
```

Include dependency graph for `shared_group.cpp`:



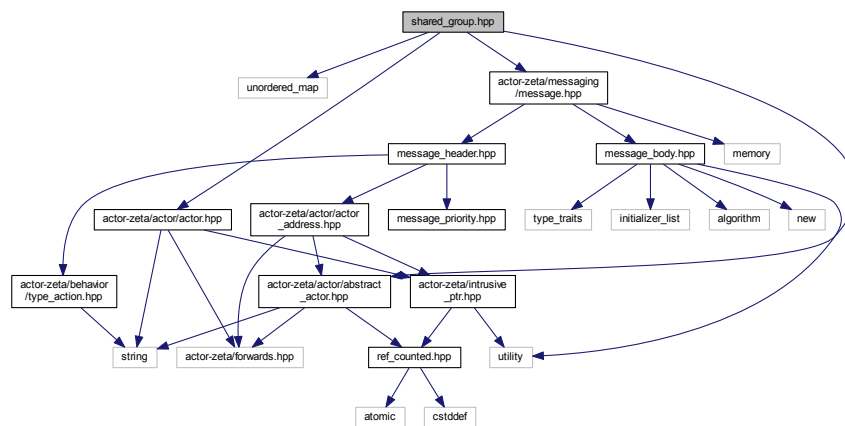
Namespaces

- [actor_zeta](#)
- [actor_zeta::environment](#)

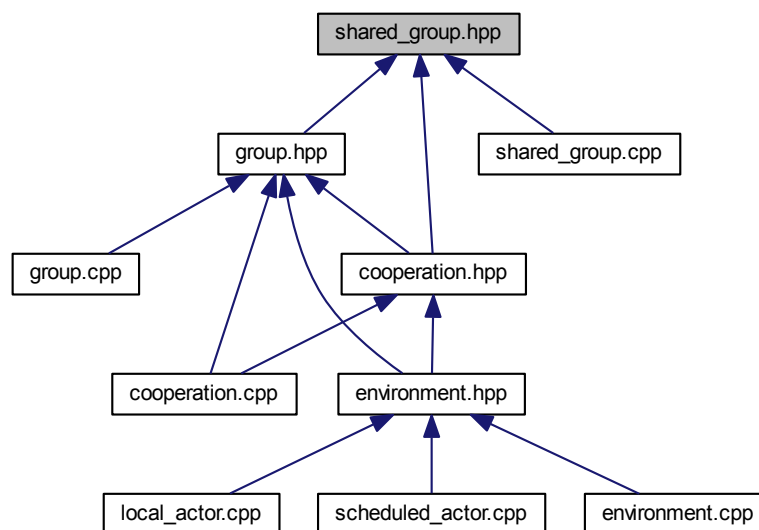
9.56 shared_group.hpp File Reference

```
#include <unordered_map>
#include "actor-zeta/actor/actor.hpp"
#include "actor-zeta/actor/abstract_actor.hpp"
#include "actor-zeta/messaging/message.hpp"
```

Include dependency graph for shared_group.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::environment::shared_group](#)

Group realisation with common resource.

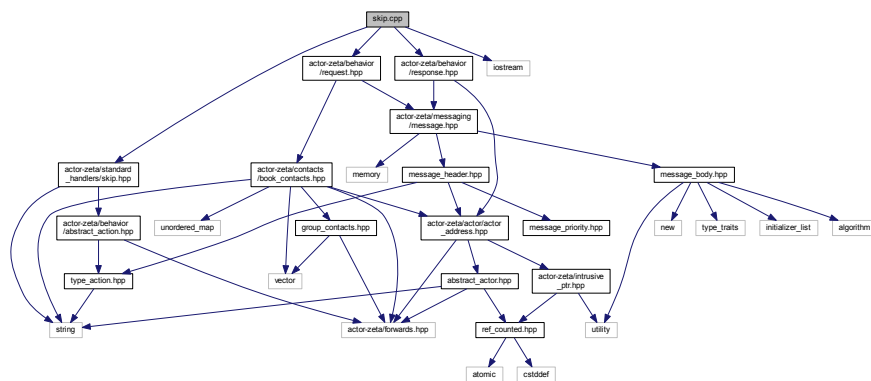
Namespaces

- [actor_zeta](#)
- [actor_zeta::environment](#)

9.57 skip.cpp File Reference

```
#include "actor-zeta/standard_handlers/skip.hpp"
#include <iostream>
#include "actor-zeta/behavior/request.hpp"
#include "actor-zeta/behavior/response.hpp"
```

Include dependency graph for skip.cpp:



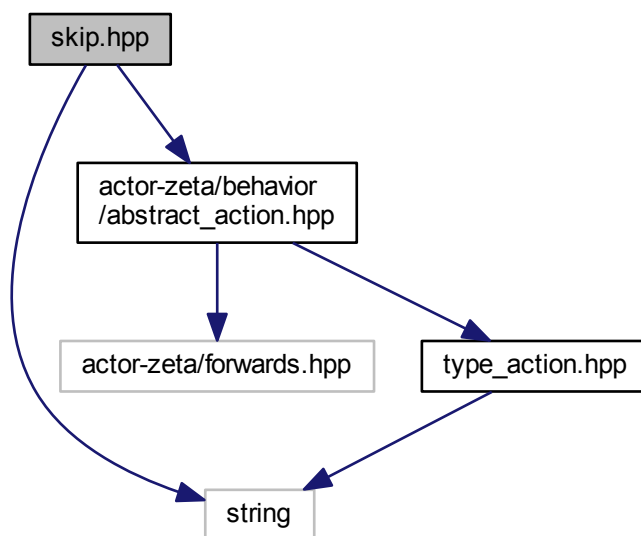
Namespaces

- [actor_zeta](#)

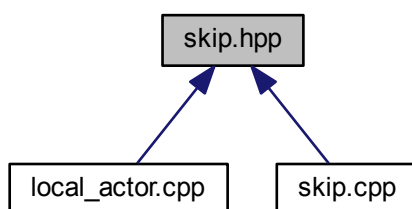
9.58 skip.hpp File Reference

```
#include <string>
#include "actor-zeta/behavior/abstract_action.hpp"
```

Include dependency graph for skip.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [actor_zeta::skip](#)
A class used for skipping action.

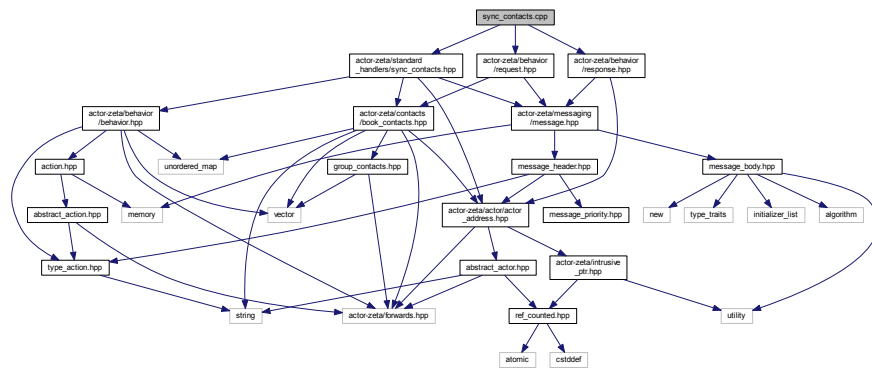
Namespaces

- [actor_zeta](#)

9.59 sync_contacts.cpp File Reference

```
#include "actor-zeta/standard_handlers/sync_contacts.hpp"
#include "actor-zeta/behavior/request.hpp"
#include "actor-zeta/behavior/response.hpp"
```

Include dependency graph for sync_contacts.cpp:



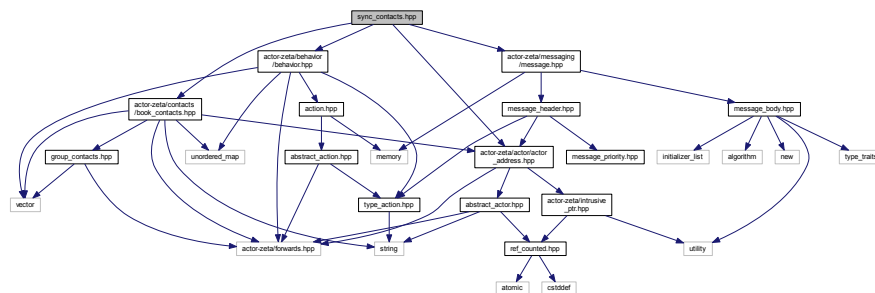
Namespaces

- [actor_zeta](#)

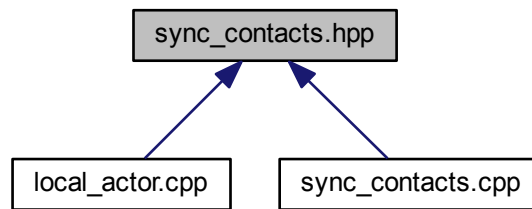
9.60 sync_contacts.hpp File Reference

```
#include "actor-zeta/behavior/behavior.hpp"
#include "actor-zeta/messaging/message.hpp"
#include "actor-zeta/contacts/book_contacts.hpp"
#include "actor-zeta/actor/actor_address.hpp"
```

Include dependency graph for sync_contacts.hpp:



This graph shows which files directly or indirectly include this file:



Classes

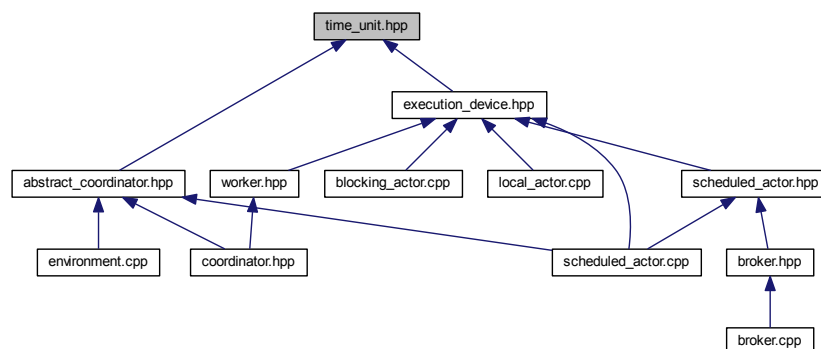
- class [actor_zeta::sync_contacts](#)
A class used for contact operations synchronization.

Namespaces

- [actor_zeta](#)

9.61 time_unit.hpp File Reference

This graph shows which files directly or indirectly include this file:



Enumerations

- enum [time_unit](#) { [time_unit::millisecond](#), [time_unit::second](#), [time_unit::minute](#) }
is a strongly typed enum class representing time unit

9.61.1 Enumeration Type Documentation

9.61.1.1 time_unit

```
enum time_unit [strong]
```

is a strongly typed enum class representing time unit

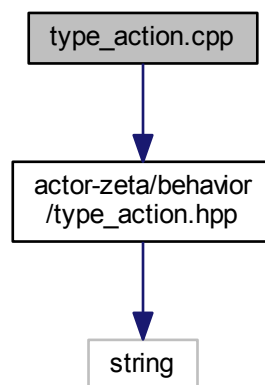
Enumerator

millisecond	is coded as std::int of value 0
second	is coded as std::int of value 1
minute	is coded as std::int of value 2

9.62 type_action.cpp File Reference

```
#include "actor-zeta/behavior/type_action.hpp"
```

Include dependency graph for type_action.cpp:



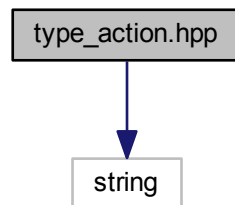
Namespaces

- [actor_zeta](#)
- [actor_zeta::behavior](#)

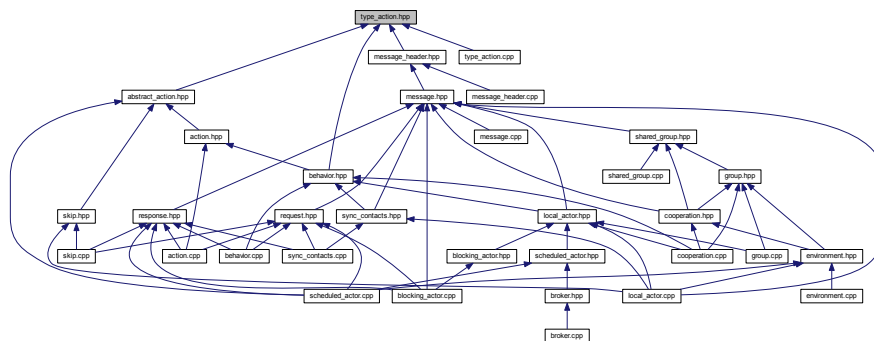
9.63 type_action.hpp File Reference

```
#include <string>
```

Include dependency graph for type_action.hpp:



This graph shows which files directly or indirectly include this file:



Classes

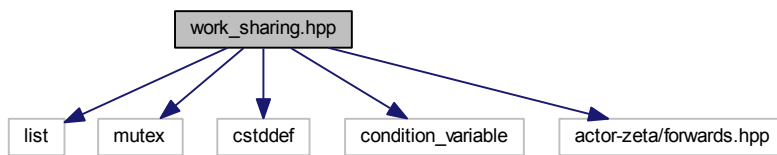
- class [actor_zeta::behavior::type_action](#)
Type of commands used for lifecycle.

Namespaces

- [actor_zeta](#)
- [actor_zeta::behavior](#)

9.64 work_sharing.hpp File Reference

```
#include <list>
#include <mutex>
#include <cstdint>
#include <condition_variable>
#include "actor-zeta/forwards.hpp"
Include dependency graph for work_sharing.hpp:
```



Classes

- class [actor_zeta::executor::work_sharing](#)
Stands for work sharing approach.
- struct [actor_zeta::executor::work_sharing::coordinator_data](#)
This structure is used as coordinator data keeper.
- struct [actor_zeta::executor::work_sharing::worker_data](#)
This structure is used as worker data keeper.

Namespaces

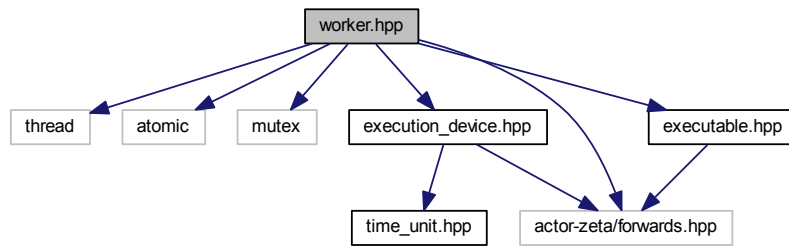
- [actor_zeta](#)
- [actor_zeta::executor](#)

9.65 worker.hpp File Reference

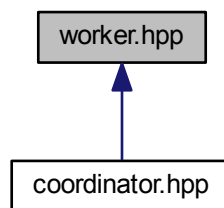
```
#include <thread>
#include <atomic>
#include <mutex>
#include "execution_device.hpp"
#include "executable.hpp"
```

```
#include "actor-zeta/forwards.hpp"
```

Include dependency graph for worker.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class `actor_zeta::executor::worker< Policy >`
A simple worker module class.

Namespaces

- `actor_zeta`
- `actor_zeta::executor`

Index

- ~abstract_action
 - actor_zeta::behavior::abstract_action, [24](#)
- ~abstract_actor
 - actor_zeta::actor::abstract_actor, [26](#)
- ~abstract_coordinator
 - actor_zeta::executor::abstract_coordinator, [28](#)
- ~action
 - actor_zeta::behavior::action, [30](#)
- ~actor
 - actor_zeta::actor::actor, [32](#)
- ~actor_address
 - actor_zeta::actor::actor_address, [35](#)
- ~behavior
 - actor_zeta::behavior::behavior, [37](#)
- ~blocking_actor
 - actor_zeta::actor::blocking_actor, [39](#)
- ~blocking_mail_queue
 - actor_zeta::messaging::blocking_mail_queue, [43](#)
- ~broker
 - actor_zeta::network::broker, [48](#)
- ~cooperation
 - actor_zeta::environment::cooperation, [52](#)
- ~environment
 - actor_zeta::environment::environment, [59](#)
- ~executable
 - actor_zeta::executor::executable, [62](#)
- ~execution_device
 - actor_zeta::executor::execution_device, [64](#)
- ~group
 - actor_zeta::environment::group, [66](#)
- ~intrusive_ptr
 - actor_zeta::intrusive_ptr, [71](#)
- ~local_actor
 - actor_zeta::actor::local_actor, [75](#)
- ~message
 - actor_zeta::messaging::message, [79](#)
- ~message_body
 - actor_zeta::messaging::message_body, [83](#)
- ~message_header
 - actor_zeta::messaging::message_header, [86](#)
- ~multiplexer
 - actor_zeta::network::multiplexer, [89](#)
- ~ref_counted
 - actor_zeta::ref_counted, [92](#)
- ~request
 - actor_zeta::behavior::request, [94](#)
- ~response
 - actor_zeta::behavior::response, [96](#)
- ~shared_group
 - actor_zeta::environment::shared_group, [101](#)
- ~type_action
 - actor_zeta::behavior::type_action, [106](#)
- ~work_sharing
 - actor_zeta::executor::work_sharing, [109](#)
- abstract_action
 - actor_zeta::behavior::abstract_action, [24](#)
- abstract_action.hpp, [117](#)
- abstract_actor
 - actor_zeta::actor::abstract_actor, [26](#)
- abstract_actor.cpp, [118](#)
- abstract_actor.hpp, [119](#)
- abstract_coordinator
 - actor_zeta::executor::abstract_coordinator, [28](#)
- abstract_coordinator.hpp, [120](#)
- act
 - actor_zeta::actor::blocking_actor, [39](#)
- action
 - actor_zeta::behavior::action, [29](#), [30](#)
- action.cpp, [121](#)
- action.hpp, [122](#)
- actor
 - actor_zeta::actor::actor, [31](#), [32](#)
- actor.cpp, [123](#)
- actor.hpp, [123](#)
- actor_address
 - actor_zeta::actor::actor_address, [34](#), [35](#)
- actor_address.cpp, [125](#)
- actor_address.hpp, [126](#)
- actor_zeta, [13](#)
 - const_pointer_cast, [14](#)
 - dynamic_pointer_cast, [14](#)
 - get_pointer, [14](#)
 - intrusive_ptr_add_ref, [14](#)
 - intrusive_ptr_release, [14](#)
 - operator!=, [15](#)
 - operator<, [15](#)
 - operator==, [15](#), [16](#)
 - static_pointer_cast, [16](#)
 - swap, [16](#)
- actor_zeta::actor, [16](#)
- actor_zeta::actor::abstract_actor, [25](#)
 - ~abstract_actor, [26](#)
 - abstract_actor, [26](#)
 - address, [26](#)
 - env, [26](#)
 - send, [27](#)
 - type, [27](#)
- actor_zeta::actor::actor, [31](#)

- ~actor, 32
- actor, 31, 32
- address, 32
- operator bool, 33
- operator!, 33
- operator->, 33
- operator=, 33
- type, 34
- actor_zeta::actor::actor_address, 34
 - ~actor_address, 35
 - actor_address, 34, 35
 - operator bool, 35
 - operator!, 35
 - operator->, 35
 - operator=, 36
- actor_zeta::actor::blocking_actor, 38
 - ~blocking_actor, 39
 - act, 39
 - blocking_actor, 39
 - launch, 40
 - run, 40
- actor_zeta::actor::local_actor, 73
 - ~local_actor, 75
 - attach, 76
 - contacts, 77
 - device, 76
 - has_next_message, 76
 - initialize, 76
 - launch, 76
 - life, 77
 - local_actor, 75
 - mailbox, 77
 - mailbox_type, 75
 - next_message, 77
 - pop_to_cache, 77
 - push_to_cache, 77
- actor_zeta::actor::scheduled_actor, 97
 - attach_to_scheduler, 99
 - detach_from_scheduler, 99
 - launch, 99
 - run, 100
 - scheduled_actor, 99
 - send, 100
- actor_zeta::behavior, 17
 - make_response, 17
- actor_zeta::behavior::abstract_action, 23
 - ~abstract_action, 24
 - abstract_action, 24
 - name, 24
 - operator(), 24
- actor_zeta::behavior::action, 29
 - ~action, 30
 - action, 29, 30
 - operator(), 30
 - operator=, 30
- actor_zeta::behavior::behavior, 36
 - ~behavior, 37
 - behavior, 37
- insert, 37
 - operator=, 37
 - run, 38
- actor_zeta::behavior::request, 93
 - ~request, 94
 - contacts, 94
 - message, 94
 - operator=, 95
 - request, 94
- actor_zeta::behavior::response, 95
 - ~response, 96
 - message, 96
 - operator=, 96, 97
 - receiver, 97
 - response, 96
- actor_zeta::behavior::type_action, 106
 - ~type_action, 106
 - hash, 107
 - operator=, 107
 - operator==, 108
 - to_string, 108
 - type_action, 106, 107
- actor_zeta::contacts, 17
- actor_zeta::contacts::book_contacts, 45
 - all_view, 45
 - book_contacts, 45
 - get, 45
 - get_group, 46
 - operator=, 46
 - put, 46
 - put_in_group, 46
- actor_zeta::contacts::group_contacts, 68
 - get, 68
 - get_all, 68
 - group_contacts, 68
 - operator=, 69
 - put, 69
- actor_zeta::environment, 18
 - send, 18
- actor_zeta::environment::cooperation, 51
 - ~cooperation, 52
 - add, 52
 - add_shared, 52
 - cooperation, 52
 - get, 52
 - operator=, 53
 - send, 53
 - send_all, 53
 - send_current, 53
- actor_zeta::environment::environment, 58
 - ~environment, 59
 - coordinator_, 60
 - environment, 58, 59
 - manager_execution_device, 59
 - manager_group, 59
 - operator=, 59, 60
 - start, 60
- actor_zeta::environment::group, 64

- ~group, 66
- add, 66
- add_shared, 66
- entry_point, 66
- group, 65
- name, 67
- operator=, 67
- send, 67
- send_all, 67
- send_current, 67
- actor_zeta::environment::shared_group, 100
 - ~shared_group, 101
 - add, 102
 - operator=, 102
 - send_all, 102
 - send_current, 102
 - shared_group, 101
- actor_zeta::executor, 18
- actor_zeta::executor::abstract_coordinator, 27
 - ~abstract_coordinator, 28
 - abstract_coordinator, 28
 - max_throughput, 28
 - num_workers, 28
 - start, 28
 - submit, 29
- actor_zeta::executor::coordinator
 - coordinator, 55
 - data, 56
 - job_ptr, 55
 - policy_data, 55
 - start, 56
 - submit, 56
 - worker_by_id, 56
 - worker_type, 55
- actor_zeta::executor::coordinator< Policy >, 54
- actor_zeta::executor::executable, 60
 - ~executable, 62
 - attach_to_scheduler, 62
 - detach_from_scheduler, 62
 - executable_result, 61
 - run, 62
- actor_zeta::executor::execution_device, 63
 - ~execution_device, 64
 - execution_device, 63, 64
 - put_execute_latest, 64
- actor_zeta::executor::work_sharing, 108
 - ~work_sharing, 109
 - central_enqueue, 109
 - dequeue, 110
 - enqueue, 110
 - external_enqueue, 110
 - foreach_central_resumable, 110
 - internal_enqueue, 110
 - queue_type, 109
 - resume_job_later, 110
- actor_zeta::executor::work_sharing::coordinator_data, 56
 - coordinator_data, 57
 - cv, 57
 - lock, 57
 - queue, 57
- actor_zeta::executor::work_sharing::worker_data, 115
 - worker_data, 115
- actor_zeta::executor::worker
 - coordinator_ptr, 112
 - data, 113
 - external_enqueue, 113
 - id, 113
 - job_ptr, 112
 - max_throughput, 114
 - operator=, 114
 - parent, 114
 - policy_data, 112
 - put_execute_latest, 114
 - start, 114
 - worker, 113
- actor_zeta::executor::worker< Policy >, 111
- actor_zeta::intrusive_ptr
 - ~intrusive_ptr, 71
 - detach, 71
 - downcast, 71
 - get, 71
 - intrusive_ptr, 70, 71
 - operator bool, 72
 - operator!, 72
 - operator*, 72
 - operator->, 72
 - operator=, 72
 - pointer, 70
 - reference, 70
 - release, 72
 - reset, 73
 - swap, 73
 - upcast, 73
- actor_zeta::intrusive_ptr< T >, 69
- actor_zeta::messaging, 19
 - enqueue_result, 19
 - make_message, 20
 - message_priority, 20
- actor_zeta::messaging::blocking_mail_queue
 - ~blocking_mail_queue, 43
 - blocking_mail_queue, 42, 43
 - cache_type, 41
 - const_pointer, 41
 - const_reference, 41
 - get, 43
 - high_priority_cache, 43
 - lock_guard, 42
 - low_priority_cache, 43
 - normal_priority_cache, 44
 - operator=, 44
 - pointer, 42
 - put, 44
 - queue_base_type, 42
 - reference, 42
 - sync, 44

- unique_lock, 42
- actor_zeta::messaging::blocking_mail_queue< T >, 40
- actor_zeta::messaging::message, 78
 - ~message, 79
 - clone, 81
 - get, 81
 - is_callback, 81
 - message, 79, 80
 - operator=, 81
 - priority, 81
 - return_address, 82
 - type, 82
- actor_zeta::messaging::message_body, 82
 - ~message_body, 83
 - clear, 84
 - empty, 84
 - get, 84
 - message_body, 83
 - operator=, 84
 - swap, 84
- actor_zeta::messaging::message_header, 85
 - ~message_header, 86
 - is_callback, 88
 - message_header, 85–87
 - operator=, 88
 - priorities, 88
 - return_address, 88
 - type, 88
- actor_zeta::network, 21
 - shared_mux_ptr, 21
 - type_connect, 21
 - unique_mux_ptr, 21
- actor_zeta::network::broker, 47
 - ~broker, 48
 - broker, 48
 - initialize, 48
 - mux_ptr, 48
- actor_zeta::network::connection_identifying, 49
 - connection_identifying, 49, 50
 - ip, 50
 - operator=, 50
 - operator==, 50
 - port, 51
 - to_string, 51
- actor_zeta::network::mux, 89
 - ~mux, 89
 - close, 89
 - new_tcp_connection, 89
 - new_tcp_listener, 89
 - registration_broker_address, 90
 - run, 90
 - write, 90
- actor_zeta::ref_counted, 90
 - ~ref_counted, 92
 - deref, 92
 - get_reference_count, 92
 - operator=, 92
 - rc_, 93
 - ref, 92
 - ref_counted, 92
 - unique, 93
- actor_zeta::skip, 103
 - operator(), 104
 - skip, 104
- actor_zeta::sync_contacts, 104
 - operator(), 105
 - sync_contacts, 105
- add
 - actor_zeta::environment::cooperation, 52
 - actor_zeta::environment::group, 66
 - actor_zeta::environment::shared_group, 102
- add_shared
 - actor_zeta::environment::cooperation, 52
 - actor_zeta::environment::group, 66
- address
 - actor_zeta::actor::abstract_actor, 26
 - actor_zeta::actor::actor, 32
- all_view
 - actor_zeta::contacts::book_contacts, 45
- attach
 - actor_zeta::actor::local_actor, 76
- attach_to_scheduler
 - actor_zeta::actor::scheduled_actor, 99
 - actor_zeta::executor::executable, 62
- behavior
 - actor_zeta::behavior::behavior, 37
- behavior.cpp, 127
- behavior.hpp, 127
- blocking_actor
 - actor_zeta::actor::blocking_actor, 39
- blocking_actor.cpp, 129
- blocking_actor.hpp, 129
- blocking_mail_queue
 - actor_zeta::messaging::blocking_mail_queue, 42, 43
- blocking_mail_queue.hpp, 130
- book_contacts
 - actor_zeta::contacts::book_contacts, 45
- book_contacts.cpp, 131
- book_contacts.hpp, 132
- broker
 - actor_zeta::network::broker, 48
- broker.cpp, 133
- broker.hpp, 134
- cache_type
 - actor_zeta::messaging::blocking_mail_queue, 41
- central_enqueue
 - actor_zeta::executor::work_sharing, 109
- clear
 - actor_zeta::messaging::message_body, 84
- clone
 - actor_zeta::messaging::message, 81
- close
 - actor_zeta::network::mux, 89
- connection_identifying

- actor_zeta::network::connection_identifying, 49, 50
- connection_identifying.cpp, 135
- connection_identifying.hpp, 136
- const_pointer
 - actor_zeta::messaging::blocking_mail_queue, 41
- const_pointer_cast
 - actor_zeta, 14
- const_reference
 - actor_zeta::messaging::blocking_mail_queue, 41
- contacts
 - actor_zeta::actor::local_actor, 77
 - actor_zeta::behavior::request, 94
- cooperation
 - actor_zeta::environment::cooperation, 52
- cooperation.cpp, 137
- cooperation.hpp, 137
- coordinator
 - actor_zeta::executor::coordinator, 55
- coordinator.hpp, 139
- coordinator_
 - actor_zeta::environment::environment, 60
- coordinator_data
 - actor_zeta::executor::work_sharing::coordinator←_data, 57
- coordinator_ptr
 - actor_zeta::executor::worker, 112
- cv
 - actor_zeta::executor::work_sharing::coordinator←_data, 57
- data
 - actor_zeta::executor::coordinator, 56
 - actor_zeta::executor::worker, 113
- dequeue
 - actor_zeta::executor::work_sharing, 110
- deref
 - actor_zeta::ref_counted, 92
- description.h, 139
- detach
 - actor_zeta::intrusive_ptr, 71
- detach_from_scheduler
 - actor_zeta::actor::scheduled_actor, 99
 - actor_zeta::executor::executable, 62
- device
 - actor_zeta::actor::local_actor, 76
- downcast
 - actor_zeta::intrusive_ptr, 71
- dynamic_pointer_cast
 - actor_zeta, 14
- empty
 - actor_zeta::messaging::message_body, 84
- enqueue
 - actor_zeta::executor::work_sharing, 110
- enqueue_result
 - actor_zeta::messaging, 19
- entry_point
 - actor_zeta::environment::group, 66
- env
 - actor_zeta::actor::abstract_actor, 26
- environment
 - actor_zeta::environment::environment, 58, 59
- environment.cpp, 139
- environment.hpp, 140
- executable.hpp, 141
- executable_result
 - actor_zeta::executor::executable, 61
- execution_device
 - actor_zeta::executor::execution_device, 63, 64
- execution_device.hpp, 142
- external_enqueue
 - actor_zeta::executor::work_sharing, 110
 - actor_zeta::executor::worker, 113
- foreach_central_resumable
 - actor_zeta::executor::work_sharing, 110
- forwards.hpp, 143, 144
- get
 - actor_zeta::contacts::book_contacts, 45
 - actor_zeta::contacts::group_contacts, 68
 - actor_zeta::environment::cooperation, 52
 - actor_zeta::intrusive_ptr, 71
 - actor_zeta::messaging::blocking_mail_queue, 43
 - actor_zeta::messaging::message, 81
 - actor_zeta::messaging::message_body, 84
- get_all
 - actor_zeta::contacts::group_contacts, 68
- get_group
 - actor_zeta::contacts::book_contacts, 46
- get_pointer
 - actor_zeta, 14
- get_reference_count
 - actor_zeta::ref_counted, 92
- group
 - actor_zeta::environment::group, 65
- group.cpp, 144
- group.hpp, 145
- group_contacts
 - actor_zeta::contacts::group_contacts, 68
- group_contacts.cpp, 146
- group_contacts.hpp, 147
- has_next_message
 - actor_zeta::actor::local_actor, 76
- hash
 - actor_zeta::behavior::type_action, 107
- high_priority_cache
 - actor_zeta::messaging::blocking_mail_queue, 43
- id
 - actor_zeta::executor::worker, 113
- initialize
 - actor_zeta::actor::local_actor, 76
 - actor_zeta::network::broker, 48
- insert
 - actor_zeta::behavior::behavior, 37
- internal_enqueue

- actor_zeta::executor::work_sharing, 110
- intrusive_ptr
 - actor_zeta::intrusive_ptr, 70, 71
- intrusive_ptr.hpp, 148
- intrusive_ptr_add_ref
 - actor_zeta, 14
- intrusive_ptr_release
 - actor_zeta, 14
- ip
 - actor_zeta::network::connection_identifying, 50
- is_callback
 - actor_zeta::messaging::message, 81
 - actor_zeta::messaging::message_header, 88
- job_ptr
 - actor_zeta::executor::coordinator, 55
 - actor_zeta::executor::worker, 112
- launch
 - actor_zeta::actor::blocking_actor, 40
 - actor_zeta::actor::local_actor, 76
 - actor_zeta::actor::scheduled_actor, 99
- life
 - actor_zeta::actor::local_actor, 77
- local_actor
 - actor_zeta::actor::local_actor, 75
- local_actor.cpp, 150
- local_actor.hpp, 150
- lock
 - actor_zeta::executor::work_sharing::coordinator←_data, 57
- lock_guard
 - actor_zeta::messaging::blocking_mail_queue, 42
- low_priority_cache
 - actor_zeta::messaging::blocking_mail_queue, 43
- mailbox
 - actor_zeta::actor::local_actor, 77
- mailbox_type
 - actor_zeta::actor::local_actor, 75
- make_message
 - actor_zeta::messaging, 20
- make_response
 - actor_zeta::behavior, 17
- manager_execution_device
 - actor_zeta::environment::environment, 59
- manager_group
 - actor_zeta::environment::environment, 59
- max_throughput
 - actor_zeta::executor::abstract_coordinator, 28
 - actor_zeta::executor::worker, 114
- message
 - actor_zeta::behavior::request, 94
 - actor_zeta::behavior::response, 96
 - actor_zeta::messaging::message, 79, 80
- message.cpp, 151
- message.hpp, 152
- message_body
 - actor_zeta::messaging::message_body, 83
- message_body.hpp, 153
 - swap, 154
- message_header
 - actor_zeta::messaging::message_header, 85–87
- message_header.cpp, 154
- message_header.hpp, 155
- message_priority
 - actor_zeta::messaging, 20
- message_priority.hpp, 156
- multiplexer.hpp, 156
- multiplexer_
 - actor_zeta::network::broker, 48
- name
 - actor_zeta::behavior::abstract_action, 24
 - actor_zeta::environment::group, 67
- new_tcp_connection
 - actor_zeta::network::multiplexer, 89
- new_tcp_listener
 - actor_zeta::network::multiplexer, 89
- next_message
 - actor_zeta::actor::local_actor, 77
- normal_priority_cache
 - actor_zeta::messaging::blocking_mail_queue, 44
- num_workers
 - actor_zeta::executor::abstract_coordinator, 28
- operator bool
 - actor_zeta::actor::actor, 33
 - actor_zeta::actor::actor_address, 35
 - actor_zeta::intrusive_ptr, 72
- operator!
 - actor_zeta::actor::actor, 33
 - actor_zeta::actor::actor_address, 35
 - actor_zeta::intrusive_ptr, 72
- operator!=
 - actor_zeta, 15
- operator<
 - actor_zeta, 15
- operator*
 - actor_zeta::intrusive_ptr, 72
- operator()
 - actor_zeta::behavior::abstract_action, 24
 - actor_zeta::behavior::action, 30
 - actor_zeta::skip, 104
 - actor_zeta::sync_contacts, 105
- operator->
 - actor_zeta::actor::actor, 33
 - actor_zeta::actor::actor_address, 35
 - actor_zeta::intrusive_ptr, 72
- operator=
 - actor_zeta::actor::actor, 33
 - actor_zeta::actor::actor_address, 36
 - actor_zeta::behavior::action, 30
 - actor_zeta::behavior::behavior, 37
 - actor_zeta::behavior::request, 95
 - actor_zeta::behavior::response, 96, 97
 - actor_zeta::behavior::type_action, 107
 - actor_zeta::contacts::book_contacts, 46

- actor_zeta::contacts::group_contacts, 69
- actor_zeta::environment::cooperation, 53
- actor_zeta::environment::environment, 59, 60
- actor_zeta::environment::group, 67
- actor_zeta::environment::shared_group, 102
- actor_zeta::executor::worker, 114
- actor_zeta::intrusive_ptr, 72
- actor_zeta::messaging::blocking_mail_queue, 44
- actor_zeta::messaging::message, 81
- actor_zeta::messaging::message_body, 84
- actor_zeta::messaging::message_header, 88
- actor_zeta::network::connection_identifying, 50
- actor_zeta::ref_counted, 92
- operator==
 - actor_zeta, 15, 16
 - actor_zeta::behavior::type_action, 108
 - actor_zeta::network::connection_identifying, 50
- parent
 - actor_zeta::executor::worker, 114
- pointer
 - actor_zeta::intrusive_ptr, 70
 - actor_zeta::messaging::blocking_mail_queue, 42
- policy_data
 - actor_zeta::executor::coordinator, 55
 - actor_zeta::executor::worker, 112
- pop_to_cache
 - actor_zeta::actor::local_actor, 77
- port
 - actor_zeta::network::connection_identifying, 51
- priorities
 - actor_zeta::messaging::message_header, 88
- priority
 - actor_zeta::messaging::message, 81
- push_to_cache
 - actor_zeta::actor::local_actor, 77
- put
 - actor_zeta::contacts::book_contacts, 46
 - actor_zeta::contacts::group_contacts, 69
 - actor_zeta::messaging::blocking_mail_queue, 44
- put_execute_latest
 - actor_zeta::executor::execution_device, 64
 - actor_zeta::executor::worker, 114
- put_in_group
 - actor_zeta::contacts::book_contacts, 46
- queue
 - actor_zeta::executor::work_sharing::coordinator↔_data, 57
- queue_base_type
 - actor_zeta::messaging::blocking_mail_queue, 42
- queue_type
 - actor_zeta::executor::work_sharing, 109
- README.md, 158
- rc_
 - actor_zeta::ref_counted, 93
- receiver
 - actor_zeta::behavior::response, 97
- ref
 - actor_zeta::ref_counted, 92
- ref_counted
 - actor_zeta::ref_counted, 92
- ref_counted.cpp, 158
- ref_counted.hpp, 159
- reference
 - actor_zeta::intrusive_ptr, 70
 - actor_zeta::messaging::blocking_mail_queue, 42
- registration_broker_address
 - actor_zeta::network::multiplexer, 90
- release
 - actor_zeta::intrusive_ptr, 72
- request
 - actor_zeta::behavior::request, 94
- request.hpp, 160
- reset
 - actor_zeta::intrusive_ptr, 73
- response
 - actor_zeta::behavior::response, 96
- response.hpp, 160
- resume_job_later
 - actor_zeta::executor::work_sharing, 110
- return_address
 - actor_zeta::messaging::message, 82
 - actor_zeta::messaging::message_header, 88
- run
 - actor_zeta::actor::blocking_actor, 40
 - actor_zeta::actor::scheduled_actor, 100
 - actor_zeta::behavior::behavior, 38
 - actor_zeta::executor::executable, 62
 - actor_zeta::network::multiplexer, 90
- scheduled_actor
 - actor_zeta::actor::scheduled_actor, 99
- scheduled_actor.cpp, 162
- scheduled_actor.hpp, 162
- send
 - actor_zeta::actor::abstract_actor, 27
 - actor_zeta::actor::scheduled_actor, 100
 - actor_zeta::environment, 18
 - actor_zeta::environment::cooperation, 53
 - actor_zeta::environment::group, 67
- send_all
 - actor_zeta::environment::cooperation, 53
 - actor_zeta::environment::group, 67
 - actor_zeta::environment::shared_group, 102
- send_current
 - actor_zeta::environment::cooperation, 53
 - actor_zeta::environment::group, 67
 - actor_zeta::environment::shared_group, 102
- shared_group
 - actor_zeta::environment::shared_group, 101
- shared_group.cpp, 163
- shared_group.hpp, 164
- shared_multiplexer_ptr
 - actor_zeta::network, 21
- skip
 - actor_zeta::skip, 104

- skip.cpp, [165](#)
- skip.hpp, [165](#)
- start
 - actor_zeta::environment::environment, [60](#)
 - actor_zeta::executor::abstract_coordinator, [28](#)
 - actor_zeta::executor::coordinator, [56](#)
 - actor_zeta::executor::worker, [114](#)
- static_pointer_cast
 - actor_zeta, [16](#)
- submit
 - actor_zeta::executor::abstract_coordinator, [29](#)
 - actor_zeta::executor::coordinator, [56](#)
- swap
 - actor_zeta, [16](#)
 - actor_zeta::intrusive_ptr, [73](#)
 - actor_zeta::messaging::message_body, [84](#)
 - message_body.hpp, [154](#)
- sync
 - actor_zeta::messaging::blocking_mail_queue, [44](#)
- sync_contacts
 - actor_zeta::sync_contacts, [105](#)
- sync_contacts.cpp, [167](#)
- sync_contacts.hpp, [167](#)
- time_unit
 - time_unit.hpp, [169](#)
- time_unit.hpp, [168](#)
- time_unit, [169](#)
- to_string
 - actor_zeta::behavior::type_action, [108](#)
 - actor_zeta::network::connection_identifying, [51](#)
- type
 - actor_zeta::actor::abstract_actor, [27](#)
 - actor_zeta::actor::actor, [34](#)
 - actor_zeta::messaging::message, [82](#)
 - actor_zeta::messaging::message_header, [88](#)
- type_action
 - actor_zeta::behavior::type_action, [106](#), [107](#)
- type_action.cpp, [169](#)
- type_action.hpp, [170](#)
- type_connect
 - actor_zeta::network, [21](#)
- unique
 - actor_zeta::ref_counted, [93](#)
- unique_lock
 - actor_zeta::messaging::blocking_mail_queue, [42](#)
- unique_muxplexer_ptr
 - actor_zeta::network, [21](#)
- upcast
 - actor_zeta::intrusive_ptr, [73](#)
- work_sharing.hpp, [171](#)
- worker
 - actor_zeta::executor::worker, [113](#)
- worker.hpp, [171](#)
- worker_by_id
 - actor_zeta::executor::coordinator, [56](#)
- worker_data
 - actor_zeta::executor::work_sharing::worker_data, [115](#)
- worker_type
 - actor_zeta::executor::coordinator, [55](#)
- write
 - actor_zeta::network::muxplexer, [90](#)