

TrickHLA

3.0: Git#2dc5c1d

Generated by Doxygen 1.8.16



---

<b>1 TrickHLA: An IEEE 1516 High Level Architecture (HLA) Simulation Interoperability Standard Implementation for Trick Base Simulations</b>	1
<b>2 Namespace Index</b>	<b>3</b>
2.1 Namespace List . . . . .	3
<b>3 Hierarchical Index</b>	<b>5</b>
3.1 Class Hierarchy . . . . .	5
<b>4 Data Structure Index</b>	<b>7</b>
4.1 Data Structures . . . . .	7
<b>5 File Index</b>	<b>9</b>
5.1 File List . . . . .	9
<b>6 Namespace Documentation</b>	<b>17</b>
6.1 DIS Namespace Reference . . . . .	17
6.1.1 Enumeration Type Documentation . . . . .	18
6.1.1.1 ExecutionModeEnum . . . . .	18
6.1.1.2 MTREnum . . . . .	19
6.1.1.3 PausePointStateEnum . . . . .	19
6.1.2 Function Documentation . . . . .	20
6.1.2.1 execution_mode_enum_to_int16() . . . . .	20
6.1.2.2 execution_mode_enum_to_string() . . . . .	20
6.1.2.3 execution_mode_int16_to_enum() . . . . .	20
6.1.2.4 from_execution_control_enum() . . . . .	21
6.1.2.5 mtr_enum_to_int16() . . . . .	21
6.1.2.6 mtr_enum_to_string() . . . . .	22
6.1.2.7 mtr_int16_to_enum() . . . . .	22
6.1.2.8 pause_pnt_state_enum_to_int16() . . . . .	22
6.1.2.9 pause_pnt_state_enum_to_string() . . . . .	23
6.1.2.10 pause_pnt_state_int16_to_enum() . . . . .	23
6.1.2.11 to_execution_control_enum() . . . . .	23
6.1.3 Variable Documentation . . . . .	24
6.1.3.1 INITIALIZE_SYNC_POINT . . . . .	24
6.1.3.2 STARTUP_FREEZE_SYNC_POINT . . . . .	24
6.1.3.3 STARTUP_SYNC_POINT . . . . .	24
6.2 DSES Namespace Reference . . . . .	24
6.2.1 Enumeration Type Documentation . . . . .	25
6.2.1.1 ExecutionModeEnum . . . . .	25
6.2.1.2 MTREnum . . . . .	26

---

---

6.2.2 Function Documentation . . . . .	26
6.2.2.1 execution_mode_enum_to_int16() . . . . .	26
6.2.2.2 execution_mode_enum_to_string() . . . . .	26
6.2.2.3 execution_mode_int16_to_enum() . . . . .	27
6.2.2.4 from_execution_contorl_enum() . . . . .	27
6.2.2.5 mtr_enum_to_int16() . . . . .	27
6.2.2.6 mtr_enum_to_string() . . . . .	28
6.2.2.7 mtr_int16_to_enum() . . . . .	28
6.2.2.8 to_execution_control_enum() . . . . .	28
6.2.3 Variable Documentation . . . . .	29
6.2.3.1 INITIALIZE_SYNC_POINT . . . . .	29
6.2.3.2 SIM_CONFIG_SYNC_POINT . . . . .	29
6.2.3.3 STARTUP_SYNC_POINT . . . . .	29
6.3 IMSim Namespace Reference . . . . .	29
6.3.1 Typedef Documentation . . . . .	31
6.3.1.1 FreezeTimeSet . . . . .	31
6.3.2 Enumeration Type Documentation . . . . .	31
6.3.2.1 ExecutionModeEnum . . . . .	31
6.3.2.2 MTREnum . . . . .	31
6.3.2.3 PausePointStateEnum . . . . .	32
6.3.3 Function Documentation . . . . .	32
6.3.3.1 execution_mode_enum_to_int16() . . . . .	32
6.3.3.2 execution_mode_enum_to_string() . . . . .	32
6.3.3.3 execution_mode_int16_to_enum() . . . . .	33
6.3.3.4 from_execution_contorl_enum() . . . . .	33
6.3.3.5 mtr_enum_to_int16() . . . . .	34
6.3.3.6 mtr_enum_to_string() . . . . .	34
6.3.3.7 mtr_int16_to_enum() . . . . .	34
6.3.3.8 pause_pnt_state_enum_to_int16() . . . . .	35
6.3.3.9 pause_pnt_state_enum_to_string() . . . . .	35
6.3.3.10 pause_pnt_state_int16_to_enum() . . . . .	35
6.3.3.11 to_execution_control_enum() . . . . .	36
6.3.4 Variable Documentation . . . . .	36
6.3.4.1 FEDRUN_SYNC_POINT . . . . .	36
6.3.4.2 FEDSAVE_SYNC_POINT . . . . .	36
6.3.4.3 INIT_COMPLETE_SYNC_POINT . . . . .	36
6.3.4.4 INITIALIZE_SYNC_POINT . . . . .	37
6.3.4.5 SIM_CONFIG_SYNC_POINT . . . . .	37
6.3.4.6 STARTUP_FREEZE_SYNC_POINT . . . . .	37

6.3.4.7 STARTUP_SYNC_POINT . . . . .	37
6.4 SpaceFOM Namespace Reference . . . . .	37
6.4.1 Enumeration Type Documentation . . . . .	38
6.4.1.1 ExecutionModeEnum . . . . .	38
6.4.1.2 MTREnum . . . . .	39
6.4.2 Function Documentation . . . . .	39
6.4.2.1 execution_mode_enum_to_int16() . . . . .	39
6.4.2.2 execution_mode_enum_to_string() . . . . .	39
6.4.2.3 execution_mode_int16_to_enum() . . . . .	40
6.4.2.4 from_execution_control_enum() . . . . .	40
6.4.2.5 from_mode_transition_enum() . . . . .	41
6.4.2.6 mtr_enum_to_int16() . . . . .	41
6.4.2.7 mtr_enum_to_string() . . . . .	41
6.4.2.8 mtr_int16_to_enum() . . . . .	42
6.4.2.9 to_execution_control_enum() . . . . .	42
6.4.3 Variable Documentation . . . . .	43
6.4.3.1 INIT_COMPLETED_SYNC_POINT . . . . .	43
6.4.3.2 INIT_STARTED_SYNC_POINT . . . . .	43
6.4.3.3 OBJECTS_DISCOVERED_SYNC_POINT . . . . .	43
6.4.3.4 ROOT_FRAME_DISCOVERED_SYNC_POINT . . . . .	43
6.5 TrickHLA Namespace Reference . . . . .	43
6.5.1 Typedef Documentation . . . . .	47
6.5.1.1 AttributeMap . . . . .	47
6.5.1.2 AttributeOwnershipMap . . . . .	47
6.5.1.3 HLAAttributeMapQueue . . . . .	47
6.5.1.4 ObjectInstanceMap . . . . .	47
6.5.1.5 THLAAttributeMap . . . . .	47
6.5.1.6 TrickHLAObjInstanceNameMap . . . . .	47
6.5.1.7 TrickRTIAmbPtr . . . . .	47
6.5.1.8 VectorOfStrings . . . . .	48
6.5.1.9 VectorOfWstrings . . . . .	48
6.5.2 Enumeration Type Documentation . . . . .	48
6.5.2.1 DataUpdateEnum . . . . .	48
6.5.2.2 DebugLevelEnum . . . . .	48
6.5.2.3 DebugSourceEnum . . . . .	49
6.5.2.4 EncodingEnum . . . . .	49
6.5.2.5 ExecutionControlEnum . . . . .	50
6.5.2.6 FederateJoinEnum . . . . .	50
6.5.2.7 LagCompensationEnum . . . . .	50

6.5.2.8 ManagerTypeOfInteractionEnum . . . . .	51
6.5.2.9 ModeTransitionEnum . . . . .	51
6.5.2.10 SyncPntStateEnum . . . . .	51
6.5.2.11 THLASaveRestoreProcEnum . . . . .	52
6.5.2.12 TransportationEnum . . . . .	52
6.5.3 Function Documentation . . . . .	53
6.5.3.1 execution_control_enum_to_int16() . . . . .	53
6.5.3.2 execution_control_enum_to_string() . . . . .	53
6.5.3.3 execution_control_int16_to_enum() . . . . .	53
6.5.3.4 mode_transition_enum_to_int16() . . . . .	54
6.5.3.5 mode_transition_enum_to_string() . . . . .	54
6.5.3.6 mode_transition_int16_to_enum() . . . . .	54
6.5.3.7 sync_pnt_state_enum_to_int16() . . . . .	55
6.5.3.8 sync_pnt_state_enum_to_string() . . . . .	55
6.5.3.9 sync_pnt_state_int16_to_enum() . . . . .	56
6.5.4 Variable Documentation . . . . .	56
6.5.4.1 def_scenario_timeline . . . . .	56
6.5.4.2 def_sim_timeline . . . . .	56
6.5.4.3 MAX_LOGICAL_TIME_SECONDS . . . . .	56
6.5.4.4 MAX_VALUE_IN_MICROS . . . . .	56
6.5.4.5 MICROS_MULTIPLIER . . . . .	57
6.6 TrickHLAModel Namespace Reference . . . . .	57
<b>7 Data Structure Documentation</b> . . . . .	<b>59</b>
7.1 TrickHLA::Attribute Class Reference . . . . .	59
7.1.1 Detailed Description . . . . .	64
7.1.2 Constructor & Destructor Documentation . . . . .	64
7.1.2.1 Attribute() [1/2] . . . . .	64
7.1.2.2 ~Attribute() . . . . .	64
7.1.2.3 Attribute() [2/2] . . . . .	65
7.1.3 Member Function Documentation . . . . .	65
7.1.3.1 byteswap_buffer_copy() . . . . .	65
7.1.3.2 calculate_number_of_items() . . . . .	65
7.1.3.3 calculate_size_and_number_of_items() . . . . .	66
7.1.3.4 calculate_static_number_of_items() . . . . .	66
7.1.3.5 check_data_cycle_ready() . . . . .	66
7.1.3.6 decode_boolean_from_buffer() . . . . .	66
7.1.3.7 decode_logical_time() . . . . .	67
7.1.3.8 decode_opaque_data_from_buffer() . . . . .	67

---

7.1.3.9 decode_string_from_buffer()	67
7.1.3.10 determine_cycle_ratio()	67
7.1.3.11 encode_boolean_to_buffer()	67
7.1.3.12 encode_logical_time()	68
7.1.3.13 encode_opaque_data_to_buffer()	68
7.1.3.14 encode_string_to_buffer()	68
7.1.3.15 ensure_buffer_capacity()	68
7.1.3.16 extract_data()	68
7.1.3.17 get_attribute_handle()	69
7.1.3.18 get_attribute_size()	69
7.1.3.19 get_attribute_value()	69
7.1.3.20 get_conditional()	70
7.1.3.21 get_configuration()	70
7.1.3.22 get_FOM_name()	70
7.1.3.23 get_preferred_order()	70
7.1.3.24 get_ref2_attributes()	71
7.1.3.25 get_rti_encoding()	71
7.1.3.26 get_sim_variable_address()	71
7.1.3.27 get_trick_name()	71
7.1.3.28 has_conditional()	71
7.1.3.29 initialize()	72
7.1.3.30 is_byteswap()	72
7.1.3.31 is_changed()	72
7.1.3.32 is_data_cycle_ready()	73
7.1.3.33 is_divest_requested()	73
7.1.3.34 is_locally_owned()	73
7.1.3.35 is_publish()	73
7.1.3.36 is_pull_requested()	74
7.1.3.37 is_push_requested()	74
7.1.3.38 is_received()	74
7.1.3.39 is_remotely_owned()	74
7.1.3.40 is_static_in_size()	74
7.1.3.41 is_subscribe()	75
7.1.3.42 is_supported_attribute_type()	75
7.1.3.43 is_update_requested()	75
7.1.3.44 mark_changed()	76
7.1.3.45 mark_locally_owned()	76
7.1.3.46 mark_remotely_owned()	76
7.1.3.47 mark_unchanged()	76

7.1.3.48 operator=( ) . . . . .	76
7.1.3.49 pack_attribute_buffer( ) . . . . .	76
7.1.3.50 print_buffer( ) . . . . .	77
7.1.3.51 set_attribute_handle( ) . . . . .	77
7.1.3.52 set_configuration( ) . . . . .	77
7.1.3.53 set_debug_level( ) . . . . .	77
7.1.3.54 set_divest_requested( ) . . . . .	78
7.1.3.55 set_encoding( ) . . . . .	78
7.1.3.56 set_preferred_order( ) . . . . .	78
7.1.3.57 set_publish( ) . . . . .	78
7.1.3.58 set_pull_requested( ) . . . . .	79
7.1.3.59 set_push_requested( ) . . . . .	79
7.1.3.60 set_subscribe( ) . . . . .	79
7.1.3.61 set_update_requested( ) . . . . .	79
7.1.3.62 unmark_locally_owned( ) . . . . .	80
7.1.3.63 unpack_attribute_buffer( ) . . . . .	80
7.1.4 Friends And Related Function Documentation . . . . .	80
7.1.4.1 init_attrTrickHLA_Attribute . . . . .	80
7.1.4.2 InputProcessor . . . . .	80
7.1.5 Field Documentation . . . . .	80
7.1.5.1 attr_handle . . . . .	80
7.1.5.2 buffer . . . . .	81
7.1.5.3 buffer_capacity . . . . .	81
7.1.5.4 byteswap . . . . .	81
7.1.5.5 conditional . . . . .	81
7.1.5.6 config . . . . .	81
7.1.5.7 cycle_cnt . . . . .	81
7.1.5.8 cycle_ratio . . . . .	82
7.1.5.9 cycle_time . . . . .	82
7.1.5.10 debug_handler . . . . .	82
7.1.5.11 divest_requested . . . . .	82
7.1.5.12 FOM_name . . . . .	82
7.1.5.13 HLAtrue . . . . .	83
7.1.5.14 initialized . . . . .	83
7.1.5.15 locally_owned . . . . .	83
7.1.5.16 num_items . . . . .	83
7.1.5.17 preferred_order . . . . .	83
7.1.5.18 publish . . . . .	84
7.1.5.19 pull_requested . . . . .	84

---

7.1.5.20 push_requested . . . . .	84
7.1.5.21 ref2 . . . . .	84
7.1.5.22 rti_encoding . . . . .	84
7.1.5.23 size . . . . .	85
7.1.5.24 size_is_static . . . . .	85
7.1.5.25 subscribe . . . . .	85
7.1.5.26 trick_name . . . . .	85
7.1.5.27 update_requested . . . . .	85
7.1.5.28 value_changed . . . . .	86
7.2 TrickHLA::BasicClock Class Reference . . . . .	86
7.2.1 Detailed Description . . . . .	86
7.2.2 Constructor & Destructor Documentation . . . . .	86
7.2.2.1 BasicClock() . . . . .	86
7.2.2.2 ~BasicClock() . . . . .	87
7.2.3 Member Function Documentation . . . . .	87
7.2.3.1 get_time() . . . . .	87
7.2.4 Friends And Related Function Documentation . . . . .	87
7.2.4.1 init_attrTrickHLA__BasicClock . . . . .	87
7.2.4.2 InputProcessor . . . . .	87
7.2.5 Field Documentation . . . . .	87
7.2.5.1 first_pass . . . . .	87
7.3 TrickHLA::Conditional Class Reference . . . . .	87
7.3.1 Detailed Description . . . . .	88
7.3.2 Constructor & Destructor Documentation . . . . .	88
7.3.2.1 Conditional() . . . . .	88
7.3.2.2 ~Conditional() . . . . .	88
7.3.3 Member Function Documentation . . . . .	89
7.3.3.1 should_send() . . . . .	89
7.3.4 Friends And Related Function Documentation . . . . .	89
7.3.4.1 init_attrTrickHLA__Conditional . . . . .	89
7.3.4.2 InputProcessor . . . . .	89
7.4 TrickHLA::CTETimelineBase Class Reference . . . . .	89
7.4.1 Detailed Description . . . . .	91
7.4.2 Constructor & Destructor Documentation . . . . .	91
7.4.2.1 CTETimelineBase() [1/2] . . . . .	91
7.4.2.2 ~CTETimelineBase() . . . . .	91
7.4.2.3 CTETimelineBase() [2/2] . . . . .	91
7.4.3 Member Function Documentation . . . . .	91
7.4.3.1 clock_init() . . . . .	92

---

7.4.3.2 <code>clock_stop()</code> . . . . .	92
7.4.3.3 <code>get_clock_ID()</code> . . . . .	92
7.4.3.4 <code>get_time()</code> . . . . .	92
7.4.3.5 <code>operator=()</code> . . . . .	93
7.4.3.6 <code>set_clock_ID()</code> . . . . .	93
7.4.3.7 <code>wall_clock_time()</code> . . . . .	93
7.4.4 Friends And Related Function Documentation . . . . .	93
7.4.4.1 <code>init_attrTrickHLA__CTETimelineBase</code> . . . . .	93
7.4.4.2 <code>InputProcessor</code> . . . . .	93
7.4.5 Field Documentation . . . . .	93
7.4.5.1 <code>clk_id</code> . . . . .	93
7.5 TrickHLA::DebugHandler Class Reference . . . . .	94
7.5.1 Detailed Description . . . . .	94
7.5.2 Constructor & Destructor Documentation . . . . .	94
7.5.2.1 <code>DebugHandler()</code> . . . . .	94
7.5.2.2 <code>~DebugHandler()</code> . . . . .	95
7.5.3 Member Function Documentation . . . . .	95
7.5.3.1 <code>get_code_section()</code> . . . . .	95
7.5.3.2 <code>get_debug_level()</code> . . . . .	95
7.5.3.3 <code>get_debug_level_as_int()</code> . . . . .	95
7.5.3.4 <code>set()</code> . . . . .	95
7.5.3.5 <code>should_print()</code> . . . . .	96
7.5.4 Friends And Related Function Documentation . . . . .	97
7.5.4.1 <code>init_attrTrickHLA__DebugHandler</code> . . . . .	97
7.5.4.2 <code>InputProcessor</code> . . . . .	97
7.5.5 Field Documentation . . . . .	97
7.5.5.1 <code>code_section</code> . . . . .	97
7.5.5.2 <code>debug_level</code> . . . . .	97
7.6 TrickHLA::DivestThreadArgs Struct Reference . . . . .	98
7.6.1 Detailed Description . . . . .	98
7.6.2 Field Documentation . . . . .	98
7.6.2.1 <code>handle_set</code> . . . . .	98
7.6.2.2 <code>trick_hla_obj</code> . . . . .	98
7.7 SpaceFOM::DynamicalEntity Class Reference . . . . .	99
7.7.1 Detailed Description . . . . .	100
7.7.2 Constructor & Destructor Documentation . . . . .	100
7.7.2.1 <code>DynamicalEntity() [1/2]</code> . . . . .	100
7.7.2.2 <code>~DynamicalEntity()</code> . . . . .	100
7.7.2.3 <code>DynamicalEntity() [2/2]</code> . . . . .	101

---

7.7.3 Member Function Documentation . . . . .	101
7.7.3.1 operator=() . . . . .	101
7.7.3.2 pack() . . . . .	101
7.7.3.3 unpack() . . . . .	101
7.7.4 Friends And Related Function Documentation . . . . .	101
7.7.4.1 init_attrSpaceFOM__DynamicalEntity . . . . .	101
7.7.4.2 InputProcessor . . . . .	101
7.7.5 Field Documentation . . . . .	101
7.7.5.1 force . . . . .	102
7.7.5.2 inertia . . . . .	102
7.7.5.3 inertia_rate . . . . .	102
7.7.5.4 mass . . . . .	102
7.7.5.5 mass_rate . . . . .	102
7.7.5.6 torque . . . . .	102
7.8 DIS::ExecutionConfiguration Class Reference . . . . .	103
7.8.1 Detailed Description . . . . .	105
7.8.2 Constructor & Destructor Documentation . . . . .	105
7.8.2.1 ExecutionConfiguration() [1/2] . . . . .	105
7.8.2.2 ~ExecutionConfiguration() . . . . .	105
7.8.2.3 ExecutionConfiguration() [2/2] . . . . .	105
7.8.3 Member Function Documentation . . . . .	106
7.8.3.1 configure_attributes() . . . . .	106
7.8.3.2 get_current_execution_mode() . . . . .	106
7.8.3.3 get_next_execution_mode() . . . . .	106
7.8.3.4 get_next_mode_cte_time() . . . . .	106
7.8.3.5 get_next_mode_scenario_time() . . . . .	107
7.8.3.6 get_root_frame_name() . . . . .	107
7.8.3.7 get_scenario_time_epoch() . . . . .	107
7.8.3.8 operator=() . . . . .	107
7.8.3.9 pack() . . . . .	107
7.8.3.10 print_execution_configuration() . . . . .	108
7.8.3.11 set_current_execution_mode() [1/2] . . . . .	108
7.8.3.12 set_current_execution_mode() [2/2] . . . . .	108
7.8.3.13 set_next_execution_mode() [1/2] . . . . .	109
7.8.3.14 set_next_execution_mode() [2/2] . . . . .	109
7.8.3.15 set_next_mode_cte_time() . . . . .	109
7.8.3.16 set_next_mode_scenario_time() . . . . .	109
7.8.3.17 set_root_frame_name() . . . . .	110
7.8.3.18 set_scenario_time_epoch() . . . . .	110

---

7.8.3.19 <code>setup_ref_attributes()</code>	110
7.8.3.20 <code>unpack()</code>	111
7.8.3.21 <code>wait_on_update()</code>	111
7.8.4 Friends And Related Function Documentation	111
7.8.4.1 <code>init_attrDIS__ExecutionConfiguration</code>	111
7.8.4.2 <code>InputProcessor</code>	111
7.8.5 Field Documentation	111
7.8.5.1 <code>current_execution_mode</code>	111
7.8.5.2 <code>next_execution_mode</code>	112
7.8.5.3 <code>next_mode_cte_time</code>	112
7.8.5.4 <code>next_mode_scenario_time</code>	112
7.8.5.5 <code>root_frame_name</code>	112
7.8.5.6 <code>scenario_time_epoch</code>	113
7.9 DSES::ExecutionConfiguration Class Reference	113
7.9.1 Detailed Description	115
7.9.2 Constructor & Destructor Documentation	116
7.9.2.1 <code>ExecutionConfiguration()</code> [1/2]	116
7.9.2.2 <code>~ExecutionConfiguration()</code>	116
7.9.2.3 <code>ExecutionConfiguration()</code> [2/2]	116
7.9.3 Member Function Documentation	116
7.9.3.1 <code>configure_attributes()</code>	116
7.9.3.2 <code>get_current_execution_mode()</code>	116
7.9.3.3 <code>get_next_execution_mode()</code>	117
7.9.3.4 <code>get_next_mode_cte_time()</code>	117
7.9.3.5 <code>get_next_mode_scenario_time()</code>	117
7.9.3.6 <code>get_root_frame_name()</code>	117
7.9.3.7 <code>get_scenario_time_epoch()</code>	118
7.9.3.8 <code>operator=()</code>	118
7.9.3.9 <code>pack()</code>	118
7.9.3.10 <code>print_execution_configuration()</code>	118
7.9.3.11 <code>set_current_execution_mode()</code> [1/2]	118
7.9.3.12 <code>set_current_execution_mode()</code> [2/2]	119
7.9.3.13 <code>set_next_execution_mode()</code> [1/2]	119
7.9.3.14 <code>set_next_execution_mode()</code> [2/2]	119
7.9.3.15 <code>set_next_mode_cte_time()</code>	120
7.9.3.16 <code>set_next_mode_scenario_time()</code>	120
7.9.3.17 <code>set_root_frame_name()</code>	120
7.9.3.18 <code>set_scenario_time_epoch()</code>	120
7.9.3.19 <code>setup_ref_attributes()</code>	121

---

7.9.3.20 <code>unpack()</code> . . . . .	121
7.9.3.21 <code>wait_on_update()</code> . . . . .	121
7.9.4 Friends And Related Function Documentation . . . . .	122
7.9.4.1 <code>init_attrDSES__ExecutionConfiguration</code> . . . . .	122
7.9.4.2 <code>InputProcessor</code> . . . . .	122
7.9.5 Field Documentation . . . . .	122
7.9.5.1 <code>current_execution_mode</code> . . . . .	122
7.9.5.2 <code>next_execution_mode</code> . . . . .	122
7.9.5.3 <code>next_mode_cte_time</code> . . . . .	122
7.9.5.4 <code>next_mode_scenario_time</code> . . . . .	123
7.9.5.5 <code>root_frame_name</code> . . . . .	123
7.9.5.6 <code>scenario_time_epoch</code> . . . . .	123
7.10 <code>IMSim::ExecutionConfiguration</code> Class Reference . . . . .	123
7.10.1 Detailed Description . . . . .	126
7.10.2 Constructor & Destructor Documentation . . . . .	126
7.10.2.1 <code>ExecutionConfiguration()</code> [1/2] . . . . .	126
7.10.2.2 <code>~ExecutionConfiguration()</code> . . . . .	126
7.10.2.3 <code>ExecutionConfiguration()</code> [2/2] . . . . .	127
7.10.3 Member Function Documentation . . . . .	127
7.10.3.1 <code>configure_attributes()</code> . . . . .	127
7.10.3.2 <code>get_current_execution_mode()</code> . . . . .	127
7.10.3.3 <code>get_least_common_time_step()</code> . . . . .	127
7.10.3.4 <code>get_next_execution_mode()</code> . . . . .	127
7.10.3.5 <code>get_next_mode_cte_time()</code> . . . . .	128
7.10.3.6 <code>get_next_mode_scenario_time()</code> . . . . .	128
7.10.3.7 <code>get_root_frame_name()</code> . . . . .	128
7.10.3.8 <code>get_scenario_time_epoch()</code> . . . . .	128
7.10.3.9 <code>operator=()</code> . . . . .	129
7.10.3.10 <code>pack()</code> . . . . .	129
7.10.3.11 <code>print_execution_configuration()</code> . . . . .	129
7.10.3.12 <code>set_current_execution_mode()</code> [1/2] . . . . .	129
7.10.3.13 <code>set_current_execution_mode()</code> [2/2] . . . . .	129
7.10.3.14 <code>set_least_common_time_step()</code> . . . . .	130
7.10.3.15 <code>set_next_execution_mode()</code> [1/2] . . . . .	130
7.10.3.16 <code>set_next_execution_mode()</code> [2/2] . . . . .	130
7.10.3.17 <code>set_next_mode_cte_time()</code> . . . . .	131
7.10.3.18 <code>set_next_mode_scenario_time()</code> . . . . .	131
7.10.3.19 <code>set_root_frame_name()</code> . . . . .	131
7.10.3.20 <code>set_scenario_time_epoch()</code> . . . . .	131

7.10.3.21 <code>setup_ref_attributes()</code> . . . . .	133
7.10.3.22 <code>unpack()</code> . . . . .	133
7.10.3.23 <code>wait_on_update()</code> . . . . .	133
7.10.4 Friends And Related Function Documentation . . . . .	134
7.10.4.1 <code>init_attrIMSim__ExecutionConfiguration</code> . . . . .	134
7.10.4.2 <code>InputProcessor</code> . . . . .	134
7.10.5 Field Documentation . . . . .	134
7.10.5.1 <code>current_execution_mode</code> . . . . .	134
7.10.5.2 <code>least_common_time_step</code> . . . . .	134
7.10.5.3 <code>next_execution_mode</code> . . . . .	134
7.10.5.4 <code>next_mode_cte_time</code> . . . . .	135
7.10.5.5 <code>next_mode_scenario_time</code> . . . . .	135
7.10.5.6 <code>root_frame_name</code> . . . . .	135
7.10.5.7 <code>scenario_time_epoch</code> . . . . .	135
7.11 SpaceFOM::ExecutionConfiguration Class Reference . . . . .	135
7.11.1 Detailed Description . . . . .	138
7.11.2 Constructor & Destructor Documentation . . . . .	138
7.11.2.1 <code>ExecutionConfiguration()</code> [1/3] . . . . .	138
7.11.2.2 <code>ExecutionConfiguration()</code> [2/3] . . . . .	138
7.11.2.3 <code>~ExecutionConfiguration()</code> . . . . .	139
7.11.2.4 <code>ExecutionConfiguration()</code> [3/3] . . . . .	139
7.11.3 Member Function Documentation . . . . .	139
7.11.3.1 <code>configure()</code> . . . . .	139
7.11.3.2 <code>configure_attributes()</code> . . . . .	139
7.11.3.3 <code>get_current_execution_mode()</code> . . . . .	140
7.11.3.4 <code>get_least_common_time_step()</code> . . . . .	140
7.11.3.5 <code>get_next_execution_mode()</code> . . . . .	140
7.11.3.6 <code>get_next_mode_cte_time()</code> . . . . .	140
7.11.3.7 <code>get_next_mode_scenario_time()</code> . . . . .	140
7.11.3.8 <code>get_root_frame_name()</code> . . . . .	141
7.11.3.9 <code>get_scenario_time_epoch()</code> . . . . .	141
7.11.3.10 <code>operator=()</code> . . . . .	141
7.11.3.11 <code>pack()</code> . . . . .	141
7.11.3.12 <code>print_execution_configuration()</code> . . . . .	142
7.11.3.13 <code>set_current_execution_mode()</code> [1/2] . . . . .	142
7.11.3.14 <code>set_current_execution_mode()</code> [2/2] . . . . .	142
7.11.3.15 <code>set_least_common_time_step()</code> . . . . .	142
7.11.3.16 <code>set_next_execution_mode()</code> [1/2] . . . . .	143
7.11.3.17 <code>set_next_execution_mode()</code> [2/2] . . . . .	143

---

7.11.3.18 set_next_mode_cte_time() . . . . .	143
7.11.3.19 set_next_mode_scenario_time() . . . . .	144
7.11.3.20 set_root_frame_name() . . . . .	144
7.11.3.21 set_scenario_time_epoch() . . . . .	144
7.11.3.22 setup_ref_attributes() . . . . .	145
7.11.3.23 unpack() . . . . .	145
7.11.3.24 wait_on_update() . . . . .	145
7.11.4 Friends And Related Function Documentation . . . . .	146
7.11.4.1 init_attrSpaceFOM__ExecutionConfiguration . . . . .	146
7.11.4.2 InputProcessor . . . . .	146
7.11.5 Field Documentation . . . . .	146
7.11.5.1 current_execution_mode . . . . .	146
7.11.5.2 least_common_time_step . . . . .	146
7.11.5.3 next_execution_mode . . . . .	147
7.11.5.4 next_mode_cte_time . . . . .	147
7.11.5.5 next_mode_scenario_time . . . . .	147
7.11.5.6 root_frame_name . . . . .	147
7.11.5.7 scenario_time_epoch . . . . .	147
7.12 TrickHLA::ExecutionConfiguration Class Reference . . . . .	148
7.12.1 Detailed Description . . . . .	149
7.12.2 Constructor & Destructor Documentation . . . . .	150
7.12.2.1 ExecutionConfiguration() [1/3] . . . . .	150
7.12.2.2 ExecutionConfiguration() [2/3] . . . . .	150
7.12.2.3 ~ExecutionConfiguration() . . . . .	150
7.12.2.4 ExecutionConfiguration() [3/3] . . . . .	150
7.12.3 Member Function Documentation . . . . .	150
7.12.3.1 configure() . . . . .	150
7.12.3.2 configure_attributes() . . . . .	151
7.12.3.3 operator=(()) . . . . .	151
7.12.3.4 pack() . . . . .	151
7.12.3.5 print_execution_configuration() . . . . .	151
7.12.3.6 setup_ref_attributes() . . . . .	151
7.12.3.7 unpack() . . . . .	152
7.12.4 Friends And Related Function Documentation . . . . .	152
7.12.4.1 init_attrTrickHLA__ExecutionConfiguration . . . . .	152
7.12.4.2 InputProcessor . . . . .	152
7.12.5 Field Documentation . . . . .	152
7.12.5.1 num_federates . . . . .	152
7.12.5.2 owner . . . . .	152

---

---

7.12.5.3 required_federates . . . . .	152
7.12.5.4 run_duration . . . . .	153
7.12.5.5 run_duration_microsec . . . . .	153
7.13 TrickHLA::ExecutionConfigurationBase Class Reference . . . . .	153
7.13.1 Detailed Description . . . . .	155
7.13.2 Constructor & Destructor Documentation . . . . .	155
7.13.2.1 ExecutionConfigurationBase() [1/3] . . . . .	156
7.13.2.2 ExecutionConfigurationBase() [2/3] . . . . .	156
7.13.2.3 ~ExecutionConfigurationBase() . . . . .	156
7.13.2.4 ExecutionConfigurationBase() [3/3] . . . . .	156
7.13.3 Member Function Documentation . . . . .	156
7.13.3.1 clear_update_pending() . . . . .	156
7.13.3.2 configure() . . . . .	157
7.13.3.3 configure_attributes() . . . . .	157
7.13.3.4 get_execution_control() . . . . .	157
7.13.3.5 get_S_define_name() . . . . .	157
7.13.3.6 operator=( ) . . . . .	157
7.13.3.7 pack() . . . . .	157
7.13.3.8 print_execution_configuration() . . . . .	158
7.13.3.9 reset_ownership_states() . . . . .	158
7.13.3.10 reset_preferred_order() . . . . .	158
7.13.3.11 set_execution_control() . . . . .	158
7.13.3.12 set_master() . . . . .	158
7.13.3.13 set_S_define_name() . . . . .	159
7.13.3.14 setup() . . . . .	159
7.13.3.15 setup_ref_attributes() . . . . .	159
7.13.3.16 unpack() . . . . .	161
7.13.3.17 update_pending() . . . . .	161
7.13.3.18 wait_on_registration() . . . . .	161
7.13.3.19 wait_on_update() . . . . .	161
7.13.4 Friends And Related Function Documentation . . . . .	162
7.13.4.1 init_attrTrickHLA__ExecutionConfigurationBase . . . . .	162
7.13.4.2 InputProcessor . . . . .	162
7.13.5 Field Documentation . . . . .	162
7.13.5.1 execution_control . . . . .	162
7.13.5.2 pending_update . . . . .	163
7.13.5.3 S_define_name . . . . .	163
7.14 DIS::ExecutionControl Class Reference . . . . .	163
7.14.1 Detailed Description . . . . .	167

---

7.14.2 Constructor & Destructor Documentation . . . . .	167
7.14.2.1 ExecutionControl() [1/2] . . . . .	167
7.14.2.2 ~ExecutionControl() . . . . .	167
7.14.2.3 ExecutionControl() [2/2] . . . . .	167
7.14.3 Member Function Documentation . . . . .	167
7.14.3.1 add_multiphase_init_sync_points() . . . . .	167
7.14.3.2 add_pause() . . . . .	168
7.14.3.3 announce_sync_point() . . . . .	168
7.14.3.4 check_freeze() . . . . .	168
7.14.3.5 check_freeze_exit() [1/2] . . . . .	168
7.14.3.6 check_freeze_exit() [2/2] . . . . .	169
7.14.3.7 check_mode_transition_request() . . . . .	169
7.14.3.8 clear_mode_values() . . . . .	169
7.14.3.9 clear_multiphase_init_sync_points() . . . . .	169
7.14.3.10 determine_federation_master() . . . . .	169
7.14.3.11 early_joiner_hla_init_process() . . . . .	170
7.14.3.12 enter_freeze() . . . . .	170
7.14.3.13 exit_freeze() . . . . .	170
7.14.3.14 freeze_mode_announce() . . . . .	170
7.14.3.15 freeze_mode_transition() . . . . .	170
7.14.3.16 get_execution_configuration() . . . . .	171
7.14.3.17 get_type() . . . . .	171
7.14.3.18 initialize() . . . . .	171
7.14.3.19 is_mtr_valid() . . . . .	172
7.14.3.20 is_save_and_restore_supported() . . . . .	172
7.14.3.21 is_save_initiated() . . . . .	172
7.14.3.22 join_federation_process() [1/2] . . . . .	172
7.14.3.23 join_federation_process() [2/2] . . . . .	173
7.14.3.24 late_joiner_hla_init_process() . . . . .	173
7.14.3.25 object_instance_name_reservation_failed() . . . . .	173
7.14.3.26 operator=( ) . . . . .	173
7.14.3.27 post_multi_phase_init_process() [1/2] . . . . .	173
7.14.3.28 post_multi_phase_init_process() [2/2] . . . . .	173
7.14.3.29 pre_multi_phase_init_processes() [1/2] . . . . .	174
7.14.3.30 pre_multi_phase_init_processes() [2/2] . . . . .	174
7.14.3.31 process_execution_control_updates() . . . . .	174
7.14.3.32 process_mode_interaction() . . . . .	175
7.14.3.33 process_mode_transition_request() . . . . .	175
7.14.3.34 publish() . . . . .	175

---

7.14.3.35 role_determination_process()	176
7.14.3.36 run_mode_transition()	176
7.14.3.37 set_mode_request_from_mtr()	176
7.14.3.38 set_next_execution_control_mode()	177
7.14.3.39 set_pending_mtr()	177
7.14.3.40 set_time_padding()	177
7.14.3.41 setup_interaction_ref_attributes()	177
7.14.3.42 setup_interaction_RTI_handles()	178
7.14.3.43 setup_object_ref_attributes()	178
7.14.3.44 setup_object_RTI_handles()	178
7.14.3.45 shutdown() [1/2]	178
7.14.3.46 shutdown() [2/2]	178
7.14.3.47 shutdown_mode_announce()	178
7.14.3.48 shutdown_mode_transition()	179
7.14.3.49 subscribe()	179
7.14.3.50 sync_point_registration_failed()	179
7.14.3.51 sync_point_registration_succeeded()	179
7.14.3.52 unpublish()	180
7.14.3.53 unsubscribe()	180
7.14.3.54 wait_on_init_sync_point()	180
7.14.4 Friends And Related Function Documentation	180
7.14.4.1 init_attrDIS__ExecutionControl	180
7.14.4.2 InputProcessor	180
7.14.5 Field Documentation	180
7.14.5.1 checktime	180
7.14.5.2 pause_sync_pts	181
7.14.5.3 pending_mtr	181
7.14.5.4 type	181
7.15 DSES::ExecutionControl Class Reference	181
7.15.1 Detailed Description	184
7.15.2 Constructor & Destructor Documentation	184
7.15.2.1 ExecutionControl() [1/2]	185
7.15.2.2 ~ExecutionControl()	185
7.15.2.3 ExecutionControl() [2/2]	185
7.15.3 Member Function Documentation	185
7.15.3.1 achieve_all_multiphase_init_sync_pnts()	185
7.15.3.2 add_multiphase_init_sync_points()	185
7.15.3.3 announce_sync_point()	186
7.15.3.4 check_freeze_exit()	186

---

7.15.3.5 check_mode_transition_request()	186
7.15.3.6 clear_mode_values()	187
7.15.3.7 determine_federation_master()	187
7.15.3.8 freeze_mode_announce()	187
7.15.3.9 freeze_mode_transition()	187
7.15.3.10 get_execution_configuration()	188
7.15.3.11 get_type()	188
7.15.3.12 initialize()	188
7.15.3.13 is_mtr_valid()	189
7.15.3.14 join_federation_process()	189
7.15.3.15 object_instance_name_reservation_failed()	189
7.15.3.16 operator=( )	189
7.15.3.17 post_multi_phase_init_process()	190
7.15.3.18 pre_multi_phase_init_processes()	190
7.15.3.19 process_execution_control_updates()	190
7.15.3.20 process_mode_interaction()	191
7.15.3.21 process_mode_transition_request()	191
7.15.3.22 publish()	192
7.15.3.23 run_mode_transition()	192
7.15.3.24 set_mode_request_from_mtr()	192
7.15.3.25 set_next_execution_control_mode()	193
7.15.3.26 set_pending_mtr()	193
7.15.3.27 set_time_padding()	193
7.15.3.28 setup_interaction_ref_attributes()	194
7.15.3.29 setup_interaction_RTI_handles()	194
7.15.3.30 setup_object_ref_attributes()	194
7.15.3.31 setup_object_RTI_handles()	194
7.15.3.32 shutdown()	194
7.15.3.33 shutdown_mode_announce()	194
7.15.3.34 shutdown_mode_transition()	195
7.15.3.35 subscribe()	195
7.15.3.36 unpublish()	195
7.15.3.37 unsubscribe()	195
7.15.3.38 wait_for_all_multiphase_init_sync_pnts()	195
7.15.4 Friends And Related Function Documentation	195
7.15.4.1 init_attrDSES__ExecutionControl	196
7.15.4.2 InputProcessor	196
7.15.5 Field Documentation	196
7.15.5.1 pending_mtr	196

---

7.15.5.2 <code>type</code>	196
7.16 <code>IMSim::ExecutionControl</code> Class Reference	196
7.16.1 Detailed Description	201
7.16.2 Constructor & Destructor Documentation	201
7.16.2.1 <code>ExecutionControl()</code> [1/2]	201
7.16.2.2 <code>~ExecutionControl()</code>	201
7.16.2.3 <code>ExecutionControl()</code> [2/2]	201
7.16.3 Member Function Documentation	201
7.16.3.1 <code>achieve_all_multiphase_init_sync_pnts()</code>	201
7.16.3.2 <code>add_freeze_scenario_time()</code>	202
7.16.3.3 <code>add_multiphase_init_sync_points()</code>	202
7.16.3.4 <code>add_pause()</code>	202
7.16.3.5 <code>announce_sync_point()</code>	203
7.16.3.6 <code>check_freeze_exit()</code>	203
7.16.3.7 <code>check_freeze_time()</code>	203
7.16.3.8 <code>check_mode_transition_request()</code>	204
7.16.3.9 <code>check_pause()</code> [1/2]	204
7.16.3.10 <code>check_pause()</code> [2/2]	204
7.16.3.11 <code>check_pause_at_init()</code> [1/2]	204
7.16.3.12 <code>check_pause_at_init()</code> [2/2]	205
7.16.3.13 <code>check_scenario_freeze_time()</code>	205
7.16.3.14 <code>clear_mode_values()</code>	205
7.16.3.15 <code>clear_pause()</code>	206
7.16.3.16 <code>convert_loggable_sync_pts()</code>	206
7.16.3.17 <code>determine_if_late_joining_or_restoring_federate()</code>	206
7.16.3.18 <code>enter_freeze()</code>	206
7.16.3.19 <code>exit_freeze()</code>	207
7.16.3.20 <code>freeze_mode_announce()</code>	207
7.16.3.21 <code>freeze_mode_transition()</code>	207
7.16.3.22 <code>get_execution_configuration()</code>	207
7.16.3.23 <code>get_type()</code>	208
7.16.3.24 <code>initialize()</code>	208
7.16.3.25 <code>is_mtr_valid()</code>	208
7.16.3.26 <code>is_save_and_restore_supported()</code>	209
7.16.3.27 <code>is_save_initiated()</code>	209
7.16.3.28 <code>join_federation_process()</code>	209
7.16.3.29 <code>mark_synchronized()</code>	209
7.16.3.30 <code>operator=()</code>	210
7.16.3.31 <code>perform_save()</code>	210

---

7.16.3.32 post_multi_phase_init_process()	210
7.16.3.33 pre_multi_phase_init_processes()	210
7.16.3.34 process_execution_control_updates()	211
7.16.3.35 process_mode_interaction() [1/2]	212
7.16.3.36 process_mode_interaction() [2/2]	212
7.16.3.37 process_mode_transition_request()	212
7.16.3.38 publish()	213
7.16.3.39 receive_interaction()	213
7.16.3.40 reinstate_logged_sync_pts()	214
7.16.3.41 run_mode_transition()	214
7.16.3.42 send_mode_transition_interaction()	214
7.16.3.43 set_mode_request_from_mtr()	214
7.16.3.44 set_next_execution_control_mode()	215
7.16.3.45 set_pending_mtr()	215
7.16.3.46 set_time_padding()	215
7.16.3.47 setup_interaction_ref_attributes()	216
7.16.3.48 setup_interaction_RTI_handles()	216
7.16.3.49 setup_object_ref_attributes()	216
7.16.3.50 setup_object_RTI_handles()	216
7.16.3.51 shutdown()	217
7.16.3.52 shutdown_mode_announce()	217
7.16.3.53 shutdown_mode_transition()	217
7.16.3.54 start_federation_save_at_scenario_time()	217
7.16.3.55 subscribe()	218
7.16.3.56 trigger_freeze_interaction()	218
7.16.3.57 un_freeze()	218
7.16.3.58 unpublish()	218
7.16.3.59 unsubscribe()	218
7.16.3.60 wait_for_all_multiphase_init_sync_pnts()	219
7.16.4 Friends And Related Function Documentation	219
7.16.4.1 init_attrIMSim__ExecutionControl	219
7.16.4.2 InputProcessor	219
7.16.5 Field Documentation	219
7.16.5.1 checktime	219
7.16.5.2 freeze_inter_count	219
7.16.5.3 freeze_interaction	220
7.16.5.4 freeze_interaction_handler	220
7.16.5.5 freeze_scenario_times	220
7.16.5.6 pause_sync_pts	220

---

7.16.5.7 pending_mtr . . . . .	220
7.16.5.8 type . . . . .	220
7.17 SpaceFOM::ExecutionControl Class Reference . . . . .	221
7.17.1 Detailed Description . . . . .	225
7.17.2 Constructor & Destructor Documentation . . . . .	225
7.17.2.1 ExecutionControl() [1/3] . . . . .	225
7.17.2.2 ExecutionControl() [2/3] . . . . .	225
7.17.2.3 ~ExecutionControl() . . . . .	225
7.17.2.4 ExecutionControl() [3/3] . . . . .	225
7.17.3 Member Function Documentation . . . . .	225
7.17.3.1 add_initialization_sync_points() . . . . .	226
7.17.3.2 announce_sync_point() . . . . .	226
7.17.3.3 check_for_shutdown() . . . . .	226
7.17.3.4 check_for_shutdown_with_termination() . . . . .	226
7.17.3.5 check_freeze_exit() . . . . .	227
7.17.3.6 check_mode_transition_request() . . . . .	227
7.17.3.7 earlyJoinerHlaInitProcess() . . . . .	227
7.17.3.8 enter_freeze() . . . . .	228
7.17.3.9 epoch_and_root_frame_discovery_process() . . . . .	228
7.17.3.10 exit_freeze() . . . . .	228
7.17.3.11 freeze_init() . . . . .	229
7.17.3.12 freeze_mode_announce() . . . . .	229
7.17.3.13 freeze_mode_transition() . . . . .	229
7.17.3.14 get_execution_configuration() . . . . .	229
7.17.3.15 get_type() . . . . .	230
7.17.3.16 initialize() . . . . .	230
7.17.3.17 is_mtr_valid() . . . . .	230
7.17.3.18 is_pacing() . . . . .	231
7.17.3.19 is_root_frame_publisher() . . . . .	231
7.17.3.20 join_federation_process() . . . . .	231
7.17.3.21 lateJoinerHlaInitProcess() . . . . .	231
7.17.3.22 operator=( ) . . . . .	232
7.17.3.23 post_multi_phase_init_processes() . . . . .	232
7.17.3.24 pre_multi_phase_init_processes() . . . . .	233
7.17.3.25 process_execution_control_updates() . . . . .	233
7.17.3.26 process_mode_interaction() . . . . .	234
7.17.3.27 process_mode_transition_request() . . . . .	234
7.17.3.28 publish() . . . . .	234
7.17.3.29 receive_init_root_ref_frame() . . . . .	235

---

7.17.3.30 receive_interaction()	235
7.17.3.31 receive_root_ref_frame()	235
7.17.3.32 role_determination_process()	236
7.17.3.33 run_mode_transition()	236
7.17.3.34 send_init_root_ref_frame()	236
7.17.3.35 send_mode_transition_interaction()	237
7.17.3.36 send_MTR_interaction()	237
7.17.3.37 send_root_ref_frame()	237
7.17.3.38 set_least_common_time_step()	237
7.17.3.39 set_mode_request_from_mtr()	238
7.17.3.40 set_next_execution_control_mode()	238
7.17.3.41 set_pending_mtr()	239
7.17.3.42 set_time_padding()	239
7.17.3.43 setup_interaction_ref_attributes()	239
7.17.3.44 setup_interaction_RTI_handles()	240
7.17.3.45 setup_object_ref_attributes()	240
7.17.3.46 setup_object_RTI_handles()	240
7.17.3.47 shutdown()	240
7.17.3.48 shutdown_mode_announce()	240
7.17.3.49 shutdown_mode_transition()	241
7.17.3.50 start_federation_save_at_scenario_time()	241
7.17.3.51 subscribe()	241
7.17.3.52 unpublish()	241
7.17.3.53 unsubscribe()	242
7.17.3.54 wait_on_root_frame_discovered_synchronization()	242
7.17.4 Friends And Related Function Documentation	242
7.17.4.1 init_attrSpaceFOM_ExecutionControl	242
7.17.4.2 InputProcessor	242
7.17.5 Field Documentation	242
7.17.5.1 mtr_interaction	242
7.17.5.2 mtr_interaction_handler	243
7.17.5.3 pacing	243
7.17.5.4 pending_mtr	243
7.17.5.5 root_frame_pub	243
7.17.5.6 root_ref_frame	243
7.17.5.7 type	243
7.18 TrickHLA::ExecutionControl Class Reference	244
7.18.1 Detailed Description	246
7.18.2 Constructor & Destructor Documentation	246

---

7.18.2.1 ExecutionControl() [1/3] . . . . .	247
7.18.2.2 ExecutionControl() [2/3] . . . . .	247
7.18.2.3 ~ExecutionControl() . . . . .	247
7.18.2.4 ExecutionControl() [3/3] . . . . .	247
7.18.3 Member Function Documentation . . . . .	247
7.18.3.1 add_initialization_sync_points() . . . . .	247
7.18.3.2 add_multiphase_init_sync_points() . . . . .	247
7.18.3.3 announce_sync_point() . . . . .	248
7.18.3.4 clear_multiphase_init_sync_points() . . . . .	248
7.18.3.5 freeze_mode_announce() . . . . .	248
7.18.3.6 freeze_mode_transition() . . . . .	248
7.18.3.7 get_execution_configuration() . . . . .	249
7.18.3.8 get_type() . . . . .	249
7.18.3.9 initialize() . . . . .	249
7.18.3.10 join_federation_process() . . . . .	249
7.18.3.11 operator=( ) . . . . .	250
7.18.3.12 post_multi_phase_init_processes() . . . . .	250
7.18.3.13 pre_multi_phase_init_processes() . . . . .	250
7.18.3.14 process_execution_control_updates() . . . . .	250
7.18.3.15 process_mode_interaction() . . . . .	251
7.18.3.16 publish() . . . . .	251
7.18.3.17 receive_interaction() . . . . .	251
7.18.3.18 run_mode_transition() . . . . .	251
7.18.3.19 send_mode_transition_interaction() . . . . .	252
7.18.3.20 set_least_common_time_step() . . . . .	252
7.18.3.21 set_next_execution_control_mode() . . . . .	252
7.18.3.22 set_time_padding() . . . . .	252
7.18.3.23 setup_interaction_ref_attributes() . . . . .	253
7.18.3.24 setup_interaction_RTI_handles() . . . . .	253
7.18.3.25 setup_object_ref_attributes() . . . . .	253
7.18.3.26 setup_object_RTI_handles() . . . . .	253
7.18.3.27 shutdown() . . . . .	253
7.18.3.28 shutdown_mode_announce() . . . . .	254
7.18.3.29 shutdown_mode_transition() . . . . .	254
7.18.3.30 start_federation_save_at_scenario_time() . . . . .	254
7.18.3.31 subscribe() . . . . .	254
7.18.3.32 unpublish() . . . . .	254
7.18.3.33 unsubscribe() . . . . .	254
7.18.3.34 wait_on_init_data() . . . . .	255

---

7.18.3.35 <code>wait_on_init_sync_point()</code>	255
7.18.4 Friends And Related Function Documentation	255
7.18.4.1 <code>init_attrTrickHLA__ExecutionControl</code>	255
7.18.4.2 <code>InputProcessor</code>	255
7.18.5 Field Documentation	255
7.18.5.1 <code>type</code>	255
7.19 <code>TrickHLA::ExecutionControlBase</code> Class Reference	256
7.19.1 Detailed Description	263
7.19.2 Constructor & Destructor Documentation	263
7.19.2.1 <code>ExecutionControlBase()</code> [1/3]	263
7.19.2.2 <code>ExecutionControlBase()</code> [2/3]	263
7.19.2.3 <code>~ExecutionControlBase()</code>	264
7.19.2.4 <code>ExecutionControlBase()</code> [3/3]	264
7.19.3 Member Function Documentation	264
7.19.3.1 <code>achieve_all_multiphase_init_sync_pnts()</code>	264
7.19.3.2 <code>add_initialization_sync_points()</code>	264
7.19.3.3 <code>add_multiphase_init_sync_points()</code>	264
7.19.3.4 <code>add_object_to_map()</code>	265
7.19.3.5 <code>check_for_shutdown()</code>	265
7.19.3.6 <code>check_for_shutdown_with_termination()</code>	265
7.19.3.7 <code>check_freeze_exit()</code>	266
7.19.3.8 <code>check_pause()</code>	266
7.19.3.9 <code>check_pause_at_init()</code>	266
7.19.3.10 <code>clear_mode_transition_requested()</code>	266
7.19.3.11 <code>clear_mode_values()</code>	267
7.19.3.12 <code>clear_multiphase_init_sync_points()</code>	267
7.19.3.13 <code>convert_loggable_sync_pts()</code>	267
7.19.3.14 <code>convert_scenario_time_to_sim_time()</code>	267
7.19.3.15 <code>convert_sim_time_to_scenario_time()</code>	268
7.19.3.16 <code>does_cte_timeline_exist()</code>	268
7.19.3.17 <code>does_init_complete_sync_point_exist()</code>	268
7.19.3.18 <code>does_scenario_timeline_exist()</code>	269
7.19.3.19 <code>does_sim_timeline_exist()</code>	269
7.19.3.20 <code>enter_freeze()</code>	269
7.19.3.21 <code>exit_freeze()</code>	269
7.19.3.22 <code>freeze_init()</code>	270
7.19.3.23 <code>freeze_mode_announce()</code>	270
7.19.3.24 <code>freeze_mode_transition()</code>	270
7.19.3.25 <code>get_cte_time()</code>	270

---

7.19.3.26 get_current_execution_control_mode()	270
7.19.3.27 get_execution_configuration()	271
7.19.3.28 get_federate()	271
7.19.3.29 get_least_common_time_step()	271
7.19.3.30 get_manager()	271
7.19.3.31 get_requested_execution_control_mode()	272
7.19.3.32 get_scenario_freeze_time()	272
7.19.3.33 get_scenario_time()	272
7.19.3.34 get_sim_time()	273
7.19.3.35 get_simulation_freeze_time()	273
7.19.3.36 get_time_padding()	273
7.19.3.37 get_trickhla_object()	273
7.19.3.38 get_type()	274
7.19.3.39 get_unregistered_object()	274
7.19.3.40 get_unregistered_remote_object()	274
7.19.3.41 initialize()	275
7.19.3.42 is_execution_configuration_used()	275
7.19.3.43 is_in_freeze()	275
7.19.3.44 is_in_reconfig()	276
7.19.3.45 is_in_restart()	276
7.19.3.46 is_initializing()	276
7.19.3.47 is_lateJoiner()	276
7.19.3.48 is_lateJoiner_determined()	277
7.19.3.49 is_master()	277
7.19.3.50 is_master_preset()	278
7.19.3.51 is_mode_transition_requested()	278
7.19.3.52 is_running()	278
7.19.3.53 is_save_and_restore_supported()	279
7.19.3.54 is_save_initiated()	279
7.19.3.55 is_shutdown()	279
7.19.3.56 join_federation_process()	279
7.19.3.57 mark_object_as_deleted_from_federation()	280
7.19.3.58 object_instance_name_reservation_failed()	280
7.19.3.59 object_instance_name_reservation_succeeded()	280
7.19.3.60 operator=()	281
7.19.3.61 perform_save()	281
7.19.3.62 post_multi_phase_init_processes()	281
7.19.3.63 pre_multi_phase_init_processes()	281
7.19.3.64 process_deleted_objects()	281

---

7.19.3.65 process_execution_control_updates()	282
7.19.3.66 process_mode_interaction()	282
7.19.3.67 provide_attribute_update()	282
7.19.3.68 publish()	282
7.19.3.69 receive_cyclic_data()	283
7.19.3.70 receive_execution_configuration()	283
7.19.3.71 receive_interaction()	283
7.19.3.72 register_interactions_with_RTI()	284
7.19.3.73 register_objects_with_RTI()	284
7.19.3.74 reinstate_logged_sync_pts()	284
7.19.3.75 remove_execution_configuration()	284
7.19.3.76 run_mode_transition()	284
7.19.3.77 send_execution_configuration()	284
7.19.3.78 send_mode_transition_interaction()	285
7.19.3.79 send_requested_data()	285
7.19.3.80 set_current_execution_control_mode()	285
7.19.3.81 set_execution_configuration()	286
7.19.3.82 set_federate()	286
7.19.3.83 set_least_common_time_step()	286
7.19.3.84 set_master()	286
7.19.3.85 set_mode_transition_requested()	287
7.19.3.86 set_next_execution_control_mode()	287
7.19.3.87 set_requested_execution_control_mode() [1/2]	287
7.19.3.88 set_requested_execution_control_mode() [2/2]	287
7.19.3.89 set_scenario_freeze_time()	288
7.19.3.90 set_scenario_timeline()	288
7.19.3.91 set_simulation_freeze_time()	288
7.19.3.92 set_time_padding()	288
7.19.3.93 set_wait_sleep()	289
7.19.3.94 set_wait_timeout()	289
7.19.3.95 setup() [1/2]	289
7.19.3.96 setup() [2/2]	290
7.19.3.97 setup_checkpoint()	290
7.19.3.98 setup_interaction_ref_attributes()	290
7.19.3.99 setup_interaction_RTI_handles()	291
7.19.3.100 setup_object_ref_attributes()	291
7.19.3.101 setup_object_RTI_handles()	291
7.19.3.102 should_print()	291
7.19.3.103 shutdown()	292

---

7.19.3.104 shutdown_mode_announce() . . . . .	292
7.19.3.105 shutdown_mode_transition() . . . . .	292
7.19.3.106 start_federation_save_at_scenario_time() . . . . .	292
7.19.3.107 subscribe() . . . . .	292
7.19.3.108 un_freeze() . . . . .	292
7.19.3.109 unpublish() . . . . .	293
7.19.3.110 unsubscribe() . . . . .	293
7.19.3.111 wait_for_all_multiphase_init_sync_pnts() . . . . .	293
7.19.3.112 wait_for_sync_point_announce() . . . . .	293
7.19.3.113 wait_on_init_data() . . . . .	293
7.19.3.114 wait_on_init_sync_point() . . . . .	294
7.19.4 Friends And Related Function Documentation . . . . .	294
7.19.4.1 init_attrTrickHLA__ExecutionControlBase . . . . .	294
7.19.4.2 InputProcessor . . . . .	294
7.19.4.3 Manager . . . . .	294
7.19.5 Field Documentation . . . . .	294
7.19.5.1 cte_timeline . . . . .	294
7.19.5.2 current_execution_control_mode . . . . .	295
7.19.5.3 execution_configuration . . . . .	295
7.19.5.4 federate . . . . .	296
7.19.5.5 init_complete_sp_exists . . . . .	296
7.19.5.6 lateJoiner . . . . .	297
7.19.5.7 lateJoiner_determined . . . . .	297
7.19.5.8 least_common_time_step . . . . .	297
7.19.5.9 loggable_sync_pts . . . . .	297
7.19.5.10 logged_sync_pts_count . . . . .	298
7.19.5.11 manager . . . . .	298
7.19.5.12 master . . . . .	298
7.19.5.13 mode_transition_requested . . . . .	298
7.19.5.14 multiphase_init_sync_pnt_list . . . . .	298
7.19.5.15 multiphase_init_sync_points . . . . .	299
7.19.5.16 next_mode_cte_time . . . . .	299
7.19.5.17 next_mode_scenario_time . . . . .	299
7.19.5.18 requested_execution_control_mode . . . . .	299
7.19.5.19 scenario_freeze_time . . . . .	299
7.19.5.20 scenario_timeline . . . . .	300
7.19.5.21 sim_timeline . . . . .	300
7.19.5.22 simulation_freeze_time . . . . .	300
7.19.5.23 time_padding . . . . .	301

---

7.19.5.24 use_preset_master . . . . .	301
7.19.5.25 wait_sleep . . . . .	301
7.19.5.26 wait_timeout . . . . .	301
7.20 TrickHLA::FedAmb Class Reference . . . . .	302
7.20.1 Detailed Description . . . . .	306
7.20.2 Constructor & Destructor Documentation . . . . .	306
7.20.2.1 FedAmb() [1/2] . . . . .	307
7.20.2.2 ~FedAmb() . . . . .	307
7.20.2.3 FedAmb() [2/2] . . . . .	307
7.20.3 Member Function Documentation . . . . .	307
7.20.3.1 announceSynchronizationPoint() . . . . .	307
7.20.3.2 attributeIsNotOwned() . . . . .	307
7.20.3.3 attributeIsOwnedByRTI() . . . . .	307
7.20.3.4 attributeOwnershipAcquisitionNotification() . . . . .	308
7.20.3.5 attributeOwnershipUnavailable() . . . . .	308
7.20.3.6 attributesInScope() . . . . .	308
7.20.3.7 attributesOutOfScope() . . . . .	308
7.20.3.8 confirmAttributeOwnershipAcquisitionCancellation() . . . . .	308
7.20.3.9 confirmAttributeTransportationTypeChange() . . . . .	309
7.20.3.10 confirmInteractionTransportationTypeChange() . . . . .	309
7.20.3.11 connectionLost() . . . . .	309
7.20.3.12 discoverObjectInstance() [1/2] . . . . .	309
7.20.3.13 discoverObjectInstance() [2/2] . . . . .	309
7.20.3.14 federationNotRestored() . . . . .	310
7.20.3.15 federationNotSaved() . . . . .	310
7.20.3.16 federationRestoreBegin() . . . . .	310
7.20.3.17 federationRestored() . . . . .	310
7.20.3.18 federationRestoreStatusResponse() . . . . .	310
7.20.3.19 federationSaved() . . . . .	310
7.20.3.20 federationSaveStatusResponse() . . . . .	311
7.20.3.21 federationSynchronized() . . . . .	311
7.20.3.22 get_manager() . . . . .	311
7.20.3.23 informAttributeOwnership() . . . . .	311
7.20.3.24 initialize() . . . . .	311
7.20.3.25 initiateFederateRestore() . . . . .	311
7.20.3.26 initiateFederateSave() [1/2] . . . . .	312
7.20.3.27 initiateFederateSave() [2/2] . . . . .	312
7.20.3.28 multipleObjectInstanceNameReservationFailed() . . . . .	312
7.20.3.29 multipleObjectInstanceNameReservationSucceeded() . . . . .	312

---

7.20.3.30 objectInstanceNameReservationFailed()	312
7.20.3.31 objectInstanceNameReservationSucceeded()	313
7.20.3.32 operator=()	313
7.20.3.33 provideAttributeValueUpdate()	313
7.20.3.34 receiveInteraction() [1/3]	313
7.20.3.35 receiveInteraction() [2/3]	313
7.20.3.36 receiveInteraction() [3/3]	314
7.20.3.37 reflectAttributeValues() [1/3]	314
7.20.3.38 reflectAttributeValues() [2/3]	314
7.20.3.39 reflectAttributeValues() [3/3]	315
7.20.3.40 removeObjectInstance() [1/3]	315
7.20.3.41 removeObjectInstance() [2/3]	315
7.20.3.42 removeObjectInstance() [3/3]	315
7.20.3.43 reportAttributeTransportationType()	316
7.20.3.44 reportFederationExecutions()	316
7.20.3.45 reportInteractionTransportationType()	316
7.20.3.46 requestAttributeOwnershipAssumption()	316
7.20.3.47 requestAttributeOwnershipRelease()	317
7.20.3.48 requestDivestitureConfirmation()	317
7.20.3.49 requestFederationRestoreFailed()	317
7.20.3.50 requestFederationRestoreSucceeded()	317
7.20.3.51 requestRetraction()	317
7.20.3.52 reset_federation_restored_rebuild_federate_handle_set()	317
7.20.3.53 set_federation_restore_status_response_to_echo()	318
7.20.3.54 set_federation_restore_status_response_to_process()	318
7.20.3.55 set_federation_restored_rebuild_federate_handle_set()	318
7.20.3.56 setup()	318
7.20.3.57 should_print()	318
7.20.3.58 startRegistrationForObjectClass()	319
7.20.3.59 stopRegistrationForObjectClass()	319
7.20.3.60 synchronizationPointRegistrationFailed()	319
7.20.3.61 synchronizationPointRegistrationSucceeded()	319
7.20.3.62 timeAdvanceGrant()	320
7.20.3.63 timeConstrainedEnabled()	320
7.20.3.64 timeRegulationEnabled()	320
7.20.3.65 turnInteractionsOff()	320
7.20.3.66 turnInteractionsOn()	320
7.20.3.67 turnUpdatesOffForObjectInstance()	320
7.20.3.68 turnUpdatesOnForObjectInstance() [1/2]	321

---

7.20.3.69 turnUpdatesOnForObjectInstance() [2/2] . . . . .	321
7.20.4 Friends And Related Function Documentation . . . . .	321
7.20.4.1 init_attrTrickHLA__FedAmb . . . . .	321
7.20.4.2 InputProcessor . . . . .	321
7.20.5 Field Documentation . . . . .	321
7.20.5.1 federate . . . . .	321
7.20.5.2 federation_restore_status_response_context_switch . . . . .	321
7.20.5.3 federation_restored_rebuild_federate_handle_set . . . . .	322
7.20.5.4 manager . . . . .	322
7.21 TrickHLA::Federate Class Reference . . . . .	322
7.21.1 Detailed Description . . . . .	336
7.21.2 Constructor & Destructor Documentation . . . . .	336
7.21.2.1 Federate() [1/2] . . . . .	337
7.21.2.2 ~Federate() . . . . .	337
7.21.2.3 Federate() [2/2] . . . . .	337
7.21.3 Member Function Documentation . . . . .	337
7.21.3.1 achieve_and_wait_for_synchronization() . . . . .	337
7.21.3.2 achieve_synchronization_point() . . . . .	338
7.21.3.3 add_a_single_entry_into_running_feds() . . . . .	338
7.21.3.4 add_federate_instance_id() . . . . .	338
7.21.3.5 add_MOM_HLAfederate_instance_id() . . . . .	338
7.21.3.6 add_MOM_HLAfederation_instance_id() . . . . .	339
7.21.3.7 announce_sync_point() . . . . .	339
7.21.3.8 ask_MOM_for_auto_provide_setting() . . . . .	339
7.21.3.9 ask_MOM_for_federate_names() . . . . .	340
7.21.3.10 backup_auto_provide_setting_from_MOM_then_disable() . . . . .	340
7.21.3.11 check_for_shutdown() . . . . .	340
7.21.3.12 check_for_shutdown_with_termination() . . . . .	340
7.21.3.13 check_freeze() . . . . .	341
7.21.3.14 check_HLA_save_directory() . . . . .	341
7.21.3.15 clear_running_feds() . . . . .	341
7.21.3.16 complete_restore() . . . . .	341
7.21.3.17 convert_sync_pts() . . . . .	341
7.21.3.18 copy_running_feds_into_known_feds() . . . . .	342
7.21.3.19 create_and_join_federation() . . . . .	342
7.21.3.20 create_federation() . . . . .	342
7.21.3.21 create_RTI_ambassador_and_connect() . . . . .	342
7.21.3.22 destroy() . . . . .	343
7.21.3.23 destroy_orphaned_federation() . . . . .	343

---

7.21.3.24 determine_federate_MOM_object_instance_names()	343
7.21.3.25 enable_async_delivery()	343
7.21.3.26 enable_MOM_auto_provide_setting()	344
7.21.3.27 enter_freeze()	344
7.21.3.28 exit_freeze()	344
7.21.3.29 federate_can_rejoin_federation()	344
7.21.3.30 federation_restored()	345
7.21.3.31 federation_saved()	345
7.21.3.32 federation_synchronized()	345
7.21.3.33 fix_FPU_control_word()	345
7.21.3.34 freeze_init()	346
7.21.3.35 get_execution_control()	346
7.21.3.36 get_fed_ambassador()	346
7.21.3.37 get_federate_name()	346
7.21.3.38 get_federate_type()	347
7.21.3.39 get_federation_name()	347
7.21.3.40 get_freeze_announced()	347
7.21.3.41 get_freeze_pending()	347
7.21.3.42 get_granted_fed_time()	348
7.21.3.43 get_granted_time()	348
7.21.3.44 get_joined_federate_handles()	348
7.21.3.45 get_lookahead()	348
7.21.3.46 get_lookahead_time()	349
7.21.3.47 get_manager()	349
7.21.3.48 get_MOM_HLAfederate_class_handle()	349
7.21.3.49 get_requested_fed_time()	349
7.21.3.50 get_requested_time()	350
7.21.3.51 get_restart()	350
7.21.3.52 get_restart_cfg()	350
7.21.3.53 get_RTI_ambassador()	350
7.21.3.54 get_running_feds_count()	351
7.21.3.55 get_stale_data_counter()	351
7.21.3.56 has_restore Been announced()	351
7.21.3.57 has_restore_process_restore_request_failed()	352
7.21.3.58 has_restore_process_restore_request_succeeded()	352
7.21.3.59 has_restore_request_failed()	352
7.21.3.60 has_restore_request_succeeded()	352
7.21.3.61 in_time_regulating_state()	353
7.21.3.62 inform_RTI_of_restore_completion()	353

---

7.21.3.63 initialize()	353
7.21.3.64 initialize_MOM_handles()	353
7.21.3.65 initiate_restore_announce()	354
7.21.3.66 initiate_save_announce()	354
7.21.3.67 is_a_required_startup_federate()	354
7.21.3.68 is_execution_member()	355
7.21.3.69 is_federate_executing()	355
7.21.3.70 is_federate_instance_id()	356
7.21.3.71 is_federation_created_by_federate()	356
7.21.3.72 is_HLA_save_and_restore_supported()	356
7.21.3.73 is_joined_federate() [1/2]	356
7.21.3.74 is_joined_federate() [2/2]	357
7.21.3.75 is_MOM_HLAfederate_class()	357
7.21.3.76 is_MOM_HLAfederation_class()	357
7.21.3.77 is_MOM_HLAfederation_instance_id()	358
7.21.3.78 is_required_federate()	358
7.21.3.79 is_start_to_restore()	359
7.21.3.80 is_time_advance_granted()	359
7.21.3.81 is_time_management_enabled()	359
7.21.3.82 is_zero_lookahead_time()	359
7.21.3.83 join_federation()	360
7.21.3.84 load_and_print_running_federate_names()	361
7.21.3.85 operator=()	361
7.21.3.86 perform_checkpoint()	361
7.21.3.87 perform_restore()	362
7.21.3.88 perform_time_advance_request()	362
7.21.3.89 post_checkpoint()	362
7.21.3.90 post_multiphase_initialization()	362
7.21.3.91 post_restore()	363
7.21.3.92 pre_multiphase_initialization()	363
7.21.3.93 print_requested_federation_restore_status()	363
7.21.3.94 print_restore_failure_reason()	363
7.21.3.95 print_save_failure_reason()	364
7.21.3.96 print_version()	364
7.21.3.97 process_requested_federation_restore_status()	364
7.21.3.98 process_requested_federation_save_status()	364
7.21.3.99 publish_interaction_class()	365
7.21.3.100 read_running_feds_file()	365
7.21.3.101 rebuild_federate_handles()	365

---

7.21.3.102 register_generic_sync_point()	366
7.21.3.103 reinstate_logged_sync_pts()	366
7.21.3.104 remove_federate_instance_id()	366
7.21.3.105 remove_MOM_HLAfederate_instance_id()	367
7.21.3.106 remove_MOM_HLAfederation_instance_id()	367
7.21.3.107 request_attribute_update()	367
7.21.3.108 request_federation_restore_status()	368
7.21.3.109 request_federation_save()	368
7.21.3.110 request_federation_save_status()	368
7.21.3.111 requested_federation_restore_status()	368
7.21.3.112 resign()	369
7.21.3.113 resign_so_we_can_rejoin()	369
7.21.3.114 restart_checkpoint()	369
7.21.3.115 restart_initialization()	369
7.21.3.116 restore_checkpoint()	370
7.21.3.117 restore_federate_handles_from_MOM()	370
7.21.3.118 restore_orig_MOM_auto_provide_setting()	370
7.21.3.119 send_interaction()	370
7.21.3.120 set_all_federate_MOM_instance_handles_by_name()	371
7.21.3.121 set_announce_save()	371
7.21.3.122 set_checkpoint_file_name()	371
7.21.3.123 set_federate_has_begun_execution()	371
7.21.3.124 set_federation_name()	372
7.21.3.125 set_freeze_announced()	372
7.21.3.126 set_granted_time() [1/2]	372
7.21.3.127 set_granted_time() [2/2]	372
7.21.3.128 set_lookahead()	373
7.21.3.129 set_MOM_HLAfederate_instance_attributes()	373
7.21.3.130 set_MOM_HLAfederation_instance_attributes()	373
7.21.3.131 set_requested_time() [1/2]	374
7.21.3.132 set_requested_time() [2/2]	374
7.21.3.133 set_restart()	374
7.21.3.134 set_restart_cfg()	374
7.21.3.135 set_restore_begun()	375
7.21.3.136 set_restore_completed()	375
7.21.3.137 set_restore_failed()	375
7.21.3.138 set_restore_is_imminent()	375
7.21.3.139 set_restore_name()	375
7.21.3.140 set_restore_request_failed()	376

---

7.21.3.141 set_restore_request_succeeded()	376
7.21.3.142 set_save_completed()	376
7.21.3.143 set_save_name()	376
7.21.3.144 set_start_to_restore()	376
7.21.3.145 set_start_to_save() [1/2]	377
7.21.3.146 set_start_to_save() [2/2]	377
7.21.3.147 set_startup()	377
7.21.3.148 set_time_advance_grant()	377
7.21.3.149 set_time_constrained_state()	378
7.21.3.150 set_time_regulation_state()	378
7.21.3.151 setup()	378
7.21.3.152 setup_checkpoint()	379
7.21.3.153 setup_restore()	379
7.21.3.154 setup_time_constrained()	379
7.21.3.155 setup_time_management()	380
7.21.3.156 setup_time_regulation()	380
7.21.3.157 should_print()	380
7.21.3.158 should_publish_data()	381
7.21.3.159 shutdown()	381
7.21.3.160 shutdown_time_constrained()	382
7.21.3.161 shutdown_time_management()	382
7.21.3.162 shutdown_time_regulating()	382
7.21.3.163 subscribe_attributes()	382
7.21.3.164 sync_point_registration_failed()	383
7.21.3.165 sync_point_registration_succeeded()	383
7.21.3.166 time_advance_request()	383
7.21.3.167 time_advance_request_to_GALT()	384
7.21.3.168 time_advance_request_to_GALT_LCTS_multiple()	384
7.21.3.169 un_freeze()	384
7.21.3.170 unfreeze()	384
7.21.3.171 unpublish_interaction_class()	385
7.21.3.172 unsubscribe_all_HLAfederate_class_attributes_from_MOM()	385
7.21.3.173 unsubscribe_all_HLAfederation_class_attributes_from_MOM()	385
7.21.3.174 unsubscribe_attributes()	385
7.21.3.175 update_running_feds()	386
7.21.3.176 wait_for_federation_restore_begun()	386
7.21.3.177 wait_for_federation_restore_failed_callback_to_complete()	386
7.21.3.178 wait_for_federation_restore_to_complete()	386
7.21.3.179 wait_for_required_federates_to_join()	386

---

7.21.3.180 wait_for_restore_request_callback()	387
7.21.3.181 wait_for_restore_status_to_complete()	387
7.21.3.182 wait_for_save_status_to_complete()	387
7.21.3.183 wait_for_time_advance_grant() [1/2]	387
7.21.3.184 wait_for_time_advance_grant() [2/2]	388
7.21.3.185 wait_until_federation_is_ready_to_restore()	388
7.21.3.186 write_running_feds_file()	388
21.4 Friends And Related Function Documentation	389
7.21.4.1 init_attrTrickHLA_Federate	389
7.21.4.2 InputProcessor	389
21.5 Field Documentation	389
7.21.5.1 all_federates_joined	389
7.21.5.2 announce_freeze	389
7.21.5.3 announce_restore	389
7.21.5.4 announce_save	389
7.21.5.5 auto_provide_setting	390
7.21.5.6 can_rejoin_federation	390
7.21.5.7 checkpoint_file_name	390
7.21.5.8 checkpoint_rt_itimer	390
7.21.5.9 cstr_restore_label	390
7.21.5.10 cstr_save_label	390
7.21.5.11 enable_known_feds	391
7.21.5.12 execution_control	391
7.21.5.13 execution_has_begun	391
7.21.5.14 federate_ambassador	391
7.21.5.15 federate_has_been_restarted	391
7.21.5.16 federate_id	392
7.21.5.17 federation_created_by_federate	392
7.21.5.18 federation_exists	392
7.21.5.19 federation_joined	392
7.21.5.20 federation_name	392
7.21.5.21 federation_restore_failed_callback_complete	392
7.21.5.22 FOM_modules	393
7.21.5.23 freeze_delay_frames	393
7.21.5.24 freeze_the_federation	393
7.21.5.25 got_startup_sp	393
7.21.5.26 granted_time	393
7.21.5.27 HLA_save_directory	394
7.21.5.28 HLA_time	394

7.21.5.29 initiate_restore_flag . . . . .	394
7.21.5.30 initiate_save_flag . . . . .	394
7.21.5.31 joined_federate_handles . . . . .	394
7.21.5.32 joined_federate_name_map . . . . .	394
7.21.5.33 joined_federate_names . . . . .	395
7.21.5.34 known_feds . . . . .	395
7.21.5.35 known_feds_count . . . . .	395
7.21.5.36 local_settings . . . . .	395
7.21.5.37 lookahead . . . . .	395
7.21.5.38 lookahead_time . . . . .	396
7.21.5.39 make_copy_of_run_directory . . . . .	396
7.21.5.40 manager . . . . .	396
7.21.5.41 MIM_module . . . . .	396
7.21.5.42 MOM_HLAautoProvide_handle . . . . .	396
7.21.5.43 MOM_HLAautoProvide_param_handle . . . . .	397
7.21.5.44 MOM_HLAfederate_class_handle . . . . .	397
7.21.5.45 MOM_HLAfederate_handle . . . . .	397
7.21.5.46 mom_HLAfederate_inst_name_map . . . . .	397
7.21.5.47 MOM_HLAfederateName_handle . . . . .	397
7.21.5.48 MOM_HLAfederatesInFederation_handle . . . . .	397
7.21.5.49 MOM_HLAfederateType_handle . . . . .	398
7.21.5.50 MOM_HLAfederation_class_handle . . . . .	398
7.21.5.51 mom_HLAfederation_instance_name_map . . . . .	398
7.21.5.52 MOM_HLAswitches_class_handle . . . . .	398
7.21.5.53 name . . . . .	398
7.21.5.54 orig_auto_provide_setting . . . . .	399
7.21.5.55 prev_restore_process . . . . .	399
7.21.5.56 publish_data . . . . .	399
7.21.5.57 requested_time . . . . .	399
7.21.5.58 restart_cfg_flag . . . . .	399
7.21.5.59 restart_flag . . . . .	399
7.21.5.60 restore_begun . . . . .	400
7.21.5.61 restore_completed . . . . .	400
7.21.5.62 restore_failed . . . . .	400
7.21.5.63 restore_in_progress . . . . .	400
7.21.5.64 restore_is_imminent . . . . .	400
7.21.5.65 restore_label_generated . . . . .	400
7.21.5.66 restore_name . . . . .	401
7.21.5.67 restore_process . . . . .	401

---

7.21.5.68 restore_request_complete . . . . .	401
7.21.5.69 RTI_ambassador . . . . .	401
7.21.5.70 running_feds . . . . .	402
7.21.5.71 running_feds_count . . . . .	402
7.21.5.72 running_feds_count_at_time_of_restore . . . . .	402
7.21.5.73 save_completed . . . . .	402
7.21.5.74 save_label_generated . . . . .	402
7.21.5.75 save_name . . . . .	403
7.21.5.76 save_request_complete . . . . .	403
7.21.5.77 shutdown_called . . . . .	403
7.21.5.78 stale_data_counter . . . . .	403
7.21.5.79 start_to_restore . . . . .	403
7.21.5.80 start_to_save . . . . .	403
7.21.5.81 str_restore_label . . . . .	404
7.21.5.82 str_save_label . . . . .	404
7.21.5.83 time_adv_grant . . . . .	404
7.21.5.84 time_constrained . . . . .	404
7.21.5.85 time_constrained_state . . . . .	404
7.21.5.86 time_management . . . . .	405
7.21.5.87 time_regulating . . . . .	405
7.21.5.88 time_regulating_state . . . . .	405
7.21.5.89 type . . . . .	405
7.21.5.90 unfreeze_after_save . . . . .	405
7.21.5.91 ws_restore_label . . . . .	405
7.21.5.92 ws_save_label . . . . .	406
7.22 IMSim::FreezeInteractionHandler Class Reference . . . . .	406
7.22.1 Detailed Description . . . . .	408
7.22.2 Constructor & Destructor Documentation . . . . .	408
7.22.2.1 FreezeInteractionHandler() [1/2] . . . . .	408
7.22.2.2 ~FreezeInteractionHandler() . . . . .	408
7.22.2.3 FreezeInteractionHandler() [2/2] . . . . .	408
7.22.3 Member Function Documentation . . . . .	408
7.22.3.1 check_values() . . . . .	408
7.22.3.2 get_address_of_interaction_time() . . . . .	409
7.22.3.3 operator=( ) . . . . .	409
7.22.3.4 receive_interaction() . . . . .	409
7.22.3.5 send_scenario_freeze_interaction() . . . . .	409
7.22.4 Friends And Related Function Documentation . . . . .	410
7.22.4.1 init_attrTrickHLA__FreezeInteractionHandler . . . . .	410

---

---

7.22.4.2 InputProcessor	410
7.22.5 Field Documentation	410
7.22.5.1 time	410
7.23 TrickHLA::Int64Interval Class Reference	411
7.23.1 Detailed Description	412
7.23.2 Constructor & Destructor Documentation	412
7.23.2.1 Int64Interval() [1/6]	412
7.23.2.2 Int64Interval() [2/6]	412
7.23.2.3 Int64Interval() [3/6]	413
7.23.2.4 Int64Interval() [4/6]	413
7.23.2.5 Int64Interval() [5/6]	413
7.23.2.6 Int64Interval() [6/6]	413
7.23.2.7 ~Int64Interval()	414
7.23.3 Member Function Documentation	414
7.23.3.1 get()	414
7.23.3.2 getDoubleTime()	414
7.23.3.3 getMicros()	414
7.23.3.4 getSeconds()	415
7.23.3.5 getTimelnMicros()	415
7.23.3.6 operator=() [1/3]	415
7.23.3.7 operator=() [2/3]	415
7.23.3.8 operator=() [3/3]	416
7.23.3.9 operator>() [1/2]	416
7.23.3.10 operator>() [2/2]	416
7.23.3.11 setTo() [1/3]	417
7.23.3.12 setTo() [2/3]	417
7.23.3.13 setTo() [3/3]	417
7.23.3.14 toMicroseconds()	418
7.23.3.15 toString()	418
7.23.4 Friends And Related Function Documentation	418
7.23.4.1 init_attrTrickHLA__Int64Interval	418
7.23.4.2 InputProcessor	418
7.23.5 Field Documentation	418
7.23.5.1 _HLAinterval	419
7.24 TrickHLA::Int64Time Class Reference	419
7.24.1 Detailed Description	421
7.24.2 Constructor & Destructor Documentation	421
7.24.2.1 Int64Time() [1/6]	421
7.24.2.2 Int64Time() [2/6]	421

---

7.24.2.3 <code>Int64Time()</code> [3/6] . . . . .	421
7.24.2.4 <code>Int64Time()</code> [4/6] . . . . .	422
7.24.2.5 <code>Int64Time()</code> [5/6] . . . . .	422
7.24.2.6 <code>Int64Time()</code> [6/6] . . . . .	422
7.24.2.7 <code>~Int64Time()</code> . . . . .	422
7.24.3 Member Function Documentation . . . . .	422
7.24.3.1 <code>decode()</code> . . . . .	423
7.24.3.2 <code>get()</code> . . . . .	423
7.24.3.3 <code>getDoubleTime()</code> . . . . .	423
7.24.3.4 <code>getMicros()</code> . . . . .	424
7.24.3.5 <code>getSeconds()</code> . . . . .	424
7.24.3.6 <code>getTimeInMicros()</code> . . . . .	424
7.24.3.7 <code>operator"!="()</code> [1/2] . . . . .	424
7.24.3.8 <code>operator"!="()</code> [2/2] . . . . .	425
7.24.3.9 <code>operator%</code> . . . . .	425
7.24.3.10 <code>operator*()</code> . . . . .	425
7.24.3.11 <code>operator+()</code> [1/2] . . . . .	426
7.24.3.12 <code>operator+()</code> [2/2] . . . . .	426
7.24.3.13 <code>operator+=()</code> [1/2] . . . . .	426
7.24.3.14 <code>operator+=()</code> [2/2] . . . . .	427
7.24.3.15 <code>operator-()</code> [1/2] . . . . .	427
7.24.3.16 <code>operator-()</code> [2/2] . . . . .	427
7.24.3.17 <code>operator/()</code> . . . . .	428
7.24.3.18 <code>operator&lt;()</code> . . . . .	428
7.24.3.19 <code>operator&lt;=()</code> . . . . .	428
7.24.3.20 <code>operator=()</code> [1/3] . . . . .	429
7.24.3.21 <code>operator=()</code> [2/3] . . . . .	429
7.24.3.22 <code>operator=()</code> [3/3] . . . . .	429
7.24.3.23 <code>operator==()</code> [1/2] . . . . .	430
7.24.3.24 <code>operator==()</code> [2/2] . . . . .	430
7.24.3.25 <code>operator&gt;()</code> [1/3] . . . . .	430
7.24.3.26 <code>operator&gt;()</code> [2/3] . . . . .	431
7.24.3.27 <code>operator&gt;()</code> [3/3] . . . . .	431
7.24.3.28 <code>operator&gt;=()</code> . . . . .	431
7.24.3.29 <code>setTo()</code> [1/4] . . . . .	432
7.24.3.30 <code>setTo()</code> [2/4] . . . . .	432
7.24.3.31 <code>setTo()</code> [3/4] . . . . .	432
7.24.3.32 <code>setTo()</code> [4/4] . . . . .	433
7.24.3.33 <code>toString()</code> . . . . .	433

---

7.24.4 Friends And Related Function Documentation . . . . .	433
7.24.4.1 init_attrTrickHLA__Int64Time . . . . .	433
7.24.4.2 InputProcessor . . . . .	433
7.24.5 Field Documentation . . . . .	433
7.24.5.1 _HLAtime . . . . .	433
7.25 TrickHLA::Interaction Class Reference . . . . .	434
7.25.1 Detailed Description . . . . .	437
7.25.2 Constructor & Destructor Documentation . . . . .	437
7.25.2.1 Interaction() [1/2] . . . . .	437
7.25.2.2 ~Interaction() . . . . .	437
7.25.2.3 Interaction() [2/2] . . . . .	438
7.25.3 Member Function Documentation . . . . .	438
7.25.3.1 extract_data() . . . . .	438
7.25.3.2 get_class_handle() . . . . .	438
7.25.3.3 get_fed_lookahead() . . . . .	438
7.25.3.4 get_federate() . . . . .	439
7.25.3.5 get_FOM_name() . . . . .	439
7.25.3.6 get_granted_fed_time() . . . . .	439
7.25.3.7 get_handler() . . . . .	440
7.25.3.8 get_manager() . . . . .	440
7.25.3.9 get_parameter_count() . . . . .	440
7.25.3.10 get_parameters() . . . . .	440
7.25.3.11 get_preferred_order() . . . . .	441
7.25.3.12 get_RTI_ambassador() . . . . .	441
7.25.3.13 initialize() . . . . .	441
7.25.3.14 is_changed() . . . . .	441
7.25.3.15 is_publish() . . . . .	442
7.25.3.16 is_subscribe() . . . . .	442
7.25.3.17 lock() . . . . .	442
7.25.3.18 mark_changed() . . . . .	442
7.25.3.19 mark_unchanged() . . . . .	442
7.25.3.20 operator=(()) . . . . .	443
7.25.3.21 process_interaction() . . . . .	443
7.25.3.22 publish_interaction() . . . . .	443
7.25.3.23 remove() . . . . .	443
7.25.3.24 send() [1/2] . . . . .	444
7.25.3.25 send() [2/2] . . . . .	444
7.25.3.26 set_class_handle() . . . . .	444
7.25.3.27 set_FOM_name() . . . . .	445

---

7.25.3.28 set_handler()	445
7.25.3.29 set_parameter_count()	445
7.25.3.30 set_parameters()	446
7.25.3.31 set_publish()	446
7.25.3.32 set_subscribe()	446
7.25.3.33 set_user_supplied_tag()	446
7.25.3.34 setup_preferred_order_with_RTI()	447
7.25.3.35 should_print()	447
7.25.3.36 subscribe_to_interaction()	447
7.25.3.37 unlock()	448
7.25.3.38 unpublish_interaction()	448
7.25.3.39 unsubscribe_from_interaction()	448
25.4 Friends And Related Function Documentation	448
7.25.4.1 init_attrTrickHLA__Interaction	448
7.25.4.2 InputProcessor	448
25.5 Field Documentation	448
7.25.5.1 changed	448
7.25.5.2 class_handle	449
7.25.5.3 FOM_name	449
7.25.5.4 handler	449
7.25.5.5 manager	449
7.25.5.6 mutex	449
7.25.5.7 param_count	450
7.25.5.8 parameters	450
7.25.5.9 preferred_order	450
7.25.5.10 publish	450
7.25.5.11 received_as_TSO	450
7.25.5.12 subscribe	450
7.25.5.13 time	451
7.25.5.14 user_supplied_tag	451
7.25.5.15 user_supplied_tag_capacity	451
7.25.5.16 user_supplied_tag_size	451
26 TrickHLA::InteractionHandler Class Reference	451
7.26.1 Detailed Description	453
7.26.2 Constructor & Destructor Documentation	453
7.26.2.1 InteractionHandler() [1/2]	453
7.26.2.2 ~InteractionHandler()	454
7.26.2.3 InteractionHandler() [2/2]	454
7.26.3 Member Function Documentation	454

---

---

7.26.3.1 <code>get_cte_time()</code> . . . . .	454
7.26.3.2 <code>get_fed_lookahead()</code> . . . . .	454
7.26.3.3 <code>get_granted_fed_time()</code> . . . . .	454
7.26.3.4 <code>get_scenario_time()</code> . . . . .	455
7.26.3.5 <code>get_sim_time()</code> . . . . .	455
7.26.3.6 <code>initialize_callback()</code> . . . . .	455
7.26.3.7 <code>operator=()</code> . . . . .	456
7.26.3.8 <code>receive_interaction()</code> . . . . .	456
7.26.3.9 <code>send_interaction() [1/4]</code> . . . . .	456
7.26.3.10 <code>send_interaction() [2/4]</code> . . . . .	456
7.26.3.11 <code>send_interaction() [3/4]</code> . . . . .	457
7.26.3.12 <code>send_interaction() [4/4]</code> . . . . .	457
7.26.3.13 <code>should_print()</code> . . . . .	457
7.26.4 Friends And Related Function Documentation . . . . .	458
7.26.4.1 <code>init_attrTrickHLA__InteractionHandler</code> . . . . .	458
7.26.4.2 <code>InputProcessor</code> . . . . .	458
7.26.5 Field Documentation . . . . .	458
7.26.5.1 <code>interaction</code> . . . . .	458
7.27 <code>TrickHLA::InteractionItem</code> Class Reference . . . . .	458
7.27.1 Detailed Description . . . . .	461
7.27.2 Constructor & Destructor Documentation . . . . .	461
7.27.2.1 <code>InteractionItem() [1/4]</code> . . . . .	461
7.27.2.2 <code>InteractionItem() [2/4]</code> . . . . .	461
7.27.2.3 <code>InteractionItem() [3/4]</code> . . . . .	461
7.27.2.4 <code>~InteractionItem()</code> . . . . .	462
7.27.2.5 <code>InteractionItem() [4/4]</code> . . . . .	462
7.27.3 Member Function Documentation . . . . .	462
7.27.3.1 <code>checkpoint_queue()</code> . . . . .	462
7.27.3.2 <code>clear_parm_items()</code> . . . . .	462
7.27.3.3 <code>initialize()</code> . . . . .	463
7.27.3.4 <code>is_receive_order()</code> . . . . .	463
7.27.3.5 <code>is_timestamp_order()</code> . . . . .	463
7.27.3.6 <code>operator=()</code> . . . . .	464
7.27.3.7 <code>restore_queue()</code> . . . . .	464
7.27.4 Friends And Related Function Documentation . . . . .	464
7.27.4.1 <code>init_attrTrickHLA__InteractionItem</code> . . . . .	464
7.27.4.2 <code>InputProcessor</code> . . . . .	464
7.27.5 Field Documentation . . . . .	464
7.27.5.1 <code>index</code> . . . . .	464

---

---

7.27.5.2 <code>interaction_type</code>	464
7.27.5.3 <code>order_is_TSO</code>	465
7.27.5.4 <code>parameter_queue</code>	465
7.27.5.5 <code>parm_items</code>	465
7.27.5.6 <code>parm_items_count</code>	465
7.27.5.7 <code>time</code>	465
7.27.5.8 <code>user_supplied_tag</code>	465
7.27.5.9 <code>user_supplied_tag_size</code>	466
7.28 TrickHLA::Item Class Reference	466
7.28.1 Detailed Description	467
7.28.2 Constructor & Destructor Documentation	467
7.28.2.1 <code>Item()</code> [1/2]	467
7.28.2.2 <code>~Item()</code>	467
7.28.2.3 <code>Item()</code> [2/2]	467
7.28.3 Member Function Documentation	467
7.28.3.1 <code>operator=()</code>	468
7.28.4 Friends And Related Function Documentation	468
7.28.4.1 <code>init_attrTrickHLA__Item</code>	468
7.28.4.2 <code>InputProcessor</code>	468
7.28.5 Field Documentation	468
7.28.5.1 <code>next</code>	468
7.29 TrickHLA::ItemQueue Class Reference	468
7.29.1 Detailed Description	470
7.29.2 Constructor & Destructor Documentation	470
7.29.2.1 <code>ItemQueue()</code> [1/2]	470
7.29.2.2 <code>~ItemQueue()</code>	470
7.29.2.3 <code>ItemQueue()</code> [2/2]	471
7.29.3 Member Function Documentation	471
7.29.3.1 <code>dump_head_pointers()</code>	471
7.29.3.2 <code>empty()</code>	471
7.29.3.3 <code>front()</code>	471
7.29.3.4 <code>lock()</code>	472
7.29.3.5 <code>next()</code>	472
7.29.3.6 <code>operator=()</code>	472
7.29.3.7 <code>pop()</code>	472
7.29.3.8 <code>push()</code>	472
7.29.3.9 <code>rewind()</code>	473
7.29.3.10 <code>size()</code>	473
7.29.3.11 <code>unlock()</code>	473

---

---

7.29.4 Friends And Related Function Documentation . . . . .	473
7.29.4.1 init_attrTrickHLA__ItemQueue . . . . .	473
7.29.4.2 InputProcessor . . . . .	473
7.29.5 Field Documentation . . . . .	474
7.29.5.1 count . . . . .	474
7.29.5.2 head . . . . .	474
7.29.5.3 mutex . . . . .	474
7.29.5.4 original_head . . . . .	474
7.29.5.5 tail . . . . .	474
7.30 TrickHLA::KnownFederate Class Reference . . . . .	474
7.30.1 Detailed Description . . . . .	475
7.30.2 Constructor & Destructor Documentation . . . . .	475
7.30.2.1 KnownFederate() [1/2] . . . . .	475
7.30.2.2 ~KnownFederate() . . . . .	475
7.30.2.3 KnownFederate() [2/2] . . . . .	476
7.30.3 Member Function Documentation . . . . .	476
7.30.3.1 operator=() . . . . .	476
7.30.4 Friends And Related Function Documentation . . . . .	476
7.30.4.1 init_attrTrickHLA__KnownFederate . . . . .	476
7.30.4.2 InputProcessor . . . . .	476
7.30.5 Field Documentation . . . . .	476
7.30.5.1 MOM_instance_name . . . . .	476
7.30.5.2 name . . . . .	476
7.30.5.3 required . . . . .	477
7.31 TrickHLA::LagCompensation Class Reference . . . . .	477
7.31.1 Detailed Description . . . . .	479
7.31.2 Constructor & Destructor Documentation . . . . .	479
7.31.2.1 LagCompensation() [1/2] . . . . .	479
7.31.2.2 ~LagCompensation() . . . . .	479
7.31.2.3 LagCompensation() [2/2] . . . . .	479
7.31.3 Member Function Documentation . . . . .	479
7.31.3.1 get_attribute() . . . . .	479
7.31.3.2 get_attribute_and_validate() . . . . .	480
7.31.3.3 get_cte_time() . . . . .	480
7.31.3.4 get_fed_lookahead() . . . . .	480
7.31.3.5 get_granted_fed_time() . . . . .	480
7.31.3.6 get_scenario_time() . . . . .	481
7.31.3.7 initialize_callback() . . . . .	481
7.31.3.8 operator=() . . . . .	481

---

---

7.31.3.9 receive_lag_compensation() . . . . .	481
7.31.3.10 send_lag_compensation() . . . . .	482
7.31.3.11 should_print() . . . . .	482
7.31.4 Friends And Related Function Documentation . . . . .	482
7.31.4.1 init_attrTrickHLA__LagCompensation . . . . .	482
7.31.4.2 InputProcessor . . . . .	482
7.31.5 Field Documentation . . . . .	482
7.31.5.1 object . . . . .	482
7.32 TrickHLA::LoggableSyncPnt Class Reference . . . . .	483
7.32.1 Detailed Description . . . . .	484
7.32.2 Constructor & Destructor Documentation . . . . .	484
7.32.2.1 LoggableSyncPnt() [1/2] . . . . .	484
7.32.2.2 ~LoggableSyncPnt() . . . . .	484
7.32.2.3 LoggableSyncPnt() [2/2] . . . . .	484
7.32.3 Member Function Documentation . . . . .	484
7.32.3.1 clear() . . . . .	484
7.32.3.2 operator=(()) . . . . .	484
7.32.4 Friends And Related Function Documentation . . . . .	485
7.32.4.1 init_attrTrickHLA__LoggableSyncPnt . . . . .	485
7.32.4.2 InputProcessor . . . . .	485
7.32.5 Field Documentation . . . . .	485
7.32.5.1 label . . . . .	485
7.32.5.2 state . . . . .	485
7.33 TrickHLA::LoggableTimedSyncPnt Class Reference . . . . .	485
7.33.1 Detailed Description . . . . .	487
7.33.2 Constructor & Destructor Documentation . . . . .	487
7.33.2.1 LoggableTimedSyncPnt() [1/2] . . . . .	487
7.33.2.2 ~LoggableTimedSyncPnt() . . . . .	487
7.33.2.3 LoggableTimedSyncPnt() [2/2] . . . . .	487
7.33.3 Member Function Documentation . . . . .	487
7.33.3.1 operator=(()) . . . . .	487
7.33.4 Friends And Related Function Documentation . . . . .	487
7.33.4.1 init_attrTrickHLA__LoggableTimedSyncPnt . . . . .	487
7.33.4.2 InputProcessor . . . . .	488
7.33.5 Field Documentation . . . . .	488
7.33.5.1 time . . . . .	488
7.34 TrickHLA::Manager Class Reference . . . . .	488
7.34.1 Detailed Description . . . . .	494
7.34.2 Constructor & Destructor Documentation . . . . .	494

---

7.34.2.1 Manager() [1/2] . . . . .	494
7.34.2.2 ~Manager() . . . . .	494
7.34.2.3 Manager() [2/2] . . . . .	494
7.34.3 Member Function Documentation . . . . .	495
7.34.3.1 add_object_to_map() . . . . .	495
7.34.3.2 clear_init_sync_points() . . . . .	495
7.34.3.3 clear_interactions() . . . . .	495
7.34.3.4 determine_job_cycle_time() . . . . .	495
7.34.3.5 discover_object_instance() . . . . .	495
7.34.3.6 dump_interactions() . . . . .	496
7.34.3.7 get_debug_handler() . . . . .	496
7.34.3.8 get_execution_configuration() . . . . .	496
7.34.3.9 get_execution_control() . . . . .	497
7.34.3.10 get_fed_lookahead() . . . . .	497
7.34.3.11 get_federate() . . . . .	497
7.34.3.12 get_granted_fed_time() . . . . .	497
7.34.3.13 get_granted_time() . . . . .	498
7.34.3.14 get_interaction_count() . . . . .	498
7.34.3.15 get_interactions() . . . . .	498
7.34.3.16 get_object_count() . . . . .	498
7.34.3.17 get_objects() . . . . .	499
7.34.3.18 get_RTI_ambassador() . . . . .	499
7.34.3.19 get_trickhla_object() [1/2] . . . . .	499
7.34.3.20 get_trickhla_object() [2/2] . . . . .	499
7.34.3.21 get_unregistered_object() . . . . .	500
7.34.3.22 get_unregistered_remote_object() . . . . .	500
7.34.3.23 grant_pull_request() . . . . .	501
7.34.3.24 has_federate_been_restored() . . . . .	501
7.34.3.25 initialization_execution_control() . . . . .	501
7.34.3.26 initialize() . . . . .	501
7.34.3.27 initiate_federation_save() . . . . .	501
7.34.3.28 is_execution_configuration_used() . . . . .	502
7.34.3.29 is_late_joining_federate() . . . . .	502
7.34.3.30 is_restore_federate() . . . . .	502
7.34.3.31 is_RTI_ready() . . . . .	502
7.34.3.32 is_shutdown() . . . . .	503
7.34.3.33 is_this_a_rejoining_federate() . . . . .	503
7.34.3.34 mark_object_as_deleted_from_federation() . . . . .	503
7.34.3.35 object_instance_name_reservation_failed() . . . . .	504

---

7.34.3.36 object_instance_name_reservation_succeeded()	504
7.34.3.37 operator=()	504
7.34.3.38 process_deleted_objects()	504
7.34.3.39 process_interactions()	505
7.34.3.40 process_ownership()	505
7.34.3.41 provide_attribute_update()	505
7.34.3.42 publish()	505
7.34.3.43 publish_and_subscribe()	506
7.34.3.44 pull_ownership()	506
7.34.3.45 pull_ownership_upon_rejoin()	506
7.34.3.46 push_ownership()	506
7.34.3.47 receive_cyclic_data()	506
7.34.3.48 receive_init_data() [1/2]	507
7.34.3.49 receive_init_data() [2/2]	507
7.34.3.50 receive_interaction()	507
7.34.3.51 register_objects_with_RTI()	508
7.34.3.52 release_ownership()	508
7.34.3.53 request_data_update() [1/2]	508
7.34.3.54 request_data_update() [2/2]	509
7.34.3.55 reserve_object_names_with_RTI()	509
7.34.3.56 reset_mgr_initialized()	509
7.34.3.57 restart_initialization()	509
7.34.3.58 restore_interactions()	510
7.34.3.59 send_cyclic_data()	510
7.34.3.60 send_init_data() [1/2]	510
7.34.3.61 send_init_data() [2/2]	510
7.34.3.62 send_requested_data()	511
7.34.3.63 set_all_object_instance_handles_by_name()	511
7.34.3.64 set_execution_configuration()	511
7.34.3.65 set_object_instance_handles_by_name()	512
7.34.3.66 setup()	512
7.34.3.67 setup_all_ref_attributes()	512
7.34.3.68 setup_all_RTI_handles()	513
7.34.3.69 setup_checkpoint()	513
7.34.3.70 setup_checkpoint_interactions()	513
7.34.3.71 setup_interaction_ref_attributes()	513
7.34.3.72 setup_interaction_RTI_handles()	514
7.34.3.73 setup_object_ref_attributes()	514
7.34.3.74 setup_object_RTI_handles()	515

---

7.34.3.75 setup_preferred_order_with_RTI()	515
7.34.3.76 should_print()	515
7.34.3.77 shutdown()	516
7.34.3.78 start_federation_save()	516
7.34.3.79 start_federation_save_at_scenario_time()	516
7.34.3.80 start_federation_save_at_sim_time()	517
7.34.3.81 subscribe()	517
7.34.3.82 unpublish()	517
7.34.3.83 unsubscribe()	517
7.34.3.84 wait_for_init_sync_point()	518
7.34.3.85 wait_on_discovery_of_objects()	518
7.34.3.86 wait_on_registration_of_required_objects()	518
7.34.3.87 wait_on_reservation_of_object_names()	518
7.34.4 Friends And Related Function Documentation	519
7.34.4.1 Federate	519
7.34.4.2 init_attrTrickHLA__Manager	519
7.34.4.3 InputProcessor	519
7.34.5 Field Documentation	519
7.34.5.1 check_interactions	519
7.34.5.2 check_interactions_count	519
7.34.5.3 debug_handler	519
7.34.5.4 execution_control	520
7.34.5.5 federate	520
7.34.5.6 federate_has_been_restored	520
7.34.5.7 initiated_a_federation_save	520
7.34.5.8 inter_count	521
7.34.5.9 interactions	521
7.34.5.10 interactions_queue	521
7.34.5.11 job_cycle_time	521
7.34.5.12 mgr_initialized	521
7.34.5.13 obj_count	522
7.34.5.14 object_map	522
7.34.5.15 objects	522
7.34.5.16 rejoining_federate	522
7.34.5.17 restore_determined	522
7.34.5.18 restore_federate	523
7.34.5.19 restore_federation	523
7.34.5.20 restore_file_name	523
7.34.5.21 shutdown_called	523

---

7.35 SpaceFOM::MTRInteractionHandler Class Reference . . . . .	523
7.35.1 Detailed Description . . . . .	525
7.35.2 Constructor & Destructor Documentation . . . . .	526
7.35.2.1 MTRInteractionHandler() [1/3] . . . . .	526
7.35.2.2 ~MTRInteractionHandler() . . . . .	526
7.35.2.3 MTRInteractionHandler() [2/3] . . . . .	526
7.35.2.4 MTRInteractionHandler() [3/3] . . . . .	526
7.35.3 Member Function Documentation . . . . .	526
7.35.3.1 get_address_of_interaction_mode() . . . . .	526
7.35.3.2 operator=( ) . . . . .	527
7.35.3.3 receive_interaction() . . . . .	527
7.35.3.4 send_interaction() . . . . .	527
7.35.3.5 set_name() . . . . .	528
7.35.4 Friends And Related Function Documentation . . . . .	528
7.35.4.1 init_attrSpaceFOM__MTRInteractionHandler . . . . .	528
7.35.4.2 InputProcessor . . . . .	528
7.35.5 Field Documentation . . . . .	528
7.35.5.1 cte_time . . . . .	528
7.35.5.2 granted_time . . . . .	528
7.35.5.3 mtr_mode . . . . .	528
7.35.5.4 mtr_mode_int . . . . .	529
7.35.5.5 name . . . . .	529
7.35.5.6 receive_cnt . . . . .	529
7.35.5.7 scenario_time . . . . .	529
7.35.5.8 send_cnt . . . . .	529
7.35.5.9 sim_time . . . . .	529
7.36 TrickHLA::Object Class Reference . . . . .	530
7.36.1 Detailed Description . . . . .	538
7.36.2 Constructor & Destructor Documentation . . . . .	538
7.36.2.1 Object() [1/2] . . . . .	538
7.36.2.2 ~Object() . . . . .	538
7.36.2.3 Object() [2/2] . . . . .	538
7.36.3 Member Function Documentation . . . . .	538
7.36.3.1 any_attribute_published() . . . . .	538
7.36.3.2 any_attribute_subscribed() . . . . .	539
7.36.3.3 any_locally_owned_attribute() . . . . .	539
7.36.3.4 any_locally_owned_published_attribute() [1/2] . . . . .	539
7.36.3.5 any_locally_owned_published_attribute() [2/2] . . . . .	539
7.36.3.6 any_locally_owned_published_cyclic_data_ready_attribute() . . . . .	540

---

7.36.3.7 any_locally_owned_published_init_attribute()	540
7.36.3.8 any_locally_owned_published_requested_attribute()	540
7.36.3.9 any_remotely_owned_subscribed_attribute() [1/2]	540
7.36.3.10 any_remotely_owned_subscribed_attribute() [2/2]	541
7.36.3.11 any_remotely_owned_subscribed_cyclic_attribute()	541
7.36.3.12 any_remotely_owned_subscribed_init_attribute()	541
7.36.3.13 build_attribute_map()	541
7.36.3.14 create_attribute_set()	542
7.36.3.15 create_requested_attribute_set()	542
7.36.3.16 extract_data()	542
7.36.3.17 get_attribute() [1/3]	542
7.36.3.18 get_attribute() [2/3]	543
7.36.3.19 get_attribute() [3/3]	543
7.36.3.20 get_attribute_count()	544
7.36.3.21 get_attribute_FOM_names()	544
7.36.3.22 get_attributes()	544
7.36.3.23 get_class_handle()	544
7.36.3.24 get_fed_lookahead()	545
7.36.3.25 get_federate()	545
7.36.3.26 get_FOM_name()	545
7.36.3.27 get_granted_fed_time()	546
7.36.3.28 get_granted_time()	546
7.36.3.29 get_instance_handle()	546
7.36.3.30 get_lag_compensation_type()	546
7.36.3.31 get_last_update_time()	547
7.36.3.32 get_name()	547
7.36.3.33 get_name_string()	547
7.36.3.34 get_RTI_ambassador()	548
7.36.3.35 get_update_time_plus_lookahead()	548
7.36.3.36 grant_pull_request()	548
7.36.3.37 grant_push_request()	549
7.36.3.38 grant_push_request_pthread()	549
7.36.3.39 initialize()	549
7.36.3.40 is_attribute_update_requested()	549
7.36.3.41 is_changed()	550
7.36.3.42 is_create_HLA_instance()	550
7.36.3.43 is_instance_handle_valid()	550
7.36.3.44 is_name_registered()	551
7.36.3.45 is_name_required()	551

---

7.36.3.46 is_object_deleted_from_RTI()	551
7.36.3.47 is_required()	551
7.36.3.48 lock()	552
7.36.3.49 mark_all_attributes_as_nonlocal()	552
7.36.3.50 mark_changed()	552
7.36.3.51 mark_required()	552
7.36.3.52 mark_unchanged()	552
7.36.3.53 negotiated_attribute_ownership_divestiture()	553
7.36.3.54 notify_attribute_ownership_changed()	553
7.36.3.55 operator=( )	553
7.36.3.56 ownership_lock()	553
7.36.3.57 ownership_unlock()	553
7.36.3.58 pack_attribute_buffers()	553
7.36.3.59 pack_cyclic_attribute_buffers()	554
7.36.3.60 pack_init_attribute_buffers()	554
7.36.3.61 pack_requested_attribute_buffers()	554
7.36.3.62 process_deleted_object()	554
7.36.3.63 provide_attribute_update()	555
7.36.3.64 publish_object_attributes()	555
7.36.3.65 pull_ownership()	555
7.36.3.66 pull_ownership_upon_rejoin()	555
7.36.3.67 push_ownership()	556
7.36.3.68 receive_cyclic_data()	556
7.36.3.69 receive_init_data()	557
7.36.3.70 register_object_with_RTI()	557
7.36.3.71 release_ownership()	557
7.36.3.72 remove()	557
7.36.3.73 remove_object_instance()	558
7.36.3.74 request_attribute_value_update()	558
7.36.3.75 reserve_object_name_with_RTI()	558
7.36.3.76 restore_ownership_transfer_checkpointed_data()	558
7.36.3.77 send_cyclic_data()	559
7.36.3.78 send_init_data()	559
7.36.3.79 send_requested_data()	560
7.36.3.80 set_class_handle()	560
7.36.3.81 set_core_job_cycle_time()	561
7.36.3.82 set_create_HLA_instance()	561
7.36.3.83 set_divest_requested()	561
7.36.3.84 set_instance_handle()	561

---

7.36.3.85 set_instance_handle_and_name()	562
7.36.3.86 set_lag_compensation_type()	562
7.36.3.87 set_last_update_time()	562
7.36.3.88 set_name()	563
7.36.3.89 set_name_and_mark_changed()	563
7.36.3.90 set_name_registered()	563
7.36.3.91 set_name_unregistered()	563
7.36.3.92 set_pull_requested()	563
7.36.3.93 set_to_unblocking_cyclic_reads()	564
7.36.3.94 setup_ownership_transfer_checkpointed_data()	564
7.36.3.95 setup_preferred_order_with_RTI()	564
7.36.3.96 should_print()	564
7.36.3.97 stop_publishing_attributes()	565
7.36.3.98 stop_subscribing_attributes()	565
7.36.3.99 subscribe_to_object_attributes()	565
7.36.3.100 unlock()	565
7.36.3.101 unpack_attribute_buffers()	566
7.36.3.102 unpack_cyclic_attribute_buffers()	566
7.36.3.103 unpack_init_attribute_buffers()	566
7.36.3.104 unpublish_all_object_attributes()	566
7.36.3.105 unsubscribe_all_object_attributes()	566
7.36.3.106 wait_on_object_name_reservation()	567
7.36.3.107 wait_on_object_registration()	567
7.36.4 Friends And Related Function Documentation	567
7.36.4.1 init_attrTrickHLA__Object	567
7.36.4.2 InputProcessor	567
7.36.5 Field Documentation	567
7.36.5.1 any_attribute_FOM_specified_order	568
7.36.5.2 any_attribute_timestamp_order	568
7.36.5.3 attr_count	568
7.36.5.4 attr_update_requested	568
7.36.5.5 attribute_FOM_names	568
7.36.5.6 attribute_values_map	569
7.36.5.7 attributes	569
7.36.5.8 blocking_cyclic_read	569
7.36.5.9 changed	569
7.36.5.10 class_handle	570
7.36.5.11 clock	570
7.36.5.12 create_HLA_instance	570

---

7.36.5.13 data_changed . . . . .	570
7.36.5.14 deleted . . . . .	570
7.36.5.15 divest_requested . . . . .	570
7.36.5.16 FOM_name . . . . .	571
7.36.5.17 instance_handle . . . . .	571
7.36.5.18 lag_comp . . . . .	571
7.36.5.19 lag_comp_type . . . . .	571
7.36.5.20 last_update_time . . . . .	571
7.36.5.21 manager . . . . .	572
7.36.5.22 mutex . . . . .	572
7.36.5.23 name . . . . .	572
7.36.5.24 name_registered . . . . .	572
7.36.5.25 name_required . . . . .	572
7.36.5.26 object_deleted_from_RTI . . . . .	573
7.36.5.27 ownership . . . . .	573
7.36.5.28 ownership_mutex . . . . .	573
7.36.5.29 packing . . . . .	573
7.36.5.30 pull_requested . . . . .	573
7.36.5.31 removed_instance . . . . .	574
7.36.5.32 required . . . . .	574
7.36.5.33 rti_ambassador . . . . .	574
7.36.5.34 thla_attribute_map . . . . .	574
7.36.5.35 thla_reflected_attributes_queue . . . . .	574
7.36.5.36 time_plus_lookahead . . . . .	574
7.37 TrickHLA::ObjectDeleted Class Reference . . . . .	575
7.37.1 Detailed Description . . . . .	575
7.37.2 Constructor & Destructor Documentation . . . . .	575
7.37.2.1 ObjectDeleted() . . . . .	575
7.37.2.2 ~ObjectDeleted() . . . . .	576
7.37.3 Member Function Documentation . . . . .	576
7.37.3.1 deleted() . . . . .	576
7.37.4 Friends And Related Function Documentation . . . . .	576
7.37.4.1 init_attrTrickHLA__ObjectDeleted . . . . .	576
7.37.4.2 InputProcessor . . . . .	576
7.38 TrickHLA::OpaqueBuffer Class Reference . . . . .	576
7.38.1 Detailed Description . . . . .	578
7.38.2 Constructor & Destructor Documentation . . . . .	578
7.38.2.1 OpaqueBuffer() . . . . .	578
7.38.2.2 ~OpaqueBuffer() . . . . .	578

---

---

7.38.3 Member Function Documentation . . . . .	578
7.38.3.1 byteswap_buffer_copy() . . . . .	579
7.38.3.2 ensure_buffer_capacity() . . . . .	579
7.38.3.3 get_byte_alignment() . . . . .	579
7.38.3.4 get_capacity() . . . . .	580
7.38.3.5 pull_from_buffer() [1/2] . . . . .	580
7.38.3.6 pull_from_buffer() [2/2] . . . . .	580
7.38.3.7 pull_pad_from_buffer() . . . . .	581
7.38.3.8 push_pad_to_buffer() . . . . .	581
7.38.3.9 push_to_buffer() [1/2] . . . . .	581
7.38.3.10 push_to_buffer() [2/2] . . . . .	581
7.38.3.11 reset_buffer_positions() . . . . .	582
7.38.3.12 reset_pull_position() . . . . .	582
7.38.3.13 reset_push_position() . . . . .	582
7.38.3.14 set_byte_alignment() . . . . .	582
7.38.4 Friends And Related Function Documentation . . . . .	583
7.38.4.1 init_attrTrickHLA__OpaqueBuffer . . . . .	583
7.38.4.2 InputProcessor . . . . .	583
7.38.5 Field Documentation . . . . .	583
7.38.5.1 alignment . . . . .	583
7.38.5.2 buffer . . . . .	583
7.38.5.3 capacity . . . . .	583
7.38.5.4 pull_pos . . . . .	584
7.38.5.5 push_pos . . . . .	584
7.39 TrickHLA::OwnershipHandler Class Reference . . . . .	584
7.39.1 Detailed Description . . . . .	587
7.39.2 Constructor & Destructor Documentation . . . . .	587
7.39.2.1 OwnershipHandler() [1/2] . . . . .	587
7.39.2.2 ~OwnershipHandler() . . . . .	587
7.39.2.3 OwnershipHandler() [2/2] . . . . .	587
7.39.3 Member Function Documentation . . . . .	587
7.39.3.1 clear_checkpoint() . . . . .	587
7.39.3.2 get_attribute() . . . . .	588
7.39.3.3 get_attribute_count() . . . . .	588
7.39.3.4 get_attribute_FOM_names() . . . . .	588
7.39.3.5 get_cte_time() . . . . .	588
7.39.3.6 get_fed_lookahead() . . . . .	589
7.39.3.7 get_granted_fed_time() . . . . .	589
7.39.3.8 get_object_FOM_name() . . . . .	589

---

7.39.3.9 <code>get_object_name()</code> . . . . .	589
7.39.3.10 <code>get_scenario_time()</code> . . . . .	590
7.39.3.11 <code>initialize_callback()</code> . . . . .	590
7.39.3.12 <code>is_locally_owned()</code> . . . . .	590
7.39.3.13 <code>is_published()</code> . . . . .	590
7.39.3.14 <code>is_remotely_owned()</code> . . . . .	591
7.39.3.15 <code>is_subscribed()</code> . . . . .	591
7.39.3.16 <code>operator=()</code> . . . . .	591
7.39.3.17 <code>pull_ownership()</code> [1/4] . . . . .	592
7.39.3.18 <code>pull_ownership()</code> [2/4] . . . . .	592
7.39.3.19 <code>pull_ownership()</code> [3/4] . . . . .	592
7.39.3.20 <code>pull_ownership()</code> [4/4] . . . . .	592
7.39.3.21 <code>push_ownership()</code> [1/4] . . . . .	593
7.39.3.22 <code>push_ownership()</code> [2/4] . . . . .	593
7.39.3.23 <code>push_ownership()</code> [3/4] . . . . .	593
7.39.3.24 <code>push_ownership()</code> [4/4] . . . . .	593
7.39.3.25 <code>restore_requests()</code> . . . . .	594
7.39.3.26 <code>setup_checkpoint_requests()</code> . . . . .	594
7.39.3.27 <code>should_print()</code> . . . . .	594
7.39.4 Friends And Related Function Documentation . . . . .	594
7.39.4.1 <code>init_attrTrickHLA_OwnershipHandler</code> . . . . .	594
7.39.4.2 <code>InputProcessor</code> . . . . .	594
7.39.4.3 <code>Object</code> . . . . .	595
7.39.5 Field Documentation . . . . .	595
7.39.5.1 <code>object</code> . . . . .	595
7.39.5.2 <code>pull_items</code> . . . . .	595
7.39.5.3 <code>pull_items_cnt</code> . . . . .	595
7.39.5.4 <code>pull_requests</code> . . . . .	595
7.39.5.5 <code>push_items</code> . . . . .	595
7.39.5.6 <code>push_items_cnt</code> . . . . .	596
7.39.5.7 <code>push_requests</code> . . . . .	596
7.40 TrickHLA::OwnershipItem Class Reference . . . . .	596
7.40.1 Detailed Description . . . . .	597
7.40.2 Constructor & Destructor Documentation . . . . .	598
7.40.2.1 <code>OwnershipItem()</code> [1/2] . . . . .	598
7.40.2.2 <code>~OwnershipItem()</code> . . . . .	598
7.40.2.3 <code>OwnershipItem()</code> [2/2] . . . . .	598
7.40.3 Member Function Documentation . . . . .	598
7.40.3.1 <code>clear()</code> . . . . .	598

---

7.40.3.2 operator=() . . . . .	598
7.40.4 Friends And Related Function Documentation . . . . .	598
7.40.4.1 init_attrTrickHLA__OwnershipItem . . . . .	598
7.40.4.2 InputProcessor . . . . .	599
7.40.5 Field Documentation . . . . .	599
7.40.5.1 FOM_name . . . . .	599
7.40.5.2 time . . . . .	599
7.41 TrickHLA::Packing Class Reference . . . . .	599
7.41.1 Detailed Description . . . . .	601
7.41.2 Constructor & Destructor Documentation . . . . .	601
7.41.2.1 Packing() [1/2] . . . . .	601
7.41.2.2 ~Packing() . . . . .	601
7.41.2.3 Packing() [2/2] . . . . .	601
7.41.3 Member Function Documentation . . . . .	601
7.41.3.1 get_attribute() . . . . .	601
7.41.3.2 get_attribute_and_validate() . . . . .	602
7.41.3.3 get_cte_time() . . . . .	602
7.41.3.4 get_scenario_time() . . . . .	602
7.41.3.5 initialize_callback() . . . . .	602
7.41.3.6 operator=() . . . . .	604
7.41.3.7 pack() . . . . .	604
7.41.3.8 should_print() . . . . .	604
7.41.3.9 unpack() . . . . .	605
7.41.4 Friends And Related Function Documentation . . . . .	605
7.41.4.1 init_attrTrickHLA__Packing . . . . .	605
7.41.4.2 InputProcessor . . . . .	605
7.41.5 Field Documentation . . . . .	605
7.41.5.1 object . . . . .	605
7.42 TrickHLA::Parameter Class Reference . . . . .	605
7.42.1 Detailed Description . . . . .	609
7.42.2 Constructor & Destructor Documentation . . . . .	609
7.42.2.1 Parameter() [1/2] . . . . .	609
7.42.2.2 ~Parameter() . . . . .	609
7.42.2.3 Parameter() [2/2] . . . . .	609
7.42.3 Member Function Documentation . . . . .	609
7.42.3.1 byteswap_buffer_copy() . . . . .	610
7.42.3.2 calculate_number_of_items() . . . . .	610
7.42.3.3 calculate_size_and_number_of_items() . . . . .	610
7.42.3.4 calculate_static_number_of_items() . . . . .	611

---

---

7.42.3.5 complete_initialization()	611
7.42.3.6 decode_boolean_from_buffer()	611
7.42.3.7 decode_logical_time()	611
7.42.3.8 decode_opaque_data_from_buffer()	612
7.42.3.9 decode_string_from_buffer()	612
7.42.3.10 encode_boolean_to_buffer()	612
7.42.3.11 encode_logical_time()	612
7.42.3.12 encode_opaque_data_to_buffer()	612
7.42.3.13 encode_string_to_buffer()	612
7.42.3.14 ensure_buffer_capacity()	613
7.42.3.15 extract_data()	613
7.42.3.16 get_encoded_parameter_value()	613
7.42.3.17 get_FOM_name()	614
7.42.3.18 get_parameter_handle()	614
7.42.3.19 get_parameter_size()	614
7.42.3.20 get_ref2_attributes()	614
7.42.3.21 get_rti_encoding()	615
7.42.3.22 get_trick_name()	615
7.42.3.23 initialize() [1/2]	615
7.42.3.24 initialize() [2/2]	616
7.42.3.25 is_byteswap()	616
7.42.3.26 is_changed()	616
7.42.3.27 is_static_in_size()	616
7.42.3.28 is_supported_parameter_type()	617
7.42.3.29 mark_changed()	617
7.42.3.30 mark_unchanged()	617
7.42.3.31 operator=( )	617
7.42.3.32 pack_parameter_buffer()	617
7.42.3.33 print_buffer()	618
7.42.3.34 set_debug_level()	618
7.42.3.35 set_encoding()	618
7.42.3.36 set_FOM_name()	618
7.42.3.37 set_parameter_handle()	619
7.42.3.38 unpack_parameter_buffer()	619
7.42.4 Friends And Related Function Documentation	619
7.42.4.1 init_attrTrickHLA_Parameter	619
7.42.4.2 InputProcessor	619
7.42.5 Field Documentation	619
7.42.5.1 address	619

---

---

7.42.5.2 attr . . . . .	620
7.42.5.3 buffer . . . . .	620
7.42.5.4 buffer_capacity . . . . .	620
7.42.5.5 byteswap . . . . .	620
7.42.5.6 debug_handler . . . . .	621
7.42.5.7 FOM_name . . . . .	621
7.42.5.8 HLAtrue . . . . .	621
7.42.5.9 interaction_FOM_name . . . . .	621
7.42.5.10 num_items . . . . .	621
7.42.5.11 param_handle . . . . .	621
7.42.5.12 rti_encoding . . . . .	622
7.42.5.13 size . . . . .	622
7.42.5.14 size_is_static . . . . .	622
7.42.5.15 trick_name . . . . .	622
7.42.5.16 value_changed . . . . .	622
7.43 TrickHLA::ParameterItem Class Reference . . . . .	623
7.43.1 Detailed Description . . . . .	624
7.43.2 Constructor & Destructor Documentation . . . . .	624
7.43.2.1 ParameterItem() [1/3] . . . . .	624
7.43.2.2 ParameterItem() [2/3] . . . . .	624
7.43.2.3 ~ParameterItem() . . . . .	625
7.43.2.4 ParameterItem() [3/3] . . . . .	625
7.43.3 Member Function Documentation . . . . .	625
7.43.3.1 clear() . . . . .	625
7.43.3.2 operator=() . . . . .	625
7.43.4 Friends And Related Function Documentation . . . . .	625
7.43.4.1 init_attrTrickHLA__ParameterItem . . . . .	625
7.43.4.2 InputProcessor . . . . .	626
7.43.5 Field Documentation . . . . .	626
7.43.5.1 data . . . . .	626
7.43.5.2 index . . . . .	626
7.43.5.3 size . . . . .	626
7.44 IMSim::PausePointList Class Reference . . . . .	626
7.44.1 Detailed Description . . . . .	629
7.44.2 Constructor & Destructor Documentation . . . . .	629
7.44.2.1 PausePointList() [1/2] . . . . .	629
7.44.2.2 ~PausePointList() . . . . .	629
7.44.2.3 PausePointList() [2/2] . . . . .	630
7.44.3 Member Function Documentation . . . . .	630

---

7.44.3.1 <code>check_state()</code> . . . . .	630
7.44.3.2 <code>clear_state()</code> . . . . .	630
7.44.3.3 <code>clear_sync_pnt()</code> . . . . .	630
7.44.3.4 <code>get_state()</code> . . . . .	631
7.44.3.5 <code>operator=()</code> . . . . .	631
7.44.3.6 <code>print_sync_pnts()</code> . . . . .	631
7.44.3.7 <code>should_exit()</code> . . . . .	631
7.44.3.8 <code>should_freeze()</code> . . . . .	631
7.44.3.9 <code>should_reconfig()</code> . . . . .	631
7.44.3.10 <code>should_restart()</code> . . . . .	631
7.44.3.11 <code>should_run()</code> . . . . .	632
7.44.3.12 <code>to_string()</code> . . . . .	632
7.44.4 Friends And Related Function Documentation . . . . .	632
7.44.4.1 <code>init_attrTrickHLA__PausePointList</code> . . . . .	632
7.44.4.2 <code>InputProcessor</code> . . . . .	632
7.44.5 Field Documentation . . . . .	632
7.44.5.1 <code>state</code> . . . . .	632
7.45 <code>DIS::PausePointList</code> Class Reference . . . . .	633
7.45.1 Detailed Description . . . . .	635
7.45.2 Constructor & Destructor Documentation . . . . .	635
7.45.2.1 <code>PausePointList()</code> [1/2] . . . . .	635
7.45.2.2 <code>~PausePointList()</code> . . . . .	635
7.45.2.3 <code>PausePointList()</code> [2/2] . . . . .	636
7.45.3 Member Function Documentation . . . . .	636
7.45.3.1 <code>check_state()</code> . . . . .	636
7.45.3.2 <code>clear_state()</code> . . . . .	636
7.45.3.3 <code>clear_sync_pnt()</code> . . . . .	636
7.45.3.4 <code>get_state()</code> . . . . .	636
7.45.3.5 <code>operator=()</code> . . . . .	637
7.45.3.6 <code>print_sync_pnts()</code> . . . . .	637
7.45.3.7 <code>should_exit()</code> . . . . .	637
7.45.3.8 <code>should_freeze()</code> . . . . .	637
7.45.3.9 <code>should_reconfig()</code> . . . . .	637
7.45.3.10 <code>should_restart()</code> . . . . .	637
7.45.3.11 <code>should_run()</code> . . . . .	637
7.45.3.12 <code>to_string()</code> . . . . .	638
7.45.4 Friends And Related Function Documentation . . . . .	638
7.45.4.1 <code>init_attrTrickHLA__PausePointList</code> . . . . .	638
7.45.4.2 <code>InputProcessor</code> . . . . .	638

7.45.5 Field Documentation . . . . .	638
7.45.5.1 state . . . . .	638
7.46 SpaceFOM::PhysicalEntityBase Class Reference . . . . .	638
7.46.1 Detailed Description . . . . .	641
7.46.2 Constructor & Destructor Documentation . . . . .	641
7.46.2.1 PhysicalEntityBase() [1/2] . . . . .	641
7.46.2.2 ~PhysicalEntityBase() . . . . .	641
7.46.2.3 PhysicalEntityBase() [2/2] . . . . .	641
7.46.3 Member Function Documentation . . . . .	641
7.46.3.1 get_name() . . . . .	641
7.46.3.2 get_parent_ref_frame() . . . . .	641
7.46.3.3 get_status() . . . . .	641
7.46.3.4 get_time() . . . . .	642
7.46.3.5 get_type() . . . . .	642
7.46.3.6 initialize() . . . . .	642
7.46.3.7 operator=(()) . . . . .	642
7.46.3.8 pack() . . . . .	642
7.46.3.9 set_name() . . . . .	642
7.46.3.10 set_parent_ref_frame() . . . . .	642
7.46.3.11 set_status() . . . . .	643
7.46.3.12 set_type() . . . . .	643
7.46.3.13 unpack() . . . . .	643
7.46.4 Friends And Related Function Documentation . . . . .	643
7.46.4.1 init_attrSpaceFOM__PhysicalEntityBase . . . . .	643
7.46.4.2 InputProcessor . . . . .	643
7.46.5 Field Documentation . . . . .	643
7.46.5.1 accel . . . . .	643
7.46.5.2 body_wrt_struct . . . . .	644
7.46.5.3 cm . . . . .	644
7.46.5.4 name . . . . .	644
7.46.5.5 parent_ref_frame . . . . .	644
7.46.5.6 quat_encoder . . . . .	644
7.46.5.7 rot_accel . . . . .	644
7.46.5.8 state . . . . .	644
7.46.5.9 status . . . . .	645
7.46.5.10 stc_encoder . . . . .	645
7.46.5.11 type . . . . .	645
7.47 QuaternionData Struct Reference . . . . .	645
7.47.1 Detailed Description . . . . .	645

---

7.47.2 Field Documentation . . . . .	646
7.47.2.1 scalar . . . . .	646
7.47.2.2 vector . . . . .	646
7.48 SpaceFOM::QuaternionEncoder Class Reference . . . . .	646
7.48.1 Detailed Description . . . . .	648
7.48.2 Constructor & Destructor Documentation . . . . .	648
7.48.2.1 QuaternionEncoder() [1/2] . . . . .	648
7.48.2.2 QuaternionEncoder() [2/2] . . . . .	648
7.48.3 Member Function Documentation . . . . .	648
7.48.3.1 decode() . . . . .	648
7.48.3.2 encode() . . . . .	648
7.48.3.3 get_data() . . . . .	649
7.48.3.4 operator=( ) . . . . .	649
7.48.4 Friends And Related Function Documentation . . . . .	649
7.48.4.1 init_attrSpaceFOM__QuaternionEncoder . . . . .	649
7.48.4.2 InputProcessor . . . . .	649
7.48.5 Field Documentation . . . . .	649
7.48.5.1 data . . . . .	649
7.48.5.2 encoder . . . . .	649
7.48.5.3 scalar_encoder . . . . .	650
7.48.5.4 vector . . . . .	650
7.48.5.5 vector_encoder . . . . .	650
7.49 SpaceFOM::RefFrameBase Class Reference . . . . .	650
7.49.1 Detailed Description . . . . .	652
7.49.2 Constructor & Destructor Documentation . . . . .	652
7.49.2.1 RefFrameBase() [1/2] . . . . .	653
7.49.2.2 ~RefFrameBase() . . . . .	653
7.49.2.3 RefFrameBase() [2/2] . . . . .	653
7.49.3 Member Function Documentation . . . . .	653
7.49.3.1 allocate_input_string() [1/2] . . . . .	653
7.49.3.2 allocate_input_string() [2/2] . . . . .	653
7.49.3.3 default_data() . . . . .	653
7.49.3.4 get_name() . . . . .	654
7.49.3.5 get_object() . . . . .	654
7.49.3.6 get_parent_name() . . . . .	654
7.49.3.7 initialize() . . . . .	655
7.49.3.8 initialize_callback() . . . . .	655
7.49.3.9 operator=( ) . . . . .	655
7.49.3.10 pack() . . . . .	655

---

7.49.3.11 set_name() . . . . .	656
7.49.3.12 set_parent_name() . . . . .	656
7.49.3.13 unpack() . . . . .	656
7.49.4 Friends And Related Function Documentation . . . . .	656
7.49.4.1 init_attrSpaceFOM__RefFrameBase . . . . .	656
7.49.4.2 InputProcessor . . . . .	656
7.49.5 Field Documentation . . . . .	657
7.49.5.1 debug . . . . .	657
7.49.5.2 initialized . . . . .	657
7.49.5.3 name . . . . .	657
7.49.5.4 parent_name . . . . .	657
7.49.5.5 ref_frame_attr . . . . .	657
7.49.5.6 ref_frame_data . . . . .	657
7.49.5.7 stc_data . . . . .	658
7.49.5.8 stc_encoder . . . . .	658
7.50 RefFrameData Struct Reference . . . . .	658
7.50.1 Detailed Description . . . . .	659
7.50.2 Field Documentation . . . . .	659
7.50.2.1 name . . . . .	659
7.50.2.2 parent_name . . . . .	659
7.50.2.3 state . . . . .	659
7.51 TrickHLA::ReflectedAttributesQueue Class Reference . . . . .	659
7.51.1 Detailed Description . . . . .	661
7.51.2 Constructor & Destructor Documentation . . . . .	661
7.51.2.1 ReflectedAttributesQueue() [1/2] . . . . .	661
7.51.2.2 ~ReflectedAttributesQueue() . . . . .	661
7.51.2.3 ReflectedAttributesQueue() [2/2] . . . . .	661
7.51.3 Member Function Documentation . . . . .	661
7.51.3.1 clear() . . . . .	661
7.51.3.2 empty() . . . . .	661
7.51.3.3 front() . . . . .	662
7.51.3.4 operator=( ) . . . . .	662
7.51.3.5 pop() . . . . .	662
7.51.3.6 push() . . . . .	662
7.51.4 Friends And Related Function Documentation . . . . .	662
7.51.4.1 init_attrTrickHLA__ReflectedAttributesQueue . . . . .	663
7.51.4.2 InputProcessor . . . . .	663
7.51.5 Field Documentation . . . . .	663
7.51.5.1 attribute_map_queue . . . . .	663

---

---

7.51.5.2 <code>queue_mutex</code> . . . . .	663
7.52 TrickHLA::ScenarioTimeline Class Reference . . . . .	663
7.52.1 Detailed Description . . . . .	665
7.52.2 Constructor & Destructor Documentation . . . . .	665
7.52.2.1 <code>ScenarioTimeline()</code> [1/2] . . . . .	665
7.52.2.2 <code>~ScenarioTimeline()</code> . . . . .	665
7.52.2.3 <code>ScenarioTimeline()</code> [2/2] . . . . .	665
7.52.3 Member Function Documentation . . . . .	666
7.52.3.1 <code>compute_HLT()</code> . . . . .	666
7.52.3.2 <code>compute_simulation_time()</code> . . . . .	666
7.52.3.3 <code>get_HLT_offset()</code> . . . . .	666
7.52.3.4 <code>get_sim_offset()</code> . . . . .	667
7.52.3.5 <code>get_time()</code> . . . . .	667
7.52.3.6 <code>operator=()</code> . . . . .	667
7.52.3.7 <code>set_HLT_offset()</code> . . . . .	667
7.52.3.8 <code>set_sim_offset()</code> . . . . .	668
7.52.3.9 <code>time_from_HLT()</code> . . . . .	668
7.52.3.10 <code>time_from_simulation_time()</code> . . . . .	668
7.52.4 Friends And Related Function Documentation . . . . .	669
7.52.4.1 <code>init_attrTrickHLA__ScenarioTimeline</code> . . . . .	669
7.52.4.2 <code>InputProcessor</code> . . . . .	669
7.52.5 Field Documentation . . . . .	669
7.52.5.1 <code>hlt_offset</code> . . . . .	669
7.52.5.2 <code>sim_offset</code> . . . . .	669
7.52.5.3 <code>sim_timeline</code> . . . . .	669
7.53 TrickHLAModel::SimpleSimConfig Class Reference . . . . .	669
7.53.1 Detailed Description . . . . .	671
7.53.2 Constructor & Destructor Documentation . . . . .	671
7.53.2.1 <code>SimpleSimConfig()</code> [1/2] . . . . .	671
7.53.2.2 <code>~SimpleSimConfig()</code> . . . . .	671
7.53.2.3 <code>SimpleSimConfig()</code> [2/2] . . . . .	672
7.53.3 Member Function Documentation . . . . .	672
7.53.3.1 <code>initialize()</code> . . . . .	672
7.53.3.2 <code>operator=()</code> . . . . .	672
7.53.3.3 <code>pack()</code> . . . . .	672
7.53.3.4 <code>unpack()</code> . . . . .	672
7.53.4 Friends And Related Function Documentation . . . . .	673
7.53.4.1 <code>init_attrTrickHLAModel__SimpleSimConfig</code> . . . . .	673
7.53.4.2 <code>InputProcessor</code> . . . . .	673

---

---

7.53.5 Field Documentation . . . . .	673
7.53.5.1 num_federates . . . . .	673
7.53.5.2 owner . . . . .	673
7.53.5.3 required_federates . . . . .	673
7.53.5.4 run_duration . . . . .	673
7.53.5.5 run_duration_microsec . . . . .	674
7.54 TrickHLA::SimTimeline Class Reference . . . . .	674
7.54.1 Detailed Description . . . . .	675
7.54.2 Constructor & Destructor Documentation . . . . .	675
7.54.2.1 SimTimeline() [1/2] . . . . .	675
7.54.2.2 ~SimTimeline() . . . . .	675
7.54.2.3 SimTimeline() [2/2] . . . . .	675
7.54.3 Member Function Documentation . . . . .	675
7.54.3.1 get_time() . . . . .	676
7.54.3.2 operator=( ) . . . . .	676
7.54.4 Friends And Related Function Documentation . . . . .	676
7.54.4.1 init_attrTrickHLA__SimTimeline . . . . .	676
7.54.4.2 InputProcessor . . . . .	676
7.55 TrickHLAModel::SineConditional Class Reference . . . . .	676
7.55.1 Detailed Description . . . . .	678
7.55.2 Constructor & Destructor Documentation . . . . .	678
7.55.2.1 SineConditional() . . . . .	678
7.55.2.2 ~SineConditional() . . . . .	678
7.55.3 Member Function Documentation . . . . .	678
7.55.3.1 convert_FOM_name_to_pos() . . . . .	678
7.55.3.2 initialize() . . . . .	679
7.55.3.3 should_send() . . . . .	679
7.55.4 Friends And Related Function Documentation . . . . .	679
7.55.4.1 init_attrTrickHLAModel__SineConditional . . . . .	680
7.55.4.2 InputProcessor . . . . .	680
7.55.5 Field Documentation . . . . .	680
7.55.5.1 attr_pos . . . . .	680
7.55.5.2 prev_sim_data . . . . .	680
7.55.5.3 sim_data . . . . .	680
7.56 TrickHLAModel::SineData Class Reference . . . . .	680
7.56.1 Detailed Description . . . . .	682
7.56.2 Constructor & Destructor Documentation . . . . .	682
7.56.2.1 SineData() [1/2] . . . . .	682
7.56.2.2 SineData() [2/2] . . . . .	683

---

---

7.56.2.3 <code>~SineData()</code> . . . . .	683
7.56.3 Member Function Documentation . . . . .	683
7.56.3.1 <code>adjust_phase()</code> [1/2] . . . . .	683
7.56.3.2 <code>adjust_phase()</code> [2/2] . . . . .	683
7.56.3.3 <code>compute_derivative()</code> [1/2] . . . . .	684
7.56.3.4 <code>compute_derivative()</code> [2/2] . . . . .	684
7.56.3.5 <code>compute_value()</code> [1/2] . . . . .	684
7.56.3.6 <code>compute_value()</code> [2/2] . . . . .	684
7.56.3.7 <code>copy_data()</code> . . . . .	684
7.56.3.8 <code>get_amplitude()</code> . . . . .	685
7.56.3.9 <code>get_derivative()</code> . . . . .	685
7.56.3.10 <code>get_frequency()</code> . . . . .	685
7.56.3.11 <code>get_name()</code> . . . . .	686
7.56.3.12 <code>get_phase()</code> . . . . .	686
7.56.3.13 <code>get_time()</code> . . . . .	686
7.56.3.14 <code>get_tolerance()</code> . . . . .	686
7.56.3.15 <code>get_value()</code> . . . . .	687
7.56.3.16 <code>integration()</code> . . . . .	687
7.56.3.17 <code>set_amplitude()</code> . . . . .	687
7.56.3.18 <code>set_derivative()</code> . . . . .	687
7.56.3.19 <code>set_frequency()</code> . . . . .	688
7.56.3.20 <code>set_name()</code> . . . . .	688
7.56.3.21 <code>set_phase()</code> . . . . .	688
7.56.3.22 <code>set_time()</code> . . . . .	688
7.56.3.23 <code>set_tolerance()</code> . . . . .	689
7.56.3.24 <code>set_value()</code> . . . . .	689
7.56.4 Friends And Related Function Documentation . . . . .	689
7.56.4.1 <code>init_attrTrickHLAModel__SineData</code> . . . . .	689
7.56.4.2 <code>InputProcessor</code> . . . . .	689
7.56.5 Field Documentation . . . . .	689
7.56.5.1 <code>amp</code> . . . . .	689
7.56.5.2 <code>dvdt</code> . . . . .	690
7.56.5.3 <code>freq</code> . . . . .	690
7.56.5.4 <code>name</code> . . . . .	690
7.56.5.5 <code>phase</code> . . . . .	690
7.56.5.6 <code>time</code> . . . . .	690
7.56.5.7 <code>tol</code> . . . . .	690
7.56.5.8 <code>value</code> . . . . .	691
7.57 <code>TrickHLAModel::SineInteractionHandler</code> Class Reference . . . . .	691

---

---

7.57.1 Detailed Description . . . . .	692
7.57.2 Constructor & Destructor Documentation . . . . .	692
7.57.2.1 SineInteractionHandler() [1/2] . . . . .	693
7.57.2.2 ~SineInteractionHandler() . . . . .	693
7.57.2.3 SineInteractionHandler() [2/2] . . . . .	693
7.57.3 Member Function Documentation . . . . .	693
7.57.3.1 operator=( ) . . . . .	693
7.57.3.2 receive_interaction() . . . . .	693
7.57.3.3 send_sine_interaction() . . . . .	693
7.57.4 Friends And Related Function Documentation . . . . .	694
7.57.4.1 init_attrTrickHLAModel__SineInteractionHandler . . . . .	694
7.57.4.2 InputProcessor . . . . .	694
7.57.5 Field Documentation . . . . .	694
7.57.5.1 message . . . . .	694
7.57.5.2 name . . . . .	694
7.57.5.3 receive_cnt . . . . .	694
7.57.5.4 send_cnt . . . . .	695
7.57.5.5 time . . . . .	695
7.57.5.6 year . . . . .	695
7.58 TrickHLAModel::SineLagCompensation Class Reference . . . . .	695
7.58.1 Detailed Description . . . . .	697
7.58.2 Constructor & Destructor Documentation . . . . .	697
7.58.2.1 SineLagCompensation() [1/2] . . . . .	697
7.58.2.2 ~SineLagCompensation() . . . . .	697
7.58.2.3 SineLagCompensation() [2/2] . . . . .	698
7.58.3 Member Function Documentation . . . . .	698
7.58.3.1 initialize( ) . . . . .	698
7.58.3.2 initialize_callback() . . . . .	698
7.58.3.3 operator=( ) . . . . .	698
7.58.3.4 receive_lag_compensation() . . . . .	699
7.58.3.5 send_lag_compensation() . . . . .	699
7.58.4 Friends And Related Function Documentation . . . . .	699
7.58.4.1 init_attrTrickHLAModel__SineLagCompensation . . . . .	699
7.58.4.2 InputProcessor . . . . .	699
7.58.5 Field Documentation . . . . .	699
7.58.5.1 amp_attr . . . . .	699
7.58.5.2 dvdt_attr . . . . .	700
7.58.5.3 freq_attr . . . . .	700
7.58.5.4 lag_comp_data . . . . .	700

---

---

7.58.5.5 name_attr . . . . .	700
7.58.5.6 phase_attr . . . . .	700
7.58.5.7 sim_data . . . . .	700
7.58.5.8 time_attr . . . . .	701
7.58.5.9 tol_attr . . . . .	701
7.58.5.10 value_attr . . . . .	701
7.59 TrickHLAModel::SineObjectDeleted Class Reference . . . . .	701
7.59.1 Detailed Description . . . . .	702
7.59.2 Constructor & Destructor Documentation . . . . .	702
7.59.2.1 SineObjectDeleted() [1/2] . . . . .	702
7.59.2.2 ~SineObjectDeleted() . . . . .	703
7.59.2.3 SineObjectDeleted() [2/2] . . . . .	703
7.59.3 Member Function Documentation . . . . .	703
7.59.3.1 deleted() . . . . .	703
7.59.3.2 operator=(()) . . . . .	703
7.59.4 Friends And Related Function Documentation . . . . .	703
7.59.4.1 init_attrTrickHLAModel__SineObjectDeleted . . . . .	703
7.59.4.2 InputProcessor . . . . .	703
7.60 TrickHLAModel::SineOwnershipHandler Class Reference . . . . .	704
7.60.1 Detailed Description . . . . .	705
7.60.2 Constructor & Destructor Documentation . . . . .	705
7.60.2.1 SineOwnershipHandler() [1/2] . . . . .	705
7.60.2.2 ~SineOwnershipHandler() . . . . .	705
7.60.2.3 SineOwnershipHandler() [2/2] . . . . .	705
7.60.3 Member Function Documentation . . . . .	705
7.60.3.1 initialize_callback() . . . . .	705
7.60.3.2 operator=(()) . . . . .	706
7.60.4 Friends And Related Function Documentation . . . . .	706
7.60.4.1 init_attrTrickHLAModel__SineOwnershipHandler . . . . .	706
7.60.4.2 InputProcessor . . . . .	706
7.61 TrickHLAModel::SinePacking Class Reference . . . . .	706
7.61.1 Detailed Description . . . . .	708
7.61.2 Constructor & Destructor Documentation . . . . .	708
7.61.2.1 SinePacking() [1/2] . . . . .	709
7.61.2.2 ~SinePacking() . . . . .	709
7.61.2.3 SinePacking() [2/2] . . . . .	709
7.61.3 Member Function Documentation . . . . .	709
7.61.3.1 initialize() . . . . .	709
7.61.3.2 initialize_callback() . . . . .	709

---

---

7.61.3.3 operator=( ) . . . . .	710
7.61.3.4 pack() . . . . .	710
7.61.3.5 unpack() . . . . .	710
7.61.4 Friends And Related Function Documentation . . . . .	710
7.61.4.1 init_attrTrickHLAModel__SinePacking . . . . .	710
7.61.4.2 InputProcessor . . . . .	711
7.61.5 Field Documentation . . . . .	711
7.61.5.1 amp_attr . . . . .	711
7.61.5.2 buff . . . . .	711
7.61.5.3 buff_size . . . . .	711
7.61.5.4 dvdt_attr . . . . .	711
7.61.5.5 freq_attr . . . . .	711
7.61.5.6 initialized . . . . .	712
7.61.5.7 name_attr . . . . .	712
7.61.5.8 pack_count . . . . .	712
7.61.5.9 phase_attr . . . . .	712
7.61.5.10 phase_deg . . . . .	712
7.61.5.11 sim_data . . . . .	712
7.61.5.12 time_attr . . . . .	713
7.61.5.13 tol_attr . . . . .	713
7.61.5.14 value_attr . . . . .	713
7.62 SpaceTimeCoordinateData Struct Reference . . . . .	713
7.62.1 Detailed Description . . . . .	713
7.62.2 Field Documentation . . . . .	714
7.62.2.1 ang_vel . . . . .	714
7.62.2.2 pos . . . . .	714
7.62.2.3 quat_scalar . . . . .	714
7.62.2.4 quat_vector . . . . .	714
7.62.2.5 time . . . . .	714
7.62.2.6 vel . . . . .	714
7.63 SpaceFOM::SpaceTimeCoordinateEncoder Class Reference . . . . .	715
7.63.1 Detailed Description . . . . .	717
7.63.2 Constructor & Destructor Documentation . . . . .	717
7.63.2.1 SpaceTimeCoordinateEncoder() [1/2] . . . . .	717
7.63.2.2 SpaceTimeCoordinateEncoder() [2/2] . . . . .	717
7.63.3 Member Function Documentation . . . . .	717
7.63.3.1 decode() . . . . .	717
7.63.3.2 encode() . . . . .	717
7.63.3.3 get_data() . . . . .	718

---

---

7.63.3.4 operator=( ) . . . . .	718
7.63.4 Friends And Related Function Documentation . . . . .	718
7.63.4.1 init_attrSpaceFOM__SpaceTimeCoordinateEncoder . . . . .	718
7.63.4.2 InputProcessor . . . . .	718
7.63.5 Field Documentation . . . . .	718
7.63.5.1 ang_vel_encoder . . . . .	718
7.63.5.2 angular_velocity . . . . .	718
7.63.5.3 data . . . . .	719
7.63.5.4 encoder . . . . .	719
7.63.5.5 position . . . . .	719
7.63.5.6 position_encoder . . . . .	719
7.63.5.7 quat_encoder . . . . .	719
7.63.5.8 quat_scalar_encoder . . . . .	719
7.63.5.9 quat_vector . . . . .	720
7.63.5.10 quat_vector_encoder . . . . .	720
7.63.5.11 rot_state_encoder . . . . .	720
7.63.5.12 time_encoder . . . . .	720
7.63.5.13 trans_state_encoder . . . . .	720
7.63.5.14 velocity . . . . .	720
7.63.5.15 velocity_encoder . . . . .	721
7.64 TrickHLA::StringUtilities Class Reference . . . . .	721
7.64.1 Detailed Description . . . . .	722
7.64.2 Constructor & Destructor Documentation . . . . .	722
7.64.2.1 StringUtilities() [1/2] . . . . .	722
7.64.2.2 ~StringUtilities() . . . . .	722
7.64.2.3 StringUtilities() [2/2] . . . . .	722
7.64.3 Member Function Documentation . . . . .	722
7.64.3.1 ip_strdup_wstring() . . . . .	723
7.64.3.2 operator=( ) . . . . .	723
7.64.3.3 to_printable_string() . . . . .	723
7.64.3.4 to_string() [1/8] . . . . .	723
7.64.3.5 to_string() [2/8] . . . . .	724
7.64.3.6 to_string() [3/8] . . . . .	724
7.64.3.7 to_string() [4/8] . . . . .	724
7.64.3.8 to_string() [5/8] . . . . .	725
7.64.3.9 to_string() [6/8] . . . . .	725
7.64.3.10 to_string() [7/8] . . . . .	725
7.64.3.11 to_string() [8/8] . . . . .	725
7.64.3.12 to_wstring() [1/2] . . . . .	726

7.64.3.13 <code>to_wstring()</code> [2/2] . . . . .	727
7.64.3.14 <code>tokenize()</code> [1/2] . . . . .	727
7.64.3.15 <code>tokenize()</code> [2/2] . . . . .	728
7.64.3.16 <code>trim_whitespace()</code> [1/2] . . . . .	728
7.64.3.17 <code>trim_whitespace()</code> [2/2] . . . . .	728
7.64.4 Friends And Related Function Documentation . . . . .	729
7.64.4.1 <code>init_attrTrickHLA__StringUtilities</code> . . . . .	729
7.64.4.2 <code>InputProcessor</code> . . . . .	729
7.65 TrickHLA::SyncPnt Class Reference . . . . .	729
7.65.1 Detailed Description . . . . .	731
7.65.2 Constructor & Destructor Documentation . . . . .	732
7.65.2.1 <code>SyncPnt()</code> [1/2] . . . . .	732
7.65.2.2 <code>SyncPnt()</code> [2/2] . . . . .	732
7.65.2.3 <code>~SyncPnt()</code> . . . . .	732
7.65.3 Member Function Documentation . . . . .	732
7.65.3.1 <code>achieve_sync_point()</code> . . . . .	732
7.65.3.2 <code>convert()</code> . . . . .	733
7.65.3.3 <code>exists()</code> . . . . .	733
7.65.3.4 <code>get_label()</code> . . . . .	733
7.65.3.5 <code>get_state()</code> . . . . .	734
7.65.3.6 <code>is_achieved()</code> . . . . .	734
7.65.3.7 <code>is_announced()</code> . . . . .	734
7.65.3.8 <code>is_error()</code> . . . . .	734
7.65.3.9 <code>is_registered()</code> . . . . .	735
7.65.3.10 <code>is_synchronized()</code> . . . . .	735
7.65.3.11 <code>is_valid()</code> . . . . .	735
7.65.3.12 <code>register_sync_point()</code> [1/2] . . . . .	735
7.65.3.13 <code>register_sync_point()</code> [2/2] . . . . .	736
7.65.3.14 <code>set_label()</code> . . . . .	736
7.65.3.15 <code>set_state()</code> . . . . .	736
7.65.3.16 <code>set_wait_sleep()</code> . . . . .	737
7.65.3.17 <code>set_wait_timeout()</code> . . . . .	737
7.65.3.18 <code>to_string()</code> . . . . .	737
7.65.3.19 <code>wait_for_announce()</code> . . . . .	738
7.65.3.20 <code>wait_for_synchronization()</code> . . . . .	738
7.65.4 Friends And Related Function Documentation . . . . .	738
7.65.4.1 <code>init_attrTrickHLA__SyncPnt</code> . . . . .	738
7.65.4.2 <code>InputProcessor</code> . . . . .	738
7.65.5 Field Documentation . . . . .	739

---

7.65.5.1 label . . . . .	739
7.65.5.2 state . . . . .	739
7.65.5.3 wait_sleep . . . . .	739
7.65.5.4 wait_timeout . . . . .	739
7.66 TrickHLA::SyncPntList Class Reference . . . . .	739
7.66.1 Detailed Description . . . . .	741
7.66.2 Constructor & Destructor Documentation . . . . .	741
7.66.2.1 SyncPntList() [1/2] . . . . .	741
7.66.2.2 ~SyncPntList() . . . . .	741
7.66.2.3 SyncPntList() [2/2] . . . . .	741
7.66.3 Member Function Documentation . . . . .	741
7.66.3.1 operator=() . . . . .	742
7.66.4 Friends And Related Function Documentation . . . . .	742
7.66.4.1 init_attrTrickHLA__SyncPntList . . . . .	742
7.66.4.2 InputProcessor . . . . .	742
7.67 TrickHLA::SyncPntListBase Class Reference . . . . .	742
7.67.1 Detailed Description . . . . .	746
7.67.2 Constructor & Destructor Documentation . . . . .	746
7.67.2.1 SyncPntListBase() [1/2] . . . . .	746
7.67.2.2 ~SyncPntListBase() . . . . .	746
7.67.2.3 SyncPntListBase() [2/2] . . . . .	746
7.67.3 Member Function Documentation . . . . .	746
7.67.3.1 achieve_all_sync_pnts() . . . . .	746
7.67.3.2 achieve_and_wait_for_synchronization() . . . . .	747
7.67.3.3 achieve_sync_pnt() [1/2] . . . . .	747
7.67.3.4 achieve_sync_pnt() [2/2] . . . . .	748
7.67.3.5 add_sync_pnt() . . . . .	748
7.67.3.6 announce_sync_point() . . . . .	748
7.67.3.7 clear_sync_pnt() . . . . .	749
7.67.3.8 contains() . . . . .	749
7.67.3.9 convert_sync_pts() . . . . .	749
7.67.3.10 get_reconfig_name() . . . . .	750
7.67.3.11 get_size() . . . . .	750
7.67.3.12 get_sync_pnt() . . . . .	750
7.67.3.13 get_sync_pnt_state() . . . . .	751
7.67.3.14 is_sync_pnt_announced() . . . . .	751
7.67.3.15 lock_read_only() . . . . .	751
7.67.3.16 lock_read_write() . . . . .	751
7.67.3.17 mark_announced() . . . . .	752

---

7.67.3.18 mark_registered()	752
7.67.3.19 mark_synchronized()	752
7.67.3.20 operator=( )	753
7.67.3.21 print_sync_pnts()	753
7.67.3.22 register_all_sync_pnts() [1/2]	753
7.67.3.23 register_all_sync_pnts() [2/2]	754
7.67.3.24 register_sync_pnt() [1/2]	754
7.67.3.25 register_sync_pnt() [2/2]	754
7.67.3.26 reset()	755
7.67.3.27 set_debug_level()	755
7.67.3.28 sync_point_registration_failed()	755
7.67.3.29 sync_point_registration_succeeded()	756
7.67.3.30 to_string()	756
7.67.3.31 unlock_read_only()	756
7.67.3.32 unlock_read_write()	756
7.67.3.33 wait_for_all_announcements()	757
7.67.3.34 wait_for_announcement()	757
7.67.3.35 wait_for_list_synchronization()	757
7.67.4 Friends And Related Function Documentation	758
7.67.4.1 init_attrTrickHLA__SyncPntListBase	758
7.67.4.2 InputProcessor	758
7.67.5 Field Documentation	758
7.67.5.1 debug_handler	758
7.67.5.2 read_locks	758
7.67.5.3 reconfig_name	759
7.67.5.4 sync_point_list	759
7.67.5.5 write_locks	759
7.68 TrickHLA::TimedSyncPnt Class Reference	759
7.68.1 Detailed Description	761
7.68.2 Constructor & Destructor Documentation	761
7.68.2.1 TimedSyncPnt() [1/3]	761
7.68.2.2 TimedSyncPnt() [2/3]	761
7.68.2.3 TimedSyncPnt() [3/3]	762
7.68.2.4 ~TimedSyncPnt()	762
7.68.3 Member Function Documentation	762
7.68.3.1 convert()	762
7.68.3.2 get_time()	762
7.68.3.3 set_time()	763
7.68.3.4 to_string()	763

---

---

7.68.4 Friends And Related Function Documentation . . . . .	763
7.68.4.1 init_attrTrickHLA__TimedSyncPnt . . . . .	763
7.68.4.2 InputProcessor . . . . .	763
7.68.5 Field Documentation . . . . .	763
7.68.5.1 time . . . . .	763
7.69 TrickHLA::TimedSyncPntList Class Reference . . . . .	764
7.69.1 Detailed Description . . . . .	766
7.69.2 Constructor & Destructor Documentation . . . . .	766
7.69.2.1 TimedSyncPntList() [1/2] . . . . .	766
7.69.2.2 ~TimedSyncPntList() . . . . .	766
7.69.2.3 TimedSyncPntList() [2/2] . . . . .	766
7.69.3 Member Function Documentation . . . . .	767
7.69.3.1 achieve_all_sync_pnts() [1/2] . . . . .	767
7.69.3.2 achieve_all_sync_pnts() [2/2] . . . . .	767
7.69.3.3 add_sync_pnt() [1/2] . . . . .	767
7.69.3.4 add_sync_pnt() [2/2] . . . . .	768
7.69.3.5 check_sync_pnts() . . . . .	768
7.69.3.6 convert_sync_pts() . . . . .	768
7.69.3.7 operator=( ) . . . . .	769
7.69.3.8 print_sync_pnts() . . . . .	769
7.69.4 Friends And Related Function Documentation . . . . .	769
7.69.4.1 init_attrTrickHLA__TimedSyncPntList . . . . .	769
7.69.4.2 InputProcessor . . . . .	769
7.70 TrickHLA::Timeline Class Reference . . . . .	769
7.70.1 Detailed Description . . . . .	771
7.70.2 Constructor & Destructor Documentation . . . . .	771
7.70.2.1 Timeline() [1/2] . . . . .	771
7.70.2.2 ~Timeline() . . . . .	771
7.70.2.3 Timeline() [2/2] . . . . .	771
7.70.3 Member Function Documentation . . . . .	771
7.70.3.1 get_elapsed_time() . . . . .	771
7.70.3.2 get_epoch() . . . . .	772
7.70.3.3 get_time() . . . . .	772
7.70.3.4 operator=( ) . . . . .	772
7.70.3.5 set_epoch() . . . . .	772
7.70.4 Friends And Related Function Documentation . . . . .	773
7.70.4.1 init_attrTrickHLA__Timeline . . . . .	773
7.70.4.2 InputProcessor . . . . .	773
7.70.5 Field Documentation . . . . .	773

---

---

7.70.5.1 epoch	773
7.71 TrickHLA::TimeOfDayTimeline Class Reference	773
7.71.1 Detailed Description	774
7.71.2 Constructor & Destructor Documentation	774
7.71.2.1 TimeOfDayTimeline() [1/2]	774
7.71.2.2 ~TimeOfDayTimeline()	775
7.71.2.3 TimeOfDayTimeline() [2/2]	775
7.71.3 Member Function Documentation	775
7.71.3.1 get_time()	775
7.71.3.2 operator=( )	775
7.71.4 Friends And Related Function Documentation	775
7.71.4.1 init_attrTrickHLA__TimeOfDayTimeline	775
7.71.4.2 InputProcessor	775
7.72 TrickHLA::Utilities Class Reference	776
7.72.1 Detailed Description	777
7.72.2 Constructor & Destructor Documentation	777
7.72.2.1 Utilities() [1/2]	777
7.72.2.2 ~Utilities()	777
7.72.2.3 Utilities() [2/2]	777
7.72.3 Member Function Documentation	777
7.72.3.1 byteswap_double()	777
7.72.3.2 byteswap_float()	778
7.72.3.3 byteswap_int()	778
7.72.3.4 byteswap_long()	778
7.72.3.5 byteswap_long_long()	779
7.72.3.6 byteswap_short()	779
7.72.3.7 byteswap_unsigned_int()	779
7.72.3.8 byteswap_unsigned_long()	780
7.72.3.9 byteswap_unsigned_long_long()	780
7.72.3.10 byteswap_unsigned_short()	780
7.72.3.11 get_endianness()	781
7.72.3.12 get_release_date()	781
7.72.3.13 get_version()	781
7.72.3.14 is_transmission_byteswap()	782
7.72.3.15 operator=( )	782
7.72.4 Friends And Related Function Documentation	782
7.72.4.1 init_attrTrickHLA__Utilities	782
7.72.4.2 InputProcessor	782

---

---

<b>8 File Documentation</b>	<b>783</b>
8.1 Attribute.cpp File Reference . . . . .	783
8.1.1 Detailed Description . . . . .	783
8.2 Attribute.hh File Reference . . . . .	784
8.2.1 Detailed Description . . . . .	785
8.3 BasicClock.hh File Reference . . . . .	786
8.3.1 Detailed Description . . . . .	787
8.4 CompileConfig.hh File Reference . . . . .	787
8.4.1 Detailed Description . . . . .	788
8.4.2 Macro Definition Documentation . . . . .	788
8.4.2.1 MIN_TRICK_MINOR . . . . .	788
8.4.2.2 MIN_TRICK_PATCH . . . . .	789
8.4.2.3 MIN_TRICK_VER . . . . .	789
8.4.2.4 NO_THLA_THREAD_WAIT_FOR_DATA . . . . .	789
8.4.2.5 NO_TRICKHLA_ENABLE_FOM_DUMP . . . . .	789
8.4.2.6 NO_TRICKHLA_ENABLE_FPU_CONTROL_WORD_VALIDATION . . . . .	789
8.4.2.7 THLA_10SEC_TIMEOUT WHILE_WAITING_FOR_DATA . . . . .	789
8.4.2.8 THLA_ENDL . . . . .	789
8.4.2.9 THLA_INTRAFRAME_BLOCKING_READ_CONFIG . . . . .	789
8.4.2.10 THLA_NEWLINE . . . . .	789
8.4.2.11 THLA_QUEUE_REFLECTED_ATTRIBUTES . . . . .	790
8.4.2.12 THLA_THREAD_TIMED_WAIT_FOR_DATA . . . . .	790
8.5 Conditional.cpp File Reference . . . . .	790
8.5.1 Detailed Description . . . . .	790
8.6 Conditional.hh File Reference . . . . .	791
8.6.1 Detailed Description . . . . .	791
8.7 Constants.hh File Reference . . . . .	792
8.7.1 Detailed Description . . . . .	792
8.8 CTETimelineBase.cpp File Reference . . . . .	793
8.8.1 Detailed Description . . . . .	793
8.9 CTETimelineBase.hh File Reference . . . . .	794
8.9.1 Detailed Description . . . . .	795
8.10 DebugHandler.hh File Reference . . . . .	796
8.10.1 Detailed Description . . . . .	797
8.11 DynamicalEntity.cpp File Reference . . . . .	797
8.11.1 Detailed Description . . . . .	798
8.12 DynamicalEntity.hh File Reference . . . . .	798
8.12.1 Detailed Description . . . . .	799
8.13 ExecutionConfiguration.cpp File Reference . . . . .	800

8.13.1 Detailed Description . . . . .	800
8.13.2 Variable Documentation . . . . .	801
8.13.2.1 attrDIS__ExecutionConfiguration . . . . .	801
8.14 ExecutionConfiguration.cpp File Reference . . . . .	801
8.14.1 Detailed Description . . . . .	802
8.14.2 Variable Documentation . . . . .	803
8.14.2.1 attrDSES__ExecutionConfiguration . . . . .	803
8.15 ExecutionConfiguration.cpp File Reference . . . . .	803
8.15.1 Detailed Description . . . . .	803
8.15.2 Variable Documentation . . . . .	804
8.15.2.1 attrIMSim__ExecutionConfiguration . . . . .	804
8.16 ExecutionConfiguration.cpp File Reference . . . . .	804
8.16.1 Detailed Description . . . . .	805
8.16.2 Variable Documentation . . . . .	806
8.16.2.1 attrSpaceFOM__ExecutionConfiguration . . . . .	806
8.17 ExecutionConfiguration.cpp File Reference . . . . .	806
8.17.1 Detailed Description . . . . .	806
8.17.2 Variable Documentation . . . . .	807
8.17.2.1 attrTrickHLA__ExecutionConfiguration . . . . .	807
8.18 ExecutionConfiguration.hh File Reference . . . . .	807
8.18.1 Detailed Description . . . . .	808
8.19 ExecutionConfiguration.hh File Reference . . . . .	809
8.19.1 Detailed Description . . . . .	810
8.20 ExecutionConfiguration.hh File Reference . . . . .	811
8.20.1 Detailed Description . . . . .	812
8.21 ExecutionConfiguration.hh File Reference . . . . .	812
8.21.1 Detailed Description . . . . .	813
8.22 ExecutionConfiguration.hh File Reference . . . . .	814
8.22.1 Detailed Description . . . . .	815
8.23 ExecutionConfigurationBase.cpp File Reference . . . . .	815
8.23.1 Detailed Description . . . . .	816
8.23.2 Variable Documentation . . . . .	817
8.23.2.1 attrTrickHLA__ExecutionConfigurationBase . . . . .	817
8.24 ExecutionConfigurationBase.hh File Reference . . . . .	817
8.24.1 Detailed Description . . . . .	817
8.25 ExecutionControl.cpp File Reference . . . . .	818
8.25.1 Detailed Description . . . . .	819
8.26 ExecutionControl.cpp File Reference . . . . .	819
8.26.1 Detailed Description . . . . .	820

---

8.27 ExecutionControl.cpp File Reference . . . . .	821
8.27.1 Detailed Description . . . . .	821
8.27.2 Variable Documentation . . . . .	822
8.27.2.1 attrTrickHLA__FreezeInteractionHandler . . . . .	822
8.28 ExecutionControl.cpp File Reference . . . . .	822
8.28.1 Detailed Description . . . . .	823
8.28.2 Variable Documentation . . . . .	824
8.28.2.1 attrSpaceFOM__MTRInteractionHandler . . . . .	824
8.28.2.2 the_clock . . . . .	824
8.29 ExecutionControl.cpp File Reference . . . . .	824
8.29.1 Detailed Description . . . . .	825
8.29.2 Variable Documentation . . . . .	825
8.29.2.1 attrTrickHLA__MTRInteractionHandler . . . . .	825
8.30 ExecutionControl.hh File Reference . . . . .	825
8.30.1 Detailed Description . . . . .	826
8.31 ExecutionControl.hh File Reference . . . . .	827
8.31.1 Detailed Description . . . . .	828
8.32 ExecutionControl.hh File Reference . . . . .	828
8.32.1 Detailed Description . . . . .	829
8.33 ExecutionControl.hh File Reference . . . . .	830
8.33.1 Detailed Description . . . . .	830
8.34 ExecutionControl.hh File Reference . . . . .	831
8.34.1 Detailed Description . . . . .	832
8.35 ExecutionControlBase.cpp File Reference . . . . .	832
8.35.1 Detailed Description . . . . .	833
8.36 ExecutionControlBase.hh File Reference . . . . .	834
8.36.1 Detailed Description . . . . .	834
8.37 FedAmb.cpp File Reference . . . . .	835
8.37.1 Detailed Description . . . . .	835
8.38 FedAmb.hh File Reference . . . . .	836
8.38.1 Detailed Description . . . . .	837
8.39 Federate.cpp File Reference . . . . .	838
8.39.1 Detailed Description . . . . .	839
8.39.2 Macro Definition Documentation . . . . .	840
8.39.2.1 THLA_TAG_USE_USLEEP . . . . .	840
8.39.3 Variable Documentation . . . . .	840
8.39.3.1 ip_mutex . . . . .	840
8.39.3.2 the_cpr . . . . .	840
8.40 Federate.hh File Reference . . . . .	841

---

---

8.40.1 Detailed Description . . . . .	841
8.41 FreezelInteractionHandler.cpp File Reference . . . . .	842
8.41.1 Detailed Description . . . . .	843
8.41.2 Macro Definition Documentation . . . . .	843
8.41.2.1 THLA_FREEZE_INTERACTION_DEBUG . . . . .	844
8.42 FreezelInteractionHandler.hh File Reference . . . . .	844
8.42.1 Detailed Description . . . . .	844
8.43 Int64Interval.cpp File Reference . . . . .	845
8.43.1 Detailed Description . . . . .	846
8.44 Int64Interval.hh File Reference . . . . .	846
8.44.1 Detailed Description . . . . .	847
8.45 Int64Time.cpp File Reference . . . . .	848
8.46 Int64Time.hh File Reference . . . . .	848
8.46.1 Detailed Description . . . . .	849
8.47 integrator.d File Reference . . . . .	850
8.47.1 Variable Documentation . . . . .	850
8.47.1.1 deriv . . . . .	850
8.47.1.2 first_step_deriv . . . . .	850
8.47.1.3 init . . . . .	850
8.47.1.4 num_state . . . . .	850
8.47.1.5 option . . . . .	850
8.47.1.6 state . . . . .	850
8.47.1.7 state_ws . . . . .	851
8.48 Interaction.cpp File Reference . . . . .	851
8.48.1 Detailed Description . . . . .	851
8.49 Interaction.hh File Reference . . . . .	852
8.49.1 Detailed Description . . . . .	853
8.50 InteractionHandler.cpp File Reference . . . . .	853
8.50.1 Detailed Description . . . . .	854
8.51 InteractionHandler.hh File Reference . . . . .	855
8.51.1 Detailed Description . . . . .	855
8.52 InteractionItem.cpp File Reference . . . . .	856
8.52.1 Detailed Description . . . . .	856
8.52.2 Variable Documentation . . . . .	857
8.52.2.1 attrParameterItem . . . . .	857
8.53 InteractionItem.hh File Reference . . . . .	857
8.53.1 Detailed Description . . . . .	858
8.54 Item.cpp File Reference . . . . .	859
8.54.1 Detailed Description . . . . .	859

---

8.55 Item.hh File Reference . . . . .	860
8.55.1 Detailed Description . . . . .	860
8.56 ItemQueue.cpp File Reference . . . . .	861
8.56.1 Detailed Description . . . . .	861
8.57 ItemQueue.hh File Reference . . . . .	862
8.57.1 Detailed Description . . . . .	862
8.58 KnownFederate.hh File Reference . . . . .	863
8.58.1 Detailed Description . . . . .	863
8.59 LagCompensation.cpp File Reference . . . . .	864
8.59.1 Detailed Description . . . . .	864
8.60 LagCompensation.hh File Reference . . . . .	865
8.60.1 Detailed Description . . . . .	866
8.61 LoggableSyncPnt.hh File Reference . . . . .	867
8.61.1 Detailed Description . . . . .	867
8.62 LoggableTimedSyncPnt.hh File Reference . . . . .	868
8.62.1 Detailed Description . . . . .	869
8.63 Manager.cpp File Reference . . . . .	869
8.63.1 Detailed Description . . . . .	870
8.63.2 Variable Documentation . . . . .	871
8.63.2.1 attrSRFOM__MTRInteractionHandler . . . . .	871
8.63.2.2 attrTrickHLA__Interaction . . . . .	871
8.63.2.3 attrTrickHLA__InteractionItem . . . . .	871
8.63.2.4 attrTrickHLA__Parameter . . . . .	871
8.64 Manager.hh File Reference . . . . .	871
8.64.1 Detailed Description . . . . .	872
8.65 MTRInteractionHandler.cpp File Reference . . . . .	872
8.65.1 Detailed Description . . . . .	873
8.66 MTRInteractionHandler.hh File Reference . . . . .	874
8.66.1 Detailed Description . . . . .	874
8.67 Object.cpp File Reference . . . . .	875
8.67.1 Detailed Description . . . . .	876
8.67.2 Macro Definition Documentation . . . . .	877
8.67.2.1 THLA_OBJ_DEBUG . . . . .	877
8.67.2.2 THLA_OBJ_DEBUG_RECEIVE . . . . .	877
8.67.2.3 THLA_OBJ_DEBUG_SEND . . . . .	877
8.67.2.4 THLA_OBJ_DEBUG_VALID_OBJECT_RECEIVE . . . . .	877
8.67.2.5 THLA_OBJ_OWNERSHIP_DEBUG . . . . .	877
8.67.3 Function Documentation . . . . .	877
8.67.3.1 grant_push_pthread_function() . . . . .	877

---

8.67.3.2 ownership_divestiture_pthread_function() . . . . .	878
8.68 Object.hh File Reference . . . . .	878
8.68.1 Detailed Description . . . . .	879
8.68.2 Macro Definition Documentation . . . . .	880
8.68.2.1 _TRICKHLA_OBJECT_HH_ . . . . .	880
8.68.2.2 THLA_USE_ATTRIBUTE_MAP_CLASS_INSTANCE . . . . .	880
8.69 ObjectDeleted.hh File Reference . . . . .	880
8.69.1 Detailed Description . . . . .	880
8.70 OpaqueBuffer.cpp File Reference . . . . .	881
8.70.1 Detailed Description . . . . .	882
8.71 OpaqueBuffer.hh File Reference . . . . .	882
8.71.1 Detailed Description . . . . .	883
8.72 OwnershipHandler.cpp File Reference . . . . .	884
8.72.1 Detailed Description . . . . .	884
8.73 OwnershipHandler.hh File Reference . . . . .	885
8.73.1 Detailed Description . . . . .	886
8.74 OwnershipItem.hh File Reference . . . . .	886
8.74.1 Detailed Description . . . . .	887
8.75 Packing.cpp File Reference . . . . .	888
8.75.1 Detailed Description . . . . .	888
8.76 Packing.hh File Reference . . . . .	889
8.76.1 Detailed Description . . . . .	889
8.77 Parameter.cpp File Reference . . . . .	890
8.77.1 Detailed Description . . . . .	891
8.78 Parameter.hh File Reference . . . . .	891
8.78.1 Detailed Description . . . . .	892
8.79 ParameterItem.cpp File Reference . . . . .	893
8.79.1 Detailed Description . . . . .	893
8.80 ParameterItem.hh File Reference . . . . .	894
8.80.1 Detailed Description . . . . .	895
8.81 PausePointList.cpp File Reference . . . . .	895
8.81.1 Detailed Description . . . . .	896
8.82 PausePointList.cpp File Reference . . . . .	896
8.82.1 Detailed Description . . . . .	897
8.83 PausePointList.hh File Reference . . . . .	897
8.83.1 Detailed Description . . . . .	898
8.84 PausePointList.hh File Reference . . . . .	899
8.84.1 Detailed Description . . . . .	900
8.85 PhysicalEntityBase.cpp File Reference . . . . .	901

---

8.85.1 Detailed Description . . . . .	901
8.85.2 Variable Documentation . . . . .	902
8.85.2.1 trick_MM . . . . .	902
8.86 PhysicalEntityBase.hh File Reference . . . . .	902
8.86.1 Detailed Description . . . . .	903
8.87 QuaternionData.h File Reference . . . . .	904
8.87.1 Detailed Description . . . . .	904
8.88 QuaternionEncoder.cpp File Reference . . . . .	905
8.88.1 Detailed Description . . . . .	905
8.89 QuaternionEncoder.hh File Reference . . . . .	906
8.89.1 Detailed Description . . . . .	907
8.90 README.md File Reference . . . . .	908
8.91 RefFrameBase.cpp File Reference . . . . .	908
8.91.1 Detailed Description . . . . .	908
8.91.2 Macro Definition Documentation . . . . .	909
8.91.2.1 REF_FRAME_PACKING_DEBUG . . . . .	909
8.91.2.2 REF_FRAME_PACKING_EXTRA_DEBUG . . . . .	909
8.91.3 Variable Documentation . . . . .	909
8.91.3.1 trick_MM . . . . .	909
8.92 RefFrameBase.hh File Reference . . . . .	909
8.92.1 Detailed Description . . . . .	910
8.93 RefFrameData.h File Reference . . . . .	911
8.93.1 Detailed Description . . . . .	912
8.94 ReflectedAttributesQueue.cpp File Reference . . . . .	912
8.94.1 Detailed Description . . . . .	913
8.95 ReflectedAttributesQueue.hh File Reference . . . . .	913
8.95.1 Detailed Description . . . . .	914
8.96 ScenarioTimeline.cpp File Reference . . . . .	915
8.96.1 Detailed Description . . . . .	915
8.97 ScenarioTimeline.hh File Reference . . . . .	916
8.97.1 Detailed Description . . . . .	916
8.98 SimpleSimConfig.cpp File Reference . . . . .	917
8.98.1 Detailed Description . . . . .	918
8.99 SimpleSimConfig.hh File Reference . . . . .	918
8.99.1 Detailed Description . . . . .	919
8.100 SimTimeline.cpp File Reference . . . . .	920
8.100.1 Detailed Description . . . . .	920
8.101 SimTimeline.hh File Reference . . . . .	921
8.102 SineConditional.cpp File Reference . . . . .	922

---

8.102.1 Detailed Description . . . . .	922
8.103 SineConditional.hh File Reference . . . . .	923
8.103.1 Detailed Description . . . . .	924
8.104 SineData.cpp File Reference . . . . .	925
8.104.1 Detailed Description . . . . .	925
8.105 SineData.hh File Reference . . . . .	926
8.105.1 Detailed Description . . . . .	926
8.106 SineInteractionHandler.cpp File Reference . . . . .	927
8.106.1 Detailed Description . . . . .	928
8.106.2 Macro Definition Documentation . . . . .	928
8.106.2.1 SINE_SEND_INTERACTION_TSO . . . . .	928
8.107 SineInteractionHandler.hh File Reference . . . . .	928
8.107.1 Detailed Description . . . . .	929
8.107.2 Macro Definition Documentation . . . . .	930
8.107.2.1 SINE_MSG_SIZE . . . . .	930
8.108 SineLagCompensation.cpp File Reference . . . . .	930
8.108.1 Detailed Description . . . . .	931
8.109 SineLagCompensation.hh File Reference . . . . .	931
8.109.1 Detailed Description . . . . .	932
8.110 SineObjectDeleted.cpp File Reference . . . . .	933
8.110.1 Detailed Description . . . . .	933
8.111 SineObjectDeleted.hh File Reference . . . . .	934
8.111.1 Detailed Description . . . . .	934
8.112 SineOwnershipHandler.cpp File Reference . . . . .	935
8.112.1 Detailed Description . . . . .	936
8.113 SineOwnershipHandler.hh File Reference . . . . .	936
8.113.1 Detailed Description . . . . .	937
8.114 SinePacking.cpp File Reference . . . . .	938
8.114.1 Detailed Description . . . . .	938
8.115 SinePacking.hh File Reference . . . . .	939
8.115.1 Detailed Description . . . . .	939
8.116 SpaceTimeCoordinateData.h File Reference . . . . .	940
8.116.1 Detailed Description . . . . .	940
8.117 SpaceTimeCoordinateEncoder.cpp File Reference . . . . .	941
8.117.1 Detailed Description . . . . .	941
8.118 SpaceTimeCoordinateEncoder.hh File Reference . . . . .	942
8.118.1 Detailed Description . . . . .	943
8.119 StandardsSupport.hh File Reference . . . . .	943
8.119.1 Detailed Description . . . . .	943

---

---

8.119.2 Macro Definition Documentation . . . . .	944
8.119.2.1 IEEE_1516_2010 . . . . .	944
8.119.2.2 RTI1516_EXCEPTION . . . . .	944
8.119.2.3 RTI1516_HEADER . . . . .	944
8.119.2.4 RTI1516_NAMESPACE . . . . .	944
8.119.2.5 RTI1516_USERDATA . . . . .	944
8.120 StringUtilities.hh File Reference . . . . .	945
8.120.1 Detailed Description . . . . .	945
8.120.2 Macro Definition Documentation . . . . .	946
8.120.2.1 WHITESPACE_CHARS . . . . .	946
8.121 SyncPnt.cpp File Reference . . . . .	946
8.121.1 Detailed Description . . . . .	947
8.122 SyncPnt.hh File Reference . . . . .	947
8.122.1 Detailed Description . . . . .	948
8.123 SyncPntList.hh File Reference . . . . .	948
8.123.1 Detailed Description . . . . .	949
8.124 SyncPntListBase.cpp File Reference . . . . .	950
8.124.1 Detailed Description . . . . .	950
8.125 SyncPntListBase.hh File Reference . . . . .	951
8.125.1 Detailed Description . . . . .	952
8.126 TimedSyncPnt.cpp File Reference . . . . .	952
8.126.1 Detailed Description . . . . .	953
8.127 TimedSyncPnt.hh File Reference . . . . .	953
8.127.1 Detailed Description . . . . .	954
8.128 TimedSyncPntList.cpp File Reference . . . . .	955
8.128.1 Detailed Description . . . . .	955
8.129 TimedSyncPntList.hh File Reference . . . . .	956
8.129.1 Detailed Description . . . . .	957
8.130 Timeline.cpp File Reference . . . . .	958
8.130.1 Detailed Description . . . . .	958
8.131 Timeline.hh File Reference . . . . .	959
8.131.1 Detailed Description . . . . .	959
8.132 TimeOfDayTimeline.cpp File Reference . . . . .	960
8.132.1 Detailed Description . . . . .	960
8.133 TimeOfDayTimeline.hh File Reference . . . . .	961
8.133.1 Detailed Description . . . . .	962
8.134 Types.cpp File Reference . . . . .	963
8.134.1 Detailed Description . . . . .	963
8.135 Types.hpp File Reference . . . . .	964

---

8.135.1 Detailed Description . . . . .	964
8.136 Types.cpp File Reference . . . . .	964
8.136.1 Detailed Description . . . . .	965
8.137 Types.cpp File Reference . . . . .	965
8.137.1 Detailed Description . . . . .	966
8.138 Types.cpp File Reference . . . . .	966
8.138.1 Detailed Description . . . . .	967
8.139 Types.hh File Reference . . . . .	967
8.139.1 Detailed Description . . . . .	969
8.140 Types.hh File Reference . . . . .	970
8.140.1 Detailed Description . . . . .	971
8.141 Types.hh File Reference . . . . .	972
8.141.1 Detailed Description . . . . .	973
8.142 Types.hh File Reference . . . . .	974
8.142.1 Detailed Description . . . . .	975
8.143 Types.hh File Reference . . . . .	976
8.143.1 Detailed Description . . . . .	979
8.144 Utilities.cpp File Reference . . . . .	979
8.144.1 Detailed Description . . . . .	979
8.145 Utilities.hh File Reference . . . . .	980
8.145.1 Detailed Description . . . . .	981
8.145.2 Macro Definition Documentation . . . . .	982
8.145.2.1 _FPU_PC_DOUBLE . . . . .	982
8.145.2.2 _FPU_PC_EXTENDED . . . . .	982
8.145.2.3 _FPU_PC_MASK . . . . .	982
8.145.2.4 _FPU_PC_SINGLE . . . . .	982
8.145.2.5 _FPU_PC_UNDEFINED . . . . .	982
8.145.2.6 TRICKHLA_INIT_FPU_CONTROL_WORD . . . . .	982
8.145.2.7 TRICKHLA_RESTORE_FPU_CONTROL_WORD . . . . .	982
8.145.2.8 TRICKHLA_SAVE_FPU_CONTROL_WORD . . . . .	982
8.145.2.9 TRICKHLA_VALIDATE_FPU_CONTROL_WORD . . . . .	983
8.146 Version.hh File Reference . . . . .	983
8.146.1 Detailed Description . . . . .	983
8.146.2 Macro Definition Documentation . . . . .	984
8.146.2.1 TRICKHLA_RELEASE_DATE . . . . .	984
8.146.2.2 TRICKHLA_VERSION . . . . .	984
Index . . . . .	985



## Chapter 1

# TrickHLA: An IEEE 1516 High Level Architecture (HLA) Simulation Interoperability Standard Implementation for Trick Base Simulations

### **Brief Abstract:**

The Trick High Level Architecture ([TrickHLA](#)) software supports IEEE-1516: High Level Architecture (HLA) based simulation interoperability with Trick based simulations. The [TrickHLA](#) software abstracts away the details of using the HLA, allowing the user to concentrate on the simulation and not worry about having to be an HLA and/or distributed simulation expert. The [TrickHLA](#) software is data driven and provides a simple Application Programming Interface (API) making it relatively easy to take an existing Trick simulation and make it HLA aware (i.e. a distributed simulation).

### **Copyright:**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### **Responsible Organization:**

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

### **Contact:**

Edwin Z. Crues  
281-483-2902  
[edwin.z.crues@nasa.gov](mailto:edwin.z.crues@nasa.gov)



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">DIS</a>	17
<a href="#">DSES</a>	24
<a href="#">IMSim</a>	29
<a href="#">SpaceFOM</a>	37
<a href="#">TrickHLA</a>	43
<a href="#">TrickHLAModel</a>	57



# Chapter 3

## Hierarchical Index

### 3.1 Class Hierarchy

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

TrickHLA::Attribute . . . . .	59
TrickHLA::BasicClock . . . . .	86
Clock	
TrickHLA::CTETimelineBase . . . . .	89
TrickHLA::Conditional . . . . .	87
TrickHLAModel::SineConditional . . . . .	676
TrickHLA::DebugHandler . . . . .	94
TrickHLA::DivestThreadArgs . . . . .	98
TrickHLA::Federate . . . . .	322
FederateAmbassador	
TrickHLA::FedAmb . . . . .	302
TrickHLA::Int64Interval . . . . .	411
TrickHLA::Int64Time . . . . .	419
TrickHLA::Interaction . . . . .	434
TrickHLA::InteractionHandler . . . . .	451
IMSim::FreezeInteractionHandler . . . . .	406
SpaceFOM::MTRInteractionHandler . . . . .	523
TrickHLAModel::SineInteractionHandler . . . . .	691
TrickHLA::Item . . . . .	466
TrickHLA::InteractionItem . . . . .	458
TrickHLA::OwnershipItem . . . . .	596
TrickHLA::ParameterItem . . . . .	623
TrickHLA::ItemQueue . . . . .	468
TrickHLA::KnownFederate . . . . .	474
TrickHLA::LagCompensation . . . . .	477
TrickHLAModel::SineLagCompensation . . . . .	695
TrickHLA::LoggableSyncPnt . . . . .	483
TrickHLA::LoggableTimedSyncPnt . . . . .	485
TrickHLA::Manager . . . . .	488
TrickHLA::Object . . . . .	530
TrickHLA::ExecutionConfigurationBase . . . . .	153
DIS::ExecutionConfiguration . . . . .	103
DSES::ExecutionConfiguration . . . . .	113
IMSim::ExecutionConfiguration . . . . .	123

SpaceFOM::ExecutionConfiguration . . . . .	135
TrickHLA::ExecutionConfiguration . . . . .	148
TrickHLA::ObjectDeleted . . . . .	575
TrickHLAModel::SineObjectDeleted . . . . .	701
TrickHLA::OpaqueBuffer . . . . .	576
SpaceFOM::PhysicalEntityBase . . . . .	638
SpaceFOM::DynamicalEntity . . . . .	99
SpaceFOM::QuaternionEncoder . . . . .	646
SpaceFOM::SpaceTimeCoordinateEncoder . . . . .	715
TrickHLA::OwnershipHandler . . . . .	584
TrickHLAModel::SineOwnershipHandler . . . . .	704
TrickHLA::Packing . . . . .	599
SpaceFOM::PhysicalEntityBase . . . . .	638
SpaceFOM::RefFrameBase . . . . .	650
TrickHLA::ExecutionConfigurationBase . . . . .	153
TrickHLAModel::SimpleSimConfig . . . . .	669
TrickHLAModel::SinePacking . . . . .	706
TrickHLA::Parameter . . . . .	605
QuaternionData . . . . .	645
RefFrameData . . . . .	658
TrickHLA::ReflectedAttributesQueue . . . . .	659
TrickHLAModel::SineData . . . . .	680
SpaceTimeCoordinateData . . . . .	713
TrickHLA::StringUtilities . . . . .	721
TrickHLA::SyncPnt . . . . .	729
TrickHLA::TimedSyncPnt . . . . .	759
TrickHLA::SyncPntListBase . . . . .	742
TrickHLA::ExecutionControlBase . . . . .	256
DIS::ExecutionControl . . . . .	163
DSES::ExecutionControl . . . . .	181
IMSim::ExecutionControl . . . . .	196
SpaceFOM::ExecutionControl . . . . .	221
TrickHLA::ExecutionControl . . . . .	244
TrickHLA::SyncPntList . . . . .	739
TrickHLA::TimedSyncPntList . . . . .	764
DIS::PausePointList . . . . .	633
IMSim::PausePointList . . . . .	626
TrickHLA::Timeline . . . . .	769
TrickHLA::CTETimelineBase . . . . .	89
TrickHLA::ScenarioTimeline . . . . .	663
TrickHLA::SimTimeline . . . . .	674
TrickHLA::TimeOfDayTimeline . . . . .	773
TrickHLA::Utilities . . . . .	776

## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

TrickHLA::Attribute	59
TrickHLA::BasicClock	86
TrickHLA::Conditional	87
TrickHLA::CTETimelineBase	89
TrickHLA::DebugHandler	94
TrickHLA::DivestThreadArgs	98
SpaceFOM::DynamicalEntity	99
DIS::ExecutionConfiguration	103
DSES::ExecutionConfiguration	113
IMSim::ExecutionConfiguration	123
SpaceFOM::ExecutionConfiguration	135
TrickHLA::ExecutionConfiguration	148
TrickHLA::ExecutionConfigurationBase	153
DIS::ExecutionControl	163
DSES::ExecutionControl	181
IMSim::ExecutionControl	196
SpaceFOM::ExecutionControl	221
TrickHLA::ExecutionControl	244
TrickHLA::ExecutionControlBase	256
TrickHLA::FedAmb	302
TrickHLA::Federate	322
IMSim::FreezeInteractionHandler	406
TrickHLA::Int64Interval	411
TrickHLA::Int64Time	419
TrickHLA::Interaction	434
TrickHLA::InteractionHandler	451
TrickHLA::InteractionItem	458
TrickHLA::Item	466
TrickHLA::ItemQueue	468
TrickHLA::KnownFederate	474
TrickHLA::LagCompensation	477
TrickHLA::LoggableSyncPnt	483
TrickHLA::LoggableTimedSyncPnt	485
TrickHLA::Manager	488
SpaceFOM::MTRInteractionHandler	523
TrickHLA::Object	530

TrickHLA::ObjectDeleted	575
TrickHLA::OpaqueBuffer	576
TrickHLA::OwnershipHandler	584
TrickHLA::OwnershipItem	596
TrickHLA::Packing	599
TrickHLA::Parameter	605
TrickHLA::ParameterItem	623
IMSim::PausePointList	626
DIS::PausePointList	633
SpaceFOM::PhysicalEntityBase	638
QuaternionData	645
SpaceFOM::QuaternionEncoder	646
SpaceFOM::RefFrameBase	650
RefFrameData	658
TrickHLA::ReflectedAttributesQueue	659
TrickHLA::ScenarioTimeline	663
TrickHLAModel::SimpleSimConfig	669
TrickHLA::SimTimeline	674
TrickHLAModel::SineConditional	676
TrickHLAModel::SineData	680
TrickHLAModel::SineInteractionHandler	691
TrickHLAModel::SineLagCompensation	695
TrickHLAModel::SineObjectDeleted	701
TrickHLAModel::SineOwnershipHandler	704
TrickHLAModel::SinePacking	706
SpaceTimeCoordinateData	713
SpaceFOM::SpaceTimeCoordinateEncoder	715
TrickHLA::StringUtilities	721
TrickHLA::SyncPnt	729
TrickHLA::SyncPntList	739
TrickHLA::SyncPntListBase	742
TrickHLA::TimedSyncPnt	759
TrickHLA::TimedSyncPntList	764
TrickHLA::Timeline	769
TrickHLA::TimeOfDayTimeline	773
TrickHLA::Utilities	776

# Chapter 5

## File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">Attribute.cpp</a>	This class represents the HLA attributes of an object that is managed by Trick	783
<a href="#">Attribute.hh</a>	This class represents the HLA attributes of an object that is managed by Trick	784
<a href="#">BasicClock.hh</a>	This class provides a function to get the current GMT time by using either the CTE external clock if enabled or the system time-of-day	786
<a href="#">CompileConfig.hh</a>	TrickHLA compile time configuration settings	787
<a href="#">Conditional.cpp</a>	This class provides a user API for the handling of a CONDITIONAL attribute	790
<a href="#">Conditional.hh</a>	This class provides a user API for the handling of a CONDITIONAL attribute	791
<a href="#">Constants.hh</a>	Define the global time constants used in TrickHLA	792
<a href="#">CTETimelineBase.cpp</a>	A base implementation of a SpaceFOM CTE timeline	793
<a href="#">CTETimelineBase.hh</a>	This class represents the CTE timeline	794
<a href="#">DebugHandler.hh</a>	Multi-level debug reporter	796
<a href="#">DynamicalEntity.cpp</a>	This class provides data packing for the SpaceFOM Reference Frames	797
<a href="#">DynamicalEntity.hh</a>	Definition of the TrickHLA SpaceFOM Dynamical entity type	798
<a href="#">DIS/ExecutionConfiguration.cpp</a>	Implementation of the TrickHLA DIS Execution Configuration Object (ExCO)	800
<a href="#">DSES/ExecutionConfiguration.cpp</a>	Implementation of the TrickHLA DSES Execution Configuration Object (ExCO)	801
<a href="#">IMSim/ExecutionConfiguration.cpp</a>	Implementation of the TrickHLA IMSim Execution Configuration Object (ExCO)	803
<a href="#">SpaceFOM/ExecutionConfiguration.cpp</a>	Implementation of the TrickHLA SpaceFOM Execution Configuration Object (ExCO)	804
<a href="#">TrickHLA/ExecutionConfiguration.cpp</a>	Implementation of the TrickHLA TrickHLA Execution Configuration Object (ExCO)	806

<a href="#">DIS/ExecutionConfiguration.h</a>	
Definition of the <a href="#">TrickHLA DIS</a> Execution Configuration Object (ExCO) . . . . .	807
<a href="#">DSES/ExecutionConfiguration.h</a>	
Definition of the <a href="#">TrickHLA DSES</a> Execution Configuration Object (ExCO) . . . . .	809
<a href="#">IMSim/ExecutionConfiguration.h</a>	
Definition of the <a href="#">TrickHLA IMSim</a> Execution Configuration Object (ExCO) . . . . .	811
<a href="#">SpaceFOM/ExecutionConfiguration.h</a>	
Definition of the <a href="#">TrickHLA SpaceFOM</a> Execution Configuration Object (ExCO) . . . . .	812
<a href="#">TrickHLA/ExecutionConfiguration.h</a>	
Definition of the <a href="#">TrickHLA</a> simple Execution Configuration Object (ExCO) . . . . .	814
<a href="#">ExecutionConfigurationBase.cpp</a>	
The abstract base class for the <a href="#">TrickHLA</a> simulation execution configuration class . . . . .	815
<a href="#">ExecutionConfigurationBase.h</a>	
The abstract base class for the <a href="#">TrickHLA</a> simulation execution configuration class . . . . .	817
<a href="#">DIS/ExecutionControl.cpp</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	818
<a href="#">DSES/ExecutionControl.cpp</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	819
<a href="#">IMSim/ExecutionControl.cpp</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	821
<a href="#">SpaceFOM/ExecutionControl.cpp</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	822
<a href="#">TrickHLA/ExecutionControl.cpp</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	824
<a href="#">DIS/ExecutionControl.h</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	825
<a href="#">DSES/ExecutionControl.h</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	827
<a href="#">IMSim/ExecutionControl.h</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	828
<a href="#">SpaceFOM/ExecutionControl.h</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	830
<a href="#">TrickHLA/ExecutionControl.h</a>	
This class provides and abstract base class as the base implementation for <a href="#">TrickHLA</a> simple execution control . . . . .	831
<a href="#">ExecutionControlBase.cpp</a>	
This class provides and abstract base class as the base implementation for managing mode transitions . . . . .	832
<a href="#">ExecutionControlBase.h</a>	
This class provides and abstract base class as the base implementation for execution control . . . . .	834
<a href="#">FedAmb.cpp</a>	
Provides methods called by the RTI Ambassador for simulation object, interaction and time management . . . . .	835
<a href="#">FedAmb.h</a>	
Provides methods called by the HLA RTI Ambassador . . . . .	836

<a href="#">Federate.cpp</a>	This class is the abstract base class for representing timelines . . . . .	838
<a href="#">Federate.hh</a>	This class provides basic services for connecting a Trick based simulation in to a HLA based distributed simulation environment . . . . .	841
<a href="#">FreezeInteractionHandler.cpp</a>	This class handles the HLA Freeze interactions . . . . .	842
<a href="#">FreezeInteractionHandler.hh</a>	This class is a specialized class for handling HLA Freeze interactions . . . . .	844
<a href="#">Int64Interval.cpp</a>	This class represents the HLA Interval time . . . . .	845
<a href="#">Int64Interval.hh</a>	This class represents the HLA Interval time . . . . .	846
<a href="#">Int64Time.cpp</a>		848
<a href="#">Int64Time.hh</a>	This class represents the HLA time . . . . .	848
<a href="#">integrator.d</a>		850
<a href="#">Interaction.cpp</a>	This class represents an HLA Interaction that is managed by Trick . . . . .	851
<a href="#">Interaction.hh</a>	This class represents an HLA Interaction that is managed by Trick . . . . .	852
<a href="#">InteractionHandler.cpp</a>	This class is the abstract base class for handling HLA interactions . . . . .	853
<a href="#">InteractionHandler.hh</a>	This class is the abstract base class for handling HLA interactions . . . . .	855
<a href="#">InteractionItem.cpp</a>	This class represents a queue for holding HLA Interactions of either Timestamp Order (TSO) or Receive Order (RO) . . . . .	856
<a href="#">InteractionItem.hh</a>	This class represents a queue for holding HLA Interactions of either Timestamp Order (TSO) or Receive Order (RO) . . . . .	857
<a href="#">Item.cpp</a>	This class represents an Item in the item queue . . . . .	859
<a href="#">Item.hh</a>	This class represents a item to be held in an Item Queue . . . . .	860
<a href="#">ItemQueue.cpp</a>	This class represents a queue for holding Items . . . . .	861
<a href="#">ItemQueue.hh</a>	This class represents a queue for holding Items . . . . .	862
<a href="#">KnownFederate.hh</a>	A class representing an HLA federate known to the Federation . . . . .	863
<a href="#">LagCompensation.cpp</a>	This class is the abstract base class for Trick HLA lag compensation . . . . .	864
<a href="#">LagCompensation.hh</a>	This class is the abstract base class for <a href="#">TrickHLA</a> lag compensation . . . . .	865
<a href="#">LoggableSyncPnt.hh</a>	This class provides a mechanism for logging sync point data and retrieving it from the log file . . . . .	867
<a href="#">LoggableTimedSyncPnt.hh</a>	This class provides a mechanism for logging timed synchronization point data and retrieving it from the log file . . . . .	868
<a href="#">Manager.cpp</a>	This class manages the interface between a Trick simulation and HLA . . . . .	869
<a href="#">Manager.hh</a>	This class manages the interface between a Trick simulation and HLA . . . . .	871

<a href="#">MTRInteractionHandler.cpp</a>	This class handles the HLA interactions for Space Reference FOM ( <a href="#">SpaceFOM</a> ) Mode Transition Request (MTR) interaction . . . . .	872
<a href="#">MTRInteractionHandler.hh</a>	This is the base implementation for the Space Reference FOM ( <a href="#">SpaceFOM</a> ) Mode Transition Request (MTR) interaction handler . . . . .	874
<a href="#">Object.cpp</a>	This class represents an HLA object that is managed by Trick . . . . .	875
<a href="#">Object.hh</a>	This class represents an HLA object that is managed by Trick . . . . .	878
<a href="#">ObjectDeleted.hh</a>	This class is the abstract base class for a callback of identification of deleted objects from the RTI .	880
<a href="#">OpaqueBuffer.cpp</a>	This class provides a generic opaque buffer that is in the Trick managed memory space . . . . .	881
<a href="#">OpaqueBuffer.hh</a>	This class provides a generic opaque buffer that is in the Trick managed memory space. This is useful for sending a fixed record of data that includes byte padding to ensure a byte alignment . .	882
<a href="#">OwnershipHandler.cpp</a>	This class represents ownership transfer of HLA attributes for a specific object . . . . .	884
<a href="#">OwnershipHandler.hh</a>	This class represents ownership transfer of HLA attributes for a specific object . . . . .	885
<a href="#">OwnershipItem.hh</a>	This class represents a queue item for holding ownership transfers of an attribute . . . . .	886
<a href="#">Packing.cpp</a>	This class is the abstract base class for Trick HLA Packing class . . . . .	888
<a href="#">Packing.hh</a>	Definition of the <a href="#">TrickHLA</a> Packing class . . . . .	889
<a href="#">Parameter.cpp</a>	This class represents the HLA parameters of an interaction that is managed by Trick . . . . .	890
<a href="#">Parameter.hh</a>	This class represents the HLA parameters of an interaction that is managed by Trick . . . . .	891
<a href="#">ParameterItem.cpp</a>	This class represents a queue for holding HLA parameters . . . . .	893
<a href="#">ParameterItem.hh</a>	This class represents a queue for holding HLA parameters . . . . .	894
<a href="#">DIS/PausePointList.cpp</a>	Represents an HLA Synchronization Point in Trick . . . . .	895
<a href="#">IMSim/PausePointList.cpp</a>	Represents an HLA Synchronization Point in Trick . . . . .	896
<a href="#">DIS/PausePointList.hh</a>	This class provides a mechanism for storing and managing HLA synchronization points for Trick . .	897
<a href="#">IMSim/PausePointList.hh</a>	This class provides a mechanism for storing and managing HLA synchronization points for Trick . .	899
<a href="#">PhysicalEntityBase.cpp</a>	This class provides data packing for the <a href="#">SpaceFOM</a> Reference Frames . . . . .	901
<a href="#">PhysicalEntityBase.hh</a>	Definition of the <a href="#">TrickHLA SpaceFOM</a> physical entity type . . . . .	902
<a href="#">QuaternionData.h</a>	A simple structure that contains the date fields required to encode and decode a SISO Space Reference FOM Quaternion data type . . . . .	904
<a href="#">QuaternionEncoder.cpp</a>	This file contains the methods for the QuaternionEncoder class . . . . .	905
<a href="#">QuaternionEncoder.hh</a>	Definition of the <a href="#">TrickHLA SpaceFOM</a> quaternion encoding utility . . . . .	906

RefFrameBase.cpp	This class provides data packing for the SpaceFOM Reference Frames . . . . .	908
RefFrameBase.hh	This class provides data packing for the SpaceFOM Reference Frames . . . . .	909
RefFrameData.h	A simple structure that contains the date fields required to encode and decode a SISO Space Reference FOM ReferenceFrame data type . . . . .	911
ReflectedAttributesQueue.cpp	This is a protected queue class to hold reflected attributes . . . . .	912
ReflectedAttributesQueue.hh	TrickHLA Queue of AttributeHandleValueMap . . . . .	913
ScenarioTimeline.cpp	This class represents the simulation timeline . . . . .	915
ScenarioTimeline.hh	This class represents the scenario timeline . . . . .	916
SimpleSimConfig.cpp	This class contains a basic simulation configuration . . . . .	917
SimpleSimConfig.hh	This class contains a basic simulation configuration . . . . .	918
SimTimeline.cpp	This class represents the simulation timeline . . . . .	920
SimTimeline.hh		921
SineConditional.cpp	Subclass the base class to provide sine wave-specific CONDITIONAL attribute . . . . .	922
SineConditional.hh	Subclass the base class to provide sine wave-specific CONDITIONAL attribute . . . . .	923
SineData.cpp	This class is the working class for the Sine HLA/RTI example problem . . . . .	925
SineData.hh	This is a container class for general test data used in the general HLA test routines . . . . .	926
SineInteractionHandler.cpp	This class handles the HLA interactions for the sine wave simulation . . . . .	927
SineInteractionHandler.hh	This class handles the HLA interactions for the sine wave simulation . . . . .	928
SineLagCompensation.cpp	This class provides lag compensation for the sine wave object . . . . .	930
SineLagCompensation.hh	Send and receiving side lag compensation . . . . .	931
SineObjectDeleted.cpp	Callback class the user writes to do something once the object has been deleted from the RTI . . . . .	933
SineObjectDeleted.hh	Callback class the user writes to do something once the object has been deleted from the RTI . . . . .	934
SineOwnershipHandler.cpp	This class handles the HLA ownership transfer for the sine wave simulation . . . . .	935
SineOwnershipHandler.hh	Ownership transfer for the HLA object attributes . . . . .	936
SinePacking.cpp	This class provides data packing for the sine wave data . . . . .	938
SinePacking.hh	This class provides data packing . . . . .	939
SpaceTimeCoordinateData.h	A simple structure that contains the date fields required to encode and decode a SISO Space Reference FOM Space/Time Coordinate data type . . . . .	940

SpaceTimeCoordinateEncoder.cpp	This file contains the methods for the SpaceTimeCoordinate encoder class . . . . .	941
SpaceTimeCoordinateEncoder.h	Definition of the <a href="#">TrickHLA SpaceFOM</a> Space/Time coordinate encoder . . . . .	942
StandardsSupport.hh	This header file provides the <a href="#">TrickHLA</a> support necessary to hide the differences between the different HLA Standards that implement the Runtime Infrastructure (RTI) . . . . .	943
StringUtilities.hh	String utilities . . . . .	945
SyncPnt.cpp	This class provides a sync-point implementation for storing and managing <a href="#">TrickHLA</a> synchronization points . . . . .	946
SyncPnt.h	This class provides a sync-point implementation for storing and managing <a href="#">TrickHLA</a> synchronization points . . . . .	947
SyncPntList.h	This class extends the <a href="#">TrickHLA::SyncPntListBase</a> class and provides an instantiable implementation for storing and managing HLA synchronization points for <a href="#">TrickHLA</a> . . . . .	948
SyncPntListBase.cpp	This class provides and abstract base class as the base implementation for storing and managing HLA synchronization points for Trick . . . . .	950
SyncPntListBase.hh	This class provides and abstract base class as the base implementation for storing and managing HLA synchronization points for Trick . . . . .	951
TimedSyncPnt.cpp	This class provides a sync-point implementation for storing and managing <a href="#">TrickHLA</a> synchronization points . . . . .	952
TimedSyncPnt.h	This class extends the basis <a href="#">TrickHLA::SyncPnt</a> synchronization point implementation to add a time stamp . . . . .	953
TimedSyncPntList.cpp	This class provides and abstract base class as the base implementation for storing and managing HLA synchronization points for Trick . . . . .	955
TimedSyncPntList.h	This class extends the <a href="#">TrickHLA::SyncPntListBase</a> class and provides an instantiable implementation for storing and managing HLA synchronization points for <a href="#">TrickHLA</a> . . . . .	956
Timeline.cpp	This class represents the HLA time . . . . .	958
Timeline.h	This class is the abstract base class for representing timelines . . . . .	959
TimeOfDayTimeline.cpp	This class represents the simulation timeline . . . . .	960
TimeOfDayTimeline.h	This class represents the time of day timeline . . . . .	961
DIS/Types.cpp	Implementation of the <a href="#">TrickHLA DIS</a> types utility functions . . . . .	963
DSES/Types.cpp	Implementation of the <a href="#">TrickHLA DSES</a> types utility functions . . . . .	964
IMSim/Types.cpp	Implementation of the <a href="#">TrickHLA IMSim</a> types utility functions . . . . .	964
SpaceFOM/Types.cpp	Implementation of the <a href="#">TrickHLA SpaceFOM</a> types utility functions . . . . .	965
TrickHLA/Types.cpp	Implementation of the <a href="#">TrickHLA</a> types utility functions . . . . .	966

---

<a href="#">DIS/Types.hh</a>	Definition of the <a href="#">TrickHLA DIS</a> enumeration types and utilities . . . . .	967
<a href="#">DSES/Types.hh</a>	Definition of the <a href="#">TrickHLA DSES</a> enumeration types and utilities . . . . .	970
<a href="#">IMSim/Types.hh</a>	Definition of the <a href="#">TrickHLA IMSim</a> enumeration types and utilities . . . . .	972
<a href="#">SpaceFOM/Types.hh</a>	Definition of the <a href="#">TrickHLA SpaceFOM</a> enumeration types and utilities . . . . .	974
<a href="#">TrickHLA/Types.hh</a>	Definition of the <a href="#">TrickHLA</a> enumeration types and utilities . . . . .	976
<a href="#">Utilities.cpp</a>	Implementation of the <a href="#">TrickHLA</a> utilities . . . . .	979
<a href="#">Utilities.hh</a>	Definition of the <a href="#">TrickHLA</a> utilities . . . . .	980
<a href="#">Version.hh</a>	Definition of the <a href="#">TrickHLA</a> version tag . . . . .	983



# Chapter 6

## Namespace Documentation

### 6.1 DIS Namespace Reference

#### Data Structures

- class `ExecutionConfiguration`
- class `ExecutionControl`
- class `PausePointList`

#### Enumerations

- enum `ExecutionModeEnum` {  
  `EXECUTION_MODE_FIRST_VALUE` = 0, `EXECUTION_MODE_UNINITIALIZED` = 0, `EXECUTION_MODE_INITIALIZING` = 1, `EXECUTION_MODE_RUNNING` = 2,  
  `EXECUTION_MODE_FREEZE` = 3, `EXECUTION_MODE_SHUTDOWN` = 4, `EXECUTION_MODE_LAST_VALUE` = 4 }  
    *Define the TrickHLA DIS execution mode enumeration values.*
- enum `MTREnum` {  
  `MTR_FIRST_VALUE` = 0, `MTR_UNINITIALIZED` = 0, `MTR_INITIALIZING` = 1, `MTR_GOTO_RUN` = 2,  
  `MTR_GOTO_FREEZE` = 3, `MTR_GOTO_SHUTDOWN` = 4, `MTR_LAST_VALUE` = 4 }  
    *Define the TrickHLA DIS Mode Transition Request state enumeration values.*
- enum `PausePointStateEnum` {  
  `PAUSE_POINT_STATE_FIRST_VALUE` = 0, `PAUSE_POINT_STATE_ERROR` = 0, `PAUSE_POINT_STATE_PENDING` = 1, `PAUSE_POINT_STATE_ACKNOWLEDGED` = 2,  
  `PAUSE_POINT_STATE_RUN` = 3, `PAUSE_POINT_STATE_FREEZE` = 4, `PAUSE_POINT_STATE_EXIT` = 5,  
  `PAUSE_POINT_STATE_RESTART` = 6,  
  `PAUSE_POINT_STATE_RECONFIG` = 7, `PAUSE_POINT_STATE_UNKNOWN` = `INT_MAX` }  
    *Define the TrickHLA synchronization point state enumeration values.*

#### Functions

- std::string `execution_mode_enum_to_string` (`ExecutionModeEnum` mode)  
    *Convert an ExecutionModeEnum value into a printable string.*
- `int16_t execution_mode_enum_to_int16` (`ExecutionModeEnum` mode)  
    *Convert an ExecutionModeEnum value into a 16 bit integer.*
- `ExecutionModeEnum execution_mode_int16_to_enum` (`int16_t int_mode`)  
    *Convert a 16 bit integer to an ExecutionModeEnum value.*
- `TrickHLA::ExecutionControlEnum to_execution_control_enum` (`ExecutionModeEnum` mode)

- Convert an [DIS::ExecutionModeEnum](#) value to a [TrickHLA::ExecutionModeEnum](#) value.
- ExecutionModeEnum from\_execution\_contorl\_enum ([TrickHLA::ExecutionControlEnum](#) mode)
  - Convert a [TrickHLA::ExecutionModeEnum](#) value to an [DIS::ExecutionModeEnum](#) value.
  - std::string mtr\_enum\_to\_string ([MTREnum](#) mtr\_enum)
    - Convert a Mode Transition Request (MTR) enum value into a printable string.
  - int16\_t mtr\_enum\_to\_int16 ([MTREnum](#) mtr\_enum)
    - Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.
  - MTREnum mtr\_int16\_to\_enum (int16\_t mtr\_int)
    - Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.
  - std::string pause\_pnt\_state\_enum\_to\_string ([PausePointStateEnum](#) state)
    - Convert a Pause Synchronization Point State enum value into a printable string.
  - int16\_t pause\_pnt\_state\_enum\_to\_int16 ([PausePointStateEnum](#) state)
    - Convert a Pause Synchronization Point State enum value into a 16 bit integer.
  - PausePointStateEnum pause\_pnt\_state\_int16\_to\_enum (int16\_t int\_state)
    - Convert an integer value to a Pause Synchronization Point State enumeration value.

## Variables

- static const std::wstring INITIALIZE\_SYNC\_POINT = L"initialize"
- static const std::wstring STARTUP\_SYNC\_POINT = L"startup"
- static const std::wstring STARTUP\_FREEZE\_SYNC\_POINT = L"pause\_0.0"

### 6.1.1 Enumeration Type Documentation

#### 6.1.1.1 ExecutionModeEnum

enum [DIS::ExecutionModeEnum](#)

Define the [TrickHLA](#) DIS execution mode enumeration values.

Define the [TrickHLA SpaceFOM](#) execution mode enumeration values.

Define the [TrickHLA IMSim](#) execution mode enumeration values.

Define the [TrickHLA DSES](#) execution mode enumeration values.

The ExecutionModeEnum enumeration defines the possible execution mode state for a [DIS](#) compliant federate. These mode states are important in the execution control process commanded by a Master federate and followed all other federates participating in a [DIS](#) compliant federation execution.

The ExecutionModeEnum enumeration defines the possible execution mode state for a [DSES](#) compliant federate. These mode states are important in the execution control process commanded by a Master federate and followed all other federates participating in a [DSES](#) compliant federation execution.

The ExecutionModeEnum enumeration defines the possible execution mode state for a [IMSim](#) compliant federate. These mode states are important in the execution control process commanded by a Master federate and followed all other federates participating in a [IMSim](#) compliant federation execution.

The ExecutionModeEnum enumeration defines the possible execution mode state for a Space Reference FOM compliant federate. These mode states are important in the execution control process commanded by a Master federate and followed all other federates participating in a Space FOM compliant federation execution.

#### Enumerator

<a href="#">EXECUTION_MODE_FIRST_VALUE</a>	Same as uninitialized.
<a href="#">EXECUTION_MODE_UNINITIALIZED</a>	Execution mode UNINITIALIZED.
<a href="#">EXECUTION_MODE_INITIALIZING</a>	Execution mode INITIALIZING.
<a href="#">EXECUTION_MODE_RUNNING</a>	Execution mode RUNNING.

## Enumerator

EXECUTION_MODE_FREEZE	Execution mode FREEZE.
EXECUTION_MODE_SHUTDOWN	Execution mode SHUTDOWN.
EXECUTION_MODE_LAST_VALUE	Same as shutdown.

Definition at line 54 of file DIS/Types.hh.

### 6.1.1.2 MTREnum

enum `DIS::MTREnum`

Define the [TrickHLA](#) DIS Mode Transition Request state enumeration values.

Define the [TrickHLA](#) SpaceFOM Mode Transition Request state enumeration values.

Define the [TrickHLA](#) IMSim Mode Transition Request state enumeration values.

Define the [TrickHLA](#) DSES Mode Transition Request state enumeration values.

The MTREnum enumeration defines the possible mode transition requests (MTRs) for a Space Reference FOM compliant federate. These mode requests are important in the execution control process involving mode requests from any federate participating in a Space FOM compliant federation execution and processed by the Master federate.

## Enumerator

MTR_FIRST_VALUE	Not a valid mode transition.
MTR_UNINITIALIZED	Not a valid mode transition.
MTR_INITIALIZING	Not a valid mode transition.
MTR_GOTO_RUN	Mode transition to RUN mode.
MTR_GOTO_FREEZE	Mode transition to FREEZE mode.
MTR_GOTO_SHUTDOWN	Mode transition to SHUTDOWN mode.
MTR_LAST_VALUE	Same as shutdown.

Definition at line 77 of file DIS/Types.hh.

### 6.1.1.3 PausePointStateEnum

enum `DIS::PausePointStateEnum`

Define the [TrickHLA](#) synchronization point state enumeration values.

The PausePointStateEnum enumeration defines the possible pause point (synchronization point) synchronization states for a [TrickHLA](#) based federate.

## Enumerator

PAUSE_POINT_STATE_FIRST_VALUE	Set to the First value in the enumeration.
PAUSE_POINT_STATE_ERROR	Pause point state error.
PAUSE_POINT_STATE_PENDING	Pause point state pending.
PAUSE_POINT_STATE_ACKNOWLEDGED	Pause point state acknowledged.
PAUSE_POINT_STATE_RUN	Pause point state run.
PAUSE_POINT_STATE_FREEZE	Pause point state freeze.
PAUSE_POINT_STATE_EXIT	Pause point state exit.
PAUSE_POINT_STATE_RESTART	Pause point state restart.
PAUSE_POINT_STATE_RECONFIG	Pause point state reconfiguration.
PAUSE_POINT_STATE_UNKNOWN	Unknown state.

Definition at line 96 of file DIS/Types.hh.

## 6.1.2 Function Documentation

### 6.1.2.1 execution\_mode\_enum\_to\_int16()

```
int16_t DIS::execution_mode_enum_to_int16 (
    ExecutionModeEnum mode )
```

Convert an ExecutionModeEnum value into a 16 bit integer.

#### Returns

DIS execution mode as a 16 bit integer representation.

#### Parameters

<i>mode</i>	Execution configuration run mode enumeration value.
-------------	---

Definition at line 66 of file DIS/Types.cpp.

Referenced by DIS::ExecutionConfiguration::set\_current\_execution\_mode(), and DIS::ExecutionConfiguration::set\_next\_execution\_mode().

### 6.1.2.2 execution\_mode\_enum\_to\_string()

```
string DIS::execution_mode_enum_to_string (
    ExecutionModeEnum mode )
```

Convert an ExecutionModeEnum value into a printable string.

#### Returns

DIS execution mode as a printable string.

#### Parameters

<i>mode</i>	Execution configuration run mode enumeration value.
-------------	---

Definition at line 33 of file DIS/Types.cpp.

References EXECUTION\_MODE\_FREEZE, EXECUTION\_MODE\_INITIALIZING, EXECUTION\_MODE\_RUNNING, EXECUTION\_MODE\_SHUTDOWN, and EXECUTION\_MODE\_UNINITIALIZED.

Referenced by DIS::ExecutionConfiguration::pack(), DIS::ExecutionConfiguration::print\_execution\_configuration(), DIS::ExecutionControl::process\_execution\_control\_updates(), and DIS::ExecutionConfiguration::unpack().

### 6.1.2.3 execution\_mode\_int16\_to\_enum()

```
ExecutionModeEnum DIS::execution_mode_int16_to_enum (
    int16_t int_mode )
```

Convert a 16 bit integer to an ExecutionModeEnum value.

**Returns**

[DIS](#) execution mode as enumeration value.

**Parameters**

<i>int_mode</i>	Execution configuration run mode as integer.
-----------------	--

Definition at line 73 of file DIS/Types.cpp.

References [EXECUTION\\_MODE\\_FREEZE](#), [EXECUTION\\_MODE\\_INITIALIZING](#), [EXECUTION\\_MODE\\_RUNNING](#), [EXECUTION\\_MODE\\_SHUTDOWN](#), and [EXECUTION\\_MODE\\_UNINITIALIZED](#).

Referenced by [DIS::ExecutionConfiguration::pack\(\)](#), [DIS::ExecutionConfiguration::print\\_execution\\_configuration\(\)](#), [DI->S::ExecutionControl::process\\_execution\\_control\\_updates\(\)](#), and [DIS::ExecutionConfiguration::unpack\(\)](#).

#### 6.1.2.4 [from\\_execution\\_contorl\\_enum\(\)](#)

```
ExecutionModeEnum DIS::from_execution_contorl_enum (
    TrickHLA::ExecutionControlEnum mode )
```

Convert a [TrickHLA::ExecutionModeEnum](#) value to an [DIS::ExecutionModeEnum](#) value.

**Returns**

[TrickHLA::ExecutionModeEnum](#) as an equivalent [DIS::ExecutionModeEnum](#) value.

**Parameters**

<i>mode</i>	TrickHLA::ExecutionModeEnum value to convert.
-------------	---

Definition at line 139 of file DIS/Types.cpp.

References [TrickHLA::EXECUTION\\_CONTROL\\_FREEZE](#), [TrickHLA::EXECUTION\\_CONTROL\\_INITIALIZING](#), [Trick->HLA::EXECUTION\\_CONTROL\\_RECONFIG](#), [TrickHLA::EXECUTION\\_CONTROL\\_RESTART](#), [TrickHLA::EXECUTIO->N\\_CONTROL\\_RUNNING](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::EXECUTION\\_CONTROL\\_-UNINITIALIZED](#), [EXECUTION\\_MODE\\_FREEZE](#), [EXECUTION\\_MODE\\_INITIALIZING](#), [EXECUTION\\_MODE\\_RUNN->ING](#), [EXECUTION\\_MODE\\_SHUTDOWN](#), and [EXECUTION\\_MODE\\_UNINITIALIZED](#).

#### 6.1.2.5 [mtr\\_enum\\_to\\_int16\(\)](#)

```
int16_t DIS::mtr_enum_to_int16 (
    MTREnum mtr_enum )
```

Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.

**Returns**

[DIS](#) Mode Transition Request (MTR) as a 16 bit integer.

**Parameters**

<i>mtr_enum</i>	MTR enumeration values.
-----------------	-------------------------

Definition at line 214 of file DIS/Types.cpp.

### 6.1.2.6 mtr\_enum\_to\_string()

```
string DIS::mtr_enum_to_string (
    MTREnum mtr_enum )
```

Convert a Mode Transition Request (MTR) enum value into a printable string.

#### Returns

[DIS](#) mode transition request (MTR) as a printable string.

#### Parameters

<i>mtr_enum</i>	MTR enumeration value to convert.
-----------------	-----------------------------------

Definition at line 180 of file DIS/Types.cpp.

References MTR\_GOTO\_FREEZE, MTR\_GOTO\_RUN, MTR\_GOTO\_SHUTDOWN, MTR\_INITIALIZING, and MTR\_UNINITIALIZED.

Referenced by DIS::ExecutionControl::check\_mode\_transition\_request().

### 6.1.2.7 mtr\_int16\_to\_enum()

```
MTREnum DIS::mtr_int16_to_enum (
    int16_t mtr_int )
```

Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.

#### Returns

[DIS](#) mode transition request (MTR) as enumeration value.

#### Parameters

<i>mtr_int</i>	MTR mode.
----------------	-----------

Definition at line 221 of file DIS/Types.cpp.

References MTR\_GOTO\_FREEZE, MTR\_GOTO\_RUN, MTR\_GOTO\_SHUTDOWN, MTR\_INITIALIZING, and MTR\_UNINITIALIZED.

### 6.1.2.8 pause\_pnt\_state\_enum\_to\_int16()

```
int16_t DIS::pause_pnt_state_enum_to_int16 (
    PausePointStateEnum state )
```

Convert a Pause Synchronization Point State enum value into a 16 bit integer.

#### Returns

[TrickHLA](#) sync point state as a 16 bit integer.

#### Parameters

<i>state</i>	Sync point state enumeration value to convert.
--------------	--

Definition at line 301 of file DIS/Types.cpp.

References state.

#### 6.1.2.9 pause\_pnt\_state\_enum\_to\_string()

```
string DIS::pause_pnt_state_enum_to_string (  
    PausePointStateEnum state )
```

Convert a Pause Synchronization Point State enum value into a printable string.

##### Returns

[TrickHLA](#) sync point state as a printable string.

##### Parameters

<i>state</i>	Sync point state enumeration value to convert.
--------------	--

Definition at line 255 of file DIS/Types.cpp.

References PAUSE\_POINT\_STATE\_ACKNOWLEDGED, PAUSE\_POINT\_STATE\_ERROR, PAUSE\_POINT\_STATE\_EXIT, PAUSE\_POINT\_STATE\_FREEZE, PAUSE\_POINT\_STATE\_PENDING, PAUSE\_POINT\_STATE\_RECONFIG, PAUSE\_POINT\_STATE\_RESTART, PAUSE\_POINT\_STATE\_RUN, and state.

#### 6.1.2.10 pause\_pnt\_state\_int16\_to\_enum()

```
PausePointStateEnum DIS::pause_pnt_state_int16_to_enum (  
    int16_t int_state )
```

Convert an integer value to a Pause Synchronization Point State enumeration value.

##### Returns

[TrickHLA](#) Synchronization Point State enum value.

##### Parameters

<i>int_state</i>	Sync point state value as a 16 bit integer.
------------------	---

Definition at line 307 of file DIS/Types.cpp.

References PAUSE\_POINT\_STATE\_ACKNOWLEDGED, PAUSE\_POINT\_STATE\_ERROR, PAUSE\_POINT\_STATE\_EXIT, PAUSE\_POINT\_STATE\_FREEZE, PAUSE\_POINT\_STATE\_PENDING, PAUSE\_POINT\_STATE\_RECONFIG, PAUSE\_POINT\_STATE\_RESTART, PAUSE\_POINT\_STATE\_RUN, and PAUSE\_POINT\_STATE\_UNKNOWN.

#### 6.1.2.11 to\_execution\_control\_enum()

```
TrickHLA::ExecutionControlEnum DIS::to_execution_control_enum (  
    ExecutionModeEnum mode )
```

Convert an [DIS::ExecutionModeEnum](#) value to a TrickHLA::ExecutionModeEnum value.

##### Returns

[DIS::ExecutionModeEnum](#) as an equivalent TrickHLA::ExecutionModeEnum value.

## Parameters

<code>mode</code>	<a href="#">DIS::ExecutionModeEnum</a> value to convert.
-------------------	--

Definition at line 106 of file [DIS/Types.cpp](#).

References [TrickHLA::EXECUTION\\_CONTROL\\_FREEZE](#), [TrickHLA::EXECUTION\\_CONTROL\\_INITIALIZING](#), [TrickHLA::EXECUTION\\_CONTROL\\_RUNNING](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::EXECUTION\\_CONTROL\\_UNINITIALIZED](#), [EXECUTION\\_MODE\\_FREEZE](#), [EXECUTION\\_MODE\\_INITIALIZING](#), [EXECUTION\\_MODE\\_RUNNING](#), [EXECUTION\\_MODE\\_SHUTDOWN](#), and [EXECUTION\\_MODE\\_UNINITIALIZED](#).

## 6.1.3 Variable Documentation

### 6.1.3.1 INITIALIZE\_SYNC\_POINT

```
const std::wstring DIS::INITIALIZE_SYNC_POINT = L"initialize" [static]
```

Definition at line 58 of file [DIS/ExecutionControl.cpp](#).

Referenced by [DIS::ExecutionControl::add\\_multiphase\\_init\\_sync\\_points\(\)](#), [DIS::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), [DIS::ExecutionControl::sync\\_point\\_registration\\_failed\(\)](#), and [DIS::ExecutionControl::sync\\_point\\_registration\\_succeeded\(\)](#).

### 6.1.3.2 STARTUP\_FREEZE\_SYNC\_POINT

```
const std::wstring DIS::STARTUP_FREEZE_SYNC_POINT = L"pause_0.0" [static]
```

Definition at line 60 of file [DIS/ExecutionControl.cpp](#).

Referenced by [DIS::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#).

### 6.1.3.3 STARTUP\_SYNC\_POINT

```
const std::wstring DIS::STARTUP_SYNC_POINT = L"startup" [static]
```

Definition at line 59 of file [DIS/ExecutionControl.cpp](#).

Referenced by [DIS::ExecutionControl::add\\_multiphase\\_init\\_sync\\_points\(\)](#), [DIS::ExecutionControl::post\\_multi\\_phase\\_init\\_process\(\)](#), [DIS::ExecutionControl::sync\\_point\\_registration\\_failed\(\)](#), and [DIS::ExecutionControl::sync\\_point\\_registration\\_succeeded\(\)](#).

## 6.2 DSES Namespace Reference

### Data Structures

- class [ExecutionConfiguration](#)
- class [ExecutionControl](#)

### Enumerations

- enum [ExecutionModeEnum](#) {
 [EXECUTION\\_MODE\\_FIRST\\_VALUE](#) = 0, [EXECUTION\\_MODE\\_UNINITIALIZED](#) = 0, [EXECUTION\\_MODE\\_INITIALIZING](#) = 1, [EXECUTION\\_MODE\\_RUNNING](#) = 2, [EXECUTION\\_MODE\\_FREEZE](#) = 3, [EXECUTION\\_MODE\\_SHUTDOWN](#) = 4, [EXECUTION\\_MODE\\_LAST\\_VALUE](#) = 4
 }

- enum `MTREnum` {
   
`MTR_FIRST_VALUE` = 0, `MTR_UNINITIALIZED` = 0, `MTR_INITIALIZING` = 1, `MTR_GOTO_RUN` = 2,
   
`MTR_GOTO_FREEZE` = 3, `MTR_GOTO_SHUTDOWN` = 4, `MTR_LAST_VALUE` = 4 }

## Functions

- `std::string execution_mode_enum_to_string (ExecutionModeEnum mode)`
  
*Convert an ExecutionModeEnum value into a printable string.*
- `int16_t execution_mode_enum_to_int16 (ExecutionModeEnum mode)`
  
*Convert an ExecutionModeEnum value into a 16 bit integer.*
- `ExecutionModeEnum execution_mode_int16_to_enum (int16_t int_mode)`
  
*Convert a 16 bit integer to an ExecutionModeEnum value.*
- `TrickHLA::ExecutionControlEnum to_execution_control_enum (ExecutionModeEnum mode)`
  
*Convert an DSES::ExecutionModeEnum value to a TrickHLA::ExecutionModeEnum value.*
- `ExecutionModeEnum from_execution_contorl_enum (TrickHLA::ExecutionControlEnum mode)`
  
*Convert a TrickHLA::ExecutionModeEnum value to an DSES::ExecutionModeEnum value.*
- `std::string mtr_enum_to_string (MTREnum mtr_enum)`
  
*Convert a Mode Transition Request (MTR) enum value into a printable string.*
- `int16_t mtr_enum_to_int16 (MTREnum mtr_enum)`
  
*Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.*
- `MTREnum mtr_int16_to_enum (int16_t mtr_int)`
  
*Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.*

## Variables

- `static const std::wstring SIM_CONFIG_SYNC_POINT = L"sim_config"`
- `static const std::wstring INITIALIZE_SYNC_POINT = L"initialize"`
- `static const std::wstring STARTUP_SYNC_POINT = L"startup"`

### 6.2.1 Enumeration Type Documentation

#### 6.2.1.1 ExecutionModeEnum

`enum DSES::ExecutionModeEnum`

##### Enumerator

<code>EXECUTION_MODE_FIRST_VALUE</code>	Same as uninitialized.
<code>EXECUTION_MODE_UNINITIALIZED</code>	Execution mode UNINITIALIZED.
<code>EXECUTION_MODE_INITIALIZING</code>	Execution mode INITIALIZING.
<code>EXECUTION_MODE_RUNNING</code>	Execution mode RUNNING.
<code>EXECUTION_MODE_FREEZE</code>	Execution mode FREEZE.
<code>EXECUTION_MODE_SHUTDOWN</code>	Execution mode SHUTDOWN.
<code>EXECUTION_MODE_LAST_VALUE</code>	Same as shutdown.

Definition at line 54 of file DSES/Types.hh.

### 6.2.1.2 MTREnum

enum `DSES::MTREnum`

#### Enumerator

<code>MTR_FIRST_VALUE</code>	Not a valid mode transition.
<code>MTR_UNINITIALIZED</code>	Not a valid mode transition.
<code>MTR_INITIALIZING</code>	Not a valid mode transition.
<code>MTR_GOTO_RUN</code>	Mode transition to RUN mode.
<code>MTR_GOTO_FREEZE</code>	Mode transition to FREEZE mode.
<code>MTR_GOTO_SHUTDOWN</code>	Mode transition to SHUTDOWN mode.
<code>MTR_LAST_VALUE</code>	Same as shutdown.

Definition at line 77 of file `DSES/Types.hh`.

## 6.2.2 Function Documentation

### 6.2.2.1 execution\_mode\_enum\_to\_int16()

```
int16_t DSES::execution_mode_enum_to_int16 (
    ExecutionModeEnum mode )
```

Convert an `ExecutionModeEnum` value into a 16 bit integer.

#### Returns

`DSES` execution mode as a 16 bit integer representation.

#### Parameters

<code>mode</code>	Execution configuration run mode enumeration value.
-------------------	---

Definition at line 66 of file `DSES/Types.cpp`.

Referenced by `DSES::ExecutionConfiguration::set_current_execution_mode()`, and `DSES::ExecutionConfiguration::set_next_execution_mode()`.

### 6.2.2.2 execution\_mode\_enum\_to\_string()

```
string DSES::execution_mode_enum_to_string (
    ExecutionModeEnum mode )
```

Convert an `ExecutionModeEnum` value into a printable string.

#### Returns

`DSES` execution mode as a printable string.

#### Parameters

<code>mode</code>	Execution configuration run mode enumeration value.
-------------------	---

Definition at line 33 of file DSES/Types.cpp.

References DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

Referenced by DSES::ExecutionConfiguration::pack(), DSES::ExecutionConfiguration::print\_execution\_configuration(), DSES::ExecutionControl::process\_execution\_control\_updates(), and DSES::ExecutionConfiguration::unpack().

### 6.2.2.3 execution\_mode\_int16\_to\_enum()

```
ExecutionModeEnum DSES::execution_mode_int16_to_enum (
    int16_t int_mode )
```

Convert a 16 bit integer to an ExecutionModeEnum value.

Returns

DSES execution mode as enumeration value.

Parameters

<i>int_mode</i>	Execution configuration run mode as integer.
-----------------	--

Definition at line 73 of file DSES/Types.cpp.

References DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

Referenced by DSES::ExecutionConfiguration::pack(), DSES::ExecutionConfiguration::print\_execution\_configuration(), DSES::ExecutionControl::process\_execution\_control\_updates(), and DSES::ExecutionConfiguration::unpack().

### 6.2.2.4 from\_execution\_contorl\_enum()

```
ExecutionModeEnum DSES::from_execution_contorl_enum (
    TrickHLA::ExecutionControlEnum mode )
```

Convert a TrickHLA::ExecutionModeEnum value to an DSES::ExecutionModeEnum value.

Returns

TrickHLA::ExecutionModeEnum as an equivalent DSES::ExecutionModeEnum value.

Parameters

<i>mode</i>	TrickHLA::ExecutionModeEnum value to convert.
-------------	---

Definition at line 139 of file DSES/Types.cpp.

References TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RECONFIG, TrickHLA::EXECUTION\_CONTROL\_RESTART, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

### 6.2.2.5 mtr\_enum\_to\_int16()

```
int16_t DSES::mtr_enum_to_int16 (
    MTREnum mtr_enum )
```

Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.

**Returns**

[DSES](#) Mode Transition Request (MTR) as a 16 bit integer.

**Parameters**

<i>mtr_enum</i>	MTR enumeration values.
-----------------	-------------------------

Definition at line 218 of file DSES/Types.cpp.

**6.2.2.6 mtr\_enum\_to\_string()**

```
string DSES::mtr_enum_to_string (
    MTREnum mtr_enum )
```

Convert a Mode Transition Request (MTR) enum value into a printable string.

**Returns**

[DSES](#) mode transition request (MTR) as a printable string.

**Parameters**

<i>mtr_enum</i>	MTR enumeration value to convert.
-----------------	-----------------------------------

Definition at line 184 of file DSES/Types.cpp.

References DIS::MTR\_GOTO\_FREEZE, DIS::MTR\_GOTO\_RUN, DIS::MTR\_GOTO\_SHUTDOWN, DIS::MTR\_INITIALIZING, and DIS::MTR\_UNINITIALIZED.

Referenced by DSES::ExecutionControl::check\_mode\_transition\_request().

**6.2.2.7 mtr\_int16\_to\_enum()**

```
MTREnum DSES::mtr_int16_to_enum (
    int16_t mtr_int )
```

Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.

**Returns**

[DSES](#) mode transition request (MTR) as enumeration value.

**Parameters**

<i>mtr_int</i>	MTR mode.
----------------	-----------

Definition at line 225 of file DSES/Types.cpp.

References DIS::MTR\_GOTO\_FREEZE, DIS::MTR\_GOTO\_RUN, DIS::MTR\_GOTO\_SHUTDOWN, DIS::MTR\_INITIALIZING, and DIS::MTR\_UNINITIALIZED.

**6.2.2.8 to\_execution\_control\_enum()**

```
TrickHLA::ExecutionControlEnum DSES::to_execution_control_enum (
    ExecutionModeEnum mode )
```

Convert an [DSES::ExecutionModeEnum](#) value to a [TrickHLA::ExecutionModeEnum](#) value.

**Returns**

[DSES::ExecutionModeEnum](#) as an equivalent [TrickHLA::ExecutionModeEnum](#) value.

**Parameters**

<i>mode</i>	<a href="#">DSES::ExecutionModeEnum</a> value to convert.
-------------	---

Definition at line 106 of file [DSES/Types.cpp](#).

References [TrickHLA::EXECUTION\\_CONTROL\\_FREEZE](#), [TrickHLA::EXECUTION\\_CONTROL\\_INITIALIZING](#), [TrickHLA::EXECUTION\\_CONTROL\\_RUNNING](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::EXECUTION\\_CONTROL\\_UNINITIALIZED](#), [DIS::EXECUTION\\_MODE\\_FREEZE](#), [DIS::EXECUTION\\_MODE\\_INITIALIZING](#), [DIS::EXECUTION\\_MODE\\_RUNNING](#), [DIS::EXECUTION\\_MODE\\_SHUTDOWN](#), and [DIS::EXECUTION\\_MODE\\_UNINITIALIZED](#).

## 6.2.3 Variable Documentation

### 6.2.3.1 INITIALIZE\_SYNC\_POINT

```
const std::wstring DSES::INITIALIZE_SYNC_POINT = L"initialize" [static]
```

Definition at line 59 of file [DSES/ExecutionControl.cpp](#).

Referenced by [DSES::ExecutionControl::achieve\\_all\\_multiphase\\_init\\_sync\\_pnts\(\)](#), [DSES::ExecutionControl::add\\_multiphase\\_init\\_sync\\_points\(\)](#), [DSES::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), and [DSES::ExecutionControl::wait\\_for\\_all\\_multiphase\\_init\\_sync\\_pnts\(\)](#).

### 6.2.3.2 SIM\_CONFIG\_SYNC\_POINT

```
const std::wstring DSES::SIM_CONFIG_SYNC_POINT = L"sim_config" [static]
```

Definition at line 58 of file [DSES/ExecutionControl.cpp](#).

Referenced by [DSES::ExecutionControl::achieve\\_all\\_multiphase\\_init\\_sync\\_pnts\(\)](#), [DSES::ExecutionControl::add\\_multiphase\\_init\\_sync\\_points\(\)](#), [DSES::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), and [DSES::ExecutionControl::wait\\_for\\_all\\_multiphase\\_init\\_sync\\_pnts\(\)](#).

### 6.2.3.3 STARTUP\_SYNC\_POINT

```
const std::wstring DSES::STARTUP_SYNC_POINT = L"startup" [static]
```

Definition at line 60 of file [DSES/ExecutionControl.cpp](#).

Referenced by [DSES::ExecutionControl::achieve\\_all\\_multiphase\\_init\\_sync\\_pnts\(\)](#), [DSES::ExecutionControl::add\\_multiphase\\_init\\_sync\\_points\(\)](#), [DSES::ExecutionControl::post\\_multi\\_phase\\_init\\_process\(\)](#), and [DSES::ExecutionControl::wait\\_for\\_all\\_multiphase\\_init\\_sync\\_pnts\(\)](#).

## 6.3 IMSim Namespace Reference

### Data Structures

- class [ExecutionConfiguration](#)
- class [ExecutionControl](#)
- class [FreezeInteractionHandler](#)
- class [PausePointList](#)

## Typedefs

- `typedef std::set< double > FreezeTimeSet`

## Enumerations

- `enum ExecutionModeEnum { EXECUTION_MODE_FIRST_VALUE = 0, EXECUTION_MODE_UNINITIALIZED = 0, EXECUTION_MODE_INITIALIZING = 1, EXECUTION_MODE_RUNNING = 2, EXECUTION_MODE_FREEZE = 3, EXECUTION_MODE_SHUTDOWN = 4, EXECUTION_MODE_LAST_VALUE = 4 }`
- `enum MTREnum { MTR_FIRST_VALUE = 0, MTR_UNINITIALIZED = 0, MTR_INITIALIZING = 1, MTR_GOTO_RUN = 2, MTR_GOTO_FREEZE = 3, MTR_GOTO_SHUTDOWN = 4, MTR_LAST_VALUE = 4 }`
- `enum PausePointStateEnum { PAUSE_POINT_STATE_FIRST_VALUE = 0, PAUSE_POINT_STATE_ERROR = 0, PAUSE_POINT_STATE_PENDING = 1, PAUSE_POINT_STATE_ACKNOWLEDGED = 2, PAUSE_POINT_STATE_RUN = 3, PAUSE_POINT_STATE_FREEZE = 4, PAUSE_POINT_STATE_EXIT = 5, PAUSE_POINT_STATE_RESTART = 6, PAUSE_POINT_STATE_RECONFIG = 7, PAUSE_POINT_STATE_UNKNOWN = INT_MAX }`

## Functions

- `std::string execution_mode_enum_to_string (ExecutionModeEnum mode)`

*Convert an ExecutionModeEnum value into a printable string.*
- `int16_t execution_mode_enum_to_int16 (ExecutionModeEnum mode)`

*Convert an ExecutionModeEnum value into a 16 bit integer.*
- `ExecutionModeEnum execution_mode_int16_to_enum (int16_t int_mode)`

*Convert a 16 bit integer to an ExecutionModeEnum value.*
- `TrickHLA::ExecutionControlEnum to_execution_control_enum (ExecutionModeEnum mode)`

*Convert an IMSim::ExecutionModeEnum value to a TrickHLA::ExecutionModeEnum value.*
- `ExecutionModeEnum from_execution_control_enum (TrickHLA::ExecutionControlEnum mode)`

*Convert a TrickHLA::ExecutionModeEnum value to an IMSim::ExecutionModeEnum value.*
- `std::string mtr_enum_to_string (MTREnum mtr_enum)`

*Convert a Mode Transition Request (MTR) enum value into a printable string.*
- `int16_t mtr_enum_to_int16 (MTREnum mtr_enum)`

*Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.*
- `MTREnum mtr_int16_to_enum (int16_t mtr_int)`

*Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.*
- `std::string pause_pnt_state_enum_to_string (PausePointStateEnum state)`

*Convert a Pause Synchronization Point State enum value into a printable string.*
- `int16_t pause_pnt_state_enum_to_int16 (PausePointStateEnum state)`

*Convert a Pause Synchronization Point State enum value into a 16 bit integer.*
- `PausePointStateEnum pause_pnt_state_int16_to_enum (int16_t int_state)`

*Convert an integer value to a Pause Synchronization Point State enumeration value.*

## Variables

- static const std::wstring **SIM\_CONFIG\_SYNC\_POINT** = L"sim\_config\_v2"
- static const std::wstring **INITIALIZE\_SYNC\_POINT** = L"initialize\_v2"
- static const std::wstring **INIT\_COMPLETE\_SYNC\_POINT** = L"initialization\_complete\_v2"
- static const std::wstring **STARTUP\_SYNC\_POINT** = L"startup\_v2"
- static const std::wstring **FEDSAVE\_SYNC\_POINT** = L"FEDSAVE\_v2"
- static const std::wstring **FEDRUN\_SYNC\_POINT** = L"FEDRUN\_v2"
- static const std::wstring **STARTUP\_FREEZE\_SYNC\_POINT** = L"pause\_0.0"

### 6.3.1 Typedef Documentation

#### 6.3.1.1 FreezeTimeSet

```
typedef std::set< double > IMSim::FreezeTimeSet
```

Definition at line 111 of file IMSim/Types.hh.

### 6.3.2 Enumeration Type Documentation

#### 6.3.2.1 ExecutionModeEnum

```
enum IMSim::ExecutionModeEnum
```

##### Enumerator

<b>EXECUTION_MODE_FIRST_VALUE</b>	Same as uninitialized.
<b>EXECUTION_MODE_UNINITIALIZED</b>	Execution mode UNINITIALIZED.
<b>EXECUTION_MODE_INITIALIZING</b>	Execution mode INITIALIZING.
<b>EXECUTION_MODE_RUNNING</b>	Execution mode RUNNING.
<b>EXECUTION_MODE_FREEZE</b>	Execution mode FREEZE.
<b>EXECUTION_MODE_SHUTDOWN</b>	Execution mode SHUTDOWN.
<b>EXECUTION_MODE_LAST_VALUE</b>	Same as shutdown.

Definition at line 54 of file IMSim/Types.hh.

#### 6.3.2.2 MTREnum

```
enum IMSim::MTREnum
```

##### Enumerator

<b>MTR_FIRST_VALUE</b>	Not a valid mode transition.
<b>MTR_UNINITIALIZED</b>	Not a valid mode transition.
<b>MTR_INITIALIZING</b>	Not a valid mode transition.
<b>MTR_GOTO_RUN</b>	Mode transition to RUN mode.
<b>MTR_GOTO_FREEZE</b>	Mode transition to FREEZE mode.
<b>MTR_GOTO_SHUTDOWN</b>	Mode transition to SHUTDOWN mode.
<b>MTR_LAST_VALUE</b>	Same as shutdown.

Definition at line 77 of file IMSim/Types.hh.

### 6.3.2.3 PausePointStateEnum

```
enum IMSim::PausePointStateEnum
```

Enumerator

PAUSE_POINT_STATE_FIRST_VALUE	Set to the First value in the enumeration.
PAUSE_POINT_STATE_ERROR	Pause point state error.
PAUSE_POINT_STATE_PENDING	Pause point state pending.
PAUSE_POINT_STATE_ACKNOWLEDGED	Pause point state acknowledged.
PAUSE_POINT_STATE_RUN	Pause point state run.
PAUSE_POINT_STATE_FREEZE	Pause point state freeze.
PAUSE_POINT_STATE_EXIT	Pause point state exit.
PAUSE_POINT_STATE_RESTART	Pause point state restart.
PAUSE_POINT_STATE_RECONFIG	Pause point state reconfiguration.
PAUSE_POINT_STATE_UNKNOWN	Unknown state.

Definition at line 96 of file IMSim/Types.hh.

### 6.3.3 Function Documentation

#### 6.3.3.1 execution\_mode\_enum\_to\_int16()

```
int16_t IMSim::execution_mode_enum_to_int16 (
    ExecutionModeEnum mode )
```

Convert an ExecutionModeEnum value into a 16 bit integer.

Returns

IMSim execution mode as a 16 bit integer representation.

Parameters

mode	Execution configuration run mode enumeration value.
------	---

Definition at line 66 of file IMSim/Types.cpp.

Referenced by IMSim::ExecutionConfiguration::set\_current\_execution\_mode(), and IMSim::ExecutionConfiguration::set\_next\_execution\_mode().

#### 6.3.3.2 execution\_mode\_enum\_to\_string()

```
string IMSim::execution_mode_enum_to_string (
    ExecutionModeEnum mode )
```

Convert an ExecutionModeEnum value into a printable string.

**Returns**

[IMSim](#) execution mode as a printable string.

**Parameters**

<i>mode</i>	Execution configuration run mode enumeration value.
-------------	---

Definition at line 33 of file IMSim/Types.cpp.

References DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

Referenced by IMSim::ExecutionConfiguration::pack(), IMSim::ExecutionConfiguration::print\_execution\_configuration(), IMSim::ExecutionControl::process\_execution\_control\_updates(), and IMSim::ExecutionConfiguration::unpack().

### 6.3.3.3 execution\_mode\_int16\_to\_enum()

```
ExecutionModeEnum IMSim::execution_mode_int16_to_enum (
    int16_t int_mode )
```

Convert a 16 bit integer to an ExecutionModeEnum value.

**Returns**

[IMSim](#) execution mode as enumeration value.

**Parameters**

<i>int_mode</i>	Execution configuration run mode as integer.
-----------------	--

Definition at line 73 of file IMSim/Types.cpp.

References DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

Referenced by IMSim::ExecutionConfiguration::pack(), IMSim::ExecutionConfiguration::print\_execution\_configuration(), IMSim::ExecutionControl::process\_execution\_control\_updates(), and IMSim::ExecutionConfiguration::unpack().

### 6.3.3.4 from\_execution\_contorl\_enum()

```
ExecutionModeEnum IMSim::from_execution_contorl_enum (
    TrickHLA::ExecutionControlEnum mode )
```

Convert a TrickHLA::ExecutionModeEnum value to an [IMSim::ExecutionModeEnum](#) value.

**Returns**

TrickHLA::ExecutionModeEnum as an equivalent [IMSim::ExecutionModeEnum](#) value.

**Parameters**

<i>mode</i>	TrickHLA::ExecutionModeEnum value to convert.
-------------	---

Definition at line 139 of file IMSim/Types.cpp.

References TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RECONFIG, TrickHLA::EXECUTION\_CONTROL\_RESTART, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED.

UNINITIALIZED, DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

### 6.3.3.5 mtr\_enum\_to\_int16()

```
int16_t IMSim::mtr_enum_to_int16 (
    MTREnum mtr_enum )
```

Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.

#### Returns

[IMSim](#) Mode Transition Request (MTR) as a 16 bit integer.

#### Parameters

<i>mtr_enum</i>	MTR enumeration values.
-----------------	-------------------------

Definition at line 214 of file IMSim/Types.cpp.

### 6.3.3.6 mtr\_enum\_to\_string()

```
string IMSim::mtr_enum_to_string (
    MTREnum mtr_enum )
```

Convert a Mode Transition Request (MTR) enum value into a printable string.

#### Returns

[IMSim](#) mode transition request (MTR) as a printable string.

#### Parameters

<i>mtr_enum</i>	MTR enumeration value to convert.
-----------------	-----------------------------------

Definition at line 180 of file IMSim/Types.cpp.

References DIS::MTR\_GOTO\_FREEZE, DIS::MTR\_GOTO\_RUN, DIS::MTR\_GOTO\_SHUTDOWN, DIS::MTR\_INITIALIZING, and DIS::MTR\_UNINITIALIZED.

Referenced by [IMSim::ExecutionControl::check\\_mode\\_transition\\_request\(\)](#).

### 6.3.3.7 mtr\_int16\_to\_enum()

```
MTREnum IMSim::mtr_int16_to_enum (
    int16_t mtr_int )
```

Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.

#### Returns

[IMSim](#) mode transition request (MTR) as enumeration value.

#### Parameters

<i>mtr_int</i>	MTR mode.
----------------	-----------

Definition at line 221 of file IMSim/Types.cpp.

References DIS::MTR\_GOTO\_FREEZE, DIS::MTR\_GOTO\_RUN, DIS::MTR\_GOTO\_SHUTDOWN, DIS::MTR\_INITIALIZING, and DIS::MTR\_UNINITIALIZED.

### 6.3.3.8 pause\_pnt\_state\_enum\_to\_int16()

```
int16_t IMSim::pause_pnt_state_enum_to_int16 (
    PausePointStateEnum state )
```

Convert a Pause Synchronization Point State enum value into a 16 bit integer.

#### Returns

[TrickHLA](#) sync point state as a 16 bit integer.

#### Parameters

<i>state</i>	Sync point state enumeration value to convert.
--------------	--

Definition at line 301 of file IMSim/Types.cpp.

References state.

### 6.3.3.9 pause\_pnt\_state\_enum\_to\_string()

```
string IMSim::pause_pnt_state_enum_to_string (
    PausePointStateEnum state )
```

Convert a Pause Synchronization Point State enum value into a printable string.

#### Returns

[TrickHLA](#) sync point state as a printable string.

#### Parameters

<i>state</i>	Sync point state enumeration value to convert.
--------------	--

Definition at line 255 of file IMSim/Types.cpp.

References DIS::PAUSE\_POINT\_STATE\_ACKNOWLEDGED, DIS::PAUSE\_POINT\_STATE\_ERROR, DIS::PAUSE\_POINT\_STATE\_EXIT, DIS::PAUSE\_POINT\_STATE\_PENDING, DIS::PAUSE\_POINT\_STATE\_RECONFIG, DIS::PAUSE\_POINT\_STATE\_RESTART, DIS::PAUSE\_POINT\_STATE\_RUN, and state.

### 6.3.3.10 pause\_pnt\_state\_int16\_to\_enum()

```
PausePointStateEnum IMSim::pause_pnt_state_int16_to_enum (
    int16_t int_state )
```

Convert an integer value to a Pause Synchronization Point State enumeration value.

#### Returns

[TrickHLA](#) Synchronization Point State enum value.

**Parameters**

<i>int_state</i>	Sync point state value as a 16 bit integer.
------------------	---

Definition at line 307 of file IMSim/Types.cpp.

References DIS::PAUSE\_POINT\_STATE\_ACKNOWLEDGED, DIS::PAUSE\_POINT\_STATE\_ERROR, DIS::PAUSE\_POINT\_STATE\_EXIT, DIS::PAUSE\_POINT\_STATE\_FREEZE, DIS::PAUSE\_POINT\_STATE\_PENDING, DIS::PAUSE\_POINT\_STATE\_RECONFIG, DIS::PAUSE\_POINT\_STATE\_RESTART, DIS::PAUSE\_POINT\_STATE\_RUN, and DIS::PAUSE\_POINT\_STATE\_UNKNOWN.

### 6.3.3.11 to\_execution\_control\_enum()

```
TrickHLA::ExecutionControlEnum IMSim::to_execution_control_enum (
    ExecutionModeEnum mode )
```

Convert an [IMSim::ExecutionModeEnum](#) value to a TrickHLA::ExecutionModeEnum value.

**Returns**

[IMSim::ExecutionModeEnum](#) as an equivalent TrickHLA::ExecutionModeEnum value.

**Parameters**

<i>mode</i>	<a href="#">IMSim::ExecutionModeEnum</a> value to convert.
-------------	--

Definition at line 106 of file IMSim/Types.cpp.

References TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

## 6.3.4 Variable Documentation

### 6.3.4.1 FEDRUN\_SYNC\_POINT

```
const std::wstring IMSim::FEDRUN_SYNC_POINT = L"FEDRUN_v2" [static]
```

Definition at line 66 of file IMSim/ExecutionControl.cpp.

Referenced by [IMSim::ExecutionControl::announce\\_sync\\_point\(\)](#), [IMSim::ExecutionControl::exit\\_freeze\(\)](#), and [IMSim::ExecutionControl::mark\\_synchronized\(\)](#).

### 6.3.4.2 FEDSAVE\_SYNC\_POINT

```
const std::wstring IMSim::FEDSAVE_SYNC_POINT = L"FEDSAVE_v2" [static]
```

Definition at line 65 of file IMSim/ExecutionControl.cpp.

Referenced by [IMSim::ExecutionControl::announce\\_sync\\_point\(\)](#), [IMSim::ExecutionControl::check\\_freeze\\_time\(\)](#), [IMSim::ExecutionControl::is\\_save\\_initiated\(\)](#), and [IMSim::ExecutionControl::mark\\_synchronized\(\)](#).

### 6.3.4.3 INIT\_COMPLETE\_SYNC\_POINT

```
const std::wstring IMSim::INIT_COMPLETE_SYNC_POINT = L"initialization_complete_v2" [static]
```

Definition at line 63 of file IMSim/ExecutionControl.cpp.

Referenced by IMSim::ExecutionControl::announce\_sync\_point(), and IMSim::ExecutionControl::post\_multi\_phase\_init\_process().

#### 6.3.4.4 INITIALIZE\_SYNC\_POINT

```
const std::wstring IMSim::INITIALIZE_SYNC_POINT = L"initialize_v2" [static]
```

Definition at line 62 of file IMSim/ExecutionControl.cpp.

Referenced by IMSim::ExecutionControl::achieve\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::add\_multiphase\_init\_sync\_points(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts().

#### 6.3.4.5 SIM\_CONFIG\_SYNC\_POINT

```
const std::wstring IMSim::SIM_CONFIG_SYNC_POINT = L"sim_config_v2" [static]
```

Definition at line 61 of file IMSim/ExecutionControl.cpp.

Referenced by IMSim::ExecutionControl::achieve\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::add\_multiphase\_init\_sync\_points(), IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts().

#### 6.3.4.6 STARTUP\_FREEZE\_SYNC\_POINT

```
const std::wstring IMSim::STARTUP_FREEZE_SYNC_POINT = L"pause_0.0" [static]
```

Definition at line 67 of file IMSim/ExecutionControl.cpp.

Referenced by IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 6.3.4.7 STARTUP\_SYNC\_POINT

```
const std::wstring IMSim::STARTUP_SYNC_POINT = L"startup_v2" [static]
```

Definition at line 64 of file IMSim/ExecutionControl.cpp.

Referenced by IMSim::ExecutionControl::achieve\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::add\_multiphase\_init\_sync\_points(), IMSim::ExecutionControl::post\_multi\_phase\_init\_process(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts().

## 6.4 SpaceFOM Namespace Reference

### Data Structures

- class [DynamicalEntity](#)
- class [ExecutionConfiguration](#)
- class [ExecutionControl](#)
- class [MTRInteractionHandler](#)
- class [PhysicalEntityBase](#)
- class [QuaternionEncoder](#)
- class [RefFrameBase](#)
- class [SpaceTimeCoordinateEncoder](#)

## Enumerations

- enum `ExecutionModeEnum` {
 `EXECUTION_MODE_FIRST_VALUE` = 0, `EXECUTION_MODE_UNINITIALIZED` = 0, `EXECUTION_MODE_INITIALIZING` = 1, `EXECUTION_MODE_RUNNING` = 2, `EXECUTION_MODE_FREEZE` = 3, `EXECUTION_MODE_SHUTDOWN` = 4, `EXECUTION_MODE_LAST_VALUE` = 4 }
- enum `MTREnum` {
 `MTR_FIRST_VALUE` = 0, `MTR_UNINITIALIZED` = 0, `MTR_INITIALIZING` = 1, `MTR_GOTO_RUN` = 2, `MTR_GOTO_FREEZE` = 3, `MTR_GOTO_SHUTDOWN` = 4, `MTR_LAST_VALUE` = 4 }

## Functions

- `std::string execution_mode_enum_to_string (ExecutionModeEnum mode)`  
*Convert an ExecutionModeEnum value into a printable string.*
- `int16_t execution_mode_enum_to_int16 (ExecutionModeEnum mode)`  
*Convert an ExecutionModeEnum value into a 16 bit integer.*
- `ExecutionModeEnum execution_mode_int16_to_enum (int16_t int_mode)`  
*Convert a 16 bit integer to an ExecutionModeEnum value.*
- `TrickHLA::ExecutionControlEnum to_execution_control_enum (ExecutionModeEnum mode)`  
*Convert an SpaceFOM::ExecutionModeEnum value to a TrickHLA::ExecutionModeEnum value.*
- `ExecutionModeEnum from_execution_control_enum (TrickHLA::ExecutionControlEnum mode)`  
*Convert a TrickHLA::ExecutionModeEnum value to an SpaceFOM::ExecutionModeEnum value.*
- `std::string mtr_enum_to_string (MTREnum mtr_enum)`  
*Convert a Mode Transition Request (MTR) enum value into a printable string.*
- `int16_t mtr_enum_to_int16 (MTREnum mtr_enum)`  
*Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.*
- `MTREnum mtr_int16_to_enum (int16_t mtr_int)`  
*Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.*
- `MTREnum from_mode_transition_enum (TrickHLA::ModeTransitionEnum mode)`  
*Convert a TrickHLA::ModeTransitionEnum value to an SpaceFOM::MTREnum value.*

## Variables

- static const `std::wstring INIT_STARTED_SYNC_POINT` = L"initialization\_started"
- static const `std::wstring INIT_COMPLETED_SYNC_POINT` = L"initialization\_completed"
- static const `std::wstring OBJECTS_DISCOVERED_SYNC_POINT` = L"objects\_discovered"
- static const `std::wstring ROOT_FRAME_DISCOVERED_SYNC_POINT` = L"root\_frame\_discovered"

### 6.4.1 Enumeration Type Documentation

#### 6.4.1.1 ExecutionModeEnum

`enum SpaceFOM::ExecutionModeEnum`

##### Enumerator

<code>EXECUTION_MODE_FIRST_VALUE</code>	Same as uninitialized.
<code>EXECUTION_MODE_UNINITIALIZED</code>	Execution mode UNINITIALIZED.
<code>EXECUTION_MODE_INITIALIZING</code>	Execution mode INITIALIZING.

## Enumerator

EXECUTION_MODE_RUNNING	Execution mode RUNNING.
EXECUTION_MODE_FREEZE	Execution mode FREEZE.
EXECUTION_MODE_SHUTDOWN	Execution mode SHUTDOWN.
EXECUTION_MODE_LAST_VALUE	Same as shutdown.

Definition at line 54 of file SpaceFOM/Types.hh.

#### 6.4.1.2 MTREnum

enum [SpaceFOM::MTREnum](#)

## Enumerator

MTR_FIRST_VALUE	Not a valid mode transition.
MTR_UNINITIALIZED	Not a valid mode transition.
MTR_INITIALIZING	Not a valid mode transition.
MTR_GOTO_RUN	Mode transition to RUN mode.
MTR_GOTO_FREEZE	Mode transition to FREEZE mode.
MTR_GOTO_SHUTDOWN	Mode transition to SHUTDOWN mode.
MTR_LAST_VALUE	Same as shutdown.

Definition at line 77 of file SpaceFOM/Types.hh.

## 6.4.2 Function Documentation

### 6.4.2.1 execution\_mode\_enum\_to\_int16()

```
int16_t SpaceFOM::execution_mode_enum_to_int16 (
    ExecutionModeEnum mode )
```

Convert an ExecutionModeEnum value into a 16 bit integer.

## Returns

[SpaceFOM](#) execution mode as a 16 bit integer representation.

## Parameters

<i>mode</i>	Execution configuration run mode enumeration value.
-------------	---

Definition at line 66 of file SpaceFOM/Types.cpp.

Referenced by [SpaceFOM::ExecutionConfiguration::set\\_current\\_execution\\_mode\(\)](#), and [SpaceFOM::ExecutionConfiguration::set\\_next\\_execution\\_mode\(\)](#).

### 6.4.2.2 execution\_mode\_enum\_to\_string()

```
string SpaceFOM::execution_mode_enum_to_string (
    ExecutionModeEnum mode )
```

Convert an ExecutionModeEnum value into a printable string.

**Returns**

[SpaceFOM](#) execution mode as a printable string.

**Parameters**

<i>mode</i>	Execution configuration run mode enumeration value.
-------------	---

Definition at line 33 of file SpaceFOM/Types.cpp.

References DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

Referenced by SpaceFOM::ExecutionConfiguration::pack(), SpaceFOM::ExecutionConfiguration::print\_execution\_configuration(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), and SpaceFOM::ExecutionConfiguration::unpack().

#### 6.4.2.3 execution\_mode\_int16\_to\_enum()

```
ExecutionModeEnum SpaceFOM::execution_mode_int16_to_enum (
    int16_t int_mode )
```

Convert a 16 bit integer to an ExecutionModeEnum value.

**Returns**

[SpaceFOM](#) execution mode as enumeration value.

**Parameters**

<i>int_mode</i>	Execution configuration run mode as integer.
-----------------	--

Definition at line 73 of file SpaceFOM/Types.cpp.

References DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

Referenced by SpaceFOM::ExecutionConfiguration::pack(), SpaceFOM::ExecutionConfiguration::print\_execution\_configuration(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), and SpaceFOM::ExecutionConfiguration::unpack().

#### 6.4.2.4 from\_execution\_control\_enum()

```
ExecutionModeEnum SpaceFOM::from_execution_control_enum (
    TrickHLA::ExecutionControlEnum mode )
```

Convert a TrickHLA::ExecutionModeEnum value to an [SpaceFOM::ExecutionModeEnum](#) value.

**Returns**

TrickHLA::ExecutionModeEnum as an equivalent [SpaceFOM::ExecutionModeEnum](#) value.

**Parameters**

<i>mode</i>	TrickHLA::ExecutionModeEnum value to convert.
-------------	---

Definition at line 139 of file SpaceFOM/Types.cpp.

References TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RECONFIG, TrickHLA::EXECUTION\_CONTROL\_RESTART, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED. Referenced by SpaceFOM::ExecutionControl::process\_execution\_control\_updates().

#### 6.4.2.5 from\_mode\_transition\_enum()

```
MTREnum SpaceFOM::from_mode_transition_enum (
    TrickHLA::ModeTransitionEnum mode )
```

Convert a [TrickHLA::ModeTransitionEnum](#) value to an [SpaceFOM::MTREnum](#) value.

##### Returns

[TrickHLA::ModeTransitionEnum](#) as an equivalent [SpaceFOM::MTREnum](#) value.

##### Parameters

<i>mode</i>	<a href="#">TrickHLA::ModeTransitionEnum</a> value to convert.
-------------	--

Definition at line 255 of file SpaceFOM/Types.cpp.

References TrickHLA::MODE\_TRANSITION\_GOTO\_FREEZE, TrickHLA::MODE\_TRANSITION\_GOTO\_RECONFIG, TrickHLA::MODE\_TRANSITION\_GOTO\_RESTART, TrickHLA::MODE\_TRANSITION\_GOTO\_RUN, TrickHLA::MODE\_TRANSITION\_GOTO\_SHUTDOWN, TrickHLA::MODE\_TRANSITION\_INITIALIZING, TrickHLA::MODE\_TRANSITION\_UNINITIALIZED, DIS::MTR\_GOTO\_FREEZE, DIS::MTR\_GOTO\_RUN, DIS::MTR\_GOTO\_SHUTDOWN, DIS::MTR\_INITIALIZING, and DIS::MTR\_UNINITIALIZED. Referenced by SpaceFOM::ExecutionControl::send\_mode\_transition\_interaction().

#### 6.4.2.6 mtr\_enum\_to\_int16()

```
int16_t SpaceFOM::mtr_enum_to_int16 (
    MTREnum mtr_enum )
```

Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.

##### Returns

[SpaceFOM](#) Mode Transition Request (MTR) as a 16 bit integer.

##### Parameters

<i>mtr_enum</i>	MTR enumeration values.
-----------------	-------------------------

Definition at line 214 of file SpaceFOM/Types.cpp.

Referenced by SpaceFOM::MTRInteractionHandler::send\_interaction().

#### 6.4.2.7 mtr\_enum\_to\_string()

```
string SpaceFOM::mtr_enum_to_string (
    MTREnum mtr_enum )
```

Convert a Mode Transition Request (MTR) enum value into a printable string.

**Returns**

[SpaceFOM](#) mode transition request (MTR) as a printable string.

**Parameters**

<i>mtr_enum</i>	MTR enumeration value to convert.
-----------------	-----------------------------------

Definition at line 180 of file SpaceFOM/Types.cpp.

References DIS::MTR\_GOTO\_FREEZE, DIS::MTR\_GOTO\_RUN, DIS::MTR\_GOTO\_SHUTDOWN, DIS::MTR\_INITIALIZING, and DIS::MTR\_UNINITIALIZED.

Referenced by [SpaceFOM::ExecutionControl::check\\_mode\\_transition\\_request\(\)](#), [SpaceFOM::MTRInteractionHandler::receive\\_interaction\(\)](#), and [SpaceFOM::MTRInteractionHandler::send\\_interaction\(\)](#).

#### 6.4.2.8 mtr\_int16\_to\_enum()

```
MTREnum SpaceFOM::mtr_int16_to_enum (
    int16_t mtr_int )
```

Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.

**Returns**

[SpaceFOM](#) mode transition request (MTR) as enumeration value.

**Parameters**

<i>mtr_int</i>	MTR mode.
----------------	-----------

Definition at line 221 of file SpaceFOM/Types.cpp.

References DIS::MTR\_GOTO\_FREEZE, DIS::MTR\_GOTO\_RUN, DIS::MTR\_GOTO\_SHUTDOWN, DIS::MTR\_INITIALIZING, and DIS::MTR\_UNINITIALIZED.

Referenced by [SpaceFOM::MTRInteractionHandler::receive\\_interaction\(\)](#).

#### 6.4.2.9 to\_execution\_control\_enum()

```
TrickHLA::ExecutionControlEnum SpaceFOM::to_execution_control_enum (
    ExecutionModeEnum mode )
```

Convert an [SpaceFOM::ExecutionModeEnum](#) value to a TrickHLA::ExecutionModeEnum value.

**Returns**

[SpaceFOM::ExecutionModeEnum](#) as an equivalent TrickHLA::ExecutionModeEnum value.

**Parameters**

<i>mode</i>	<a href="#">SpaceFOM::ExecutionModeEnum</a> value to convert.
-------------	---

Definition at line 106 of file SpaceFOM/Types.cpp.

References TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_STOPPING.

ON\_CONTROL\_UNINITIALIZED, DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, and DIS::EXECUTION\_MODE\_UNINITIALIZED.

### 6.4.3 Variable Documentation

#### 6.4.3.1 INIT\_COMPLETED\_SYNC\_POINT

```
const std::wstring SpaceFOM::INIT_COMPLETED_SYNC_POINT = L"initialization_completed" [static]
```

Definition at line 64 of file SpaceFOM/ExecutionControl.cpp.

Referenced by SpaceFOM::ExecutionControl::add\_initialization\_sync\_points(), SpaceFOM::ExecutionControl::announce\_sync\_point(), and SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes().

#### 6.4.3.2 INIT\_STARTED\_SYNC\_POINT

```
const std::wstring SpaceFOM::INIT_STARTED_SYNC_POINT = L"initialization_started" [static]
```

Definition at line 63 of file SpaceFOM/ExecutionControl.cpp.

Referenced by SpaceFOM::ExecutionControl::add\_initialization\_sync\_points(), SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), and SpaceFOM::ExecutionControl::role\_determination\_process().

#### 6.4.3.3 OBJECTS\_DISCOVERED\_SYNC\_POINT

```
const std::wstring SpaceFOM::OBJECTS_DISCOVERED_SYNC_POINT = L"objects_discovered" [static]
```

Definition at line 65 of file SpaceFOM/ExecutionControl.cpp.

Referenced by SpaceFOM::ExecutionControl::add\_initialization\_sync\_points(), SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), and SpaceFOM::ExecutionControl::role\_determination\_process().

#### 6.4.3.4 ROOT\_FRAME\_DISCOVERED\_SYNC\_POINT

```
const std::wstring SpaceFOM::ROOT_FRAME_DISCOVERED_SYNC_POINT = L"root_frame_discovered" [static]
```

Definition at line 66 of file SpaceFOM/ExecutionControl.cpp.

Referenced by SpaceFOM::ExecutionControl::add\_initialization\_sync\_points(), SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::role\_determination\_process(), and SpaceFOM::ExecutionControl::wait\_on\_root\_frame\_discovered\_synchronization().

## 6.5 TrickHLA Namespace Reference

### Data Structures

- class [Attribute](#)
- class [BasicClock](#)
- class [Conditional](#)
- class [CTETimelineBase](#)
- class [DebugHandler](#)
- struct [DivestThreadArgs](#)
- class [ExecutionConfiguration](#)
- class [ExecutionConfigurationBase](#)
- class [ExecutionControl](#)

- class [ExecutionControlBase](#)
- class [FedAmb](#)
- class [Federate](#)
- class [Int64Interval](#)
- class [Int64Time](#)
- class [Interaction](#)
- class [InteractionHandler](#)
- class [InteractionItem](#)
- class [Item](#)
- class [ItemQueue](#)
- class [KnownFederate](#)
- class [LagCompensation](#)
- class [LoggableSyncPnt](#)
- class [LoggableTimedSyncPnt](#)
- class [Manager](#)
- class [Object](#)
- class [ObjectDeleted](#)
- class [OpaqueBuffer](#)
- class [OwnershipHandler](#)
- class [OwnershipItem](#)
- class [Packing](#)
- class [Parameter](#)
- class [ParameterItem](#)
- class [ReflectedAttributesQueue](#)
- class [ScenarioTimeline](#)
- class [SimTimeline](#)
- class [StringUtilities](#)
- class [SyncPnt](#)
- class [SyncPntList](#)
- class [SyncPntListBase](#)
- class [TimedSyncPnt](#)
- class [TimedSyncPntList](#)
- class [Timeline](#)
- class [TimeOfDayTimeline](#)
- class [Utilities](#)

## Typedefs

- `typedef std::map< RTI1516_NAMESPACE::AttributeHandle, Attribute * > AttributeMap`

*Data I/O: \*\*  
Map of attributes.*
- `typedef std::map< RTI1516_NAMESPACE::ObjectInstanceHandle, Object * > ObjectInstanceMap`

*trick\_io{\*\*} Map of TrickHLA objects.*
- `typedef std::map< std::string, Attribute * > THLAAttributeMap`
- `typedef std::map< double, THLAAttributeMap *, std::less< double > > AttributeOwnershipMap`
- `typedef std::auto_ptr< RTI1516_NAMESPACE::RTIambassador > TrickRTIAmbPtr`
- `typedef std::queue< RTI1516_NAMESPACE::AttributeHandleValueMap > HLAAttributeMapQueue`
- `typedef std::map< RTI1516_NAMESPACE::ObjectInstanceHandle, std::wstring > TrickHLAOBJInstanceNameMap`
- `typedef std::vector< std::string > VectorOfStrings`
- `typedef std::vector< std::wstring > VectorOfWstrings`

## Enumerations

- enum THLASaveRestoreProcEnum {
 `No_Restore` = 0, `Restore_Request_Failed` = 1, `Restore_Request_Succeeded` = 2, `Initiate_Restore` = 3, `Restore_In_Progress` = 4, `Restore_Complete` = 5, `Restore_Failed` = 6
 }
- enum ManagerTypeOfInteractionEnum { `TRICKHLA_MANAGER_USER_DEFINED_INTERACTION` = 0, `TRICKHLA_MANAGER_BUILTIN_FREEZE_INTERACTION` = 1, `TRICKHLA_MANAGER_BUILTIN_MTR_INTERACTION` = 2 }
- enum DataUpdateEnum {
 `CONFIG_NONE` = 0x0001, `CONFIG_INITIALIZE` = 0x0002, `CONFIG_INTERMITTENT` = 0x0004, `CONFIG_CYCLIC` = 0x0008, `CONFIG_MAX_VALUE`
}

*Define the `TrickHLA` attribute update reflection type.*
- enum EncodingEnum {
 `ENCODING_FIRST_VALUE` = 0, `ENCODING_UNKNOWN` = 0, `ENCODING_BIG_ENDIAN` = 1, `ENCODING_LITTLE_ENDIAN` = 2, `ENCODING_LOGICAL_TIME` = 3, `ENCODING_C_STRING` = 4, `ENCODING_UNICODE_STRING` = 5, `ENCODING_ASCII_STRING` = 6, `ENCODING_OPAQUE_DATA` = 7, `ENCODING_BOOLEAN` = 8, `ENCODING_NO_ENCODING` = 9, `ENCODING_LAST_VALUE` = 9
}

*Define the `TrickHLA` data encoding type.*
- enum TransportationEnum {
 `TRANSPORT_FIRST_VALUE` = 0, `TRANSPORT_SPECIFIED_IN_FOM` = 0, `TRANSPORT_TIMESTAMP_ORDER` = 1, `TRANSPORT_RECEIVE_ORDER` = 2, `TRANSPORT_LAST_VALUE` = 3
}

*Define the `TrickHLA` data transportation type.*
- enum LagCompensationEnum {
 `LAG_COMPENSATION_FIRST_VALUE` = 0, `LAG_COMPENSATION_NONE` = 0, `LAG_COMPENSATION_SEND_SIDE` = 1, `LAG_COMPENSATION_RECEIVE_SIDE` = 2, `LAG_COMPENSATION_LAST_VALUE` = 2
}

*Define the `TrickHLA` latency (lag) compensation type.*
- enum DebugLevelEnum {
 `DEBUG_LEVEL_NO_TRACE` = 0, `DEBUG_LEVEL_0_TRACE` = 0, `DEBUG_LEVEL_1_TRACE` = 1, `DEBUG_LEVEL_2_TRACE` = 2, `DEBUG_LEVEL_3_TRACE` = 3, `DEBUG_LEVEL_4_TRACE` = 4, `DEBUG_LEVEL_5_TRACE` = 5, `DEBUG_LEVEL_6_TRACE` = 6, `DEBUG_LEVEL_7_TRACE` = 7, `DEBUG_LEVEL_8_TRACE` = 8, `DEBUG_LEVEL_9_TRACE` = 9, `DEBUG_LEVEL_10_TRACE` = 10, `DEBUG_LEVEL_11_TRACE` = 11, `DEBUG_LEVEL_FULL_TRACE` = 11
}

*Define the `TrickHLA` level for debug messages.*
- enum DebugSourceEnum {
 `DEBUG_SOURCE_FED_AMB` = 0x00000001, `DEBUG_SOURCE_FEDERATE` = 0x00000002, `DEBUG_SOURCE_MANAGER` = 0x00000004, `DEBUG_SOURCE_OBJECT` = 0x00000008, `DEBUG_SOURCE_INTERACTION` = 0x00000010, `DEBUG_SOURCE_ATTRIBUTE` = 0x00000020, `DEBUG_SOURCE_PARAMETER` = 0x00000040, `DEBUG_SOURCE_SYNCPOINT` = 0x00000080, `DEBUG_SOURCE_OWNERSHIP` = 0x00000100, `DEBUG_SOURCE_PACKING` = 0x00000200, `DEBUG_SOURCE_LAG_COMPENSATION` = 0x00000400, `DEBUG_SOURCE_ALL_MODULES` = 0x7FFFFFFF
}

*Define the `TrickHLA` source for debug messages.*
- enum FederateJoinEnum {
 `FEDERATE_JOIN_FIRST_VALUE` = 0, `FEDERATE_JOIN NOMINAL` = 0, `FEDERATE_JOIN_` `EARLY` = 0, `FEDERATE_JOIN_LATE` = 1, `FEDERATE_JOIN_RESTORING` = 2, `FEDERATE_JOIN_UNKNOWN` = 3, `FEDERATE_JOIN_LAST_VALUE` = 3
}

Define the *TrickHLA* federate join enumeration values.

- enum `ExecutionControlEnum` {
`EXECUTION_CONTROL_FIRST_VALUE` = 0, `EXECUTION_CONTROL_UNINITIALIZED` = 0, `EXECUTION_CONTROL_INITIALIZING` = 1, `EXECUTION_CONTROL_RUNNING` = 2, `EXECUTION_CONTROL_FREEZE` = 3, `EXECUTION_CONTROL_RESTART` = 4, `EXECUTION_CONTROL_RECONFIG` = 5, `EXECUTION_CONTROL_SHUTDOWN` = 6, `EXECUTION_CONTROL_LAST_VALUE` = 6 }

Define the *TrickHLA* execution control enumeration values.

- enum `ModeTransitionEnum` {
`MODE_TRANSITION_FIRST_VALUE` = 0, `MODE_TRANSITION_UNINITIALIZED` = 0, `MODE_TRANSITION_INITIALIZING` = 1, `MODE_TRANSITION_GOTO_RUN` = 2, `MODE_TRANSITION_GOTO_FREEZE` = 3, `MODE_TRANSITION_GOTO_RESTART` = 4, `MODE_TRANSITION_GOTO_RECONFIG` = 5, `MODE_TRANSITION_GOTO_SHUTDOWN` = 6, `MODE_TRANSITION_LAST_VALUE` = 6 }

Define the *TrickHLA* Mode Transition state enumeration values.

- enum `SyncPntStateEnum` {
`SYNC_PNT_STATE_FIRST_VALUE` = 0, `SYNC_PNT_STATE_ERROR` = 0, `SYNC_PNT_STATE_EXISTS` = 1, `SYNC_PNT_STATE_REGISTERED` = 2, `SYNC_PNT_STATE_ANNOUNCED` = 3, `SYNC_PNT_STATE_ACHIEVED` = 4, `SYNC_PNT_STATE_SYNCHRONIZED` = 5, `SYNC_PNT_STATE_LAST_VALUE` = 5, `SYNC_PNT_STATE_UNKNOWN` = `INT_MAX` }

Define the *TrickHLA* synchronization point state enumeration values.

## Functions

- `std::string execution_control_enum_to_string (ExecutionControlEnum mode)`  
Convert an `ExecutionModeEnum` value into a printable string.
- `int16_t execution_control_enum_to_int16 (ExecutionControlEnum mode)`  
Convert an `ExecutionModeEnum` value into a 16 bit integer.
- `ExecutionControlEnum execution_control_int16_to_enum (int16_t int_mode)`  
Convert a 16 bit integer to an `ExecutionModeEnum` value.
- `std::string mode_transition_enum_to_string (ModeTransitionEnum mode)`  
Convert an `ModeTransitionEnum` value into a printable string.
- `int16_t mode_transition_enum_to_int16 (ModeTransitionEnum mode)`  
Convert an `ModeTransitionEnum` value into a 16 bit integer.
- `ModeTransitionEnum mode_transition_int16_to_enum (int16_t int_mode)`  
Convert a 16 bit integer to an `ModeTransitionEnum` value.
- `std::string sync_pnt_state_enum_to_string (SyncPntStateEnum state)`  
Convert a `Synchronization Point State` enum value into a printable string.
- `int16_t sync_pnt_state_enum_to_int16 (SyncPntStateEnum state)`  
Convert a `Synchronization Point State` enum value into a 16 bit integer.
- `SyncPntStateEnum sync_pnt_state_int16_to_enum (int16_t int_state)`  
Convert an integer value to a `Synchronization Point State` enumeration value.

## Variables

- `const int64_t MICROS_MULTIPLIER = 1000000`  
**Units:** –
- `const int64_t MAX_VALUE_IN_MICROS = std::numeric_limits< int64_t >::max()`

*Units: us*

- const double `MAX_LOGICAL_TIME_SECONDS` = (double)`MAX_VALUE_IN_MICROS / MICROS_MULTIPLIER`  
*Units: s*

- `SimTimeline def_sim_timeline`
- `ScenarioTimeline def_scenario_timeline (def_sim_timeline)`

## 6.5.1 Typedef Documentation

### 6.5.1.1 AttributeMap

```
typedef std::map< RTI1516_NAMESPACE::AttributeHandle, Attribute * > TrickHLA::AttributeMap
```

**Data I/O: \*\***

Map of attributes.

Definition at line 442 of file Attribute.hh.

### 6.5.1.2 AttributeOwnershipMap

```
typedef std::map< double, THLAAttributeMap * , std::less< double > > TrickHLA::AttributeOwnershipMap
```

Definition at line 66 of file OwnershipHandler.hh.

### 6.5.1.3 HLAAttributeMapQueue

```
typedef std::queue< RTI1516_NAMESPACE::AttributeHandleValueMap > TrickHLA::HLAAttributeMapQueue
```

Definition at line 245 of file TrickHLA/Types.hh.

### 6.5.1.4 ObjectInstanceMap

```
typedef std::map< RTI1516_NAMESPACE::ObjectInstanceHandle, Object * > TrickHLA::ObjectInstanceMap
```

trick\_io{\*\*} Map of `TrickHLA` objects.  
 Definition at line 738 of file Object.hh.

### 6.5.1.5 THLAAttributeMap

```
typedef std::map< std::string, Attribute * > TrickHLA::THLAAttributeMap
```

Definition at line 58 of file OwnershipHandler.hh.

### 6.5.1.6 TrickHLAObjInstanceNameMap

```
typedef std::map< RTI1516_NAMESPACE::ObjectInstanceHandle, std::wstring > TrickHLA::TrickHLAObjInstanceNameMap
```

Definition at line 247 of file TrickHLA/Types.hh.

### 6.5.1.7 TrickRTIAmbPtr

```
typedef std::auto_ptr< RTI1516_NAMESPACE::RTIambassador > TrickHLA::TrickRTIAmbPtr
```

Definition at line 243 of file TrickHLA/Types.hh.

### 6.5.1.8 VectorOfStrings

```
typedef std::vector< std::string > TrickHLA::VectorOfStrings
Definition at line 249 of file TrickHLA/Types.hh.
```

### 6.5.1.9 VectorOfWstrings

```
typedef std::vector< std::wstring > TrickHLA::VectorOfWstrings
Definition at line 251 of file TrickHLA/Types.hh.
```

## 6.5.2 Enumeration Type Documentation

### 6.5.2.1 DataUpdateEnum

```
enum TrickHLA::DataUpdateEnum
```

Define the [TrickHLA](#) attribute update reflection type.

#### Enumerator

CONFIG_NONE	No configuration.
CONFIG_INITIALIZE	Dynamic simulation initialization.
CONFIG_INTERMITTENT	Intermittent updates.
CONFIG_CYCLIC	Cyclic updates.
CONFIG_MAX_VALUE	Maximum configuration value.

Definition at line 54 of file TrickHLA/Types.hh.

### 6.5.2.2 DebugLevelEnum

```
enum TrickHLA::DebugLevelEnum
```

Define the [TrickHLA](#) level for debug messages.

#### Enumerator

DEBUG_LEVEL_NO_TRACE	Default: No <a href="#">TrickHLA</a> output is displayed; user messages will still be printed.
DEBUG_LEVEL_0_TRACE	Default: No <a href="#">TrickHLA</a> output is displayed; user messages will still be printed.
DEBUG_LEVEL_1_TRACE	Adds initialization complete and Time Advance Grant messages.
DEBUG_LEVEL_2_TRACE	Adds initialization messages as well as the standard complement of execution messages.
DEBUG_LEVEL_3_TRACE	Adds Ownership Transfer messages.
DEBUG_LEVEL_4_TRACE	Adds HLA Time Advancement, Freeze job, and additional Shutdown job messages.
DEBUG_LEVEL_5_TRACE	Adds additional HLA Time Advancement, <a href="#">Interaction</a> , InitSyncPts and SyncPts messages.
DEBUG_LEVEL_6_TRACE	Adds Packing/LagCompensation subclass messages.
DEBUG_LEVEL_7_TRACE	Adds the names of all Attributes/Parameters sent to other federates.
DEBUG_LEVEL_8_TRACE	Adds FederateAmbassador and RTI callback messages.
DEBUG_LEVEL_9_TRACE	Adds Trick Ref-Attributes and RTI Handles (both during initialization).

## Enumerator

DEBUG_LEVEL_10_TRACE	Adds internal state of all Attributes and Parameters.
DEBUG_LEVEL_11_TRACE	Adds buffer contents of all Attributes and Parameters.
DEBUG_LEVEL_FULL_TRACE	Outputs All debug messages.

Definition at line 120 of file TrickHLA/Types.hh.

### 6.5.2.3 DebugSourceEnum

enum [TrickHLA::DebugSourceEnum](#)

Define the [TrickHLA](#) source for debug messages.

## Enumerator

DEBUG_SOURCE_FED_AMB	Adds <a href="#">TrickHLA::FedAmb</a> debug messages.
DEBUG_SOURCE_FEDERATE	Adds <a href="#">TrickHLA::Federate</a> debug messages.
DEBUG_SOURCE_MANAGER	Adds <a href="#">TrickHLA::Manager</a> debug messages.
DEBUG_SOURCE_OBJECT	Adds <a href="#">TrickHLA::Object</a> (and subclass) debug messages.
DEBUG_SOURCE_INTERACTION	Adds <a href="#">TrickHLA::Interaction</a> (and subclass) debug messages.
DEBUG_SOURCE_ATTRIBUTE	Adds <a href="#">TrickHLA::Attribute</a> debug messages.
DEBUG_SOURCE_PARAMETER	Adds <a href="#">TrickHLA::Parameter</a> debug messages.
DEBUG_SOURCE_SYNCPOINT	Adds <a href="#">TrickHLA::SyncPoint</a> debug messages.
DEBUG_SOURCE_OWNERSHIP	Adds <a href="#">TrickHLA::OwnershipHandler</a> debug messages.
DEBUG_SOURCE_PACKING	Adds <a href="#">TrickHLA::Packing</a> (and subclass) debug messages.
DEBUG_SOURCE_LAG_COMPENSATION	Adds <a href="#">TrickHLA::LagCompensation</a> (and subclass) debug messages.
DEBUG_SOURCE_ALL_MODULES	Default: Add debug messages from all code modules.

Definition at line 145 of file TrickHLA/Types.hh.

### 6.5.2.4 EncodingEnum

enum [TrickHLA::EncodingEnum](#)

Define the [TrickHLA](#) data encoding type.

## Enumerator

ENCODING_FIRST_VALUE	Set to the First value in the enumeration.
ENCODING_UNKNOWN	Default encoding. The software automatically determines it for you. Otherwise, specify to one of the below values.
ENCODING_BIG_ENDIAN	Big Endian.
ENCODING_LITTLE_ENDIAN	Little Endian.
ENCODING_LOGICAL_TIME	64-bit Big Endian encoded integer representing microseconds.
ENCODING_C_STRING	Null terminated C string (i.e. char *).
ENCODING_UNICODE_STRING	Variable length HLA Unicode string encoding.
ENCODING_ASCII_STRING	Variable length HLA ASCII string encoding.
ENCODING_OPAQUE_DATA	Variable length HLA Opaque data for a "char *" type.

## Enumerator

ENCODING_BOOLEAN	Boolean c++ type configured in the FOM to use HLAboolean HLA data type encoded as an HLAinteger32BE.
ENCODING_NO_ENCODING	Fixed length array of data for "char *" type sent as is.
ENCODING_LAST_VALUE	Set to the Last value in the enumeration.

Definition at line 71 of file TrickHLA/Types.hh.

#### 6.5.2.5 ExecutionControlEnum

enum [TrickHLA::ExecutionControlEnum](#)

Define the [TrickHLA](#) execution control enumeration values.

## Enumerator

EXECUTION_CONTROL_FIRST_VALUE	Set to the First value in the enumeration.
EXECUTION_CONTROL_UNINITIALIZED	Execution control state is uninitialized.
EXECUTION_CONTROL_INITIALIZING	Execution control state is initializing.
EXECUTION_CONTROL_RUNNING	Execution control state is running.
EXECUTION_CONTROL_FREEZE	Execution control state is freeze.
EXECUTION_CONTROL_RESTART	Execution control state is restart.
EXECUTION_CONTROL_RECONFIG	Execution control state is reconfigure.
EXECUTION_CONTROL_SHUTDOWN	Execution control state is shutdown.
EXECUTION_CONTROL_LAST_VALUE	Set to the Last value in the enumeration.

Definition at line 182 of file TrickHLA/Types.hh.

#### 6.5.2.6 FederateJoinEnum

enum [TrickHLA::FederateJoinEnum](#)

Define the [TrickHLA](#) federate join enumeration values.

## Enumerator

FEDERATE_JOIN_FIRST_VALUE	Set to the First value in the enumeration.
FEDERATE_JOIN_NOMINAL	Normal <a href="#">Federate</a> Execution (neither late joiner nor federate restore).
FEDERATE_JOIN_EARLY	Early joining <a href="#">Federate</a> .
FEDERATE_JOIN_LATE	Late Joining <a href="#">Federate</a> .
FEDERATE_JOIN_RESTORING	<a href="#">Federate</a> Restore.
FEDERATE_JOIN_UNKNOWN	Unknown <a href="#">Federate</a> state.
FEDERATE_JOIN_LAST_VALUE	Set to the Last value in the enumeration.

Definition at line 166 of file TrickHLA/Types.hh.

#### 6.5.2.7 LagCompensationEnum

enum [TrickHLA::LagCompensationEnum](#)

Define the [TrickHLA](#) latency (lag) compensation type.

#### Enumerator

LAG_COMPENSATION_FIRST_VALUE	Set to the First value in the enumeration.
LAG_COMPENSATION_NONE	No lag compensation.
LAG_COMPENSATION_SEND_SIDE	Send-side lag compensation.
LAG_COMPENSATION_RECEIVE_SIDE	Receive-side lag compensation.
LAG_COMPENSATION_LAST_VALUE	Set to the Last value in the enumeration.

Definition at line 106 of file TrickHLA/Types.hh.

#### 6.5.2.8 ManagerTypeOfInteractionEnum

```
enum TrickHLA::ManagerTypeOfInteractionEnum
```

#### Enumerator

TRICKHLA_MANAGER_USER_DEFINED_INTERACTION	Interaction must be defined by the user in the input file.
TRICKHLA_MANAGER_BUILTIN_FREEZE_INTERACTION	Freeze Interaction internal to TrickHLA.
TRICKHLA_MANAGER_BUILTIN_MTR_INTERACTION	MTR Interaction internal to TrickHLA.

Definition at line 80 of file Manager.hh.

#### 6.5.2.9 ModeTransitionEnum

```
enum TrickHLA::ModeTransitionEnum
```

Define the [TrickHLA](#) Mode Transition state enumeration values.

The [TrickHLA::ModeTransitionEnum](#) enumeration defines the possible mode transition for the [TrickHLA::ExecutionControl](#) executive. These mode requests are important in the execution control process involving mode requests from any federate participating in a controlled federation execution and usually processed by the Master federate.

#### Enumerator

MODE_TRANSITION_FIRST_VALUE	Not a valid mode transition.
MODE_TRANSITION_UNINITIALIZED	Not a valid mode transition.
MODE_TRANSITION_INITIALIZING	Not a valid mode transition.
MODE_TRANSITION_GOTO_RUN	Mode transition to RUN mode.
MODE_TRANSITION_GOTO_FREEZE	Mode transition to FREEZE mode.
MODE_TRANSITION_GOTO_RESTART	Mode transition to RESTART mode.
MODE_TRANSITION_GOTO_RECONFIG	Mode transition to RECONFIG mode.
MODE_TRANSITION_GOTO_SHUTDOWN	Mode transition to SHUTDOWN mode.
MODE_TRANSITION_LAST_VALUE	Same as shutdown.

Definition at line 207 of file TrickHLA/Types.hh.

#### 6.5.2.10 SyncPntStateEnum

```
enum TrickHLA::SyncPntStateEnum
```

Define the [TrickHLA](#) synchronization point state enumeration values.

The SyncPntStateEnum enumeration defines the possible synchronization point (sync-point) synchronization states for a [TrickHLA](#) based federate. These sync-point states correspond directly to the sync-point states in HLA.

**Enumerator**

SYNC_PNT_STATE_FIRST_VALUE	Set to the First value in the enumeration.
SYNC_PNT_STATE_ERROR	Sync-point error.
SYNC_PNT_STATE_EXISTS	Sync-point exists.
SYNC_PNT_STATE_REGISTERED	Sync-point registered.
SYNC_PNT_STATE_ANNOUNCED	Sync-point announced.
SYNC_PNT_STATE_ACHIEVED	Sync-point achieved.
SYNC_PNT_STATE_SYNCHRONIZED	Sync-point synchronized.
SYNC_PNT_STATE_LAST_VALUE	Set to the Last value in the enumeration.
SYNC_PNT_STATE_UNKNOWN	Unknown state.

Definition at line 229 of file TrickHLA/Types.hh.

#### 6.5.2.11 THLASaveRestoreProcEnum

```
enum TrickHLA::THLASaveRestoreProcEnum
```

**Enumerator**

No_Restore	
Restore_Request_Failed	
Restore_Request_Succeeded	
Initiate_Restore	
Restore_In_Progress	
Restore_Complete	
Restore_Failed	

Definition at line 72 of file Federate.hh.

#### 6.5.2.12 TransportationEnum

```
enum TrickHLA::TransportationEnum
```

Define the [TrickHLA](#) data transportation type.

**Enumerator**

TRANSPORT_FIRST_VALUE	Set to the First value in the enumeration.
TRANSPORT_SPECIFIED_IN_FOM	Indicates which attributes or interactions use the order specified in the FOM.
TRANSPORT_TIMESTAMP_ORDER	Indicates which attributes or interactions are Timestamp Order.
TRANSPORT_RECEIVE_ORDER	Indicates which attributes or interactions are Receive Order.
TRANSPORT_LAST_VALUE	Set to the Last value in the enumeration.

Definition at line 92 of file TrickHLA/Types.hh.

### 6.5.3 Function Documentation

#### 6.5.3.1 execution\_control\_enum\_to\_int16()

```
int16_t TrickHLA::execution_control_enum_to_int16 (
```

`ExecutionControlEnum mode` )

Convert an ExecutionModeEnum value into a 16 bit integer.

##### Returns

Execution control mode as a 16 bit integer representation.

##### Parameters

<code>mode</code>	Execution control mode enumeration value.
-------------------	---

Definition at line 79 of file TrickHLA/Types.cpp.

#### 6.5.3.2 execution\_control\_enum\_to\_string()

```
string TrickHLA::execution_control_enum_to_string (
```

`ExecutionControlEnum mode` )

Convert an ExecutionModeEnum value into a printable string.

##### Returns

Execution control mode as a printable string.

##### Parameters

<code>mode</code>	Execution configuration run mode enumeration value.
-------------------	---

Definition at line 38 of file TrickHLA/Types.cpp.

References EXECUTION\_CONTROL\_FREEZE, EXECUTION\_CONTROL\_INITIALIZING, EXECUTION\_CONTROL\_RECONFIG, EXECUTION\_CONTROL\_RESTART, EXECUTION\_CONTROL\_RUNNING, EXECUTION\_CONTROL\_SHUTDOWN, and EXECUTION\_CONTROL\_UNINITIALIZED.

Referenced by SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), and IMSim::ExecutionControl::process\_execution\_control\_updates().

#### 6.5.3.3 execution\_control\_int16\_to\_enum()

```
ExecutionControlEnum TrickHLA::execution_control_int16_to_enum (
```

`int16_t int_mode` )

Convert a 16 bit integer to an ExecutionModeEnum value.

##### Returns

Execution control mode as enumeration value.

**Parameters**

<i>int_mode</i>	Execution control mode as integer.
-----------------	------------------------------------

Definition at line 86 of file TrickHLA/Types.cpp.

References EXECUTION\_CONTROL\_FREEZE, EXECUTION\_CONTROL\_INITIALIZING, EXECUTION\_CONTROL↔\_RECONFIG, EXECUTION\_CONTROL\_RESTART, EXECUTION\_CONTROL\_RUNNING, EXECUTION\_CONTROL↔\_SHUTDOWN, and EXECUTION\_CONTROL\_UNINITIALIZED.

Referenced by TrickHLA::ExecutionControlBase::set\_requested\_execution\_control\_mode().

**6.5.3.4 mode\_transition\_enum\_to\_int16()**

```
int16_t TrickHLA::mode_transition_enum_to_int16 (
    ModeTransitionEnum mode )
```

Convert an ModeTransitionEnum value into a 16 bit integer.

**Returns**

Mode transition as a 16 bit integer representation.

**Parameters**

<i>mode</i>	Mode transition enumeration value.
-------------	------------------------------------

Definition at line 168 of file TrickHLA/Types.cpp.

**6.5.3.5 mode\_transition\_enum\_to\_string()**

```
string TrickHLA::mode_transition_enum_to_string (
    ModeTransitionEnum mode )
```

Convert an ModeTransitionEnum value into a printable string.

**Returns**

Mode transition as a printable string.

**Parameters**

<i>mode</i>	Mode transition enumeration value.
-------------	------------------------------------

Definition at line 127 of file TrickHLA/Types.cpp.

References MODE\_TRANSITION\_GOTO\_FREEZE, MODE\_TRANSITION\_GOTO\_RECONFIG, MODE\_TRANSITION↔\_ON\_GOTO\_RESTART, MODE\_TRANSITION\_GOTO\_RUN, MODE\_TRANSITION\_GOTO\_SHUTDOWN, MODE\_T↔\_TRANSITION\_INITIALIZING, and MODE\_TRANSITION\_UNINITIALIZED.

**6.5.3.6 mode\_transition\_int16\_to\_enum()**

```
ModeTransitionEnum TrickHLA::mode_transition_int16_to_enum (
    int16_t int_mode )
```

Convert a 16 bit integer to an ModeTransitionEnum value.

**Returns**

Mode transition as enumeration value.

**Parameters**

<i>int_mode</i>	Mode transition as integer.
-----------------	-----------------------------

Definition at line 175 of file TrickHLA/Types.cpp.

References MODE\_TRANSITION\_GOTO\_FREEZE, MODE\_TRANSITION\_GOTO\_RECONFIG, MODE\_TRANSITION\_GOTO\_RESTART, MODE\_TRANSITION\_GOTO\_RUN, MODE\_TRANSITION\_GOTO\_SHUTDOWN, MODE\_TRANSITION\_INITIALIZING, and MODE\_TRANSITION\_UNINITIALIZED.

### 6.5.3.7 sync\_pnt\_state\_enum\_to\_int16()

```
int16_t TrickHLA::sync_pnt_state_enum_to_int16 (
    SyncPntStateEnum state )
```

Convert a Synchronization Point State enum value into a 16 bit integer.

**Returns**

TrickHLA sync point state as a 16 bit integer.

**Parameters**

<i>state</i>	Sync point state enumeration value to convert.
--------------	--

Definition at line 254 of file TrickHLA/Types.cpp.

References state.

### 6.5.3.8 sync\_pnt\_state\_enum\_to\_string()

```
string TrickHLA::sync_pnt_state_enum_to_string (
    SyncPntStateEnum state )
```

Convert a Synchronization Point State enum value into a printable string.

**Returns**

TrickHLA sync point state as a printable string.

**Parameters**

<i>state</i>	Sync point state enumeration value to convert.
--------------	--

Definition at line 216 of file TrickHLA/Types.cpp.

References state, SYNC\_PNT\_STATE\_ACHIEVED, SYNC\_PNT\_STATE\_ANNOUNCED, SYNC\_PNT\_STATE\_EROR, SYNC\_PNT\_STATE\_EXISTS, SYNC\_PNT\_STATE\_REGISTERED, and SYNC\_PNT\_STATE\_SYNCHRONIZED.

### 6.5.3.9 sync\_pnt\_state\_int16\_to\_enum()

```
SyncPntStateEnum TrickHLA::sync_pnt_state_int16_to_enum (
    int16_t int_state )
```

Convert an integer value to a Synchronization Point State enumeration value.

#### Returns

[TrickHLA](#) Synchronization Point State enum value.

#### Parameters

<i>int_state</i>	Sync point state value as a 16 bit integer.
------------------	---

Definition at line 260 of file TrickHLA/Types.cpp.

References SYNC\_PNT\_STATE\_ACHIEVED, SYNC\_PNT\_STATE\_ANNOUNCED, SYNC\_PNT\_STATE\_ERROR, SYNC\_PNT\_STATE\_EXISTS, SYNC\_PNT\_STATE\_REGISTERED, SYNC\_PNT\_STATE\_SYNCHRONIZED, and SYNC\_PNT\_STATE\_UNKNOWN.

## 6.5.4 Variable Documentation

### 6.5.4.1 def\_scenario\_timeline

```
ScenarioTimeline TrickHLA::def_scenario_timeline(def_sim_timeline)
```

Referenced by TrickHLA::ExecutionControlBase::initialize().

### 6.5.4.2 def\_sim\_timeline

```
SimTimeline TrickHLA::def_sim_timeline
```

Definition at line 61 of file ExecutionControlBase.cpp.

### 6.5.4.3 MAX\_LOGICAL\_TIME\_SECONDS

```
const double TrickHLA::MAX_LOGICAL_TIME_SECONDS = (double)MAX_VALUE_IN_MICROS / MICROS_MULTIPLIER
Units: s
```

Definition at line 44 of file Int64Interval.cpp.

Referenced by TrickHLA::Parameter::encode\_logical\_time(), TrickHLA::Attribute::encode\_logical\_time(), TrickHLA::A::LagCompensation::get\_granted\_fed\_time(), TrickHLA::InteractionHandler::get\_granted\_fed\_time(), TrickHLA::OwnershipHandler::get\_granted\_fed\_time(), TrickHLA::Interaction::get\_granted\_fed\_time(), TrickHLA::Manager::get\_granted\_fed\_time(), TrickHLA::Object::get\_granted\_fed\_time(), TrickHLAModel::SimpleSimConfig::pack(), TrickHLA::ExecutionConfiguration::pack(), and TrickHLA::Int64Interval::toMicroseconds().

### 6.5.4.4 MAX\_VALUE\_IN\_MICROS

```
const int64_t TrickHLA::MAX_VALUE_IN_MICROS = std::numeric_limits< int64_t >::max()
```

**Units:** us

Definition at line 43 of file Int64Interval.cpp.

Referenced by TrickHLA::Parameter::decode\_logical\_time(), TrickHLA::Attribute::decode\_logical\_time(), TrickHLA::Parameter::encode\_logical\_time(), TrickHLA::Attribute::encode\_logical\_time(), TrickHLAModel::SimpleSimConfig::pack(), TrickHLA::ExecutionConfiguration::pack(), and TrickHLA::Int64Interval::toMicroseconds().

#### 6.5.4.5 MICROS\_MULTIPLIER

```
const int64_t TrickHLA::MICROS_MULTIPLIER = 1000000
```

**Units:** –

Definition at line 42 of file Int64Interval.cpp.

Referenced by TrickHLA::Parameter::decode\_logical\_time(), TrickHLA::Attribute::decode\_logical\_time(), TrickHLA::Parameter::encode\_logical\_time(), TrickHLA::Attribute::encode\_logical\_time(), TrickHLA::Int64Interval::getDoubleTime(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Int64Interval::getMicros(), TrickHLA::Int64Time::getMicros(), TrickHLA::Int64Interval::getSeconds(), TrickHLA::Int64Time::getSeconds(), TrickHLAModel::SimpleSimConfig::pack(), TrickHLA::ExecutionConfiguration::pack(), TrickHLA::Int64Interval::toMicroseconds(), TrickHLAModel::SimpleSimConfig::unpack(), and TrickHLA::ExecutionConfiguration::unpack().

## 6.6 TrickHLAModel Namespace Reference

### Data Structures

- class [SimpleSimConfig](#)
- class [SineConditional](#)
- class [SineData](#)
- class [SineInteractionHandler](#)
- class [SineLagCompensation](#)
- class [SineObjectDeleted](#)
- class [SineOwnershipHandler](#)
- class [SinePacking](#)



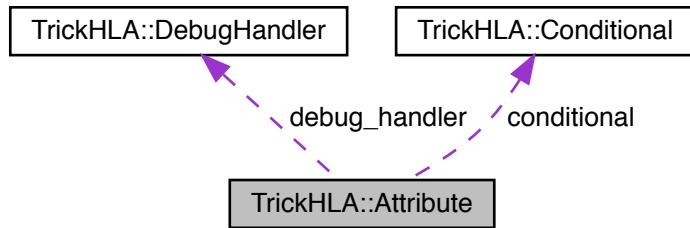
## Chapter 7

# Data Structure Documentation

### 7.1 TrickHLA::Attribute Class Reference

```
#include <Attribute.hh>
```

Collaboration diagram for TrickHLA::Attribute:



#### Public Member Functions

- **Attribute ()**  
*Default constructor for the [TrickHLA Attribute](#) class.*
- **virtual ~Attribute ()**  
*Destructor for the [TrickHLA Attribute](#) class.*
- **void initialize (const char \*obj\_FOM\_name, const int object\_index, const int attribute\_index)**  
*Initializes the [TrickHLA Attribute](#).*
- **const DataUpdateEnum get\_configuration () const**  
*Get the reflection rate configuration type.*
- **void set\_configuration (const DataUpdateEnum c)**  
*Set the reflection rate configuration type.*
- **void determine\_cycle\_ratio (double core\_job\_cycle\_time)**  
*Determine the cycle-ratio given the core job cycle rate and the cycle-time for this attribute.*
- **void pack\_attribute\_buffer ()**  
*Pack the attribute into the buffer using the appropriate encoding.*

- void `unpack_attribute_buffer ()`  
*Unpack the attribute from the buffer into the trick-variable using the appropriate decoding.*
- RTI1516\_NAMESPACE::VariableLengthData `get_attribute_value ()`  
*Gets the encoded attribute value.*
- void `extract_data` (RTI1516\_NAMESPACE::VariableLengthData \*attr\_value)  
*Extract the data out of the HLA `Attribute` Value.*
- bool `is_received () const`  
*Determine if an attribute was received from another federate.*
- bool `is_changed () const`  
*Determine if an attribute value has changed.*
- void `mark_changed ()`  
*Mark the attribute value as changed.*
- void `mark_unchanged ()`  
*Mark the attribute value as unchanged.*
- void `set_debug_level` (DebugHandler hndlr)  
*Set the debug handler settings for this attribute.*
- const char \* `get_FOM_name () const`  
*Get the Federation `Object` Model attribute name.*
- const char \* `get_trick_name () const`  
*Get the associated Trick variable space name.*
- bool `is_publish () const`  
*Determine if the attribute is published.*
- void `set_publish` (bool enable)  
*Set the attribute publish flag.*
- bool `is_subscribe () const`  
*Determine if the attribute is subscribed.*
- void `set_subscribe` (bool enable)  
*Set the attribute is subscribed flag.*
- bool `is_locally_owned () const`  
*Determine if the attribute is locally owned.*
- void `mark_locally_owned ()`  
*Mark the attribute is locally owned flag.*
- void `unmark_locally_owned ()`  
*Mark the attribute is NOT locally owned flag.*
- bool `is_remotely_owned () const`  
*Determine if the attribute is remotely owned.*
- void `mark_remotely_owned ()`  
*Mark the attribute as remotely owned.*
- bool `is_pull_requested () const`  
*Determine if someone is requesting an ownership transfer of this attribute.*
- void `set_pull_requested` (bool enable)  
*Set the pull requested flag.*
- bool `is_push_requested () const`  
*Determine if this federate is trying to push ownership of this attribute.*
- void `set_push_requested` (bool enable)  
*Set the ownership push flag.*
- bool `is_divest_requested () const`

*Determine if this federate is requesting to divest ownership of this attribute.*

- void [set\\_divest\\_requested](#) (bool enable)

*Set the ownership divest requested flag.*

- bool [is\\_byteswap](#) () const

*Determine if byteswapping is required.*

- bool [is\\_data\\_cycle\\_ready](#) () const

*Determine is the data cycle is ready for sending data.*

- bool [check\\_data\\_cycle\\_ready](#) ()

*Determine is the data cycle is ready for sending data.*

- void [set\\_preferred\\_order](#) (TransportationEnum order)

*Set the preferred transportation order.*

- TransportationEnum [get\\_preferred\\_order](#) () const

*Get the current preferred transportation order.*

- bool [is\\_update\\_requested](#) () const

*Determine if an update is requested.*

- void [set\\_update\\_requested](#) (bool request\_update)

*Set the attribute update requested flag.*

- RTI1516\_NAMESPACE::AttributeHandle [get\\_attribute\\_handle](#) () const

*Get the RTI attribute handle.*

- void [set\\_attribute\\_handle](#) (RTI1516\_NAMESPACE::AttributeHandle id)

*Set the RTI attribute handle.*

- void \* [get\\_sim\\_variable\\_address](#) ()

*Get the Trick simulation variable associated with this attribute.*

- void [print\\_buffer](#) () const

*Prints the contents of buffer used to encode/decode the attribute to the console on standard out.*

- bool [has\\_conditional](#) () const

*Check if attribute is sent conditionally.*

- Conditional \* [get\\_conditional](#) ()

*Get the associated conditionality handler.*

- ATTRIBUTES [get\\_ref2\\_attributes](#) () const

*Get the Trick "Ref Attributes" associated with this attribute.*

- EncodingEnum [get\\_rti\\_encoding](#) () const

*Get the RTI encoding for this attribute.*

- void [set\\_encoding](#) (EncodingEnum in\_type)

- bool [is\\_static\\_in\\_size](#) () const

*Determines if the attribute is static in size.*

- size\_t [calculate\\_number\\_of\\_items](#) ()

*Calculate the number of items in the attribute.*

- size\_t [get\\_attribute\\_size](#) ()

*Gets the attribute size in bytes.*

## Data Fields

- `char * trick_name`

**Units:** –  
*Trick name for the attribute.*
- `char * FOM_name`

**Units:** –  
*FOM name for the attribute.*
- `DataUpdateEnum config`

**Units:** –  
*The attribute configuration.*
- `TransportationEnum preferred_order`

**Units:** –  
*Either Timestamp (default) or Receive Order.*
- `bool publish`

**Units:** –  
*True to publish attribute that is owned locally.*
- `bool subscribe`

**Units:** –  
*True to subscribe to attribute.*
- `bool locally_owned`

**Units:** –  
*Flag to indicate attribute is locally owned.*
- `EncodingEnum rti_encoding`

**Units:** –  
*RTI encoding of the data.*
- `double cycle_time`

**Units:** s  
*Send the cyclic attribute at the specified rate.*
- `Conditional * conditional`

**Units:** –  
*Handler for a conditional attribute*

## Private Member Functions

- `void calculate_size_and_number_of_items ()`

*Calculates the attribute size in bytes and the number of items it contains.*
- `void calculate_static_number_of_items ()`

*Calculates the number of items contained by the attribute.*
- `void ensure_buffer_capacity (size_t capacity)`

*Ensure the attribute buffer has at least the specified capacity.*
- `bool is_supported_attribute_type () const`

*Determines if the HLA object attribute type is supported given the RTI encoding.*
- `void encode_boolean_to_buffer ()`

*Encode a boolean attribute into the buffer using the HLAboolean data type which is encoded as a HLAinteger32BE.*
- `void decode_boolean_from_buffer () const`

*Decode a boolean attribute from the buffer using the HLAboolean data type which is encoded as a HLAinteger32BE.*
- `void encode_logical_time () const`

*Encode the object attribute using the HLAlogicalTime 64-bit integer encoding.*
- `void decode_logical_time ()`

- Decode the object attribute that is using the HLAlogicalTime 64-bit integer encoding.*
- void [encode\\_opaque\\_data\\_to\\_buffer \(\)](#)  
*Encode the data as HLA opaque data into the buffer.*
  - void [decode\\_opaque\\_data\\_from\\_buffer \(\)](#)  
*Decode the opaque data in the buffer.*
  - void [encode\\_string\\_to\\_buffer \(\)](#)  
*Encode a string attribute into the buffer using the appropriate encoding.*
  - void [decode\\_string\\_from\\_buffer \(\)](#)  
*Decode a string from the buffer into the attribute using the appropriate decoding.*
  - void [byteswap\\_buffer\\_copy \(void \\*dest, void \\*src, int type, size\\_t length, size\\_t num\\_bytes\) const](#)  
*Copy the data from the source to the destination and byteswap as needed.*
  - [Attribute \(const Attribute &rhs\)](#)  
*Copy constructor for [Attribute](#) class.*
  - [Attribute & operator= \(const Attribute &rhs\)](#)  
*Assignment operator for [Attribute](#) class.*

## Private Attributes

- unsigned char \* [buffer](#)

**Units:** –  
*Byte buffer for the attribute value bytes.*
- size\_t [buffer\\_capacity](#)

**Units:** count  
*The capacity of the buffer.*
- bool [size\\_is\\_static](#)

**Units:** –  
*Flag to indicate the size of this attribute is static.*
- size\_t [size](#)

**Units:** count  
*The size of the attribute in bytes.*
- size\_t [num\\_items](#)

**Units:** count  
*Number of attribute items, length of the array.*
- bool [value\\_changed](#)

**Units:** –  
*Flag to indicate the attribute value changed.*
- bool [update\\_requested](#)

**Units:** –  
*Flag to indicate another federate has requested an attribute update.*
- unsigned int [HLAtrue](#)

**Units:** –  
*A 32-bit integer with a value of 1 on a Big Endian computer.*
- bool [byteswap](#)

**Units:** –  
*Flag to indicate byte-swap before RTI Rx/Tx.*
- int [cycle\\_ratio](#)

**Units:** –  
*Ratio of the attribute cycle-time to the send\_cyclic\_data job cycle time.*
- int [cycle\\_cnt](#)

- **REF2 \* ref2**

**Units:** count  
*Internal cycle counter used to determine when cyclic data will be sent.*
- **RTI1516\_NAMESPACE::AttributeHandle attr\_handle**

**Data I/O:** \*\*  
*The ref\_attributes of the given trick\_name.*
- **bool pull\_requested**

**Units:** –  
*Has someone asked to own us?*
- **bool push\_requested**

**Units:** –  
*Is someone giving up ownership?*
- **bool divest\_requested**

**Units:** –  
*Are we releasing ownership?*
- **bool initialized**

**Units:** –  
*Has this attribute been initialized?*
- **DebugHandler debug\_handler**

**Units:** –  
*Prints out multiple debug levels*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_Attribute \(\)](#)

### 7.1.1 Detailed Description

Definition at line 58 of file Attribute.hh.

### 7.1.2 Constructor & Destructor Documentation

#### 7.1.2.1 Attribute() [1/2]

`Attribute::Attribute ( )`

Default constructor for the [TrickHLA Attribute](#) class.

The endianess of the computer is determined as part of the [Attribute](#) construction process. **Trick Job Class: initialization**

Definition at line 62 of file Attribute.cpp.

References [TrickHLA::Utilities::get\\_endianness\(\)](#), and [HLAtrue](#).

#### 7.1.2.2 ~Attribute()

`Attribute::~Attribute ( ) [virtual]`

Destructor for the [TrickHLA Attribute](#) class.

The buffer and ref2 values are freed and nulled. **Trick Job Class: shutdown**

Definition at line 99 of file Attribute.cpp.

References buffer, buffer\_capacity, and ref2.

### 7.1.2.3 Attribute() [2/2]

```
TrickHLA::Attribute::Attribute (
    const Attribute & rhs ) [private]
```

Copy constructor for [Attribute](#) class.

This constructor is private to prevent inadvertent copies.

## 7.1.3 Member Function Documentation

### 7.1.3.1 byteswap\_buffer\_copy()

```
void Attribute::byteswap_buffer_copy (
    void * dest,
    void * src,
    int type,
    size_t length,
    size_t num_bytes ) const [private]
```

Copy the data from the source to the destination and byteswap as needed.

#### Parameters

<i>dest</i>	Destination to copy data to.
<i>src</i>	Source of the data to byteswap and copy from.
<i>type</i>	The type of the data.
<i>length</i>	The length/number of entries in the source array.
<i>num_bytes</i>	The number of bytes in the source array.

#### Assumptions and Limitations:

- The destination must be large enough to hold num\_bytes of data.
- Only primitive types and static arrays of primitive type are supported for now.

Definition at line 3159 of file Attribute.cpp.

References TrickHLA::Utilities::byteswap\_double(), TrickHLA::Utilities::byteswap\_float(), TrickHLA::Utilities::byteswap\_int(), TrickHLA::Utilities::byteswap\_long(), TrickHLA::Utilities::byteswap\_long\_long(), TrickHLA::Utilities::byteswap\_short(), TrickHLA::Utilities::byteswap\_unsigned\_int(), TrickHLA::Utilities::byteswap\_unsigned\_long(), TrickHLA::Utilities::byteswap\_unsigned\_long\_long(), TrickHLA::Utilities::byteswap\_unsigned\_short(), TrickHLA::DEBUG\_LEVEL\_11\_TRACE, TrickHLA::DEBUG\_SOURCE\_ATTRIBUTE, and TrickHLA::ENCODING\_NO\_ENCODING.

Referenced by pack\_attribute\_buffer(), and unpack\_attribute\_buffer().

### 7.1.3.2 calculate\_number\_of\_items()

```
size_t TrickHLA::Attribute::calculate_number_of_items ( ) [inline]
```

Calculate the number of items in the attribute.

**Returns**

Number of items in the attribute.

Definition at line 331 of file Attribute.hh.

References calculate\_size\_and\_number\_of\_items(), and num\_items.

**7.1.3.3 calculate\_size\_and\_number\_of\_items()**

```
void Attribute::calculate_size_and_number_of_items ( ) [private]
```

Calculates the attribute size in bytes and the number of items it contains.

Definition at line 841 of file Attribute.cpp.

References buffer\_capacity, calculate\_static\_number\_of\_items(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_10\_T←RACE, TrickHLA::DEBUG\_SOURCE\_ATTRIBUTE, TrickHLA::ENCODING\_NO\_ENCODING, TrickHLA::ENCODING←\_OPAQUE\_DATA, FOM\_name, is\_byteswap(), locally\_owned, num\_items, publish, ref2, rti\_encoding, TrickHLA::DebugHandler::should\_print(), size, size\_is\_static, subscribe, and trick\_name.

Referenced by calculate\_number\_of\_items(), get\_attribute\_size(), initialize(), pack\_attribute\_buffer(), and unpack\_attribute\_buffer().

**7.1.3.4 calculate\_static\_number\_of\_items()**

```
void Attribute::calculate_static_number_of_items ( ) [private]
```

Calculates the number of items contained by the attribute.

If the attribute is not for an array then a value of one is set in the num\_items variable. Otherwise the number of items in the static array are set in the num\_items variable.

**Assumptions and Limitations:**

- Only static arrays are supported for now.

Definition at line 986 of file Attribute.cpp.

References num\_items, and ref2.

Referenced by calculate\_size\_and\_number\_of\_items(), and unpack\_attribute\_buffer().

**7.1.3.5 check\_data\_cycle\_ready()**

```
bool TrickHLA::Attribute::check_data_cycle_ready ( ) [inline]
```

Determine is the data cycle is ready for sending data.

**Returns**

True if the data cycle is ready for a send, false otherwise.

Definition at line 251 of file Attribute.hh.

References cycle\_cnt, and cycle\_ratio.

**7.1.3.6 decode\_boolean\_from\_buffer()**

```
void Attribute::decode_boolean_from_buffer ( ) const [private]
```

Decode a boolean attribute from the buffer using the HLAboolean data type which is encoded as a HLAinteger32BE.

Definition at line 1392 of file Attribute.cpp.

References buffer, num\_items, and ref2.

Referenced by unpack\_attribute\_buffer().

### 7.1.3.7 decode\_logical\_time()

```
void Attribute::decode_logical_time ( ) [private]
Decode the object attribute that is using the HLAlogicalTime 64-bit integer encoding.
Definition at line 1510 of file Attribute.cpp.
References buffer, FOM_name, TrickHLA::MAX_VALUE_IN_MICROS, TrickHLA::MICROS_MULTIPLIER, ref2, THL←
A_ENDL, and trick_name.
Referenced by unpack_attribute_buffer().
```

### 7.1.3.8 decode\_opaque\_data\_from\_buffer()

```
void Attribute::decode_opaque_data_from_buffer ( ) [private]
Decode the opaque data in the buffer.
Definition at line 1665 of file Attribute.cpp.
References buffer, decode_string_from_buffer(), FOM_name, TrickHLA::Utilities::get_endianness(), ref2, size, THL←
_ENDL, THLA_NEWLINE, and trick_name.
Referenced by unpack_attribute_buffer().
```

### 7.1.3.9 decode\_string\_from\_buffer()

```
void Attribute::decode_string_from_buffer ( ) [private]
Decode a string from the buffer into the attribute using the appropriate decoding.
Definition at line 2312 of file Attribute.cpp.
References buffer, TrickHLA::ENCODING_ASCII_STRING, TrickHLA::ENCODING_C_STRING, TrickHLA::ENCODI←
NG_NO_ENCODING, TrickHLA::ENCODING_OPAQUE_DATA, TrickHLA::ENCODING_UNICODE_STRING, FOM←
name, TrickHLA::Utilities::get_endianness(), num_items, ref2, rti_encoding, size, size_is_static, THLA_ENDL, THLA←
_NEWLINE, and trick_name.
Referenced by decode_opaque_data_from_buffer(), and unpack_attribute_buffer().
```

### 7.1.3.10 determine\_cycle\_ratio()

```
void Attribute::determine_cycle_ratio (
    double core_job_cycle_time )
```

Determine the cycle-ratio given the core job cycle rate and the cycle-time for this attribute.

#### Parameters

core_job_cycle_time	Core job cycle time in seconds.
---------------------	---------------------------------

Definition at line 540 of file Attribute.cpp.

```
References cycle_cnt, cycle_ratio, cycle_time, debug_handler, TrickHLA::DEBUG_LEVEL_9_TRACE, TrickHLA::DE←
BUG_SOURCE_ATTRIBUTE, FOM_name, TrickHLA::DebugHandler::should_print(), THLA_ENDL, and trick_name.
Referenced by TrickHLA::Object::set_core_job_cycle_time().
```

### 7.1.3.11 encode\_boolean\_to\_buffer()

```
void Attribute::encode_boolean_to_buffer ( ) [private]
Encode a boolean attribute into the buffer using the HLAboolean data type which is encoded as a HLAinteger32BE.
Definition at line 1364 of file Attribute.cpp.
References buffer, ensure_buffer_capacity(), HLAtrue, num_items, and ref2.
Referenced by pack_attribute_buffer().
```

### 7.1.3.12 encode\_logical\_time()

void Attribute::encode\_logical\_time ( ) const [private]

Encode the object attribute using the HLAlogicalTime 64-bit integer encoding.

Definition at line 1416 of file Attribute.cpp.

References buffer, FOM\_name, TrickHLA::MAX\_LOGICAL\_TIME\_SECONDS, TrickHLA::MAX\_VALUE\_IN\_MICROS, TrickHLA::MICROS\_MULTIPLIER, ref2, THLA\_ENDL, and trick\_name.

Referenced by pack\_attribute\_buffer().

### 7.1.3.13 encode\_opaque\_data\_to\_buffer()

void Attribute::encode\_opaque\_data\_to\_buffer ( ) [private]

Encode the data as HLA opaque data into the buffer.

Definition at line 1597 of file Attribute.cpp.

References buffer, encode\_string\_to\_buffer(), ensure\_buffer\_capacity(), TrickHLA::Utilities::get\_endianness(), ref2, and size.

Referenced by pack\_attribute\_buffer().

### 7.1.3.14 encode\_string\_to\_buffer()

void Attribute::encode\_string\_to\_buffer ( ) [private]

Encode a string attribute into the buffer using the appropriate encoding.

Definition at line 1784 of file Attribute.cpp.

References buffer, TrickHLA::ENCODING\_ASCII\_STRING, TrickHLA::ENCODING\_C\_STRING, TrickHLA::ENCODING\_G\_NO\_ENCODING, TrickHLA::ENCODING\_OPAQUE\_DATA, TrickHLA::ENCODING\_UNICODE\_STRING, ensure\_buffer\_capacity(), FOM\_name, TrickHLA::Utilities::get\_endianness(), num\_items, ref2, rti\_encoding, size, THLA\_ENDL, and trick\_name.

Referenced by encode\_opaque\_data\_to\_buffer(), and pack\_attribute\_buffer().

### 7.1.3.15 ensure\_buffer\_capacity()

void Attribute::ensure\_buffer\_capacity ( size\_t capacity ) [private]

Ensure the attribute buffer has at least the specified capacity.

#### Parameters

capacity	Desired capacity of the buffer in bytes.
----------	--

Definition at line 810 of file Attribute.cpp.

References buffer, buffer\_capacity, FOM\_name, THLA\_ENDL, and trick\_name.

Referenced by encode\_boolean\_to\_buffer(), encode\_opaque\_data\_to\_buffer(), encode\_string\_to\_buffer(), extract\_data(), initialize(), and pack\_attribute\_buffer().

### 7.1.3.16 extract\_data()

void Attribute::extract\_data ( RTI1516\_NAMESPACE::VariableLengthData \* attr\_value )

Extract the data out of the HLA [Attribute](#) Value.

**Parameters**

<i>attr_value</i>	The variable length data buffer containing the attribute value.
-------------------	---

Definition at line 625 of file Attribute.cpp.

References `buffer`, `debug_handler`, `TrickHLA::DEBUG_LEVEL_11_TRACE`, `TrickHLA::DEBUG_LEVEL_7_TRACE`, `TrickHLA::DEBUG_SOURCE_ATTRIBUTE`, `TrickHLA::ENCODING_BOOLEAN`, `TrickHLA::ENCODING_LOGICAL←TIME`, `TrickHLA::ENCODING_NO_ENCODING`, `TrickHLA::ENCODING_OPAQUE_DATA`, `TrickHLA::ENCODING_UNICODE_STRING`, `ensure_buffer_capacity()`, `FOM_name`, `get_attribute_size()`, `get_FOM_name()`, `get_trick_name()`, `mark_changed()`, `print_buffer()`, `ref2`, `rti_encoding`, `TrickHLA::DebugHandler::should_print()`, `size`, `size_is_static`, `THL←A_ENDL`, `THLA_NEWLINE`, and `trick_name`.

Referenced by `TrickHLA::Object::extract_data()`.

**7.1.3.17 `get_attribute_handle()`**

`RTI1516_NAMESPACE::AttributeHandle TrickHLA::Attribute::get_attribute_handle ( ) const [inline]`  
Get the RTI attribute handle.

**Returns**

The RTI AttributeHandle associate with this attribute.

Definition at line 278 of file Attribute.hh.

References `attr_handle`.

Referenced by `TrickHLA::Object::build_attribute_map()`, `TrickHLA::Object::create_attribute_set()`, `TrickHLA::Object::create_requested_attribute_set()`, `TrickHLA::Object::provide_attribute_update()`, `TrickHLA::Object::pull_ownership()`, `TrickHLA::Object::pull_ownership_upon_rejoin()`, and `TrickHLA::Object::push_ownership()`.

**7.1.3.18 `get_attribute_size()`**

`size_t Attribute::get_attribute_size ( )`  
Gets the attribute size in bytes.

**Returns**

The size in bytes of the attribute.

If the attribute is static in size it uses a cached size value otherwise the size is calculated.

Definition at line 953 of file Attribute.cpp.

References `calculate_size_and_number_of_items()`, `size`, and `size_is_static`.

Referenced by `extract_data()`, and `TrickHLAModel::SinePacking::pack()`.

**7.1.3.19 `get_attribute_value()`**

`VariableLengthData Attribute::get_attribute_value ( )`  
Gets the encoded attribute value.

**Returns**

The attribute value that contains the buffer of the encoded attribute.

Definition at line 614 of file Attribute.cpp.

References `buffer`, `TrickHLA::ENCODING_BOOLEAN`, `rti_encoding`, and `size`.

Referenced by `TrickHLA::Object::create_attribute_set()`, and `TrickHLA::Object::create_requested_attribute_set()`.

### 7.1.3.20 `get_conditional()`

```
Conditional* TrickHLA::Attribute::get_conditional ( ) const [inline]
Get the associated conditionality handler.
```

#### Returns

The conditionality handler class associated with this attribute.

Definition at line 305 of file Attribute.hh.

References conditional.

### 7.1.3.21 `get_configuration()`

```
const DataUpdateEnum TrickHLA::Attribute::get_configuration ( ) const [inline]
Get the reflection rate configuration type.
```

#### Returns

The reflection rate configuration type enumeration value.

Definition at line 118 of file Attribute.hh.

References config.

Referenced by TrickHLA::ExecutionConfigurationBase::reset\_ownership\_states().

### 7.1.3.22 `get_FOM_name()`

```
const char* TrickHLA::Attribute::get_FOM_name ( ) const [inline]
Get the Federation Object Model attribute name.
```

#### Returns

FOM name for the attribute.

Definition at line 163 of file Attribute.hh.

References FOM\_name.

Referenced by TrickHLA::FedAmb::attributeOwnershipAcquisitionNotification(), extract\_data(), TrickHLA::Object::grant\_pull\_request(), print\_buffer(), TrickHLA::OwnershipHandler::pull\_ownership(), TrickHLA::Object::pull\_ownership(), TrickHLA::OwnershipHandler::push\_ownership(), TrickHLA::Object::push\_ownership(), TrickHLA::Object::release\_ownership(), TrickHLA::FedAmb::requestAttributeOwnershipAssumption(), TrickHLA::FedAmb::requestAttributeOwnershipRelease(), TrickHLA::FedAmb::requestDivestitureConfirmation(), TrickHLA::Manager::setup\_object\_RTI\_handles(), SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes(), and TrickHLAModel::SineConditional::should\_send().

### 7.1.3.23 `get_preferred_order()`

```
TransportationEnum TrickHLA::Attribute::get_preferred_order ( ) const [inline]
Get the current preferred transportation order.
```

#### Returns

The preferred transportation order enumeration value.

Definition at line 266 of file Attribute.hh.

References preferred\_order.

### 7.1.3.24 `get_ref2_attributes()`

```
ATTRIBUTES TrickHLA::Attribute::get_ref2_attributes ( ) const [inline]  
Get the Trick "Ref Attributes" associated with this attribute.
```

#### Returns

A pointer to the Trick "Ref Attributes" class.

Definition at line 309 of file Attribute.hh.

References ref2.

### 7.1.3.25 `get_rti_encoding()`

```
EncodingEnum TrickHLA::Attribute::get_rti_encoding ( ) const [inline]  
Get the RTI encoding for this attribute.
```

#### Returns

The RTI encoding type enumeration value.

Definition at line 313 of file Attribute.hh.

References rti\_encoding.

### 7.1.3.26 `get_sim_variable_address()`

```
void* TrickHLA::Attribute::get_sim_variable_address ( ) [inline]  
Get the Trick simulation variable associated with this attribute.  
Definition at line 285 of file Attribute.hh.  
References ref2.  
Referenced by TrickHLAModel::SinePacking::pack(), and TrickHLAModel::SinePacking::unpack().
```

### 7.1.3.27 `get_trick_name()`

```
const char* TrickHLA::Attribute::get_trick_name ( ) const [inline]  
Get the associated Trick variable space name.
```

#### Returns

The Trick variable space name associated with this attribute.

Definition at line 167 of file Attribute.hh.

References trick\_name.

Referenced by extract\_data(), TrickHLA::Object::mark\_all\_attributes\_as\_nonlocal(), TrickHLA::Manager::setup\_←  
object\_ref\_attributes(), and TrickHLA::Object::setup\_preferred\_order\_with\_RTI().

### 7.1.3.28 `has_conditional()`

```
bool TrickHLA::Attribute::has_conditional ( ) const [inline]  
Check if attribute is sent conditionally.
```

#### Returns

True if attribute is to be sent conditionally.

Definition at line 301 of file Attribute.hh.

References conditional.

### 7.1.3.29 initialize()

```
void Attribute::initialize (
    const char * obj_FOM_name,
    const int object_index,
    const int attribute_index )
```

Initializes the [TrickHLA Attribute](#).

#### Parameters

<i>obj_FOM_name</i>	The FOM name of the parent object.
<i>object_index</i>	The array index to the parent <a href="#">Object</a> .
<i>attribute_index</i>	The array index to this <a href="#">Attribute</a> .

Definition at line 115 of file Attribute.cpp.

References buffer\_capacity, byteswap, calculate\_size\_and\_number\_of\_items(), config, TrickHLA::CONFIG\_MAX\_VA\_LUE, TrickHLA::CONFIG\_NONE, cycle\_time, debug\_handler, TrickHLA::DEBUG\_LEVEL\_9\_TRACE, TrickHLA::DEB\_UG\_SOURCE\_ATTRIBUTE, TrickHLA::ENCODING\_ASCII\_STRING, TrickHLA::ENCODING\_BIG\_ENDIAN, TrickHLA::ENCODING\_BOOLEAN, TrickHLA::ENCODING\_C\_STRING, TrickHLA::ENCODING\_FIRST\_VALUE, TrickHLA::ENCODING\_LAST\_VALUE, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_LOGICAL\_TIME, TrickHLA::ENCODING\_NO\_ENCODING, TrickHLA::ENCODING\_OPAQUE\_DATA, TrickHLA::ENCODING\_UNICODE\_STRING, TrickHLA::ENCODING\_UNKNOWN, ensure\_buffer\_capacity(), FOM\_name, initialized, is\_byteswap(), is\_static\_in\_size(), is\_supported\_attribute\_type(), TrickHLA::Utilities::is\_transmission\_byteswap(), locally\_owned, num\_items, preferred\_order, publish, ref2, rti\_encoding, TrickHLA::DebugHandler::should\_print(), size, size\_is\_static, subscribe, THLA\_ENDL, TrickHLA::TRANSPORT\_FIRST\_VALUE, TrickHLA::TRANSPORT\_LAST\_VALUE, trick\_name, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::Manager::setup\_object\_ref\_attributes(), and SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

### 7.1.3.30 is\_byteswap()

```
bool TrickHLA::Attribute::is_byteswap ( ) const [inline]
```

Determine if byteswapping is required.

#### Returns

True is byte swapping of the attribute date is required.

Definition at line 240 of file Attribute.hh.

References byteswap.

Referenced by calculate\_size\_and\_number\_of\_items(), initialize(), pack\_attribute\_buffer(), print\_buffer(), and unpack\_attribute\_buffer().

### 7.1.3.31 is\_changed()

```
bool TrickHLA::Attribute::is_changed ( ) const [inline]
```

Determine if an attribute value has changed.

#### Returns

True if the attribute value is marked as changed.

Definition at line 149 of file Attribute.hh.

References value\_changed.

### 7.1.3.32 `is_data_cycle_ready()`

```
bool TrickHLA::Attribute::is_data_cycle_ready ( ) const [inline]  
Determine is the data cycle is ready for sending data.
```

#### Returns

True if the data cycle is ready for a send, false otherwise.

Definition at line 244 of file Attribute.hh.

References `cycle_cnt`, and `cycle_ratio`.

### 7.1.3.33 `is_divest_requested()`

```
bool TrickHLA::Attribute::is_divest_requested ( ) const [inline]  
Determine if this federate is requesting to divest ownership of this attribute.
```

#### Returns

True if divest ownership is requested.

Definition at line 232 of file Attribute.hh.

References `divest_requested`.

### 7.1.3.34 `is_locally_owned()`

```
bool TrickHLA::Attribute::is_locally_owned ( ) const [inline]  
Determine if the attribute is locally owned.
```

#### Returns

True is attribute is locally owned.

Definition at line 187 of file Attribute.hh.

References `locally_owned`.

Referenced by `TrickHLA::FedAmb::attributeOwnershipAcquisitionNotification()`, `TrickHLA::OwnershipHandler::is_locally_owned()`, `TrickHLA::Object::mark_all_attributes_as_nonlocal()`, `TrickHLAModel::SinePacking::pack()`, `pack_attribute_buffer()`, `TrickHLA::Object::pull_ownership_upon_rejoin()`, `TrickHLA::Object::push_ownership()`, `TrickHLA::FedAmb::requestAttributeOwnershipAssumption()`, `TrickHLA::FedAmb::requestAttributeOwnershipRelease()`, `TrickHLA::FedAmb::requestDivestitureConfirmation()`, and `unpack_attribute_buffer()`.

### 7.1.3.35 `is_publish()`

```
bool TrickHLA::Attribute::is_publish ( ) const [inline]  
Determine if the attribute is published.
```

#### Returns

True if attribute is published.

Definition at line 171 of file Attribute.hh.

References `publish`.

Referenced by `TrickHLA::FedAmb::attributeOwnershipAcquisitionNotification()`, `TrickHLA::OwnershipHandler::is_published()`, `TrickHLAModel::SinePacking::pack()`, `TrickHLA::Object::pull_ownership()`, and `TrickHLA::FedAmb::requestAttributeOwnershipAssumption()`.

### 7.1.3.36 is\_pull\_requested()

```
bool TrickHLA::Attribute::is_pull_requested ( ) const [inline]  
Determine if someone is requesting an ownership transfer of this attribute.
```

#### Returns

True if an ownership pull is requested.

Definition at line 216 of file Attribute.hh.

References pull\_requested.

### 7.1.3.37 is\_push\_requested()

```
bool TrickHLA::Attribute::is_push_requested ( ) const [inline]  
Determine if this federate is trying to push ownership of this attribute.
```

#### Returns

True if an ownership push is requested.

Definition at line 224 of file Attribute.hh.

References push\_requested.

### 7.1.3.38 is\_received()

```
bool TrickHLA::Attribute::is_received ( ) const [inline]  
Determine if an attribute was received from another federate.
```

#### Returns

True if new attribute value has been received.

Definition at line 145 of file Attribute.hh.

References locally\_owned, and value\_changed.

Referenced by TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLAModel::SinePacking::unpack(), and SpaceFOM::RefFrameBase::unpack().

### 7.1.3.39 is\_remotely\_owned()

```
bool TrickHLA::Attribute::is_remotely_owned ( ) const [inline]  
Determine if the attribute is remotely owned.
```

#### Returns

True if attribute is remotely owned.

Definition at line 209 of file Attribute.hh.

Referenced by TrickHLA::FedAmb::attributeOwnershipAcquisitionNotification(), TrickHLA::OwnershipHandler::is\_remotely\_owned(), TrickHLA::Object::pull\_ownership(), TrickHLA::FedAmb::requestAttributeOwnershipAssumption(), TrickHLA::FedAmb::requestAttributeOwnershipRelease(), and TrickHLA::FedAmb::requestDivestitureConfirmation().

### 7.1.3.40 is\_static\_in\_size()

```
bool Attribute::is_static_in_size ( ) const  
Determines if the attribute is static in size.
```

**Returns**

True if attribute size is static.

Definition at line 961 of file Attribute.cpp.

References is\_supported\_attribute\_type(), and ref2.

Referenced by initialize().

**7.1.3.41 is\_subscribe()**

```
bool TrickHLA::Attribute::is_subscribe ( ) const [inline]
```

Determine if the attribute is subscribed.

**Returns**

True if attribute is subscribed.

Definition at line 179 of file Attribute.hh.

References subscribe.

Referenced by TrickHLA::OwnershipHandler::is\_subscribed().

**7.1.3.42 is\_supported\_attribute\_type()**

```
bool Attribute::is_supported_attribute_type ( ) const [private]
```

Determines if the HLA object attribute type is supported given the RTI encoding.

**Returns**

True if supported, false otherwise.

**Assumptions and Limitations:**

- Only primitive types and static arrays of primitive type are supported for now.

Definition at line 3331 of file Attribute.cpp.

References TrickHLA::ENCODING\_ASCII\_STRING, TrickHLA::ENCODING\_BIG\_ENDIAN, TrickHLA::ENCODING\_B←  
OOLEAN, TrickHLA::ENCODING\_C\_STRING, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_LO←  
GICAL\_TIME, TrickHLA::ENCODING\_NO\_ENCODING, TrickHLA::ENCODING\_OPAQUE\_DATA, TrickHLA::ENCOD←  
ING\_UNICODE\_STRING, TrickHLA::ENCODING\_UNKNOWN, ref2, and rti\_encoding.

Referenced by initialize(), and is\_static\_in\_size().

**7.1.3.43 is\_update\_requested()**

```
bool TrickHLA::Attribute::is_update_requested ( ) const [inline]
```

Determine if an update is requested.

**Returns**

True if an update is requested.

Definition at line 270 of file Attribute.hh.

References update\_requested.

#### 7.1.3.44 `mark_changed()`

```
void TrickHLA::Attribute::mark_changed ( ) [inline]
Mark the attribute value as changed.
Definition at line 152 of file Attribute.hh.
References value_changed.
Referenced by extract_data().
```

#### 7.1.3.45 `mark_locally_owned()`

```
void TrickHLA::Attribute::mark_locally_owned ( ) [inline]
Mark the attribute is locally owned flag.
Definition at line 190 of file Attribute.hh.
References cycle_cnt, and locally_owned.
Referenced by TrickHLA::FedAmb::attributeOwnershipAcquisitionNotification(), TrickHLA::ExecutionConfigurationBase::reset_ownership_states(), and TrickHLA::ExecutionConfigurationBase::set_master().
```

#### 7.1.3.46 `mark_remotely_owned()`

```
void TrickHLA::Attribute::mark_remotely_owned ( ) [inline]
Mark the attribute as remotely owned.
Definition at line 212 of file Attribute.hh.
References unmark_locally_owned().
Referenced by TrickHLA::Object::grant_pull_request(), TrickHLA::Object::release_ownership(), and TrickHLA::ExecutionConfigurationBase::set_master().
```

#### 7.1.3.47 `mark_unchanged()`

```
void TrickHLA::Attribute::mark_unchanged ( ) [inline]
Mark the attribute value as unchanged.
Definition at line 155 of file Attribute.hh.
References value_changed.
Referenced by TrickHLA::Object::mark_unchanged().
```

#### 7.1.3.48 `operator=()`

```
Attribute& TrickHLA::Attribute::operator=
    const Attribute & rhs ) [private]
Assignment operator for Attribute class.
This assignment operator is private to prevent inadvertent copies.
```

#### 7.1.3.49 `pack_attribute_buffer()`

```
void Attribute::pack_attribute_buffer ( )
Pack the attribute into the buffer using the appropriate encoding.
Definition at line 1002 of file Attribute.cpp.
References buffer, buffer_capacity, byteswap_buffer_copy(), calculate_size_and_number_of_items(), debug_handler,
TrickHLA::DEBUG_LEVEL_10_TRACE, TrickHLA::DEBUG_LEVEL_11_TRACE, TrickHLA::DEBUG_SOURCE_ATTRIBUTE,
encode_boolean_to_buffer(), encode_logical_time(), encode_opaque_data_to_buffer(), encode_string_to_buffer(),
TrickHLA::ENCODING_BOOLEAN, TrickHLA::ENCODING_LOGICAL_TIME, TrickHLA::ENCODING_OPAQUE_DATA,
ensure_buffer_capacity(), FOM_name, is_byteswap(), is_locally_owned(), locally_owned, num_items,
```

print\_buffer(), publish, ref2, rti\_encoding, TrickHLA::DebugHandler::should\_print(), size, size\_is\_static, subscribe, and trick\_name.

Referenced by TrickHLA::Object::pack\_attribute\_buffers(), and TrickHLA::Object::pack\_requested\_attribute\_buffers().

### 7.1.3.50 print\_buffer()

```
void Attribute::print_buffer ( ) const
```

Prints the contents of buffer used to encode/decode the attribute to the console on standard out.

Definition at line 3401 of file Attribute.cpp.

References buffer, TrickHLA::Utilities::byteswap\_double(), get\_FOM\_name(), is\_byteswap(), num\_items, ref2, and size.

Referenced by extract\_data(), TrickHLAModel::SinePacking::pack(), pack\_attribute\_buffer(), TrickHLAModel::SinePacking::unpack(), and unpack\_attribute\_buffer().

### 7.1.3.51 set\_attribute\_handle()

```
void TrickHLA::Attribute::set_attribute_handle (
    RTI1516_NAMESPACE::AttributeHandle id ) [inline]
```

Set the RTI attribute handle.

#### Parameters

<i>id</i>	The RTI attribute handle associated with this attribute.
-----------	--

Definition at line 282 of file Attribute.hh.

References attr\_handle.

Referenced by TrickHLA::Manager::setup\_object\_RTI\_handles().

### 7.1.3.52 set\_configuration()

```
void TrickHLA::Attribute::set_configuration (
    const DataUpdateEnum c ) [inline]
```

Set the reflection rate configuration type.

Definition at line 121 of file Attribute.hh.

Referenced by TrickHLA::ExecutionConfigurationBase::reset\_ownership\_states().

### 7.1.3.53 set\_debug\_level()

```
void TrickHLA::Attribute::set_debug_level (
    DebugHandler hndlr ) [inline]
```

Set the debug handler settings for this attribute.

#### Parameters

<i>hndlr</i>	DebugHandler from which to copy settings.
--------------	---

Definition at line 159 of file Attribute.hh.

References debug\_handler, and TrickHLA::DebugHandler::set().

Referenced by TrickHLA::Manager::setup\_object\_ref\_attributes(), and SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

### 7.1.3.54 `set_divest_requested()`

```
void TrickHLA::Attribute::set_divest_requested (
    bool enable )  [inline]
```

Set the ownership divest requested flag.

#### Parameters

<i>enable</i>	Flag to set divest requested state.
---------------	-------------------------------------

Definition at line 236 of file Attribute.hh.

References `divest_requested`.

Referenced by `TrickHLA::Object::release_ownership()`, and `TrickHLA::FedAmb::requestDivestitureConfirmation()`.

### 7.1.3.55 `set_encoding()`

```
void TrickHLA::Attribute::set_encoding (
    EncodingEnum in_type )  [inline]
```

Definition at line 317 of file Attribute.hh.

References `byteswap`, `TrickHLA::Utilities::is_transmission_byteswap()`, and `rti_encoding`.

### 7.1.3.56 `set_preferred_order()`

```
void TrickHLA::Attribute::set_preferred_order (
    TransportationEnum order )  [inline]
```

Set the preferred transportation order.

#### Parameters

<i>order</i>	The transportation type enumeration value.
--------------	--

Definition at line 262 of file Attribute.hh.

Referenced by `TrickHLA::ExecutionConfigurationBase::reset_preferred_order()`.

### 7.1.3.57 `set_publish()`

```
void TrickHLA::Attribute::set_publish (
    bool enable )  [inline]
```

Set the attribute publish flag.

#### Parameters

<i>enable</i>	Flag to set the publish state.
---------------	--------------------------------

Definition at line 175 of file Attribute.hh.

Referenced by `TrickHLA::ExecutionConfigurationBase::set_master()`, and `TrickHLA::Object::stop_publishing_attributes()`.

### 7.1.3.58 set\_pull\_requested()

```
void TrickHLA::Attribute::set_pull_requested (
    bool enable ) [inline]
```

Set the pull requested flag.

#### Parameters

<i>enable</i>	Flag to set the pull requested state.
---------------	---------------------------------------

Definition at line 220 of file Attribute.hh.

References pull\_requested.

Referenced by TrickHLA::Object::grant\_pull\_request(), and TrickHLA::FedAmb::requestAttributeOwnershipRelease().

### 7.1.3.59 set\_push\_requested()

```
void TrickHLA::Attribute::set_push_requested (
    bool enable ) [inline]
```

Set the ownership push flag.

#### Parameters

<i>enable</i>	Flag to set the push requested state.
---------------	---------------------------------------

Definition at line 228 of file Attribute.hh.

References push\_requested.

Referenced by TrickHLA::Object::grant\_push\_request(), and TrickHLA::FedAmb::requestAttributeOwnershipAssumption().

### 7.1.3.60 set\_subscribe()

```
void TrickHLA::Attribute::set_subscribe (
    bool enable ) [inline]
```

Set the attribute is subscribed flag.

#### Parameters

<i>enable</i>	Flag to set the subscribe state.
---------------	----------------------------------

Definition at line 183 of file Attribute.hh.

Referenced by TrickHLA::ExecutionConfigurationBase::set\_master(), and TrickHLA::Object::stop\_subscribing\_attributes().

### 7.1.3.61 set\_update\_requested()

```
void TrickHLA::Attribute::set_update_requested (
    bool request_update ) [inline]
```

Set the attribute update requested flag.

#### Parameters

<i>request_update</i>	Request update flag.
-----------------------	----------------------

Definition at line 274 of file Attribute.hh.

References update\_requested.

Referenced by TrickHLA::Object::create\_attribute\_set(), TrickHLA::Object::create\_requested\_attribute\_set(), and TrickHLA::Object::provide\_attribute\_update().

### 7.1.3.62 `unmark_locally_owned()`

```
void TrickHLA::Attribute::unmark_locally_owned ( ) [inline]
```

Mark the attribute is NOT locally owned flag.

Definition at line 200 of file Attribute.hh.

References cycle\_cnt.

Referenced by TrickHLA::Object::mark\_all\_attributes\_as\_nonlocal(), mark\_remotely\_owned(), and TrickHLA::Object::pull\_ownership\_upon\_rejoin().

### 7.1.3.63 `unpack_attribute_buffer()`

```
void Attribute::unpack_attribute_buffer ( )
```

Unpack the attribute from the buffer into the trick-variable using the appropriate decoding.

Definition at line 1201 of file Attribute.cpp.

References buffer, buffer\_capacity, byteswap\_buffer\_copy(), calculate\_size\_and\_number\_of\_items(), calculate\_static\_number\_of\_items(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_10\_TRACE, TrickHLA::DEBUG\_LEVEL\_11\_TRACE, TrickHLA::DEBUG\_SOURCE\_ATTRIBUTE, decode\_boolean\_from\_buffer(), decode\_logical\_time(), decode\_opaque\_data\_from\_buffer(), decode\_string\_from\_buffer(), TrickHLA::ENCODING\_BOOLEAN, TrickHLA::ENCODING\_LOGICAL\_TIME, TrickHLA::ENCODING\_OPAQUE\_DATA, FOM\_name, is\_byteswap(), is\_locally\_owned(), locally\_owned, num\_items, print\_buffer(), publish, ref2, rti\_encoding, TrickHLA::DebugHandler::should\_print(), size, size\_is\_static, subscribe, and trick\_name.

Referenced by TrickHLA::Object::unpack\_attribute\_buffers().

## 7.1.4 Friends And Related Function Documentation

### 7.1.4.1 `init_attrTrickHLA__Attribute`

```
void init_attrTrickHLA__Attribute ( ) [friend]
```

### 7.1.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 65 of file Attribute.hh.

## 7.1.5 Field Documentation

### 7.1.5.1 `attr_handle`

```
RTI1516_NAMESPACE::AttributeHandle TrickHLA::Attribute::attr_handle [private]
```

**Data I/O:** \*\*

The RTI attribute handle.

Definition at line 422 of file Attribute.hh.

Referenced by get\_attribute\_handle(), and set\_attribute\_handle().

### 7.1.5.2 buffer

```
unsigned char* TrickHLA::Attribute::buffer [private]
```

**Units:** –

Byte buffer for the attribute value bytes.

Definition at line 401 of file Attribute.hh.

Referenced by decode\_boolean\_from\_buffer(), decode\_logical\_time(), decode\_opaque\_data\_from\_buffer(), decode\_string\_from\_buffer(), encode\_boolean\_to\_buffer(), encode\_logical\_time(), encode\_opaque\_data\_to\_buffer(), encode\_string\_to\_buffer(), ensure\_buffer\_capacity(), extract\_data(), get\_attribute\_value(), pack\_attribute\_buffer(), print\_buffer(), unpack\_attribute\_buffer(), and ~Attribute().

### 7.1.5.3 buffer\_capacity

```
size_t TrickHLA::Attribute::buffer_capacity [private]
```

**Units:** *count*

The capacity of the buffer.

Definition at line 402 of file Attribute.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), ensure\_buffer\_capacity(), initialize(), pack\_attribute\_buffer(), unpack\_attribute\_buffer(), and ~Attribute().

### 7.1.5.4 byteswap

```
bool TrickHLA::Attribute::byteswap [private]
```

**Units:** –

Flag to indicate byte-swap before RTI Rx/Tx.

Definition at line 415 of file Attribute.hh.

Referenced by initialize(), is\_byteswap(), and set\_encoding().

### 7.1.5.5 conditional

```
Conditional* TrickHLA::Attribute::conditional
```

**Units:** –

Handler for a conditional attribute

Definition at line 91 of file Attribute.hh.

Referenced by get\_conditional(), and has\_conditional().

### 7.1.5.6 config

```
DataUpdateEnum TrickHLA::Attribute::config
```

**Units:** –

The attribute configuration.

Definition at line 77 of file Attribute.hh.

Referenced by get\_configuration(), initialize(), and SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

### 7.1.5.7 cycle\_cnt

```
int TrickHLA::Attribute::cycle_cnt [private]
```

**Units:** *count*

Internal cycle counter used to determine when cyclic data will be sent.

Definition at line 418 of file Attribute.hh.

Referenced by `check_data_cycle_ready()`, `determine_cycle_ratio()`, `is_data_cycle_ready()`, `mark_locally_owned()`, and `unmark_locally_owned()`.

**7.1.5.8 cycle\_ratio**

```
int TrickHLA::Attribute::cycle_ratio [private]
```

**Units:** –

Ratio of the attribute cycle-time to the `send_cyclic_data` job cycle time.

Definition at line 417 of file Attribute.hh.

Referenced by `check_data_cycle_ready()`, `determine_cycle_ratio()`, and `is_data_cycle_ready()`.

**7.1.5.9 cycle\_time**

```
double TrickHLA::Attribute::cycle_time
```

**Units:** *s*

Send the cyclic attribute at the specified rate.

Definition at line 88 of file Attribute.hh.

Referenced by `determine_cycle_ratio()`, and `initialize()`.

**7.1.5.10 debug\_handler**

```
DebugHandler TrickHLA::Attribute::debug_handler [private]
```

**Units:** –

Prints out multiple debug levels

Definition at line 430 of file Attribute.hh.

Referenced by `calculate_size_and_number_of_items()`, `determine_cycle_ratio()`, `extract_data()`, `initialize()`, `pack_attribute_buffer()`, `set_debug_level()`, and `unpack_attribute_buffer()`.

**7.1.5.11 divest\_requested**

```
bool TrickHLA::Attribute::divest_requested [private]
```

**Units:** –

Are we releasing ownership?

Definition at line 426 of file Attribute.hh.

Referenced by `is_divest_requested()`, and `set_divest_requested()`.

**7.1.5.12 FOM\_name**

```
char* TrickHLA::Attribute::FOM_name
```

**Units:** –

FOM name for the attribute.

Definition at line 75 of file Attribute.hh.

Referenced by `calculate_size_and_number_of_items()`, `TrickHLA::ExecutionConfiguration::configure_attributes()`, `DSES::ExecutionConfiguration::configure_attributes()`, `DIS::ExecutionConfiguration::configure_attributes()`, `IMSim::ExecutionConfiguration::configure_attributes()`, `SpaceFOM::ExecutionConfiguration::configure_attributes()`, `decode_logical_time()`, `decode_opaque_data_from_buffer()`, `decode_string_from_buffer()`, `determine_cycle_ratio()`, `encode`

`_logical_time()`, `encode_string_to_buffer()`, `ensure_buffer_capacity()`, `extract_data()`, `get_FOM_name()`, `initialize()`, `pack_attribute_buffer()`, `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`, and `unpack_attribute_buffer()`.

### 7.1.5.13 HLAtrue

```
unsigned int TrickHLA::Attribute::HLAtrue [private]
```

**Units:** –

A 32-bit integer with a value of 1 on a Big Endian computer.

Definition at line 413 of file Attribute.hh.

Referenced by `Attribute()`, and `encode_boolean_to_buffer()`.

### 7.1.5.14 initialized

```
bool TrickHLA::Attribute::initialized [private]
```

**Units:** –

Has this attribute been initialized?

Definition at line 428 of file Attribute.hh.

Referenced by `initialize()`.

### 7.1.5.15 locally\_owned

```
bool TrickHLA::Attribute::locally_owned
```

**Units:** –

Flag to indicate attribute is locally owned.

Definition at line 84 of file Attribute.hh.

Referenced by `calculate_size_and_number_of_items()`, `initialize()`, `is_locally_owned()`, `is_received()`, `mark_locally_owned()`, `pack_attribute_buffer()`, `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`, and `unpack_attribute_buffer()`.

### 7.1.5.16 num\_items

```
size_t TrickHLA::Attribute::num_items [private]
```

**Units:** *count*

Number of attribute items, length of the array.

Definition at line 407 of file Attribute.hh.

Referenced by `calculate_number_of_items()`, `calculate_size_and_number_of_items()`, `calculate_static_number_of_items()`, `decode_boolean_from_buffer()`, `decode_string_from_buffer()`, `encode_boolean_to_buffer()`, `encode_string_to_buffer()`, `initialize()`, `pack_attribute_buffer()`, `print_buffer()`, and `unpack_attribute_buffer()`.

### 7.1.5.17 preferred\_order

```
TransportationEnum TrickHLA::Attribute::preferred_order
```

**Units:** –

Either `Timestamp` (default) or `Receive Order`.

Definition at line 79 of file Attribute.hh.

Referenced by `get_preferred_order()`, and `initialize()`.

### 7.1.5.18 publish

```
bool TrickHLA::Attribute::publish
```

**Units:** –

True to publish attribute that is owned locally.

Definition at line 81 of file Attribute.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), initialize(), is\_publish(), pack\_attribute\_buffer(), SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes(), and unpack\_attribute\_buffer().

### 7.1.5.19 pull\_requested

```
bool TrickHLA::Attribute::pull_requested [private]
```

**Units:** –

Has someone asked to own us?

Definition at line 424 of file Attribute.hh.

Referenced by is\_pull\_requested(), and set\_pull\_requested().

### 7.1.5.20 push\_requested

```
bool TrickHLA::Attribute::push_requested [private]
```

**Units:** –

Is someone giving up ownership?

Definition at line 425 of file Attribute.hh.

Referenced by is\_push\_requested(), and set\_push\_requested().

### 7.1.5.21 ref2

```
REF2* TrickHLA::Attribute::ref2 [private]
```

**Data I/O:** \*\*

The ref\_attributes of the given trick\_name.

Definition at line 420 of file Attribute.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), calculate\_static\_number\_of\_items(), decode\_boolean\_from\_buffer(), decode\_logical\_time(), decode\_opaque\_data\_from\_buffer(), decode\_string\_from\_buffer(), encode\_boolean\_to\_buffer(), encode\_logical\_time(), encode\_opaque\_data\_to\_buffer(), encode\_string\_to\_buffer(), extract\_data(), get\_ref2\_attributes(), get\_sim\_variable\_address(), initialize(), is\_static\_in\_size(), is\_supported\_attribute\_type(), pack\_attribute\_buffer(), print\_buffer(), unpack\_attribute\_buffer(), and ~Attribute().

### 7.1.5.22 rti\_encoding

```
EncodingEnum TrickHLA::Attribute::rti_encoding
```

**Units:** –

RTI encoding of the data.

Definition at line 86 of file Attribute.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), TrickHLA::ExecutionConfiguration::configure\_attributes(), D::ExecutionConfiguration::configure\_attributes(), DSES::ExecutionConfiguration::configure\_attributes(), IMSim::ExecutionConfiguration::configure\_attributes(), SpaceFOM::ExecutionConfiguration::configure\_attributes(), decode\_string\_from\_buffer(), encode\_string\_to\_buffer(), extract\_data(), get\_attribute\_value(), get\_rti\_encoding(), initialize(), is\_supported\_attribute\_type(), pack\_attribute\_buffer(), set\_encoding(), SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes(), and unpack\_attribute\_buffer().

### 7.1.5.23 size

```
size_t TrickHLA::Attribute::size [private]
```

**Units:** count

The size of the attribute in bytes.

Definition at line 406 of file Attribute.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), decode\_opaque\_data\_from\_buffer(), decode\_string\_from\_buffer(), encode\_opaque\_data\_to\_buffer(), encode\_string\_to\_buffer(), extract\_data(), get\_attribute\_size(), get\_attribute\_value(), initialize(), pack\_attribute\_buffer(), print\_buffer(), and unpack\_attribute\_buffer().

### 7.1.5.24 size\_is\_static

```
bool TrickHLA::Attribute::size_is_static [private]
```

**Units:** –

Flag to indicate the size of this attribute is static.

Definition at line 404 of file Attribute.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), decode\_string\_from\_buffer(), extract\_data(), get\_attribute\_size(), initialize(), pack\_attribute\_buffer(), and unpack\_attribute\_buffer().

### 7.1.5.25 subscribe

```
bool TrickHLA::Attribute::subscribe
```

**Units:** –

True to subscribe to attribute.

Definition at line 82 of file Attribute.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), initialize(), is\_subscribe(), pack\_attribute\_buffer(), SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes(), and unpack\_attribute\_buffer().

### 7.1.5.26 trick\_name

```
char* TrickHLA::Attribute::trick_name
```

**Units:** –

Trick name for the attribute.

Definition at line 74 of file Attribute.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), TrickHLA::ExecutionConfiguration::configure\_attributes(), DLS::ExecutionConfiguration::configure\_attributes(), DSES::ExecutionConfiguration::configure\_attributes(), IMSim::ExecutionConfiguration::configure\_attributes(), SpaceFOM::ExecutionConfiguration::configure\_attributes(), decode\_logical\_time(), decode\_opaque\_data\_from\_buffer(), decode\_string\_from\_buffer(), determine\_cycle\_ratio(), encode\_logical\_time(), encode\_string\_to\_buffer(), ensure\_buffer\_capacity(), extract\_data(), get\_trick\_name(), initialize(), pack\_attribute\_buffer(), and unpack\_attribute\_buffer().

### 7.1.5.27 update\_requested

```
bool TrickHLA::Attribute::update_requested [private]
```

**Units:** –

Flag to indicate another federate has requested an attribute update.

Definition at line 411 of file Attribute.hh.

Referenced by is\_update\_requested(), and set\_update\_requested().

### 7.1.5.28 `value_changed`

```
bool TrickHLA::Attribute::value_changed [private]
```

**Units:** –

Flag to indicate the attribute value changed.

Definition at line 409 of file Attribute.hh.

Referenced by `is_changed()`, `is_received()`, `mark_changed()`, and `mark_unchanged()`.

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

- [Attribute.hh](#)
- [Attribute.cpp](#)

## 7.2 TrickHLA::BasicClock Class Reference

```
#include <BasicClock.hh>
```

### Public Member Functions

- [BasicClock \(\)](#)  
*Default constructor for the `TrickHLA BasicClock` class.*
- [virtual ~BasicClock \(\)](#)  
*Destructor for the `TrickHLA BasicClock` class.*

### Static Public Member Functions

- [static double get\\_time \(\)](#)  
*Get the computer system time in seconds.*

### Private Attributes

- [bool first\\_pass](#)  
*Data I/O: \*\**  
*Flag indicates first pass to determine external clock.*

### Friends

- [class InputProcessor](#)
- [void init\\_attrTrickHLA\\_BasicClock \(\)](#)

### 7.2.1 Detailed Description

Definition at line 44 of file BasicClock.hh.

### 7.2.2 Constructor & Destructor Documentation

#### 7.2.2.1 `BasicClock()`

```
TrickHLA::BasicClock::BasicClock ( ) [inline]
```

Default constructor for the `TrickHLA BasicClock` class.

Definition at line 59 of file BasicClock.hh.

### 7.2.2.2 ~BasicClock()

```
virtual TrickHLA::BasicClock::~BasicClock ( ) [inline], [virtual]
```

Destructor for the [TrickHLA BasicClock](#) class.

Definition at line 61 of file BasicClock.hh.

## 7.2.3 Member Function Documentation

### 7.2.3.1 get\_time()

```
static double TrickHLA::BasicClock::get_time ( ) [inline], [static]
```

Get the computer system time in seconds.

Returns

Computer system time in seconds.

Definition at line 68 of file BasicClock.hh.

Referenced by [TrickHLA::Object::receive\\_cyclic\\_data\(\)](#).

## 7.2.4 Friends And Related Function Documentation

### 7.2.4.1 init\_attrTrickHLA\_\_BasicClock

```
void init_attrTrickHLA__BasicClock ( ) [friend]
```

### 7.2.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 51 of file BasicClock.hh.

## 7.2.5 Field Documentation

### 7.2.5.1 first\_pass

```
bool TrickHLA::BasicClock::first_pass [private]
```

**Data I/O: \*\***

Flag indicates first pass to determine external clock.

Definition at line 75 of file BasicClock.hh.

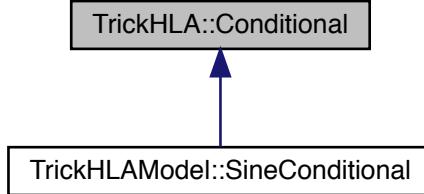
The documentation for this class was generated from the following file:

- [BasicClock.hh](#)

## 7.3 TrickHLA::Conditional Class Reference

```
#include <Conditional.hh>
```

Inheritance diagram for TrickHLA::Conditional:



## Public Member Functions

- [Conditional \(\)](#)  
*Default constructor for the `TrickHLA Conditional` class.*
- [virtual ~Conditional \(\)](#)  
*Destructor for the `TrickHLA Conditional` class.*
- [virtual bool should\\_send \(Attribute \\*attr\)](#)  
*Default implementation of a virtual function, returning true.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_\\_Conditional \(\)](#)

### 7.3.1 Detailed Description

Definition at line 44 of file `Conditional.hh`.

### 7.3.2 Constructor & Destructor Documentation

#### 7.3.2.1 [Conditional\(\)](#)

`Conditional::Conditional ( )`  
Default constructor for the `TrickHLA Conditional` class.  
**Trick Job Class:** *initialization*  
Definition at line 47 of file `Conditional.cpp`.

#### 7.3.2.2 [~Conditional\(\)](#)

`Conditional::~Conditional ( ) [virtual]`  
Destructor for the `TrickHLA Conditional` class.  
**Trick Job Class:** *shutdown*  
Definition at line 54 of file `Conditional.cpp`.

### 7.3.3 Member Function Documentation

#### 7.3.3.1 `should_send()`

```
bool Conditional::should_send (
    Attribute * attr ) [virtual]
```

Default implementation of a virtual function, returning true.

**Returns**

Always returns boolean true.

**Parameters**

<code>attr</code>	Pointer to attribute.
-------------------	-----------------------

Reimplemented in [TrickHLAModel::SineConditional](#).

Definition at line 58 of file Conditional.cpp.

### 7.3.4 Friends And Related Function Documentation

#### 7.3.4.1 `init_attrTrickHLA__Conditional`

```
void init_attrTrickHLA__Conditional ( ) [friend]
```

#### 7.3.4.2 `InputProcessor`

friend class InputProcessor [friend]

Definition at line 51 of file Conditional.hh.

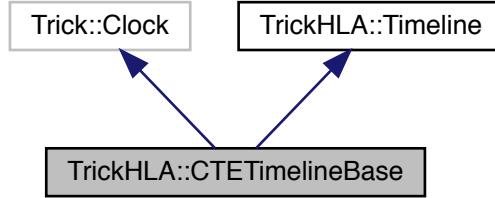
The documentation for this class was generated from the following files:

- [Conditional.hh](#)
- [Conditional.cpp](#)

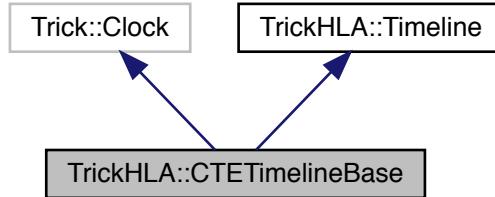
## 7.4 TrickHLA::CTETimelineBase Class Reference

```
#include <CTETimelineBase.hh>
```

Inheritance diagram for TrickHLA::CTETimelineBase:



Collaboration diagram for TrickHLA::CTETimelineBase:



## Public Member Functions

- **CTETimelineBase ()**  
*Default constructor for the [TrickHLA CTETimelineBase](#) class.*
- **virtual ~CTETimelineBase ()**  
*Destructor for the [TrickHLA CTETimelineBase](#) class.*
- **virtual double get\_time ()**  
*Get the current CTE time.*
- **virtual int clock\_init ()**  
*Initialize the [Trick::Clock](#) functions.*
- **virtual long long wall\_clock\_time ()**  
*Get the wall clock time.*
- **virtual int clock\_stop ()**  
*Stop the CTE clock.*
- **virtual void set\_clock\_ID (clockid\_t id)**  
*Sets the clock ID (system clock type).*
- **virtual clockid\_t get\_clock\_ID ()**  
*Gets the current clock ID (system clock type).*

## Protected Attributes

- `clockid_t clk_id`

**Data I/O:** \*\*

*System clock type used.*

## Private Member Functions

- `CTETimelineBase (const CTETimelineBase &rhs)`  
*Copy constructor for `CTETimelineBase` class.*
- `CTETimelineBase & operator= (const CTETimelineBase &rhs)`  
*Assignment operator for `CTETimelineBase` class.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA__CTETimelineBase ()`

### 7.4.1 Detailed Description

Definition at line 58 of file `CTETimelineBase.hh`.

### 7.4.2 Constructor & Destructor Documentation

#### 7.4.2.1 `CTETimelineBase()` [1/2]

`CTETimelineBase::CTETimelineBase ( )`

Default constructor for the `TrickHLA CTETimelineBase` class.

**Trick Job Class:** *initialization*

Definition at line 54 of file `CTETimelineBase.cpp`.

#### 7.4.2.2 `~CTETimelineBase()`

`CTETimelineBase::~CTETimelineBase ( ) [virtual]`

Destructor for the `TrickHLA CTETimelineBase` class.

**Trick Job Class:** *shutdown*

Definition at line 64 of file `CTETimelineBase.cpp`.

#### 7.4.2.3 `CTETimelineBase()` [2/2]

`TrickHLA::CTETimelineBase::CTETimelineBase (`  
`const CTETimelineBase & rhs ) [private]`

Copy constructor for `CTETimelineBase` class.

This constructor is private to prevent inadvertent copies.

### 7.4.3 Member Function Documentation

#### 7.4.3.1 `clock_init()`

```
int CTETimelineBase::clock_init ( ) [virtual]
```

Initialize the Trick::Clock functions.

Set the global "the\_clock" pointer to this instance.

Definition at line 72 of file CTETimelineBase.cpp.

Referenced by TrickHLA::ExecutionControlBase::initialize().

#### 7.4.3.2 `clock_stop()`

```
int CTETimelineBase::clock_stop ( ) [virtual]
```

Stop the CTE clock.

##### Returns

Default implementation always returns 0.

This function is empty

Definition at line 101 of file CTETimelineBase.cpp.

#### 7.4.3.3 `get_clock_ID()`

```
clockid_t CTETimelineBase::get_clock_ID ( ) [virtual]
```

Gets the current clock ID (system clock type).

##### Returns

The system clock type in use.

Definition at line 124 of file CTETimelineBase.cpp.

References clk\_id.

#### 7.4.3.4 `get_time()`

```
double CTETimelineBase::get_time ( ) [virtual]
```

Get the current CTE time.

##### Returns

Current time of day in seconds.

Get the global time base on the CTE.

Implements [TrickHLA::Timeline](#).

Definition at line 81 of file CTETimelineBase.cpp.

Referenced by TrickHLA::ExecutionControlBase::get\_cte\_time(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init←\_process(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process←\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::Execution←Control::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DS←ES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition←\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), and IMSim::ExecutionControl::process←\_mode\_transition\_request().

**7.4.3.5 operator=( )**

```
CTETimelineBase& TrickHLA::CTETimelineBase::operator= (
    const CTETimelineBase & rhs ) [private]
```

Assignment operator for `CTETimelineBase` class.

This assignment operator is private to prevent inadvertent copies.

**7.4.3.6 set\_clock\_ID()**

```
void CTETimelineBase::set_clock_ID (
    clockid_t id ) [virtual]
```

Sets the clock ID (system clock type).

Definition at line 106 of file `CTETimelineBase.cpp`.

**7.4.3.7 wall\_clock\_time()**

```
long long CTETimelineBase::wall_clock_time ( ) [virtual]
```

Get the wall clock time.

Returns

The current real time as a count of microseconds.

Call the system `clock_gettime` to get the current real time.

Definition at line 91 of file `CTETimelineBase.cpp`.

References `clk_id`.

**7.4.4 Friends And Related Function Documentation****7.4.4.1 init\_attrTrickHLA\_\_CTETimelineBase**

```
void init_attrTrickHLA__CTETimelineBase ( ) [friend]
```

**7.4.4.2 InputProcessor**

```
friend class InputProcessor [friend]
```

Definition at line 65 of file `CTETimelineBase.hh`.

**7.4.5 Field Documentation****7.4.5.1 clk\_id**

```
clockid_t TrickHLA::CTETimelineBase::clk_id [protected]
```

**Data I/O: \*\***

System clock type used.

The default clock ID is `CLOCK_REALTIME`.

Definition at line 106 of file `CTETimelineBase.hh`.

Referenced by `get_clock_ID()`, and `wall_clock_time()`.

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

- [CTETimelineBase.hh](#)
- [CTETimelineBase.cpp](#)

## 7.5 TrickHLA::DebugHandler Class Reference

```
#include <DebugHandler.hh>
```

### Public Member Functions

- `DebugHandler ()`  
*Initialization constructor for the `TrickHLA DebugHandler` class.*
- `virtual ~DebugHandler ()`  
*Destructor for the `TrickHLA DebugHandler` class.*
- `const DebugLevelEnum & get_debug_level () const`  
*Get current debug level as an enumeration value.*
- `const int get_debug_level_as_int () const`  
*Get the current debug level as an integer value.*
- `const DebugSourceEnum & get_code_section () const`  
*Get the code section for this debug handler.*
- `bool should_print (const DebugLevelEnum &level, const DebugSourceEnum &code) const`  
*Conditional test to see if a debug message should print.*
- `void set (const DebugHandler &in)`  
*Set the debug handler from existing debug handler.*

### Data Fields

- `DebugLevelEnum debug_level`

**Units:** –  
*Maximum debug report level requested by the user, default: `THLA_NO_TRACE`*
- `DebugSourceEnum code_section`

**Units:** –  
*Code section(s) for which to activate debug messages, default: `THLA_ALL_MODULES`*

### Friends

- `class InputProcessor`
- `void init_attrTrickHLA__DebugHandler ()`

#### 7.5.1 Detailed Description

Definition at line 46 of file `DebugHandler.hh`.

#### 7.5.2 Constructor & Destructor Documentation

##### 7.5.2.1 DebugHandler()

```
TrickHLA::DebugHandler::DebugHandler ( ) [inline]
Initialization constructor for the TrickHLA DebugHandler class.
Definition at line 63 of file DebugHandler.hh.
```

### 7.5.2.2 ~DebugHandler()

```
virtual TrickHLA::DebugHandler::~DebugHandler ( ) [inline], [virtual]
```

Destructor for the [TrickHLA DebugHandler](#) class.

Definition at line 68 of file DebugHandler.hh.

## 7.5.3 Member Function Documentation

### 7.5.3.1 get\_code\_section()

```
const DebugSourceEnum& TrickHLA::DebugHandler::get_code_section ( ) const [inline]
```

Get the code section for this debug handler.

#### Returns

Debug handler code module as a DebugSourceEnum tag.

Definition at line 83 of file DebugHandler.hh.

References code\_section.

Referenced by set().

### 7.5.3.2 get\_debug\_level()

```
const DebugLevelEnum& TrickHLA::DebugHandler::get_debug_level ( ) const [inline]
```

Get current debug level as an enumeration value.

#### Returns

Current debug level as a DebugLevelEnum tag.

Definition at line 75 of file DebugHandler.hh.

References debug\_level.

Referenced by set().

### 7.5.3.3 get\_debug\_level\_as\_int()

```
const int TrickHLA::DebugHandler::get_debug_level_as_int ( ) const [inline]
```

Get the current debug level as an integer value.

#### Returns

Current debug level as an integer value.

Definition at line 79 of file DebugHandler.hh.

References debug\_level.

Referenced by TrickHLA::Manager::initialize().

### 7.5.3.4 set()

```
void TrickHLA::DebugHandler::set ( const DebugHandler & in ) [inline]
```

Set the debug handler from existing debug handler.

## Parameters

<i>in</i>	Debug handler from which to copy the level and code section.
-----------	--

Definition at line 98 of file DebugHandler.hh.

References code\_section, debug\_level, get\_code\_section(), and get\_debug\_level().

Referenced by TrickHLA::Parameter::set\_debug\_level(), TrickHLA::Attribute::set\_debug\_level(), and TrickHLA::SyncPntListBase::set\_debug\_level().

7.5.3.5 **should\_print()**

```
bool TrickHLA::DebugHandler::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const [inline]
```

**Conditional** test to see if a debug message should print.

## Returns

Returns true if the requested message should print level.

## Parameters

<i>level</i>	Debug level of incoming message.
<i>code</i>	Debug code source area of the incoming message.

Definition at line 89 of file DebugHandler.hh.

References code\_section, and debug\_level.

Referenced by DIS::ExecutionControl::announce\_sync\_point(), DSES::ExecutionControl::announce\_sync\_point(), IMSim::ExecutionControl::announce\_sync\_point(), SpaceFOM::ExecutionControl::announce\_sync\_point(), TrickHLA::SyncPntListBase::announce\_sync\_point(), TrickHLA::Parameter::byteswap\_buffer\_copy(), TrickHLA::Parameter::calculate\_size\_and\_number\_of\_items(), TrickHLA::Attribute::calculate\_size\_and\_number\_of\_items(), TrickHLA::ExecutionControl::clear\_multiphase\_init\_sync\_points(), DIS::ExecutionControl::clear\_multiphase\_init\_sync\_points(), TrickHLA::ExecutionControlBase::clear\_multiphase\_init\_sync\_points(), TrickHLA::Parameter::complete\_initialization(), TrickHLA::Attribute::determine\_cycle\_ratio(), DSES::ExecutionControl::determine\_federation\_master(), DIS::ExecutionControl::determine\_federation\_master(), IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate(), TrickHLA::Manager::determine\_job\_cycle\_time(), TrickHLA::Manager::discover\_object\_instance(), TrickHLA::Attribute::extract\_data(), TrickHLA::Parameter::extract\_data(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), TrickHLA::Attribute::initialize(), TrickHLA::ExecutionControlBase::mark\_object\_as\_deleted\_from\_federation(), TrickHLA::Manager::mark\_object\_as\_deleted\_from\_federation(), TrickHLA::ExecutionControlBase::object\_instance\_name\_reservation\_succeeded(), TrickHLA::Manager::object\_instance\_name\_reservation\_succeeded(), TrickHLA::Attribute::pack\_attribute\_buffer(), TrickHLA::Parameter::pack\_parameter\_buffer(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), TrickHLA::Manager::process\_interactions(), TrickHLA::Manager::publish\_and\_subscribe(), TrickHLA::Manager::receive\_cyclic\_data(), TrickHLA::ExecutionControlBase::receive\_execution\_configuration(), TrickHLA::Manager::receive\_init\_data(), SpaceFOM::ExecutionControl::receive\_init\_root\_ref\_frame(), SpaceFOM::ExecutionControl::receive\_interaction(), IMSim::ExecutionControl::receive\_interaction(), TrickHLA::Manager::receive\_interaction(), SpaceFOM::ExecutionControl::receive\_root\_ref\_frame(), TrickHLA::ExecutionControlBase::register\_objects\_with\_RTI(), TrickHLA::Manager::register\_objects\_with\_RTI(), TrickHLA::Manager::request\_data\_update(), TrickHLA::Manager::reserve\_object\_names\_with\_RTI(), TrickHLA::Manager::restart\_initialization(), TrickHLA::Manager::restore\_interactions(), SpaceFOM::ExecutionControl::role\_determination\_process(), DSES::ExecutionControl::run\_mode\_transition(), DIS::ExecutionControl::run\_mode\_transition(), IMSim::ExecutionControl::run\_mode\_transition()

```
::run_mode_transition(), SpaceFOM::ExecutionControl::run_mode_transition(), TrickHLA::Manager::send_cyclic_data(),
TrickHLA::ExecutionControlBase::send_execution_configuration(), TrickHLA::Manager::send_init_data(),
SpaceFOM::ExecutionControl::send_init_root_ref_frame(), TrickHLA::Manager::send_requested_data(),
SpaceFOM::ExecutionControl::send_root_ref_frame(), TrickHLA::Manager::set_object_instance_handles_by_name(),
TrickHLA::Manager::setup_all_ref_attributes(), TrickHLA::Manager::setup_all_RTI_handles(),
TrickHLA::Manager::setup_checkpoint_interactions(), IMSim::ExecutionControl::setup_interaction_ref_attributes(),
SpaceFOM::ExecutionControl::setup_interaction_ref_attributes(), TrickHLA::Manager::setup_interaction_ref_attributes(),
TrickHLA::Manager::setup_interaction_RTI_handles(), TrickHLA::Manager::setup_object_ref_attributes(),
TrickHLA::Manager::setup_object_RTI_handles(), TrickHLA::Manager::setup_preferred_order_with_RTI(),
SpaceFOM::ExecutionConfiguration::setup_ref_attributes(), TrickHLA::Manager::should_print(),
IMSim::ExecutionControl::start_federation_save_at_scenario_time(),
TrickHLA::SyncPntListBase::sync_point_registration_failed(),
TrickHLA::SyncPntListBase::sync_point_registration_succeeded(),
TrickHLA::Attribute::unpack_attribute_buffer(),
TrickHLA::Parameter::unpack_parameter_buffer(),
TrickHLA::SyncPntListBase::wait_for_all_announcements(),
TrickHLA::SyncPntListBase::wait_for_announcement(),
TrickHLA::Manager::wait_for_init_sync_point(),
TrickHLA::Manager::wait_on_discovery_of_objects(),
TrickHLA::Manager::wait_on_registration_of_required_objects(),
TrickHLA::Manager::wait_on_reservation_of_object_names(),
and SpaceFOM::ExecutionControl::wait_on_root_frame_discovered_synchronization().
```

## 7.5.4 Friends And Related Function Documentation

### 7.5.4.1 init\_attrTrickHLA\_\_DebugHandler

```
void init_attrTrickHLA__DebugHandler () [friend]
```

### 7.5.4.2 InputProcessor

```
friend class InputProcessor [friend]
Definition at line 53 of file DebugHandler.hh.
```

## 7.5.5 Field Documentation

### 7.5.5.1 code\_section

```
DebugSourceEnum TrickHLA::DebugHandler::code_section
```

#### Units: –

Code section(s) for which to activate debug messages, default: THLA\_ALL\_MODULES

Definition at line 107 of file DebugHandler.hh.

Referenced by `get_code_section()`, `set()`, and `should_print()`.

### 7.5.5.2 debug\_level

```
DebugLevelEnum TrickHLA::DebugHandler::debug_level
```

#### Units: –

Maximum debug report level requested by the user, default: THLA\_NO\_TRACE

Definition at line 106 of file DebugHandler.hh.

Referenced by `get_debug_level()`, `get_debug_level_as_int()`, `set()`, and `should_print()`.

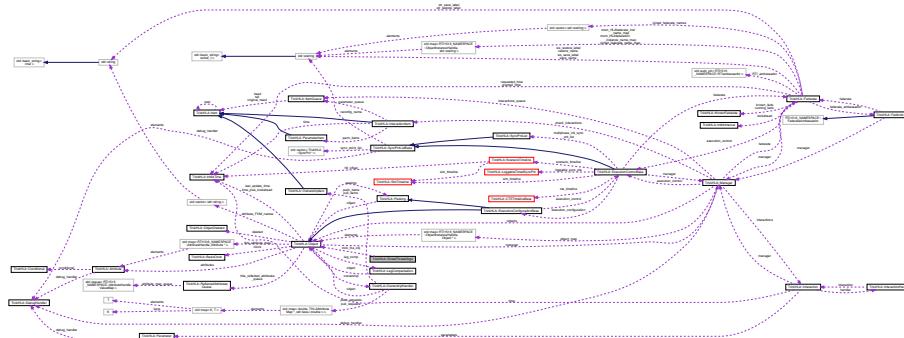
The documentation for this class was generated from the following file:

- [DebugHandler.hh](#)

## 7.6 TrickHLA::DivestThreadArgs Struct Reference

```
#include <Object.hh>
```

Collaboration diagram for TrickHLA::DivestThreadArgs:



### Data Fields

- `Object * trick_hla_obj`  
`trick_io{**}` Pointer to `TrickHLA` object.
- `RTI1516_NAMESPACE::AttributeHandleSet * handle_set`  
`trick_io{**}` Pointer to attribute handle set to divest ownership of.

#### 7.6.1 Detailed Description

Definition at line 740 of file Object.hh.

#### 7.6.2 Field Documentation

##### 7.6.2.1 handle\_set

`RTI1516_NAMESPACE::AttributeHandleSet* TrickHLA::DivestThreadArgs::handle_set`  
`trick_io{**}` Pointer to attribute handle set to divest ownership of.

Definition at line 742 of file Object.hh.

Referenced by `ownership_divestiture_pthread_function()`, and `TrickHLA::Object::push_ownership()`.

##### 7.6.2.2 trick\_hla\_obj

`Object* TrickHLA::DivestThreadArgs::trick_hla_obj`  
`trick_io{**}` Pointer to `TrickHLA` object.

Definition at line 741 of file Object.hh.

Referenced by `ownership_divestiture_pthread_function()`, and `TrickHLA::Object::push_ownership()`.

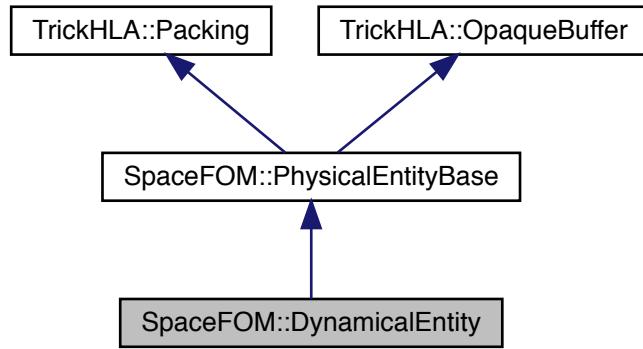
The documentation for this struct was generated from the following file:

- `Object.hh`

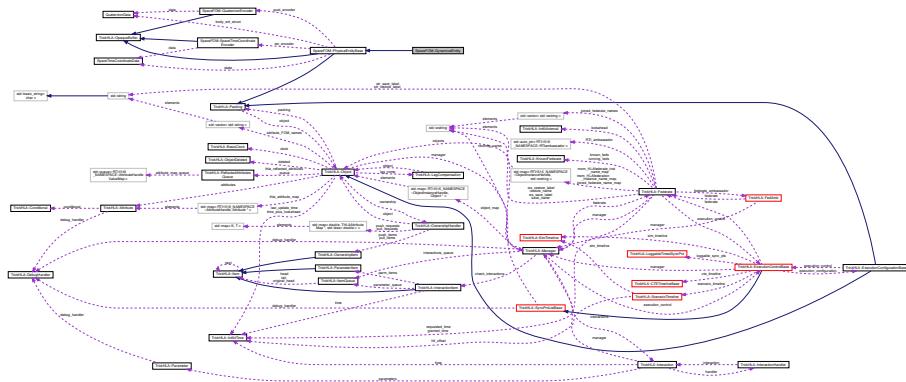
## 7.7 SpaceFOM::DynamicalEntity Class Reference

```
#include <DynamicalEntity.hh>
```

Inheritance diagram for SpaceFOM::DynamicalEntity:



Collaboration diagram for SpaceFOM::DynamicalEntity:



### Public Member Functions

- [DynamicalEntity \(\)](#)
- [virtual ~DynamicalEntity \(\)](#)
- [virtual void pack \(\)](#)

*Pack the data before being sent.*

- [virtual void unpack \(\)](#)

*Unpack the received data. The default.*

### Protected Attributes

- double [force](#) [3]

**Units:** *N**Total external force on vehicle applied through the vehicle center of mass.*

- double `torque` [3]

**Units:** *N\*m**Total external torques on vehicle.*

- double `mass`

**Units:** *kg**Vehicle mass.*

- double `mass_rate`

- double `inertia` [3][3]

*trick\_units{kg/s} Vehicle mass flow rate.*

- double `inertia_rate` [3][3]

*trick\_units{kg\*m2} Inertia matrix in element body frame.***Private Member Functions**

- `DynamicalEntity` (const `DynamicalEntity` &)

**Units:** *kg\*m2/s**Inertia matrix in element body frame.*

- `DynamicalEntity` & `operator=` (const `DynamicalEntity` &)

**Friends**

- class `InputProcessor`
- void `init_attrSpaceFOM__DynamicalEntity` ()

**Additional Inherited Members****7.7.1 Detailed Description**Definition at line 46 of file `DynamicalEntity.hh`.**7.7.2 Constructor & Destructor Documentation****7.7.2.1 `DynamicalEntity()` [1/2]**`DynamicalEntity::DynamicalEntity ( )`**Trick Job Class:** *initialization*Definition at line 53 of file `DynamicalEntity.cpp`.References `force`, `inertia`, `inertia_rate`, and `torque`.**7.7.2.2 `~DynamicalEntity()`**`DynamicalEntity::~DynamicalEntity ( ) [virtual]`**Trick Job Class:** *shutdown*Definition at line 66 of file `DynamicalEntity.cpp`.

### 7.7.2.3 **DynamicalEntity()** [2/2]

```
SpaceFOM::DynamicalEntity::DynamicalEntity (
    const DynamicalEntity & ) [private]
```

**Units:** *kg\*m2/s*

Inertia matrix in element body frame.

## 7.7.3 Member Function Documentation

### 7.7.3.1 **operator=( )**

```
DynamicalEntity& SpaceFOM::DynamicalEntity::operator= (
    const DynamicalEntity & ) [private]
```

### 7.7.3.2 **pack()**

```
void DynamicalEntity::pack ( ) [virtual]
```

Pack the data before being sent.

Reimplemented from [SpaceFOM::PhysicalEntityBase](#).

Definition at line 70 of file [DynamicalEntity.cpp](#).

References [SpaceFOM::PhysicalEntityBase::pack\(\)](#).

### 7.7.3.3 **unpack()**

```
void DynamicalEntity::unpack ( ) [virtual]
```

Unpack the received data. The default.

Reimplemented from [SpaceFOM::PhysicalEntityBase](#).

Definition at line 75 of file [DynamicalEntity.cpp](#).

References [SpaceFOM::PhysicalEntityBase::unpack\(\)](#).

## 7.7.4 Friends And Related Function Documentation

### 7.7.4.1 **init\_attrSpaceFOM\_\_DynamicalEntity**

```
void init_attrSpaceFOM__DynamicalEntity ( ) [friend]
```

### 7.7.4.2 **InputProcessor**

```
friend class InputProcessor [friend]
```

Definition at line 53 of file [DynamicalEntity.hh](#).

## 7.7.5 Field Documentation

### 7.7.5.1 force

```
double SpaceFOM::DynamicalEntity::force[3] [protected]
```

**Units:**  $N$

Total external force on vehicle applied through the vehicle center of mass.

Expressed in the vehicle struct frame.

Definition at line 68 of file `DynamicalEntity.hh`.

Referenced by `DynamicalEntity()`.

### 7.7.5.2 inertia

```
double SpaceFOM::DynamicalEntity::inertia[3][3] [protected]
```

trick\_units{kg/s} Vehicle mass flow rate.

Definition at line 75 of file `DynamicalEntity.hh`.

Referenced by `DynamicalEntity()`.

### 7.7.5.3 inertia\_rate

```
double SpaceFOM::DynamicalEntity::inertia_rate[3][3] [protected]
```

trick\_units{kg\*m2} Inertia matrix in element body frame.

Definition at line 76 of file `DynamicalEntity.hh`.

Referenced by `DynamicalEntity()`.

### 7.7.5.4 mass

```
double SpaceFOM::DynamicalEntity::mass [protected]
```

**Units:**  $kg$

Vehicle mass.

Definition at line 73 of file `DynamicalEntity.hh`.

### 7.7.5.5 mass\_rate

```
double SpaceFOM::DynamicalEntity::mass_rate [protected]
```

Definition at line 74 of file `DynamicalEntity.hh`.

### 7.7.5.6 torque

```
double SpaceFOM::DynamicalEntity::torque[3] [protected]
```

**Units:**  $N*m$

Total external torques on vehicle.

Expressed in the vehicle struct frame.

Definition at line 71 of file `DynamicalEntity.hh`.

Referenced by `DynamicalEntity()`.

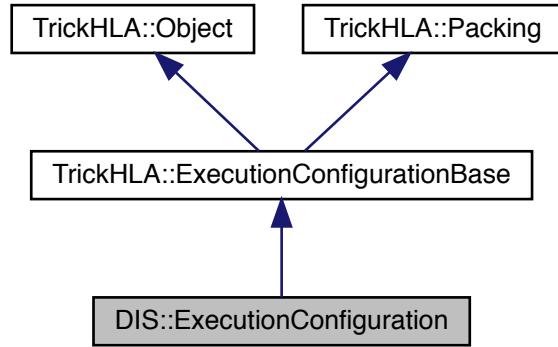
The documentation for this class was generated from the following files:

- [DynamicalEntity.hh](#)
- [DynamicalEntity.cpp](#)

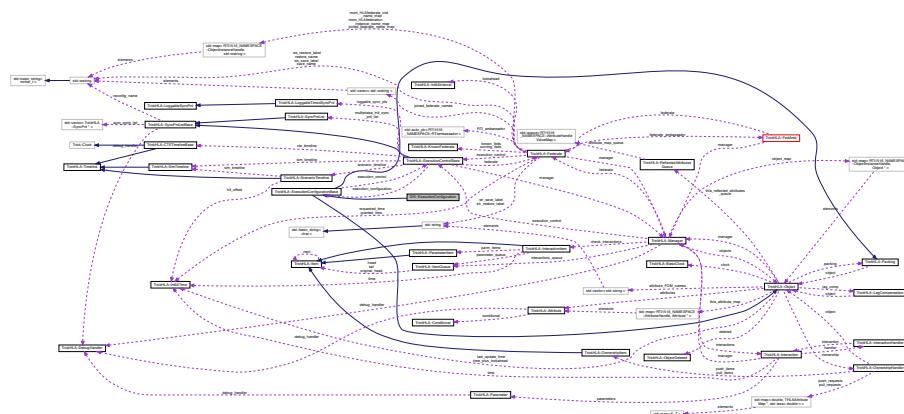
## 7.8 DIS::ExecutionConfiguration Class Reference

```
#include <ExecutionConfiguration.hh>
```

Inheritance diagram for DIS::ExecutionConfiguration:



Collaboration diagram for DIS::ExecutionConfiguration:



### Public Member Functions

- `ExecutionConfiguration ()`  
*Default constructor for the `DIS ExecutionConfiguration` class.*
- `virtual ~ExecutionConfiguration ()`  
*Pure virtual destructor for the `DIS ExecutionConfiguration` class.*
- `virtual void configure_attributes (const char *exco_name)`  
*Sets up the attributes for the ExCO using default values. These can be overridden in the input file.*
- `virtual void pack ()`
- `virtual void unpack ()`
- `virtual void set_root_frame_name (const char *name)`

- `virtual const char * get_root_frame_name ()`

*Set the root reference frame name.*
- `virtual void set_scenario_time_epoch (double scenario_time)`

*Get the root reference frame name.*
- `virtual double get_scenario_time_epoch ()`

*Set the scenario time line epoch.*
- `virtual void set_next_mode_scenario_time (double next_mode_time)`

*Get the scenario time line epoch.*
- `virtual double get_next_mode_scenario_time ()`

*Set the scenario time for the next mode transition.*
- `virtual void set_next_mode_cte_time (double cte_time)`

*Get the next mode scenario time.*
- `virtual double get_next_mode_cte_time ()`

*Set the next mode CTE time.*
- `virtual void set_current_execution_mode (short mode)`

*Sets the current ExCO run mode.*
- `virtual void set_current_execution_mode (DIS::ExecutionModeEnum mode)`

*Sets the current ExCO run mode.*
- `virtual short get_current_execution_mode ()`

*Get the current execution mode.*
- `virtual void set_next_execution_mode (short mode)`

*Sets the next ExCO execution mode.*
- `virtual void set_next_execution_mode (DIS::ExecutionModeEnum mode)`

*Sets the next ExCO execution mode.*
- `virtual short get_next_execution_mode ()`

*Get the next execution mode.*
- `virtual void setup_ref_attributes (Packing *packing_obj)`

*Setup the Trick Ref Attributes for the ExCO object.*
- `virtual void print_execution_configuration ()`

*Print the current ExCO state to the console.*
- `virtual bool wait_on_update ()`

*Wait on an ExCO update.*

## Data Fields

- `char * root_frame_name`

**Units:** –  
*Specifies the name of the root coordinate frame in the federation execution's reference frame tree.*
- `double scenario_time_epoch`

**Units:** s  
*Federation execution scenario time epoch.*
- `double next_mode_scenario_time`

**Units:** s  
*The time for the next federation execution mode change expressed as a federation scenario time reference.*
- `double next_mode_cte_time`

**Units:** s  
*The time for the next federation execution mode change expressed as a Central Timing Equipment (CTE) time reference.*

- short `current_execution_mode`

**Units:** –

Defines the current running state of the federation execution in terms of a finite set of states expressed in the `RunMode` enumeration.

- short `next_execution_mode`

**Units:** –

Defines the next running state of the federation execution in terms of a finite set of states expressed in the `RunMode` enumeration.

## Private Member Functions

- `ExecutionConfiguration (const ExecutionConfiguration &rhs)`
- `ExecutionConfiguration & operator= (const ExecutionConfiguration &rhs)`

## Friends

- class `InputProcessor`
- void `init_attrDIS__ExecutionConfiguration ()`

## Additional Inherited Members

### 7.8.1 Detailed Description

Definition at line 50 of file `DIS/ExecutionConfiguration.hh`.

### 7.8.2 Constructor & Destructor Documentation

#### 7.8.2.1 `ExecutionConfiguration()` [1/2]

`ExecutionConfiguration::ExecutionConfiguration ( )`

Default constructor for the `DIS ExecutionConfiguration` class.

**Trick Job Class:** *initialization*

Definition at line 83 of file `DIS/ExecutionConfiguration.cpp`.

References `TrickHLA::Object::name`, and `TrickHLA::Object::packing`.

#### 7.8.2.2 `~ExecutionConfiguration()`

`ExecutionConfiguration::~ExecutionConfiguration ( ) [virtual]`

Pure virtual destructor for the `DIS ExecutionConfiguration` class.

Even though this is a pure virtual destructor, we provide a default implementation that can be called from an inheriting class. **Trick Job Class:** *shutdown*

Definition at line 108 of file `DIS/ExecutionConfiguration.cpp`.

References `root_frame_name`.

#### 7.8.2.3 `ExecutionConfiguration()` [2/2]

```
DIS::ExecutionConfiguration::ExecutionConfiguration ( 
    const ExecutionConfiguration & rhs ) [private]
```

### 7.8.3 Member Function Documentation

#### 7.8.3.1 configure\_attributes()

```
void ExecutionConfiguration::configure_attributes (
    const char * exco_name )  [virtual]
```

Sets up the attributes for the ExCO using default values. These can be overridden in the input file.

##### Parameters

<code>exco_name</code>	S_define level Trick name for ExCO.
------------------------	-------------------------------------

These can be overridden in the input file. **Trick Job Class:** `default_data`

Definition at line 124 of file DIS/ExecutionConfiguration.cpp.

References TrickHLA::Object::attr\_count, TrickHLA::Object::attributes, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_UNICODE\_STRING, TrickHLA::Attribute::FOM\_name, TrickHLA::Object::FOM\_name, TrickHLA::Object::name, TrickHLA::Object::packing, root\_frame\_name, TrickHLA::Attribute::rti\_encoding, trick\_MM, and TrickHLA::Attribute::trick\_name.

#### 7.8.3.2 get\_current\_execution\_mode()

```
virtual short DIS::ExecutionConfiguration::get_current_execution_mode ( )  [inline], [virtual]
```

Get the current execution mode.

##### Returns

The current execution mode as an integer.

Definition at line 153 of file DIS/ExecutionConfiguration.hh.

References `current_execution_mode`.

#### 7.8.3.3 get\_next\_execution\_mode()

```
virtual short DIS::ExecutionConfiguration::get_next_execution_mode ( )  [inline], [virtual]
```

Get the next execution mode.

##### Returns

The next execution mode as an integer.

Definition at line 163 of file DIS/ExecutionConfiguration.hh.

References `next_execution_mode`.

#### 7.8.3.4 get\_next\_mode\_cte\_time()

```
virtual double DIS::ExecutionConfiguration::get_next_mode_cte_time ( )  [inline], [virtual]
```

Get the next mode CTE time.

**Returns**

The next mode CTE time.

Definition at line 143 of file DIS/ExecutionConfiguration.hh.

References next\_mode\_cte\_time.

Referenced by DIS::ExecutionControl::run\_mode\_transition(), and DIS::ExecutionControl::set\_next\_execution\_control\_mode().

**7.8.3.5 get\_next\_mode\_scenario\_time()**

```
virtual double DIS::ExecutionConfiguration::get_next_mode_scenario_time () [inline], [virtual]  
Get the next mode scenario time.
```

**Returns**

The next mode scenario time.

Definition at line 136 of file DIS/ExecutionConfiguration.hh.

References next\_mode\_scenario\_time.

**7.8.3.6 get\_root\_frame\_name()**

```
virtual const char* DIS::ExecutionConfiguration::get_root_frame_name () [inline], [virtual]  
Get the root reference frame name.
```

**Returns**

Root Reference Frame name as a constant string.

Definition at line 122 of file DIS/ExecutionConfiguration.hh.

References root\_frame\_name.

**7.8.3.7 get\_scenario\_time\_epoch()**

```
virtual double DIS::ExecutionConfiguration::get_scenario_time_epoch () [inline], [virtual]  
Get the scenario time line epoch.
```

**Returns**

The scenario time line epoch.

Definition at line 129 of file DIS/ExecutionConfiguration.hh.

References scenario\_time\_epoch.

**7.8.3.8 operator=()**

```
ExecutionConfiguration& DIS::ExecutionConfiguration::operator= (   
    const ExecutionConfiguration & rhs ) [private]
```

**7.8.3.9 pack()**

```
void ExecutionConfiguration::pack () [virtual]
```

This function is called before the data is sent to the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 195 of file DIS/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, DIS::execution\_mode\_enum\_to\_string(), DIS::execution\_mode\_int16\_to\_enum(), TrickHLA::Object::get\_federate(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Federate::get\_looking\_ahead(), TrickHLA::Object::get\_name(), TrickHLA::Federate::get\_requested\_time(), TrickHLA::Int64Interval::get\_time\_in\_millis(), next\_execution\_mode, next\_mode\_cte\_time, next\_mode\_scenario\_time, root\_frame\_name, scenario\_time\_epoch, TrickHLA::Packing::should\_print(), and THLA-ENDL.

#### 7.8.3.10 print\_execution\_configuration()

```
void ExecutionConfiguration::print_execution_configuration ( ) [virtual]
```

Print the current ExCO state to the console.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 611 of file DIS/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, DIS::execution\_mode\_enum\_to\_string(), DIS::execution\_mode\_int16\_to\_enum(), TrickHLA::Object::get\_name(), next\_execution\_mode, next\_mode\_cte\_time, next\_mode\_scenario\_time, root\_frame\_name, scenario\_time\_epoch, and THLA-ENDL.

#### 7.8.3.11 set\_current\_execution\_mode() [1/2]

```
void ExecutionConfiguration::set_current_execution_mode (
    DIS::ExecutionModeEnum mode ) [virtual]
```

Sets the current ExCO run mode.

##### Parameters

<i>mode</i>	Current Execution configuration run mode enumeration value.
-------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 410 of file DIS/ExecutionConfiguration.cpp.

References DIS::execution\_mode\_enum\_to\_int16(), and set\_current\_execution\_mode().

#### 7.8.3.12 set\_current\_execution\_mode() [2/2]

```
void ExecutionConfiguration::set_current_execution_mode (
    short mode ) [virtual]
```

Sets the current ExCO run mode.

##### Parameters

<i>mode</i>	Current Execution configuration run mode integer value.
-------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 397 of file DIS/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::ExecutionConfigurationBase::execution\_control, and TrickHLA::ExecutionControlBase::is\_master().

Referenced by DIS::ExecutionControl::freeze\_mode\_transition(), DIS::ExecutionControl::run\_mode\_transition(), and set\_current\_execution\_mode().

**7.8.3.13 set\_next\_execution\_mode() [1/2]**

```
void ExecutionConfiguration::set_next_execution_mode (
    DIS::ExecutionModeEnum mode ) [virtual]
```

Sets the next ExCO execution mode.

**Parameters**

<i>mode</i>	Next Execution configuration execution mode from an enumeration value.
-------------	--

WARNING: Only the Master federate should ever set this.

Definition at line 434 of file DIS/ExecutionConfiguration.cpp.

References DIS::execution\_mode\_enum\_to\_int16(), and set\_next\_execution\_mode().

**7.8.3.14 set\_next\_execution\_mode() [2/2]**

```
void ExecutionConfiguration::set_next_execution_mode (
    short mode ) [virtual]
```

Sets the next ExCO execution mode.

**Parameters**

<i>mode</i>	Next Execution configuration execution mode from an integer.
-------------	--

WARNING: Only the Master federate should ever set this.

Definition at line 421 of file DIS/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_execution\_mode.

Referenced by DIS::ExecutionControl::set\_next\_execution\_control\_mode(), and set\_next\_execution\_mode().

**7.8.3.15 set\_next\_mode\_cte\_time()**

```
void ExecutionConfiguration::set_next_mode_cte_time (
    double cte_time ) [virtual]
```

Set the next mode CTE time.

**Parameters**

<i>cte_time</i>	CTE time for next mode transition.
-----------------	------------------------------------

WARNING: Only the Master federate should ever set this.

Definition at line 383 of file DIS/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_mode\_cte\_time.

Referenced by DIS::ExecutionControl::set\_next\_execution\_control\_mode().

**7.8.3.16 set\_next\_mode\_scenario\_time()**

```
void ExecutionConfiguration::set_next_mode_scenario_time (
    double next_mode_time ) [virtual]
```

Set the scenario time for the next mode transition.

**Parameters**

<i>next_mode_time</i>	Scenario time for next mode transition.
-----------------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 369 of file DIS/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_mode\_scenario\_time.

Referenced by DIS::ExecutionControl::set\_next\_execution\_control\_mode().

**7.8.3.17 set\_root\_frame\_name()**

```
void ExecutionConfiguration::set_root_frame_name (
    const char * name ) [virtual]
```

Set the root reference frame name.

**Parameters**

<i>name</i>	Root reference frame name.
-------------	----------------------------

Definition at line 336 of file DIS/ExecutionConfiguration.cpp.

References TrickHLA::Object::name, and root\_frame\_name.

**7.8.3.18 set\_scenario\_time\_epoch()**

```
void ExecutionConfiguration::set_scenario_time_epoch (
    double scenario_time ) [virtual]
```

Set the scenario time line epoch.

**Parameters**

<i>scenario_time</i>	Scenario time line epoch.
----------------------	---------------------------

WARNING: Only the Master federate should ever set this.

Definition at line 356 of file DIS/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and scenario\_time\_epoch.

Referenced by DIS::ExecutionControl::set\_next\_execution\_control\_mode().

**7.8.3.19 setup\_ref\_attributes()**

```
virtual void DIS::ExecutionConfiguration::setup_ref_attributes (
    Packing * packing_obj ) [virtual]
```

Setup the Trick Ref Attributes for the ExCO object.

**Parameters**

<i>packing_obj</i>	Associated packing object.
--------------------	----------------------------

### 7.8.3.20 `unpack()`

```
void ExecutionConfiguration::unpack ( ) [virtual]
```

This function is called after data is received from the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 245 of file `DIS/ExecutionConfiguration.cpp`.

References `current_execution_mode`, `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_SOURCE_PACKING`, `DIS::execution_mode_enum_to_string()`, `DIS::execution_mode_int16_to_enum()`, `TrickHLA::Object::get_federate()`, `TrickHLA::Federate::get_granted_time()`, `TrickHLA::Federate::get_looking_ahead()`, `TrickHLA::Object::get_name()`, `TrickHLA::Federate::get_requested_time()`, `TrickHLA::Int64Interval::getTimeInMicros()`, `next_execution_mode`, `next_mode_cte_time`, `next_mode_scenario_time`, `TrickHLA::ExecutionConfigurationBase::pending_update`, `root_frame_name`, `scenario_time_epoch`, `TrickHLA::Packing::should_print()`, and `THLA_ENDL`.

### 7.8.3.21 `wait_on_update()`

```
bool ExecutionConfiguration::wait_on_update ( ) [virtual]
```

Wait on an ExCO update.

#### Returns

True for successful wait.

Reimplemented from [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 632 of file `DIS/ExecutionConfiguration.cpp`.

References `TrickHLA::Object::any_remotely_owned_subscribed_init_attribute()`, `TrickHLA::Federate::check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionConfigurationBase::execution_control`, `TrickHLA::Object::get_federate()`, `TrickHLA::Object::is_changed()`, `TrickHLA::Federate::is_execution_member()`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::Object::receive_init_data()`, `TrickHLA::Federate::should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `DIS::ExecutionControl::run_mode_transition()`.

## 7.8.4 Friends And Related Function Documentation

### 7.8.4.1 `init_attrDIS__ExecutionConfiguration`

```
void init_attrDIS__ExecutionConfiguration ( ) [friend]
```

### 7.8.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 58 of file `DIS/ExecutionConfiguration.hh`.

## 7.8.5 Field Documentation

### 7.8.5.1 `current_execution_mode`

```
short DIS::ExecutionConfiguration::current_execution_mode
```

**Units:** –

Defines the current running state of the federation execution in terms of a finite set of states expressed in the `RunMode` enumeration.

Definition at line 88 of file DIS/ExecutionConfiguration.hh.

Referenced by `get_current_execution_mode()`, `DIS::ExecutionControl::is_mtr_valid()`, `pack()`, `print_execution_configuration()`, `DIS::ExecutionControl::process_execution_control_updates()`, `set_current_execution_mode()`, and `unpack()`.

#### 7.8.5.2 `next_execution_mode`

```
short DIS::ExecutionConfiguration::next_execution_mode
```

**Units:** –

Defines the next running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.

This is used in conjunction with the `cte_mode_time`, `sim_mode_time` and associated sync point mechanisms to coordinate federation execution mode transitions.

Definition at line 92 of file DIS/ExecutionConfiguration.hh.

Referenced by `get_next_execution_mode()`, `pack()`, `print_execution_configuration()`, `DIS::ExecutionControl::process_execution_control_updates()`, `set_next_execution_mode()`, and `unpack()`.

#### 7.8.5.3 `next_mode_cte_time`

```
double DIS::ExecutionConfiguration::next_mode_cte_time
```

**Units:** s

The time for the next federation execution mode change expressed as a Central Timing Equipment (CTE) time reference.

The standard for this reference shall be defined in the federation agreement when CTE is used.

Definition at line 83 of file DIS/ExecutionConfiguration.hh.

Referenced by `get_next_mode_cte_time()`, `pack()`, `print_execution_configuration()`, `DIS::ExecutionControl::process_execution_control_updates()`, `DIS::ExecutionControl::process_mode_transition_request()`, `set_next_mode_cte_time()`, and `unpack()`.

#### 7.8.5.4 `next_mode_scenario_time`

```
double DIS::ExecutionConfiguration::next_mode_scenario_time
```

**Units:** s

The time for the next federation execution mode change expressed as a federation scenario time reference.

Note: this value is only meaningful for going into freeze; exiting freeze is coordinated through a sync point mechanism.

Definition at line 77 of file DIS/ExecutionConfiguration.hh.

Referenced by `get_next_mode_scenario_time()`, `pack()`, `print_execution_configuration()`, `DIS::ExecutionControl::process_execution_control_updates()`, `DIS::ExecutionControl::process_mode_transition_request()`, `set_next_mode_scenario_time()`, and `unpack()`.

#### 7.8.5.5 `root_frame_name`

```
char* DIS::ExecutionConfiguration::root_frame_name
```

**Units:** –

Specifies the name of the root coordinate frame in the federation execution's reference frame tree.

This frame shall remain fixed throughout the federation execution.

Definition at line 65 of file DIS/ExecutionConfiguration.hh.

Referenced by `configure_attributes()`, `get_root_frame_name()`, `pack()`, `print_execution_configuration()`, `set_root_frame_name()`, `unpack()`, and `~ExecutionConfiguration()`.

### 7.8.5.6 scenario\_time\_epoch

```
double DIS::ExecutionConfiguration::scenario_time_epoch
```

**Units:** s

Federation execution scenario time epoch.

This is the beginning epoch expressed in Terrestrial Time (TT) that corresponds to HLA logical time 0. All joining federates shall use this time to coordinate the offset between their local simulation scenario times, their local simulation execution times and the HLA logical time.

Definition at line 70 of file DIS/ExecutionConfiguration.hh.

Referenced by `get_scenario_time_epoch()`, `pack()`, `print_execution_configuration()`, `DIS::ExecutionControl::process_<→ execution_control_updates()`, `DIS::ExecutionControl::process_mode_transition_request()`, `set_scenario_time_epoch()`, and `unpack()`.

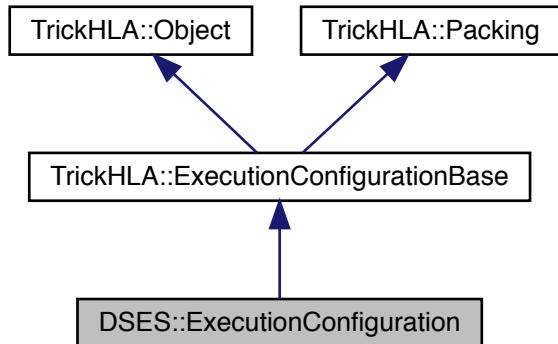
The documentation for this class was generated from the following files:

- [DIS/ExecutionConfiguration.hh](#)
- [DIS/ExecutionConfiguration.cpp](#)

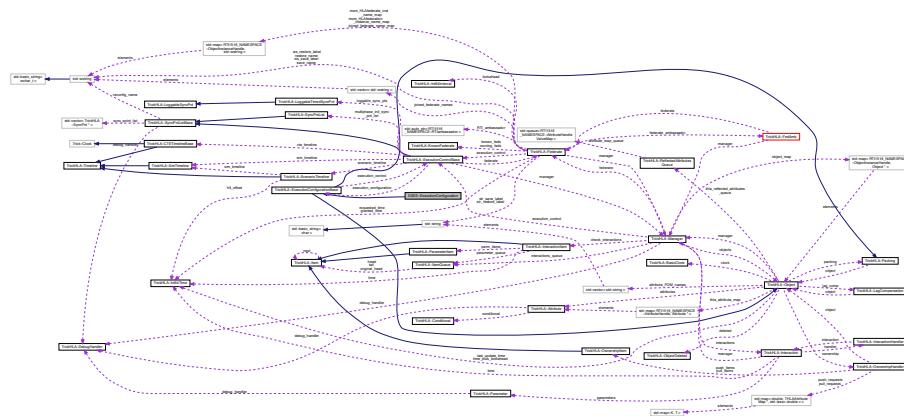
## 7.9 DSES::ExecutionConfiguration Class Reference

```
#include <ExecutionConfiguration.hh>
```

Inheritance diagram for DSES::ExecutionConfiguration:



## Collaboration diagram for DSES::ExecutionConfiguration:



## Public Member Functions

- **ExecutionConfiguration ()**  
*Default constructor for the DSES ExecutionConfiguration class.*
  - **virtual ~ExecutionConfiguration ()**  
*Pure virtual destructor for the DSES ExecutionConfiguration class.*
  - **virtual void configure\_attributes (const char \*exco\_name)**  
*Sets up the attributes for the ExCO using default values. These can be overridden in the input file.*
  - **virtual void pack ()**
  - **virtual void unpack ()**
  - **virtual void set\_root\_frame\_name (const char \*name)**  
*Set the root reference frame name.*
  - **virtual const char \* get\_root\_frame\_name ()**  
*Get the root reference frame name.*
  - **virtual void set\_scenario\_time\_epoch (double scenario\_time)**  
*Set the scenario time line epoch.*
  - **virtual double get\_scenario\_time\_epoch ()**  
*Get the scenario time line epoch.*
  - **virtual void set\_next\_mode\_scenario\_time (double next\_mode\_time)**  
*Set the scenario time for the next mode transition.*
  - **virtual double get\_next\_mode\_scenario\_time ()**  
*Get the next mode scenario time.*
  - **virtual void set\_next\_mode\_cte\_time (double cte\_time)**  
*Set the next mode CTE time.*
  - **virtual double get\_next\_mode\_cte\_time ()**  
*Get the next mode CTE time.*
  - **virtual void set\_current\_execution\_mode (short mode)**  
*Sets the current ExCO run mode.*
  - **virtual void set\_current\_execution\_mode (DSES::ExecutionModeEnum mode)**  
*Sets the current ExCO run mode.*
  - **virtual short get\_current\_execution\_mode ()**  
*Get the current execution mode.*

- virtual void `set_next_execution_mode` (short mode)  
*Sets the next ExCO execution mode.*
- virtual void `set_next_execution_mode` (DSES::ExecutionModeEnum mode)  
*Sets the next ExCO execution mode.*
- virtual short `get_next_execution_mode` ()  
*Get the next execution mode.*
- virtual void `setup_ref_attributes` (Packing \*packing\_obj)  
*Setup the Trick Ref Attributes for the ExCO object.*
- virtual void `print_execution_configuration` ()  
*Print the current ExCO state to the console.*
- virtual bool `wait_on_update` ()  
*Wait on an ExCO update.*

## Data Fields

- char \* `root_frame_name`  
**Units:** –  
*Specifies the name of the root coordinate frame in the federation execution's reference frame tree.*
- double `scenario_time_epoch`  
**Units:** s  
*Federation execution scenario time epoch.*
- double `next_mode_scenario_time`  
**Units:** s  
*The time for the next federation execution mode change expressed as a federation scenario time reference.*
- double `next_mode_cte_time`  
**Units:** s  
*The time for the next federation execution mode change expressed as a Central Timing Equipment (CTE) time reference.*
- short `current_execution_mode`  
**Units:** –  
*Defines the current running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.*
- short `next_execution_mode`  
**Units:** –  
*Defines the next running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.*

## Private Member Functions

- `ExecutionConfiguration` (const `ExecutionConfiguration` &rhs)
- `ExecutionConfiguration` & `operator=` (const `ExecutionConfiguration` &rhs)

## Friends

- class `InputProcessor`
- void `init_attrDSES__ExecutionConfiguration` ()

## Additional Inherited Members

### 7.9.1 Detailed Description

Definition at line 50 of file DSES/ExecutionConfiguration.hh.

## 7.9.2 Constructor & Destructor Documentation

### 7.9.2.1 ExecutionConfiguration() [1/2]

ExecutionConfiguration::ExecutionConfiguration ( )  
 Default constructor for the [DSES ExecutionConfiguration](#) class.

**Trick Job Class:** *initialization*

Definition at line 83 of file DSES/ExecutionConfiguration.cpp.

References TrickHLA::Object::name, and TrickHLA::Object::packing.

### 7.9.2.2 ~ExecutionConfiguration()

ExecutionConfiguration::~ExecutionConfiguration ( ) [virtual]  
 Pure virtual destructor for the [DSES ExecutionConfiguration](#) class.

Even though this is a pure virtual destructor, we provide a default implementation that can be called from an inheriting class. **Trick Job Class:** *shutdown*

Definition at line 108 of file DSES/ExecutionConfiguration.cpp.

References root\_frame\_name.

### 7.9.2.3 ExecutionConfiguration() [2/2]

DSES::ExecutionConfiguration::ExecutionConfiguration (   
 const [ExecutionConfiguration](#) & rhs ) [private]

## 7.9.3 Member Function Documentation

### 7.9.3.1 configure\_attributes()

void ExecutionConfiguration::configure\_attributes (   
 const char \* exco\_name ) [virtual]

Sets up the attributes for the ExCO using default values. These can be overridden in the input file.

#### Parameters

exco_name	S_define level Trick name for ExCO.
-----------	-------------------------------------

These can be overridden in the input file. **Trick Job Class:** *default\_data*

Definition at line 124 of file DSES/ExecutionConfiguration.cpp.

References TrickHLA::Object::attr\_count, TrickHLA::Object::attributes, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_UNICODE\_STRING, TrickHLA::Attribute::FOM\_name, TrickHLA::Object::FOM\_name, TrickHLA::Object::name, TrickHLA::Object::packing, root\_frame\_name, TrickHLA::Attribute::rti\_encoding, trick\_MM, and TrickHLA::Attribute::trick\_name.

### 7.9.3.2 get\_current\_execution\_mode()

virtual short DSES::ExecutionConfiguration::get\_current\_execution\_mode ( ) [inline], [virtual]  
 Get the current execution mode.

**Returns**

The current execution mode as an integer.

Definition at line 153 of file DSES/ExecutionConfiguration.hh.  
References current\_execution\_mode.

**7.9.3.3 `get_next_execution_mode()`**

```
virtual short DSES::ExecutionConfiguration::get_next_execution_mode ( ) [inline], [virtual]  
Get the next execution mode.
```

**Returns**

The next execution mode as an integer.

Definition at line 163 of file DSES/ExecutionConfiguration.hh.  
References next\_execution\_mode.

**7.9.3.4 `get_next_mode_cte_time()`**

```
virtual double DSES::ExecutionConfiguration::get_next_mode_cte_time ( ) [inline], [virtual]  
Get the next mode CTE time.
```

**Returns**

The next mode CTE time.

Definition at line 143 of file DSES/ExecutionConfiguration.hh.  
References next\_mode\_cte\_time.  
Referenced by DSES::ExecutionControl::run\_mode\_transition(), and DSES::ExecutionControl::set\_next\_execution\_cte\_time().

**7.9.3.5 `get_next_mode_scenario_time()`**

```
virtual double DSES::ExecutionConfiguration::get_next_mode_scenario_time ( ) [inline], [virtual]  
Get the next mode scenario time.
```

**Returns**

The next mode scenario time.

Definition at line 136 of file DSES/ExecutionConfiguration.hh.  
References next\_mode\_scenario\_time.

**7.9.3.6 `get_root_frame_name()`**

```
virtual const char* DSES::ExecutionConfiguration::get_root_frame_name ( ) [inline], [virtual]  
Get the root reference frame name.
```

**Returns**

Root Reference Frame name as a constant string.

Definition at line 122 of file DSES/ExecutionConfiguration.hh.  
References root\_frame\_name.

### 7.9.3.7 `get_scenario_time_epoch()`

```
virtual double DSES::ExecutionConfiguration::get_scenario_time_epoch ( ) [inline], [virtual]
Get the scenario time line epoch.
```

#### Returns

The scenario time line epoch.

Definition at line 129 of file DSES/ExecutionConfiguration.hh.  
 References scenario\_time\_epoch.

### 7.9.3.8 `operator=( )`

```
ExecutionConfiguration& DSES::ExecutionConfiguration::operator= (
    const ExecutionConfiguration & rhs ) [private]
```

### 7.9.3.9 `pack()`

```
void ExecutionConfiguration::pack ( ) [virtual]
```

This function is called before the data is sent to the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 195 of file DSES/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, DSES::execution\_mode\_enum\_to\_string(), DSES::execution\_mode\_int16\_to\_enum(), TrickHLA::Object::get\_federate(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Federate::get\_lookahead(), TrickHLA::Object::get\_name(), TrickHLA::Federate::get\_requested\_time(), TrickHLA::Int64Interval::getTimelnMicros(), next\_execution\_mode, next\_mode\_cte\_time, next\_mode\_scenario\_time, root\_frame\_name, scenario\_time\_epoch, TrickHLA::Packing::should\_print(), and THLA\_ENDL.

### 7.9.3.10 `print_execution_configuration()`

```
void ExecutionConfiguration::print_execution_configuration ( ) [virtual]
```

Print the current ExCO state to the console.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 611 of file DSES/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, DSES::execution\_mode\_enum\_to\_string(), DSES::execution\_mode\_int16\_to\_enum(), TrickHLA::Object::get\_name(), next\_execution\_mode, next\_mode\_cte\_time, next\_mode\_scenario\_time, root\_frame\_name, scenario\_time\_epoch, and THLA\_ENDL.

### 7.9.3.11 `set_current_execution_mode() [1/2]`

```
void ExecutionConfiguration::set_current_execution_mode (
    DSES::ExecutionModeEnum mode ) [virtual]
```

Sets the current ExCO run mode.

#### Parameters

<code>mode</code>	Current Execution configuration run mode enumeration value.
-------------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 410 of file DSES/ExecutionConfiguration.cpp.

References DSES::execution\_mode\_enum\_to\_int16(), and set\_current\_execution\_mode().

#### 7.9.3.12 set\_current\_execution\_mode() [2/2]

```
void ExecutionConfiguration::set_current_execution_mode (
    short mode ) [virtual]
```

Sets the current ExCO run mode.

##### Parameters

<i>mode</i>	Current Execution configuration run mode integer value.
-------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 397 of file DSES/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::ExecutionConfigurationBase::execution\_control, and TrickHLA::ExecutionControlBase::is\_master().

Referenced by DSES::ExecutionControl::freeze\_mode\_transition(), DSES::ExecutionControl::run\_mode\_transition(), and set\_current\_execution\_mode().

#### 7.9.3.13 set\_next\_execution\_mode() [1/2]

```
void ExecutionConfiguration::set_next_execution_mode (
    DSES::ExecutionModeEnum mode ) [virtual]
```

Sets the next ExCO execution mode.

##### Parameters

<i>mode</i>	Next Execution configuration execution mode from an enumeration value.
-------------	--

WARNING: Only the Master federate should ever set this.

Definition at line 434 of file DSES/ExecutionConfiguration.cpp.

References DSES::execution\_mode\_enum\_to\_int16(), and set\_next\_execution\_mode().

#### 7.9.3.14 set\_next\_execution\_mode() [2/2]

```
void ExecutionConfiguration::set_next_execution_mode (
    short mode ) [virtual]
```

Sets the next ExCO execution mode.

##### Parameters

<i>mode</i>	Next Execution configuration execution mode from an integer.
-------------	--

WARNING: Only the Master federate should ever set this.

Definition at line 421 of file DSES/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_execution\_mode.

Referenced by DSES::ExecutionControl::set\_next\_execution\_control\_mode(), and set\_next\_execution\_mode().

### 7.9.3.15 `set_next_mode_cte_time()`

```
void ExecutionConfiguration::set_next_mode_cte_time (
    double cte_time ) [virtual]
```

Set the next mode CTE time.

#### Parameters

<i>cte_time</i>	CTE time for next mode transition.
-----------------	------------------------------------

WARNING: Only the Master federate should ever set this.

Definition at line 383 of file DSES/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_mode\_cte\_time.

Referenced by DSES::ExecutionControl::set\_next\_execution\_control\_mode().

### 7.9.3.16 `set_next_mode_scenario_time()`

```
void ExecutionConfiguration::set_next_mode_scenario_time (
    double next_mode_time ) [virtual]
```

Set the scenario time for the next mode transition.

#### Parameters

<i>next_mode_time</i>	Scenario time for next mode transition.
-----------------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 369 of file DSES/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_mode\_scenario\_time.

Referenced by DSES::ExecutionControl::set\_next\_execution\_control\_mode().

### 7.9.3.17 `set_root_frame_name()`

```
void ExecutionConfiguration::set_root_frame_name (
    const char * name ) [virtual]
```

Set the root reference frame name.

#### Parameters

<i>name</i>	Root reference frame name.
-------------	----------------------------

Definition at line 336 of file DSES/ExecutionConfiguration.cpp.

References TrickHLA::Object::name, and root\_frame\_name.

### 7.9.3.18 `set_scenario_time_epoch()`

```
void ExecutionConfiguration::set_scenario_time_epoch (
    double scenario_time ) [virtual]
```

Set the scenario time line epoch.

**Parameters**

<i>scenario_time</i>	Scenario time line epoch.
----------------------	---------------------------

**WARNING:** Only the Master federate should ever set this.

Definition at line 356 of file DSES/ExecutionConfiguration.cpp.

References [TrickHLA::ExecutionConfigurationBase::execution\\_control](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), and [scenario\\_time\\_epoch](#).

Referenced by [DSES::ExecutionControl::set\\_next\\_execution\\_control\\_mode\(\)](#).

**7.9.3.19 setup\_ref\_attributes()**

```
virtual void DSES::ExecutionConfiguration::setup_ref_attributes (
    Packing * packing_obj ) [virtual]
```

Setup the Trick Ref Attributes for the ExCO object.

**Parameters**

<i>packing_obj</i>	Associated packing object.
--------------------	----------------------------

**7.9.3.20 unpack()**

```
void ExecutionConfiguration::unpack ( ) [virtual]
```

This function is called after data is received from the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 245 of file DSES/ExecutionConfiguration.cpp.

References [current\\_execution\\_mode](#), [TrickHLA::DEBUG\\_LEVEL\\_1\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_PACKING](#), [DSES::execution\\_mode\\_enum\\_to\\_string\(\)](#), [DSES::execution\\_mode\\_int16\\_to\\_enum\(\)](#), [TrickHLA::Object::get\\_federate\(\)](#), [TrickHLA::Federate::get\\_granted\\_time\(\)](#), [TrickHLA::Federate::get\\_lookahead\(\)](#), [TrickHLA::Object::get\\_name\(\)](#), [TrickHLA::Federate::get\\_requested\\_time\(\)](#), [TrickHLA::Int64Interval::getTimeInMicros\(\)](#), [next\\_execution\\_mode](#), [next\\_mode\\_cte\\_time](#), [next\\_mode\\_scenario\\_time](#), [TrickHLA::ExecutionConfigurationBase::pending\\_update](#), [root\\_frame\\_name](#), [scenario\\_time\\_epoch](#), [TrickHLA::Packing::should\\_print\(\)](#), and [THLA\\_ENDL](#).

**7.9.3.21 wait\_on\_update()**

```
bool ExecutionConfiguration::wait_on_update ( ) [virtual]
```

Wait on an ExCO update.

**Returns**

True for successful wait.

Reimplemented from [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 632 of file DSES/ExecutionConfiguration.cpp.

References [TrickHLA::Object::any\\_remotely\\_owned\\_subscribed\\_init\\_attribute\(\)](#), [TrickHLA::Federate::check\\_for\\_shutdown\\_with\\_termination\(\)](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::ExecutionConfigurationBase::execution\\_control](#), [TrickHLA::Object::get\\_federate\(\)](#), [TrickHLA::Object::is\\_changed\(\)](#), [TrickHLA::Federate::is\\_execution\\_member\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::Object::receive\\_init\\_data\(\)](#), [TrickHLA::Federate::should\\_print\(\)](#), [THLA\\_ENDL](#), and [THLA\\_NEWLINE](#).

Referenced by [DSES::ExecutionControl::run\\_mode\\_transition\(\)](#).

## 7.9.4 Friends And Related Function Documentation

### 7.9.4.1 init\_attrDSES\_\_ExecutionConfiguration

```
void init_attrDSES__ExecutionConfiguration ( ) [friend]
```

### 7.9.4.2 InputProcessor

```
friend class InputProcessor [friend]
Definition at line 58 of file DSES/ExecutionConfiguration.hh.
```

## 7.9.5 Field Documentation

### 7.9.5.1 current\_execution\_mode

```
short DSES::ExecutionConfiguration::current_execution_mode
```

#### Units: –

Defines the current running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.

Definition at line 88 of file DSES/ExecutionConfiguration.hh.

Referenced by `get_current_execution_mode()`, `DSES::ExecutionControl::is_mtr_valid()`, `pack()`, `print_execution_configuration()`, `DSES::ExecutionControl::process_execution_control_updates()`, `set_current_execution_mode()`, and `unpack()`.

### 7.9.5.2 next\_execution\_mode

```
short DSES::ExecutionConfiguration::next_execution_mode
```

#### Units: –

Defines the next running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.

This is used in conjunction with the `cte_mode_time`, `sim_mode_time` and associated sync point mechanisms to coordinate federation execution mode transitions.

Definition at line 92 of file DSES/ExecutionConfiguration.hh.

Referenced by `get_next_execution_mode()`, `pack()`, `print_execution_configuration()`, `DSES::ExecutionControl::process_execution_control_updates()`, `set_next_execution_mode()`, and `unpack()`.

### 7.9.5.3 next\_mode\_cte\_time

```
double DSES::ExecutionConfiguration::next_mode_cte_time
```

#### Units: s

The time for the next federation execution mode change expressed as a Central Timing Equipment (CTE) time reference. The standard for this reference shall be defined in the federation agreement when CTE is used.

Definition at line 83 of file DSES/ExecutionConfiguration.hh.

Referenced by `get_next_mode_cte_time()`, `pack()`, `print_execution_configuration()`, `DSES::ExecutionControl::process_execution_control_updates()`, `DSES::ExecutionControl::process_mode_transition_request()`, `set_next_mode_cte_time()`, and `unpack()`.

#### 7.9.5.4 `next_mode_scenario_time`

```
double DSES::ExecutionConfiguration::next_mode_scenario_time
```

**Units:** s

The time for the next federation execution mode change expressed as a federation scenario time reference.

Note: this value is only meaningful for going into freeze; exiting freeze is coordinated through a sync point mechanism.

Definition at line 77 of file DSES/ExecutionConfiguration.hh.

Referenced by `get_next_mode_scenario_time()`, `pack()`, `print_execution_configuration()`, `DSES::ExecutionControl::process_execution_control_updates()`, `DSES::ExecutionControl::process_mode_transition_request()`, `set_next_mode_scenario_time()`, and `unpack()`.

#### 7.9.5.5 `root_frame_name`

```
char* DSES::ExecutionConfiguration::root_frame_name
```

**Units:** –

Specifies the name of the root coordinate frame in the federation execution's reference frame tree.

This frame shall remain fixed throughout the federation execution.

Definition at line 65 of file DSES/ExecutionConfiguration.hh.

Referenced by `configure_attributes()`, `get_root_frame_name()`, `pack()`, `print_execution_configuration()`, `set_root_frame_name()`, `unpack()`, and `~ExecutionConfiguration()`.

#### 7.9.5.6 `scenario_time_epoch`

```
double DSES::ExecutionConfiguration::scenario_time_epoch
```

**Units:** s

Federation execution scenario time epoch.

This is the beginning epoch expressed in Terrestrial Time (TT) that corresponds to HLA logical time 0. All joining federates shall use this time to coordinate the offset between their local simulation scenario times, their local simulation execution times and the HLA logical time.

Definition at line 70 of file DSES/ExecutionConfiguration.hh.

Referenced by `get_scenario_time_epoch()`, `pack()`, `print_execution_configuration()`, `DSES::ExecutionControl::process_execution_control_updates()`, `DSES::ExecutionControl::process_mode_transition_request()`, `set_scenario_time_epoch()`, and `unpack()`.

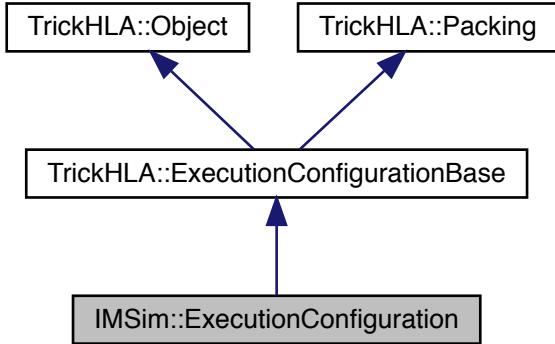
The documentation for this class was generated from the following files:

- [DSES/ExecutionConfiguration.hh](#)
- [DSES/ExecutionConfiguration.cpp](#)

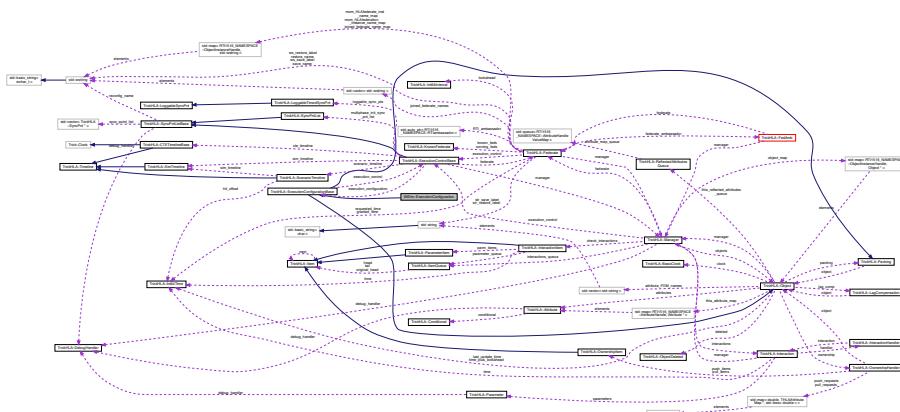
## 7.10 IMSim::ExecutionConfiguration Class Reference

```
#include <ExecutionConfiguration.hh>
```

## Inheritance diagram for IMSim::ExecutionConfiguration:



## Collaboration diagram for IMSim::ExecutionConfiguration:



## Public Member Functions

- [ExecutionConfiguration \(\)](#)

*Default constructor for the `IMSim ExecutionConfiguration` class.*
  - virtual [~ExecutionConfiguration \(\)](#)

*Pure virtual destructor for the `IMSim ExecutionConfiguration` class.*
  - virtual void [configure\\_attributes](#) (const char \*exco\_name)

*Sets up the attributes for the ExCO using default values. These can be overridden in the input file.*
  - virtual void [pack \(\)](#)
  - virtual void [unpack \(\)](#)
  - virtual void [set\\_root\\_frame\\_name](#) (const char \*name)

*Set the root reference frame name.*
  - virtual const char \* [get\\_root\\_frame\\_name \(\)](#)

- `virtual void set_scenario_time_epoch (double scenario_time)`

*Get the root reference frame name.*
- `virtual double get_scenario_time_epoch ()`

*Set the scenario time line epoch.*
- `virtual void set_next_mode_scenario_time (double next_mode_time)`

*Get the scenario time line epoch.*
- `virtual double get_next_mode_scenario_time ()`

*Set the scenario time for the next mode transition.*
- `virtual void set_next_mode_cte_time (double cte_time)`

*Get the next mode scenario time.*
- `virtual double get_next_mode_cte_time ()`

*Set the next mode CTE time.*
- `virtual void set_current_execution_mode (short mode)`

*Sets the current ExCO run mode.*
- `virtual void set_next_execution_mode (IMSim::ExecutionModeEnum mode)`

*Sets the current ExCO run mode.*
- `virtual short get_current_execution_mode ()`

*Get the current execution mode.*
- `virtual void set_next_execution_mode (short mode)`

*Sets the next ExCO execution mode.*
- `virtual void set_next_execution_mode (IMSim::ExecutionModeEnum mode)`

*Sets the next ExCO execution mode.*
- `virtual short get_next_execution_mode ()`

*Get the next execution mode.*
- `virtual void set_least_common_time_step (int64_t lcts)`

*Set the least common time step in microseconds for the federation.*
- `virtual int64_t get_least_common_time_step ()`

*Get the value of the least common time step.*
- `virtual void setup_ref_attributes (Packing *packing_obj)`

*Setup the Trick Ref Attributes for the ExCO object.*
- `virtual void print_execution_configuration ()`

*Print the current ExCO state to the console.*
- `virtual bool wait_on_update ()`

*Wait on an ExCO update.*

## Data Fields

- `char * root_frame_name`

**Units:** –

*Specifies the name of the root coordinate frame in the federation execution's reference frame tree.*
- `double scenario_time_epoch`

**Units:** s

*Federation execution scenario time epoch.*
- `double next_mode_scenario_time`

**Units:** s

*The time for the next federation execution mode change expressed as a federation scenario time reference.*
- `double next_mode_cte_time`

**Units:** s

The time for the next federation execution mode change expressed as a Central Timing Equipment (CTE) time reference.

- short `current_execution_mode`

**Units:** –

Defines the current running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.

- short `next_execution_mode`

**Units:** –

Defines the next running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.

- int64\_t `least_common_time_step`

**Units:** –

A 64 bit integer time that represents microseconds for the least common value of all the time step values in the federation execution (LCTS).

## Private Member Functions

- `ExecutionConfiguration (const ExecutionConfiguration &rhs)`
- `ExecutionConfiguration & operator= (const ExecutionConfiguration &rhs)`

## Friends

- class `InputProcessor`
- void `init_attrIMSim__ExecutionConfiguration ()`

## Additional Inherited Members

### 7.10.1 Detailed Description

Definition at line 50 of file IMSim/ExecutionConfiguration.hh.

### 7.10.2 Constructor & Destructor Documentation

#### 7.10.2.1 `ExecutionConfiguration()` [1/2]

`ExecutionConfiguration::ExecutionConfiguration ( )`  
Default constructor for the `IMSim ExecutionConfiguration` class.

**Trick Job Class:** *initialization*

Definition at line 83 of file IMSim/ExecutionConfiguration.cpp.

References `TrickHLA::Object::name`, and `TrickHLA::Object::packing`.

#### 7.10.2.2 `~ExecutionConfiguration()`

`ExecutionConfiguration::~ExecutionConfiguration ( ) [virtual]`  
Pure virtual destructor for the `IMSim ExecutionConfiguration` class.  
Even though this is a pure virtual destructor, we provide a default implementation that can be called from an inheriting class. **Trick Job Class:** *shutdown*  
Definition at line 108 of file IMSim/ExecutionConfiguration.cpp.  
References `root_frame_name`.

## 7.10.2.3 ExecutionConfiguration() [2/2]

```
IMSim::ExecutionConfiguration::ExecutionConfiguration (
    const ExecutionConfiguration & rhs ) [private]
```

## 7.10.3 Member Function Documentation

## 7.10.3.1 configure\_attributes()

```
void ExecutionConfiguration::configure_attributes (
    const char * exco_name ) [virtual]
```

Sets up the attributes for the ExCO using default values. These can be overridden in the input file.

## Parameters

exco_name	S_define level Trick name for ExCO.
-----------	-------------------------------------

These can be overridden in the input file. **Trick Job Class:** *default\_data*

Definition at line 124 of file IMSim/ExecutionConfiguration.cpp.

References TrickHLA::Object::attr\_count, TrickHLA::Object::attributes, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_UNICODE\_STRING, TrickHLA::Attribute::FOM\_name, TrickHLA::Object::FOM\_name, TrickHLA::Object::name, TrickHLA::Object::packing, root\_frame\_name, TrickHLA::Attribute::rti\_encoding, trick\_MM, and TrickHLA::Attribute::trick\_name.

## 7.10.3.2 get\_current\_execution\_mode()

```
virtual short IMSim::ExecutionConfiguration::get_current_execution_mode ( ) [inline], [virtual]
```

Get the current execution mode.

## Returns

The current execution mode as an integer.

Definition at line 165 of file IMSim/ExecutionConfiguration.hh.

References current\_execution\_mode.

## 7.10.3.3 get\_least\_common\_time\_step()

```
virtual int64_t IMSim::ExecutionConfiguration::get_least_common_time_step ( ) [inline], [virtual]
```

Get the value of the least common time step.

## Returns

The value of the least common time step.

Definition at line 182 of file IMSim/ExecutionConfiguration.hh.

References least\_common\_time\_step.

## 7.10.3.4 get\_next\_execution\_mode()

```
virtual short IMSim::ExecutionConfiguration::get_next_execution_mode ( ) [inline], [virtual]
```

Get the next execution mode.

**Returns**

The next execution mode as an integer.

Definition at line 175 of file IMSim/ExecutionConfiguration.hh.  
References next\_execution\_mode.

**7.10.3.5 get\_next\_mode\_cte\_time()**

```
virtual double IMSim::ExecutionConfiguration::get_next_mode_cte_time ( ) [inline], [virtual]  
Get the next mode CTE time.
```

**Returns**

The next mode CTE time.

Definition at line 155 of file IMSim/ExecutionConfiguration.hh.  
References next\_mode\_cte\_time.  
Referenced by IMSim::ExecutionControl::run\_mode\_transition(), and IMSim::ExecutionControl::set\_next\_execution\_cte\_time().

**7.10.3.6 get\_next\_mode\_scenario\_time()**

```
virtual double IMSim::ExecutionConfiguration::get_next_mode_scenario_time ( ) [inline], [virtual]  
Get the next mode scenario time.
```

**Returns**

The next mode scenario time.

Definition at line 148 of file IMSim/ExecutionConfiguration.hh.  
References next\_mode\_scenario\_time.

**7.10.3.7 get\_root\_frame\_name()**

```
virtual const char* IMSim::ExecutionConfiguration::get_root_frame_name ( ) [inline], [virtual]  
Get the root reference frame name.
```

**Returns**

Root Reference Frame name as a constant string.

Definition at line 134 of file IMSim/ExecutionConfiguration.hh.  
References root\_frame\_name.

**7.10.3.8 get\_scenario\_time\_epoch()**

```
virtual double IMSim::ExecutionConfiguration::get_scenario_time_epoch ( ) [inline], [virtual]  
Get the scenario time line epoch.
```

**Returns**

The scenario time line epoch.

Definition at line 141 of file IMSim/ExecutionConfiguration.hh.  
References scenario\_time\_epoch.

**7.10.3.9 operator=( )**

```
ExecutionConfiguration& IMSim::ExecutionConfiguration::operator= (
    const ExecutionConfiguration & rhs ) [private]
```

**7.10.3.10 pack()**

void ExecutionConfiguration::pack ( ) [virtual]

This function is called before the data is sent to the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 195 of file IMSim/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, TrickHLA::ExecutionConfigurationBase::execution\_control, IMSim::execution\_mode\_enum\_to\_string(), IMSim::execution\_mode\_int16\_to\_enum(), TrickHLA::Object::get\_federate(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Federate::get\_lookahead(), TrickHLA::Object::get\_name(), TrickHLA::Federate::get\_requested\_time(), TrickHLA::ScenarioTimeline::get\_time(), TrickHLA::Int64Interval::getTimeInMicros(), least\_common\_time\_step, next\_execution\_mode, next\_mode\_cte\_time, next\_mode\_scenario\_time, root\_frame\_name, scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Packing::should\_print(), and THLA\_ENDL.

**7.10.3.11 print\_execution\_configuration()**

void ExecutionConfiguration::print\_execution\_configuration ( ) [virtual]

Print the current ExCO state to the console.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 611 of file IMSim/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, IMSim::execution\_mode\_enum\_to\_string(), IMSim::execution\_mode\_int16\_to\_enum(), TrickHLA::Object::get\_name(), least\_common\_time\_step, next\_execution\_mode, next\_mode\_cte\_time, next\_mode\_scenario\_time, root\_frame\_name, scenario\_time\_epoch, and THLA\_ENDL.

**7.10.3.12 set\_current\_execution\_mode() [1/2]**

```
void ExecutionConfiguration::set_current_execution_mode (
    IMSim::ExecutionModeEnum mode ) [virtual]
```

Sets the current ExCO run mode.

**Parameters**

<i>mode</i>	Current Execution configuration run mode enumeration value.
-------------	---

**WARNING:** Only the Master federate should ever set this.

Definition at line 410 of file IMSim/ExecutionConfiguration.cpp.

References IMSim::execution\_mode\_enum\_to\_int16(), and set\_current\_execution\_mode().

**7.10.3.13 set\_current\_execution\_mode() [2/2]**

```
void ExecutionConfiguration::set_current_execution_mode (
    short mode ) [virtual]
```

Sets the current ExCO run mode.

**Parameters**

<i>mode</i>	Current Execution configuration run mode integer value.
-------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 397 of file IMSim/ExecutionConfiguration.cpp.

References `current_execution_mode`, `TrickHLA::ExecutionConfigurationBase::execution_control`, and `TrickHLA::ExecutionControlBase::is_master()`.

Referenced by `IMSim::ExecutionControl::freeze_mode_transition()`, `IMSim::ExecutionControl::run_mode_transition()`, and `set_current_execution_mode()`.

**7.10.3.14 set\_least\_common\_time\_step()**

```
virtual void IMSim::ExecutionConfiguration::set_least_common_time_step (
    int64_t lcts ) [virtual]
```

Set the least common time step in microseconds for the federation.

**Parameters**

<i>lcts</i>	Least Common Time Step time in microseconds.
-------------	--

**7.10.3.15 set\_next\_execution\_mode() [1/2]**

```
void ExecutionConfiguration::set_next_execution_mode (
    IMSim::ExecutionModeEnum mode ) [virtual]
```

Sets the next ExCO execution mode.

**Parameters**

<i>mode</i>	Next Execution configuration execution mode from an enumeration value.
-------------	--

WARNING: Only the Master federate should ever set this.

Definition at line 434 of file IMSim/ExecutionConfiguration.cpp.

References `IMSim::execution_mode_enum_to_int16()`, and `set_next_execution_mode()`.

**7.10.3.16 set\_next\_execution\_mode() [2/2]**

```
void ExecutionConfiguration::set_next_execution_mode (
    short mode ) [virtual]
```

Sets the next ExCO execution mode.

**Parameters**

<i>mode</i>	Next Execution configuration execution mode from an integer.
-------------	--

WARNING: Only the Master federate should ever set this.

Definition at line 421 of file IMSim/ExecutionConfiguration.cpp.

References `TrickHLA::ExecutionConfigurationBase::execution_control`, `TrickHLA::ExecutionControlBase::is_master()`, and `next_execution_mode()`.

Referenced by `IMSim::ExecutionControl::set_next_execution_control_mode()`, and `set_next_execution_mode()`.

#### 7.10.3.17 set\_next\_mode\_cte\_time()

```
void ExecutionConfiguration::set_next_mode_cte_time (
```

double	cte_time	) [virtual]
--------	----------	-------------

Set the next mode CTE time.

##### Parameters

<i>cte_time</i>	CTE time for next mode transition.
-----------------	------------------------------------

WARNING: Only the Master federate should ever set this.

Definition at line 383 of file IMSim/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_mode\_cte\_time.

Referenced by IMSim::ExecutionControl::set\_next\_execution\_control\_mode().

#### 7.10.3.18 set\_next\_mode\_scenario\_time()

```
void ExecutionConfiguration::set_next_mode_scenario_time (
```

double	next_mode_time	) [virtual]
--------	----------------	-------------

Set the scenario time for the next mode transition.

##### Parameters

<i>next_mode_time</i>	Scenario time for next mode transition.
-----------------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 369 of file IMSim/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_mode\_scenario\_time.

Referenced by IMSim::ExecutionControl::set\_next\_execution\_control\_mode().

#### 7.10.3.19 set\_root\_frame\_name()

```
void ExecutionConfiguration::set_root_frame_name (
```

const char *	<i>name</i>	) [virtual]
--------------	-------------	-------------

Set the root reference frame name.

##### Parameters

<i>name</i>	Root reference frame name.
-------------	----------------------------

Definition at line 336 of file IMSim/ExecutionConfiguration.cpp.

References TrickHLA::Object::name, and root\_frame\_name.

#### 7.10.3.20 set\_scenario\_time\_epoch()

```
void ExecutionConfiguration::set_scenario_time_epoch (
```

double	scenario_time	) [virtual]
--------	---------------	-------------

Set the scenario time line epoch.

**Parameters**

<i>scenario_time</i>	Scenario time line epoch.
----------------------	---------------------------

**WARNING:** Only the Master federate should ever set this.

Definition at line 356 of file IMSim/ExecutionConfiguration.cpp.

References [TrickHLA::ExecutionConfigurationBase::execution\\_control](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), and [scenario\\_time\\_epoch](#).

Referenced by [IMSim::ExecutionControl::set\\_next\\_execution\\_control\\_mode\(\)](#).

**7.10.3.21 setup\_ref\_attributes()**

```
virtual void IMSim::ExecutionConfiguration::setup_ref_attributes (
    Packing * packing_obj ) [virtual]
```

Setup the Trick Ref Attributes for the ExCO object.

**Parameters**

<i>packing_obj</i>	Associated packing object.
--------------------	----------------------------

**7.10.3.22 unpack()**

```
void ExecutionConfiguration::unpack ( ) [virtual]
```

This function is called after data is received from the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 245 of file IMSim/ExecutionConfiguration.cpp.

References [current\\_execution\\_mode](#), [TrickHLA::DEBUG\\_LEVEL\\_1\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_PACKING](#), [TrickHLA::ExecutionConfigurationBase::execution\\_control](#), [IMSim::execution\\_mode\\_enum\\_to\\_string\(\)](#), [IMSim::execution\\_mode\\_int16\\_to\\_enum\(\)](#), [TrickHLA::Object::get\\_federate\(\)](#), [TrickHLA::Federate::get\\_granted\\_time\(\)](#), [TrickHLA::Federate::get\\_lookahead\(\)](#), [TrickHLA::Object::get\\_name\(\)](#), [TrickHLA::Federate::get\\_requested\\_time\(\)](#), [TrickHLA::ScenarioTimeline::get\\_time\(\)](#), [TrickHLA::Int64Interval::getTimeInMicros\(\)](#), [least\\_common\\_time\\_step](#), [next\\_execution\\_mode](#), [next\\_mode\\_cte\\_time](#), [next\\_mode\\_scenario\\_time](#), [TrickHLA::ExecutionConfigurationBase::pending\\_update](#), [root\\_frame\\_name](#), [scenario\\_time\\_epoch](#), [TrickHLA::ExecutionControlBase::scenario\\_timeline](#), [TrickHLA::Packing::should\\_print\(\)](#), and [THLA\\_ENDL](#).

**7.10.3.23 wait\_on\_update()**

```
bool ExecutionConfiguration::wait_on_update ( ) [virtual]
```

Wait on an ExCO update.

**Returns**

True for successful wait.

Reimplemented from [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 632 of file IMSim/ExecutionConfiguration.cpp.

References [TrickHLA::Object::any\\_remotely\\_owned\\_subscribed\\_init\\_attribute\(\)](#), [TrickHLA::Federate::check\\_for\\_shutdown\\_with\\_termination\(\)](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::ExecutionConfigurationBase::execution\\_control](#), [TrickHLA::Object::get\\_federate\(\)](#), [TrickHLA::Object::is\\_changed\(\)](#), [TrickHLA::Federate::is\\_execution\\_member\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::Object::receive\\_init\\_data\(\)](#), [TrickHLA::Federate::should\\_print\(\)](#), [THLA\\_ENDL](#), and [THLA\\_NEWLINE](#).

Referenced by [IMSim::ExecutionControl::run\\_mode\\_transition\(\)](#).

## 7.10.4 Friends And Related Function Documentation

### 7.10.4.1 `init_attrIMSim__ExecutionConfiguration`

```
void init_attrIMSim__ExecutionConfiguration ( ) [friend]
```

### 7.10.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 58 of file IMSim/ExecutionConfiguration.hh.

## 7.10.5 Field Documentation

### 7.10.5.1 `current_execution_mode`

```
short IMSim::ExecutionConfiguration::current_execution_mode
```

**Units:** –

Defines the current running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.

Definition at line 88 of file IMSim/ExecutionConfiguration.hh.

Referenced by `get_current_execution_mode()`, `IMSim::ExecutionControl::is_mtr_valid()`, `pack()`, `print_execution_configuration()`, `IMSim::ExecutionControl::process_execution_control_updates()`, `set_current_execution_mode()`, and `unpack()`.

### 7.10.5.2 `least_common_time_step`

```
int64_t IMSim::ExecutionConfiguration::least_common_time_step
```

**Units:** –

A 64 bit integer time that represents microseconds for the least common value of all the time step values in the federation execution (LCTS).

This value is set by the Master Federate and does not change during the federation execution. This is used in the computation to find the next HLA Logical Time Boundary (HLTB) available to all federates in the federation execution. The basic equation is  $HLTB = (\text{floor}(GALT/LCTS) + 1) * LCTS$ , where GALT is the greatest available logical time. This is used to synchronize the federates in a federation execution to be on a common logical time boundary.

Definition at line 99 of file IMSim/ExecutionConfiguration.hh.

Referenced by `get_least_common_time_step()`, `pack()`, `print_execution_configuration()`, and `unpack()`.

### 7.10.5.3 `next_execution_mode`

```
short IMSim::ExecutionConfiguration::next_execution_mode
```

**Units:** –

Defines the next running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.

This is used in conjunction with the `cte_mode_time`, `sim_mode_time` and associated sync point mechanisms to coordinate federation execution mode transitions.

Definition at line 92 of file IMSim/ExecutionConfiguration.hh.

Referenced by `get_next_execution_mode()`, `pack()`, `print_execution_configuration()`, `IMSim::ExecutionControl::process_execution_control_updates()`, `set_next_execution_mode()`, and `unpack()`.

#### 7.10.5.4 next\_mode\_cte\_time

```
double IMSim::ExecutionConfiguration::next_mode_cte_time
```

**Units:** s

The time for the next federation execution mode change expressed as a Central Timing Equipment (CTE) time reference. The standard for this reference shall be defined in the federation agreement when CTE is used.

Definition at line 83 of file IMSim/ExecutionConfiguration.hh.

Referenced by `get_next_mode_cte_time()`, `pack()`, `print_execution_configuration()`, `IMSim::ExecutionControl::process_execution_control_updates()`, `IMSim::ExecutionControl::process_mode_transition_request()`, `set_next_mode_cte_time()`, and `unpack()`.

#### 7.10.5.5 next\_mode\_scenario\_time

```
double IMSim::ExecutionConfiguration::next_mode_scenario_time
```

**Units:** s

The time for the next federation execution mode change expressed as a federation scenario time reference.

Note: this value is only meaningful for going into freeze; exiting freeze is coordinated through a sync point mechanism. Definition at line 77 of file IMSim/ExecutionConfiguration.hh.

Referenced by `get_next_mode_scenario_time()`, `pack()`, `print_execution_configuration()`, `IMSim::ExecutionControl::process_execution_control_updates()`, `IMSim::ExecutionControl::process_mode_transition_request()`, `set_next_mode_scenario_time()`, and `unpack()`.

#### 7.10.5.6 root\_frame\_name

```
char* IMSim::ExecutionConfiguration::root_frame_name
```

**Units:** –

Specifies the name of the root coordinate frame in the federation execution's reference frame tree.

This frame shall remain fixed throughout the federation execution.

Definition at line 65 of file IMSim/ExecutionConfiguration.hh.

Referenced by `configure_attributes()`, `get_root_frame_name()`, `pack()`, `print_execution_configuration()`, `set_root_frame_name()`, `unpack()`, and `~ExecutionConfiguration()`.

#### 7.10.5.7 scenario\_time\_epoch

```
double IMSim::ExecutionConfiguration::scenario_time_epoch
```

**Units:** s

Federation execution scenario time epoch.

This is the beginning epoch expressed in Terrestrial Time (TT) that corresponds to HLA logical time 0. All joining federates shall use this time to coordinate the offset between their local simulation scenario times, their local simulation execution times and the HLA logical time.

Definition at line 70 of file IMSim/ExecutionConfiguration.hh.

Referenced by `get_scenario_time_epoch()`, `pack()`, `print_execution_configuration()`, `IMSim::ExecutionControl::process_execution_control_updates()`, `IMSim::ExecutionControl::process_mode_transition_request()`, `set_scenario_time_epoch()`, and `unpack()`.

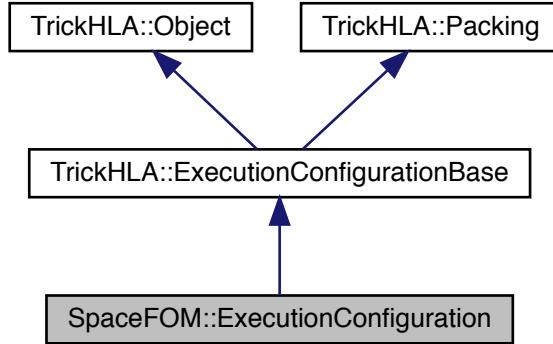
The documentation for this class was generated from the following files:

- [IMSim/ExecutionConfiguration.hh](#)
- [IMSim/ExecutionConfiguration.cpp](#)

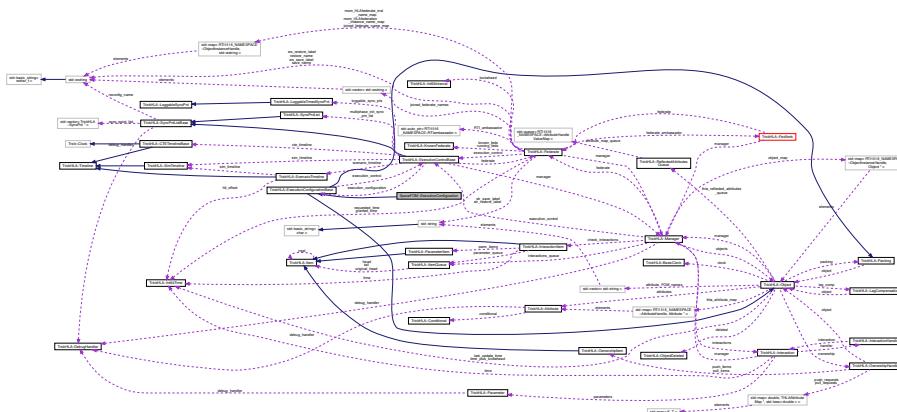
## 7.11 SpaceFOM::ExecutionConfiguration Class Reference

```
#include <ExecutionConfiguration.hh>
```

Inheritance diagram for SpaceFOM::ExecutionConfiguration:



Collaboration diagram for SpaceFOM::ExecutionConfiguration:



## Public Member Functions

- `ExecutionConfiguration ()`  
*Default constructor for the `SpaceFOM ExecutionConfiguration` class.*
- `ExecutionConfiguration (const char *s_define_name)`  
*Initialization constructor for the `TrickHLA ExecutionConfiguration` class.*
- `virtual ~ExecutionConfiguration ()`  
*Pure virtual destructor for the `SpaceFOM ExecutionConfiguration` class.*
- `virtual void configure_attributes ()`  
*Sets up the attributes for the ExCO using default values. These can be overridden in the input file.*
- `virtual void configure ()`  
*Configure the execution configuration object.*
- `virtual void pack ()`

- virtual void [unpack \(\)](#)
- virtual void [set\\_root\\_frame\\_name \(const char \\*name\)](#)  
*Set the root reference frame name.*
- virtual const char \* [get\\_root\\_frame\\_name \(\)](#)  
*Get the root reference frame name.*
- virtual void [set\\_scenario\\_time\\_epoch \(double scenario\\_time\)](#)  
*Set the scenario time line epoch.*
- virtual double [get\\_scenario\\_time\\_epoch \(\)](#)  
*Get the scenario time line epoch.*
- virtual void [set\\_next\\_mode\\_scenario\\_time \(double next\\_mode\\_time\)](#)  
*Set the scenario time for the next mode transition.*
- virtual double [get\\_next\\_mode\\_scenario\\_time \(\)](#)  
*Get the next mode scenario time.*
- virtual void [set\\_next\\_mode\\_cte\\_time \(double cte\\_time\)](#)  
*Set the next mode CTE time.*
- virtual double [get\\_next\\_mode\\_cte\\_time \(\)](#)  
*Get the next mode CTE time.*
- virtual void [set\\_current\\_execution\\_mode \(short mode\)](#)  
*Sets the current ExCO run mode.*
- virtual void [set\\_current\\_execution\\_mode \(SpaceFOM::ExecutionModeEnum mode\)](#)  
*Sets the current ExCO run mode.*
- virtual short [get\\_current\\_execution\\_mode \(\)](#)  
*Get the current execution mode.*
- virtual void [set\\_next\\_execution\\_mode \(short mode\)](#)  
*Sets the next ExCO execution mode.*
- virtual void [set\\_next\\_execution\\_mode \(SpaceFOM::ExecutionModeEnum mode\)](#)  
*Sets the next ExCO execution mode.*
- virtual short [get\\_next\\_execution\\_mode \(\)](#)  
*Get the next execution mode.*
- virtual void [set\\_least\\_common\\_time\\_step \(int64\\_t lcts\)](#)  
*Set the least common time step in microseconds for the federation.*
- virtual int64\_t [get\\_least\\_common\\_time\\_step \(\)](#)  
*Get the value of the least common time step.*
- virtual void [setup\\_ref\\_attributes \(TrickHLA::Packing \\*packing\\_obj\)](#)  
*Setup the Trick Ref Attributes for the ExCO object.*
- virtual void [print\\_execution\\_configuration \(\)](#)  
*Print the current ExCO state to the console.*
- virtual bool [wait\\_on\\_update \(\)](#)  
*Wait on an ExCO update.*

## Data Fields

- char \* [root\\_frame\\_name](#)

**Units:** –  
*Specifies the name of the root coordinate frame in the federation execution's reference frame tree.*
- double [scenario\\_time\\_epoch](#)

**Units:** s  
*Federation execution scenario time epoch.*

- double `next_mode_scenario_time`

**Units:** s  
*The time for the next federation execution mode change expressed as a federation scenario time reference.*
- double `next_mode_cte_time`

**Units:** s  
*The time for the next federation execution mode change expressed as a Central Timing Equipment (CTE) time reference.*
- short `current_execution_mode`

**Units:** –  
*Defines the current running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.*
- short `next_execution_mode`

**Units:** –  
*Defines the next running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.*
- int64\_t `least_common_time_step`

**Units:** –  
*A 64 bit integer time that represents microseconds for the least common value of all the time step values in the federation execution (LCTS).*

## Private Member Functions

- `ExecutionConfiguration (const ExecutionConfiguration &rhs)`
- `ExecutionConfiguration & operator= (const ExecutionConfiguration &rhs)`

## Friends

- class `InputProcessor`
- void `init_attrSpaceFOM__ExecutionConfiguration ()`

## Additional Inherited Members

### 7.11.1 Detailed Description

Definition at line 50 of file SpaceFOM/ExecutionConfiguration.hh.

### 7.11.2 Constructor & Destructor Documentation

#### 7.11.2.1 `ExecutionConfiguration()` [1/3]

`ExecutionConfiguration::ExecutionConfiguration ( )`  
 Default constructor for the `SpaceFOM ExecutionConfiguration` class.

**Trick Job Class:** `initialization`

Definition at line 83 of file SpaceFOM/ExecutionConfiguration.cpp.

#### 7.11.2.2 `ExecutionConfiguration()` [2/3]

`ExecutionConfiguration::ExecutionConfiguration (`  
`const char * s_define_name ) [explicit]`  
 Initialization constructor for the `TrickHLA ExecutionConfiguration` class.

**Parameters**

<code>s_define_name</code>	Full path name in the S_define for this <a href="#">ExecutionConfiguration</a> instance.
----------------------------	--

**Trick Job Class:** *initialization*

Definition at line 98 of file SpaceFOM/ExecutionConfiguration.cpp.

**7.11.2.3 ~ExecutionConfiguration()**

```
ExecutionConfiguration::~ExecutionConfiguration ( ) [virtual]
```

Pure virtual destructor for the [SpaceFOM ExecutionConfiguration](#) class.

Even though this is a pure virtual destructor, we provide a default implementation that can be called from an inheriting class. **Trick Job Class:** *shutdown*

Definition at line 117 of file SpaceFOM/ExecutionConfiguration.cpp.

References `root_frame_name`.

**7.11.2.4 ExecutionConfiguration() [3/3]**

```
SpaceFOM::ExecutionConfiguration::ExecutionConfiguration ( 
    const ExecutionConfiguration & rhs ) [private]
```

**7.11.3 Member Function Documentation****7.11.3.1 configure()**

```
void ExecutionConfiguration::configure ( ) [virtual]
```

Configure the execution configuration object.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 214 of file SpaceFOM/ExecutionConfiguration.cpp.

References `TrickHLA::LAG_COMPENSATION_NONE`, `TrickHLA::Object::manager`, `TrickHLA::Object::name`, `TrickHLA::Object::ownership`, `TrickHLA::ExecutionConfigurationBase::reset_preferred_order()`, `TrickHLA::Object::set_lag_compensation_type()`, and `THLA_ENDL`.

**7.11.3.2 configure\_attributes()**

```
void ExecutionConfiguration::configure_attributes ( ) [virtual]
```

Sets up the attributes for the ExCO using default values. These can be overridden in the input file.

These can be overridden in the input file. **Trick Job Class:** *default\_data*

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 133 of file SpaceFOM/ExecutionConfiguration.cpp.

References `TrickHLA::Object::attr_count`, `TrickHLA::Object::attributes`, `TrickHLA::ENCODING_LITTLE_ENDIAN`, `TrickHLA::ENCODING_UNICODE_STRING`, `TrickHLA::Attribute::FOM_name`, `TrickHLA::Object::FOM_name`, `TrickHLA::Object::name`, `TrickHLA::Object::packing`, `root_frame_name`, `TrickHLA::Attribute::rti_encoding`, `TrickHLA::ExecutionConfigurationBase::S_define_name`, `THLA_ENDL`, `trick_MM`, and `TrickHLA::Attribute::trick_name`.

### 7.11.3.3 `get_current_execution_mode()`

```
virtual short SpaceFOM::ExecutionConfiguration::get_current_execution_mode ( ) [inline], [virtual]  
Get the current execution mode.
```

#### Returns

The current execution mode as an integer.

Definition at line 170 of file SpaceFOM/ExecutionConfiguration.hh.

References `current_execution_mode`.

### 7.11.3.4 `get_least_common_time_step()`

```
virtual int64_t SpaceFOM::ExecutionConfiguration::get_least_common_time_step ( ) [inline], [virtual]  
Get the value of the least common time step.
```

#### Returns

The value of the least common time step.

Definition at line 187 of file SpaceFOM/ExecutionConfiguration.hh.

References `least_common_time_step`.

Referenced by `SpaceFOM::ExecutionControl::pre_multi_phase_init_processes()`.

### 7.11.3.5 `get_next_execution_mode()`

```
virtual short SpaceFOM::ExecutionConfiguration::get_next_execution_mode ( ) [inline], [virtual]  
Get the next execution mode.
```

#### Returns

The next execution mode as an integer.

Definition at line 180 of file SpaceFOM/ExecutionConfiguration.hh.

References `next_execution_mode`.

### 7.11.3.6 `get_next_mode_cte_time()`

```
virtual double SpaceFOM::ExecutionConfiguration::get_next_mode_cte_time ( ) [inline], [virtual]  
Get the next mode CTE time.
```

#### Returns

The next mode CTE time.

Definition at line 160 of file SpaceFOM/ExecutionConfiguration.hh.

References `next_mode_cte_time`.

Referenced by `SpaceFOM::ExecutionControl::run_mode_transition()`, and `SpaceFOM::ExecutionControl::set_next_execution_control_mode()`.

### 7.11.3.7 `get_next_mode_scenario_time()`

```
virtual double SpaceFOM::ExecutionConfiguration::get_next_mode_scenario_time ( ) [inline], [virtual]  
Get the next mode scenario time.
```

**Returns**

The next mode scenario time.

Definition at line 153 of file SpaceFOM/ExecutionConfiguration.hh.

References next\_mode\_scenario\_time.

**7.11.3.8 get\_root\_frame\_name()**

```
virtual const char* SpaceFOM::ExecutionConfiguration::get_root_frame_name ( ) [inline], [virtual]
Get the root reference frame name.
```

**Returns**

Root Reference Frame name as a constant string.

Definition at line 139 of file SpaceFOM/ExecutionConfiguration.hh.

References root\_frame\_name.

**7.11.3.9 get\_scenario\_time\_epoch()**

```
virtual double SpaceFOM::ExecutionConfiguration::get_scenario_time_epoch ( ) [inline], [virtual]
Get the scenario time line epoch.
```

**Returns**

The scenario time line epoch.

Definition at line 146 of file SpaceFOM/ExecutionConfiguration.hh.

References scenario\_time\_epoch.

Referenced by SpaceFOM::ExecutionControl::epoch\_and\_root\_frame\_discovery\_process(), and SpaceFOM::ExecutionControl::late\_joine\_hla\_init\_process().

**7.11.3.10 operator=()**

```
ExecutionConfiguration& SpaceFOM::ExecutionConfiguration::operator= (
    const ExecutionConfiguration & rhs ) [private]
```

**7.11.3.11 pack()**

```
void ExecutionConfiguration::pack ( ) [virtual]
```

This function is called before the data is sent to the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 253 of file SpaceFOM/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PAC KING, TrickHLA::ExecutionConfigurationBase::execution\_control, SpaceFOM::execution\_mode\_enum\_to\_string(), SpaceFOM::execution\_mode\_int16\_to\_enum(), TrickHLA::Object::get\_federate(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Federate::get\_looking(), TrickHLA::Object::get\_name(), TrickHLA::Federate::get\_requested\_time(), TrickHLA::ScenarioTimeline::get\_time(), TrickHLA::Int64Interval::getTimeInMicros(), least\_common\_time\_step, next\_execution\_mode, next\_mode\_cte\_time, next\_mode\_scenario\_time, root\_frame\_name, scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Packing::should\_print(), and THLA\_ENDL.

### 7.11.3.12 print\_execution\_configuration()

```
void ExecutionConfiguration::print_execution_configuration ( ) [virtual]
```

Print the current ExCO state to the console.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 678 of file SpaceFOM/ExecutionConfiguration.cpp.

References `current_execution_mode`, `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_SOURCE_PACKING`, `SpaceFOM::execution_mode_enum_to_string()`, `SpaceFOM::execution_mode_int16_to_enum()`, `TrickHLA::Object::get_name()`, `least_common_time_step`, `next_execution_mode`, `next_mode_cte_time`, `next_mode_scenario_time`, `root_frame_name`, `scenario_time_epoch`, and `THLA_ENDL`.

Referenced by `SpaceFOM::ExecutionControl::post_multi_phase_init_processes()`.

### 7.11.3.13 set\_current\_execution\_mode() [1/2]

```
void ExecutionConfiguration::set_current_execution_mode (
    short mode ) [virtual]
```

Sets the current ExCO run mode.

#### Parameters

<code>mode</code>	Current Execution configuration run mode integer value.
-------------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 455 of file SpaceFOM/ExecutionConfiguration.cpp.

References `current_execution_mode`, `TrickHLA::ExecutionConfigurationBase::execution_control`, and `TrickHLA::ExecutionControlBase::is_master()`.

Referenced by `SpaceFOM::ExecutionControl::freeze_mode_transition()`, `SpaceFOM::ExecutionControl::pre_multi_phase_init_processes()`, `SpaceFOM::ExecutionControl::run_mode_transition()`, and `set_current_execution_mode()`.

### 7.11.3.14 set\_current\_execution\_mode() [2/2]

```
void ExecutionConfiguration::set_current_execution_mode (
    SpaceFOM::ExecutionModeEnum mode ) [virtual]
```

Sets the current ExCO run mode.

#### Parameters

<code>mode</code>	Current Execution configuration run mode enumeration value.
-------------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 468 of file SpaceFOM/ExecutionConfiguration.cpp.

References `SpaceFOM::execution_mode_enum_to_int16()`, and `set_current_execution_mode()`.

### 7.11.3.15 set\_least\_common\_time\_step()

```
void ExecutionConfiguration::set_least_common_time_step (
    int64_t lcts ) [virtual]
```

Set the least common time step in microseconds for the federation.

#### Parameters

<code>lcts</code>	Least Common Time Step time in microseconds.
-------------------	--

WARNING: Only the Master federate should ever set this.

Definition at line 503 of file SpaceFOM/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and least\_common\_time\_step.

Referenced by SpaceFOM::ExecutionControl::set\_least\_common\_time\_step().

#### 7.11.3.16 set\_next\_execution\_mode() [1/2]

```
void ExecutionConfiguration::set_next_execution_mode (
    short mode ) [virtual]
```

Sets the next ExCO execution mode.

##### Parameters

<i>mode</i>	Next Execution configuration execution mode from an integer.
-------------	--

WARNING: Only the Master federate should ever set this.

Definition at line 479 of file SpaceFOM/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_execution\_mode.

Referenced by SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), and set\_next\_execution\_mode().

#### 7.11.3.17 set\_next\_execution\_mode() [2/2]

```
void ExecutionConfiguration::set_next_execution_mode (
    SpaceFOM::ExecutionModeEnum mode ) [virtual]
```

Sets the next ExCO execution mode.

##### Parameters

<i>mode</i>	Next Execution configuration execution mode from an enumeration value.
-------------	--

WARNING: Only the Master federate should ever set this.

Definition at line 492 of file SpaceFOM/ExecutionConfiguration.cpp.

References SpaceFOM::execution\_mode\_enum\_to\_int16(), and set\_next\_execution\_mode().

#### 7.11.3.18 set\_next\_mode\_cte\_time()

```
void ExecutionConfiguration::set_next_mode_cte_time (
    double cte_time ) [virtual]
```

Set the next mode CTE time.

##### Parameters

<i>cte_time</i>	CTE time for next mode transition.
-----------------	------------------------------------

WARNING: Only the Master federate should ever set this.

Definition at line 441 of file SpaceFOM/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_mode\_cte\_time.

Referenced by SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode().

### 7.11.3.19 set\_next\_mode\_scenario\_time()

```
void ExecutionConfiguration::set_next_mode_scenario_time (
    double next_mode_time ) [virtual]
```

Set the scenario time for the next mode transition.

#### Parameters

<i>next_mode_time</i>	Scenario time for next mode transition.
-----------------------	---

WARNING: Only the Master federate should ever set this.

Definition at line 427 of file SpaceFOM/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and next\_mode\_scenario\_time.

Referenced by SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode().

### 7.11.3.20 set\_root\_frame\_name()

```
void ExecutionConfiguration::set_root_frame_name (
    const char * name ) [virtual]
```

Set the root reference frame name.

#### Parameters

<i>name</i>	Root reference frame name.
-------------	----------------------------

Definition at line 394 of file SpaceFOM/ExecutionConfiguration.cpp.

References TrickHLA::Object::name, and root\_frame\_name.

Referenced by SpaceFOM::ExecutionControl::epoch\_and\_root\_frame\_discovery\_process().

### 7.11.3.21 set\_scenario\_time\_epoch()

```
void ExecutionConfiguration::set_scenario_time_epoch (
    double scenario_time ) [virtual]
```

Set the scenario time line epoch.

#### Parameters

<i>scenario_time</i>	Scenario time line epoch.
----------------------	---------------------------

WARNING: Only the Master federate should ever set this.

Definition at line 414 of file SpaceFOM/ExecutionConfiguration.cpp.

References TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::ExecutionControlBase::is\_master(), and scenario\_time\_epoch.

Referenced by SpaceFOM::ExecutionControl::epoch\_and\_root\_frame\_discovery\_process(), and SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode().

### 7.11.3.22 setup\_ref\_attributes()

```
void ExecutionConfiguration::setup_ref_attributes (
    TrickHLA::Packing * packing_obj ) [virtual]
```

Setup the Trick Ref Attributes for the ExCO object.

#### Parameters

<i>packing_obj</i>	Associated packing object.
--------------------	----------------------------

WARNING: This function is BROKEN!

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 445 of file DIS/ExecutionConfiguration.cpp.

References TrickHLA::Object::attr\_count, attrDIS\_\_ExecutionConfiguration, TrickHLA::Object::attributes, TrickHLA::Object::blocking\_cyclic\_read, TrickHLA::Attribute::config, TrickHLA::CONFIG\_INTERMITTENT, TrickHLA::Object::create\_HLA\_instance, TrickHLA::Object::data\_changed, TrickHLA::Manager::debug\_handler, TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_LEVEL\_9\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::Object::deleted, TrickHLA::ENCODING\_UNICODE\_STRING, TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::Attribute::FOM\_name, TrickHLA::Object::FOM\_name, TrickHLA::Object::get\_federate(), TrickHLA::Attribute::get\_FOM\_name(), TrickHLA::Federate::get\_manager(), TrickHLA::Attribute::initialize(), TrickHLA::Object::initialize(), TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::Object::lag\_comp, TrickHLA::Object::lag\_comp\_type, TrickHLA::LAG\_COMPENSATION\_NONE, TrickHLA::Attribute::locally\_owned, TrickHLA::Object::name, TrickHLA::Object::name\_required, TrickHLA::Object::object\_deleted\_from\_RTI, TrickHLA::Object::ownership, TrickHLA::Object::packing, TrickHLA::Attribute::publish, TrickHLA::Object::required, TrickHLA::Attribute::rti\_encoding, TrickHLA::Attribute::set\_debug\_level(), TrickHLA::DebugHandler::should\_print(), TrickHLA::Federate::should\_print(), TrickHLA::Attribute::subscribe, THLA\_NEWLINE, and trick\_MM.

### 7.11.3.23 unpack()

```
void ExecutionConfiguration::unpack ( ) [virtual]
```

This function is called after data is received from the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 303 of file SpaceFOM/ExecutionConfiguration.cpp.

References current\_execution\_mode, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PAC KING, TrickHLA::ExecutionConfigurationBase::execution\_control, SpaceFOM::execution\_mode\_enum\_to\_string(), SpaceFOM::execution\_mode\_int16\_to\_enum(), TrickHLA::Object::get\_federate(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Federate::get\_looking\_ahead(), TrickHLA::Object::get\_name(), TrickHLA::Federate::get\_requested\_time(), TrickHLA::ScenarioTimeline::get\_time(), TrickHLA::Int64Interval::getTimeInMicros(), least\_common\_time\_step, next\_execution\_mode, next\_mode\_cte\_time, next\_mode\_scenario\_time, TrickHLA::ExecutionConfigurationBase::pending\_update, root\_frame\_name, scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Packing::should\_print(), and THLA\_ENDL.

### 7.11.3.24 wait\_on\_update()

```
bool ExecutionConfiguration::wait_on_update ( ) [virtual]
```

Wait on an ExCO update.

#### Returns

True for successful wait.

Reimplemented from [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 699 of file SpaceFOM/ExecutionConfiguration.cpp.

References TrickHLA::Object::any\_remotely\_owned\_subscribed\_init\_attribute(), TrickHLA::Federate::check\_for\_shutdown\_with\_termination(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ExecutionConfigurationBase::execution\_control, TrickHLA::Object::get\_federate(), TrickHLA::Object::is\_changed(), TrickHLA::Federate::is\_execution\_member(), TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::Object::receive\_init\_data(), TrickHLA::Federate::should\_print(), THLA\_ENDL, and THLA\_NEWLINE.  
 Referenced by SpaceFOM::ExecutionControl::epoch\_and\_root\_frame\_discovery\_process(), SpaceFOM::ExecutionControl::late\_joinder\_hla\_init\_process(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), and SpaceFOM::ExecutionControl::run\_mode\_transition().

## 7.11.4 Friends And Related Function Documentation

### 7.11.4.1 init\_attrSpaceFOM\_ExecutionConfiguration

```
void init_attrSpaceFOM_ExecutionConfiguration ( ) [friend]
```

### 7.11.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 58 of file SpaceFOM/ExecutionConfiguration.hh.

## 7.11.5 Field Documentation

### 7.11.5.1 current\_execution\_mode

```
short SpaceFOM::ExecutionConfiguration::current_execution_mode
```

#### Units: –

Defines the current running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.

Definition at line 88 of file SpaceFOM/ExecutionConfiguration.hh.

Referenced by get\_current\_execution\_mode(), SpaceFOM::ExecutionControl::is\_mtr\_valid(), pack(), print\_execution\_configuration(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), set\_current\_execution\_mode(), and unpack().

### 7.11.5.2 least\_common\_time\_step

```
int64_t SpaceFOM::ExecutionConfiguration::least_common_time_step
```

#### Units: –

A 64 bit integer time that represents microseconds for the least common value of all the time step values in the federation execution (LCTS).

This value is set by the Master Federate and does not change during the federation execution. This is used in the computation to find the next HLA Logical Time Boundary (HLTB) available to all federates in the federation execution. The basic equation is HLTB = ( floor(GALT/LCTS) + 1 ) \* LCTS, where GALT is the greatest available logical time. This is used to synchronize the federates in a federation execution to be on a common logical time boundary.

Definition at line 99 of file SpaceFOM/ExecutionConfiguration.hh.

Referenced by get\_least\_common\_time\_step(), pack(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), print\_execution\_configuration(), set\_least\_common\_time\_step(), and unpack().

### 7.11.5.3 `next_execution_mode`

```
short SpaceFOM::ExecutionConfiguration::next_execution_mode
```

**Units:** –

Defines the next running state of the federation execution in terms of a finite set of states expressed in the RunMode enumeration.

This is used in conjunction with the cte\_mode\_time, sim\_mode\_time and associated sync point mechanisms to coordinate federation execution mode transitions.

Definition at line 92 of file SpaceFOM/ExecutionConfiguration.hh.

Referenced by `get_next_execution_mode()`, `SpaceFOM::ExecutionControl::lateJoinerHlaInitProcess()`, `pack()`, `printExecutionConfiguration()`, `SpaceFOM::ExecutionControl::processExecutionControlUpdates()`, `setNextExecutionMode()`, and `unpack()`.

### 7.11.5.4 `next_mode_cte_time`

```
double SpaceFOM::ExecutionConfiguration::next_mode_cte_time
```

**Units:** s

The time for the next federation execution mode change expressed as a Central Timing Equipment (CTE) time reference.

The standard for this reference shall be defined in the federation agreement when CTE is used.

Definition at line 83 of file SpaceFOM/ExecutionConfiguration.hh.

Referenced by `get_next_mode_cte_time()`, `pack()`, `printExecutionConfiguration()`, `SpaceFOM::ExecutionControl::processExecutionControlUpdates()`, `SpaceFOM::ExecutionControl::processModeTransitionRequest()`, `setNextMode_cte_time()`, and `unpack()`.

### 7.11.5.5 `next_mode_scenario_time`

```
double SpaceFOM::ExecutionConfiguration::next_mode_scenario_time
```

**Units:** s

The time for the next federation execution mode change expressed as a federation scenario time reference.

Note: this value is only meaningful for going into freeze; exiting freeze is coordinated through a sync point mechanism.

Definition at line 77 of file SpaceFOM/ExecutionConfiguration.hh.

Referenced by `get_next_mode_scenario_time()`, `pack()`, `printExecutionConfiguration()`, `SpaceFOM::ExecutionControl::processExecutionControlUpdates()`, `SpaceFOM::ExecutionControl::processModeTransitionRequest()`, `setNextMode_scenario_time()`, and `unpack()`.

### 7.11.5.6 `root_frame_name`

```
char* SpaceFOM::ExecutionConfiguration::root_frame_name
```

**Units:** –

Specifies the name of the root coordinate frame in the federation execution's reference frame tree.

This frame shall remain fixed throughout the federation execution.

Definition at line 65 of file SpaceFOM/ExecutionConfiguration.hh.

Referenced by `configureAttributes()`, `getRootFrameName()`, `pack()`, `printExecutionConfiguration()`, `setRootFrameName()`, `unpack()`, and `~ExecutionConfiguration()`.

### 7.11.5.7 `scenario_time_epoch`

```
double SpaceFOM::ExecutionConfiguration::scenario_time_epoch
```

**Units:** s

Federation execution scenario time epoch.

This is the beginning epoch expressed in Terrestrial Time (TT) that corresponds to HLA logical time 0. All joining federates shall use this time to coordinate the offset between their local simulation scenario times, their local simulation execution times and the HLA logical time.

Definition at line 70 of file SpaceFOM/ExecutionConfiguration.hh.

Referenced by `get_scenario_time_epoch()`, `SpaceFOM::ExecutionControl::late_joiner_hla_init_process()`, `pack()`, `SpaceFOM::ExecutionControl::post_multi_phase_init_processes()`, `print_execution_configuration()`, `SpaceFOM::ExecutionControl::process_execution_control_updates()`, `SpaceFOM::ExecutionControl::process_mode_transition_request()`, `set_scenario_time_epoch()`, and `unpack()`.

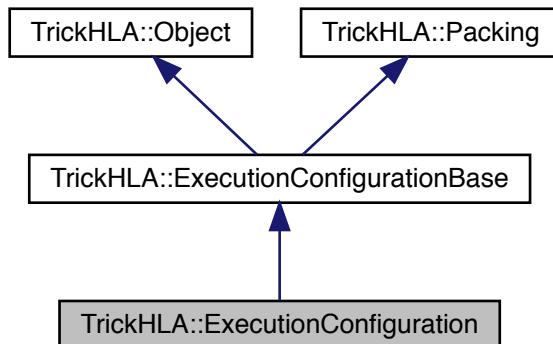
The documentation for this class was generated from the following files:

- SpaceFOM/ExecutionConfiguration.hh
  - DIS/ExecutionConfiguration.cpp

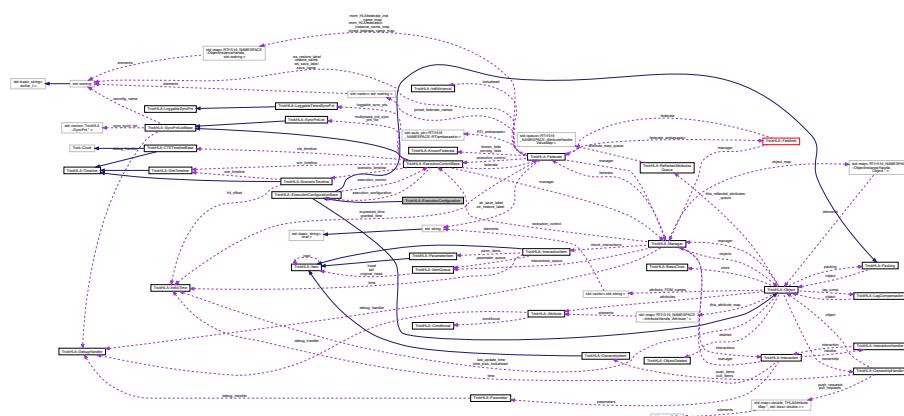
## 7.12 TrickHLA::ExecutionConfiguration Class Reference

```
#include <ExecutionConfiguration.hh>
```

## Inheritance diagram for TrickHLA::ExecutionConfiguration:



## Collaboration diagram for TrickHLA::ExecutionConfiguration:



## Public Member Functions

- [ExecutionConfiguration \(\)](#)  
*Default constructor for the `TrickHLA ExecutionConfiguration` class.*
- [ExecutionConfiguration \(const char \\*s\\_define\\_name\)](#)  
*Initialization constructor for the `TrickHLA ExecutionConfiguration` class.*
- [virtual ~ExecutionConfiguration \(\)](#)  
*Pure virtual destructor for the `TrickHLA ExecutionConfiguration` class.*
- [virtual void configure\\_attributes \(\)](#)  
*Sets up the attributes for the ExCO using default values. These can be overridden in the input file.*
- [virtual void configure \(\)](#)  
*Configure the execution configuration object.*
- [virtual void pack \(\)](#)
- [virtual void unpack \(\)](#)
- [virtual void setup\\_ref\\_attributes \(Packing \\*packing\\_obj\)](#)  
*Setup the Trick Ref Attributes for the `ExecutionConfiguration` object.*
- [virtual void print\\_execution\\_configuration \(\)](#)  
*Print the current Execution Configuration object to the console.*

## Data Fields

- [double run\\_duration](#)  
**Units:** s  
*The run duration of the simulation.*
- [long long run\\_duration\\_microsec](#)  
**Units:** us  
*The run duration in microseconds.*
- [int num\\_federates](#)  
**Units:** –  
*Number of required federates.*
- [char \\* required\\_federates](#)  
**Units:** –  
*Comma-separated list of required federates.*
- [char \\* owner](#)  
**Units:** –  
*Federate's name publishing the object.*

## Private Member Functions

- [ExecutionConfiguration \(const ExecutionConfiguration &rhs\)](#)
- [ExecutionConfiguration & operator= \(const ExecutionConfiguration &rhs\)](#)

## Friends

- [class InputProcessor](#)
- [void init\\_attrTrickHLA\\_\\_ExecutionConfiguration \(\)](#)

## Additional Inherited Members

### 7.12.1 Detailed Description

Definition at line 50 of file TrickHLA/ExecutionConfiguration.hh.

## 7.12.2 Constructor & Destructor Documentation

### 7.12.2.1 ExecutionConfiguration() [1/3]

```
ExecutionConfiguration::ExecutionConfiguration ( )
Default constructor for the TrickHLA ExecutionConfiguration class.
Trick Job Class: initialization
Definition at line 83 of file TrickHLA/ExecutionConfiguration.cpp.
```

### 7.12.2.2 ExecutionConfiguration() [2/3]

```
ExecutionConfiguration::ExecutionConfiguration (
    const char * s_define_name ) [explicit]
Initialization constructor for the TrickHLA ExecutionConfiguration class.
```

#### Parameters

<code>s_define_name</code>	Full path name in the S_define for this <a href="#">ExecutionConfiguration</a> instance.
----------------------------	--

#### **Trick Job Class:** *initialization*

Definition at line 96 of file TrickHLA/ExecutionConfiguration.cpp.

### 7.12.2.3 ~ExecutionConfiguration()

```
ExecutionConfiguration::~ExecutionConfiguration ( ) [virtual]
Pure virtual destructor for the TrickHLA ExecutionConfiguration class.
Even though this is a pure virtual destructor, we provide a default implementation that can be called from an inheriting class. Trick Job Class: shutdown
Definition at line 113 of file TrickHLA/ExecutionConfiguration.cpp.
References owner, and required_federates.
```

### 7.12.2.4 ExecutionConfiguration() [3/3]

```
TrickHLA::ExecutionConfiguration::ExecutionConfiguration (
    const ExecutionConfiguration & rhs ) [private]
```

## 7.12.3 Member Function Documentation

### 7.12.3.1 configure()

```
void ExecutionConfiguration::configure ( ) [virtual]
Configure the execution configuration object.
```

#### **Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 193 of file TrickHLA/ExecutionConfiguration.cpp.

References [TrickHLA::Manager::get\\_federate\(\)](#), [TrickHLA::Object::initialize\(\)](#), [TrickHLA::Federate::known\\_feds](#), [TrickHLA::Federate::known\\_feds\\_count](#), [TrickHLA::Object::manager](#), [TrickHLA::KnownFederate::name](#), [num\\_federates](#), [TrickHLA::KnownFederate::required](#), [required\\_federates](#), and [THLA\\_ENDL](#).

### 7.12.3.2 configure\_attributes()

```
void ExecutionConfiguration::configure_attributes ( ) [virtual]
```

Sets up the attributes for the ExCO using default values. These can be overridden in the input file.

These can be overridden in the input file. **Trick Job Class:** *default\_data*

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 135 of file TrickHLA/ExecutionConfiguration.cpp.

References TrickHLA::Object::attr\_count, TrickHLA::Object::attributes, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_UNICODE\_STRING, TrickHLA::Attribute::FOM\_name, TrickHLA::Object::FOM\_name, TrickHLA::Object::name, TrickHLA::Object::packing, TrickHLA::Attribute::rti\_encoding, TrickHLA::ExecutionConfigurationBase::S\_define\_name, THLA\_ENDL, trick\_MM, and TrickHLA::Attribute::trick\_name.

### 7.12.3.3 operator=()

```
ExecutionConfiguration& TrickHLA::ExecutionConfiguration::operator= (
    const ExecutionConfiguration & rhs ) [private]
```

### 7.12.3.4 pack()

```
void ExecutionConfiguration::pack ( ) [virtual]
```

This function is called before the data is sent to the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 245 of file TrickHLA/ExecutionConfiguration.cpp.

References TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, TrickHLA::MAX\_LOGICAL\_TIME\_SECONDS, TrickHLA::MAX\_VALUE\_IN\_MICROS, TrickHLA::MICROS\_MULTIPLIER, num\_federates, owner, required\_federates, run\_duration, run\_duration\_microsec, and TrickHLA::Packing::should\_print().

### 7.12.3.5 print\_execution\_configuration()

```
void ExecutionConfiguration::print_execution_configuration ( ) [virtual]
```

Print the current Execution Configuration object to the console.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 344 of file TrickHLA/ExecutionConfiguration.cpp.

References TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, TrickHLA::Object::get\_name(), num\_federates, owner, required\_federates, run\_duration, run\_duration\_microsec, TrickHLA::Packing::should\_print(), and THLA\_ENDL.

### 7.12.3.6 setup\_ref\_attributes()

```
void ExecutionConfiguration::setup_ref_attributes (
    Packing * packing_obj ) [virtual]
```

Setup the Trick Ref Attributes for the [ExecutionConfiguration](#) object.

#### Parameters

<i>packing_obj</i>	Associated packing object.
--------------------	----------------------------

WARNING: This function is BROKEN!

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 338 of file TrickHLA/ExecutionConfiguration.cpp.

#### 7.12.3.7 `unpack()`

```
void ExecutionConfiguration::unpack ( ) [virtual]
```

This function is called after data is received from the RTI.

Implements [TrickHLA::ExecutionConfigurationBase](#).

Definition at line 299 of file TrickHLA/ExecutionConfiguration.cpp.

References `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_SOURCE_PACKING`, `TrickHLA::MICROS_`←  
`MULTIPLIER`, `num_federates`, `owner`, `TrickHLA::ExecutionConfigurationBase::pending_update`, `required_federates`,  
`run_duration`, `run_duration_microsec`, and `TrickHLA::Packing::should_print()`.

### 7.12.4 Friends And Related Function Documentation

#### 7.12.4.1 `init_attrTrickHLA__ExecutionConfiguration`

```
void init_attrTrickHLA__ExecutionConfiguration ( ) [friend]
```

#### 7.12.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 58 of file TrickHLA/ExecutionConfiguration.hh.

### 7.12.5 Field Documentation

#### 7.12.5.1 `num_federates`

```
int TrickHLA::ExecutionConfiguration::num_federates
```

**Units:** –

Number of required federates.

Definition at line 67 of file TrickHLA/ExecutionConfiguration.hh.

Referenced by `configure()`, `pack()`, `print_execution_configuration()`, and `unpack()`.

#### 7.12.5.2 `owner`

```
char* TrickHLA::ExecutionConfiguration::owner
```

**Units:** –

[Federate](#)'s name publishing the object.

Definition at line 70 of file TrickHLA/ExecutionConfiguration.hh.

Referenced by `pack()`, `print_execution_configuration()`, `unpack()`, and `~ExecutionConfiguration()`.

#### 7.12.5.3 `required_federates`

```
char* TrickHLA::ExecutionConfiguration::required_federates
```

**Units:** –

Comma-separated list of required federates.

Definition at line 68 of file TrickHLA/ExecutionConfiguration.hh.

Referenced by `configure()`, `pack()`, `print_execution_configuration()`, `unpack()`, and `~ExecutionConfiguration()`.

#### 7.12.5.4 `run_duration`

```
double TrickHLA::ExecutionConfiguration::run_duration
```

**Units:** *s*

The run duration of the simulation.

Definition at line 64 of file `TrickHLA/ExecutionConfiguration.hh`.

Referenced by `pack()`, `print_execution_configuration()`, and `unpack()`.

#### 7.12.5.5 `run_duration_microsec`

```
long long TrickHLA::ExecutionConfiguration::run_duration_microsec
```

**Units:** *us*

The run duration in microseconds.

Definition at line 65 of file `TrickHLA/ExecutionConfiguration.hh`.

Referenced by `pack()`, `print_execution_configuration()`, and `unpack()`.

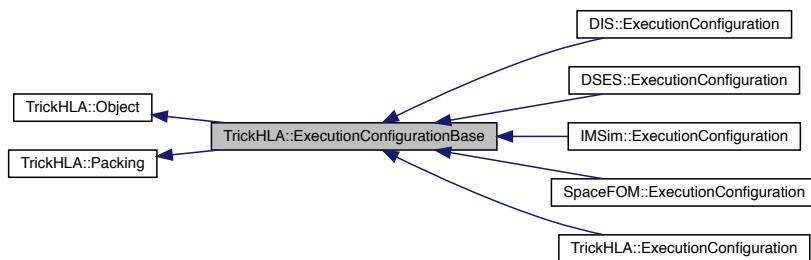
The documentation for this class was generated from the following files:

- [TrickHLA/ExecutionConfiguration.hh](#)
- [TrickHLA/ExecutionConfiguration.cpp](#)

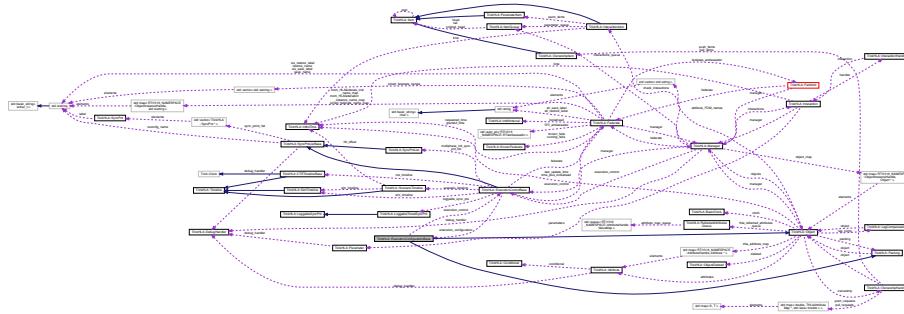
## 7.13 TrickHLA::ExecutionConfigurationBase Class Reference

```
#include <ExecutionConfigurationBase.hh>
```

Inheritance diagram for `TrickHLA::ExecutionConfigurationBase`:



Collaboration diagram for TrickHLA::ExecutionConfigurationBase:



## Public Member Functions

- **ExecutionConfigurationBase ()**  
*Default constructor for the [TrickHLA ExecutionConfigurationBase](#) class.*
- **ExecutionConfigurationBase (const char \*s\_define\_name)**  
*Initialization constructor for the [TrickHLA ExecutionConfigurationBase](#) class.*
- **virtual ~ExecutionConfigurationBase ()=0**  
*Pure virtual destructor for the [TrickHLA ExecutionConfigurationBase](#) class.*
- **virtual void setup (TrickHLA::ExecutionControlBase &exec\_control)**  
*Sets up the attributes for this Execution Configuration object using default values. These can be overridden in the input file.*
- **virtual void configure\_attributes ()=0**  
*Sets up the attributes for this Execution Configuration object using default values. These can be overridden in the input file.*
- **virtual void configure ()=0**  
*Configure the execution configuration object.*
- **virtual void set\_S\_define\_name (const char \*new\_name)**  
*Set the full path name in the S\_define to the [ExecutionConfiguration](#) object instance.*
- **virtual const char \* get\_S\_define\_name ()**  
*Get the full path name in the S\_define to the [ExecutionConfiguration](#) object instance.*
- **virtual void pack ()=0**  
*Pack the data before being sent.*
- **virtual void unpack ()=0**  
*Unpack the received data. The default.*
- **virtual void reset\_preferred\_order ()**  
*Resets the object and attribute preferred-order flags to Receive-Order.*
- **virtual void reset\_ownership\_states ()**  
*Resets the object and attribute ownership flags to locally owned and enabled the CONFIG\_TYPE\_INITIALIZE flag for each attribute.*
- **virtual void set\_master (bool is\_master)**  
*The Execution Configuration is published by the master federate and subscribed to by the non-master federates.*
- **virtual void setup\_ref\_attributes (Packing \*packing\_obj)=0**  
*Setup the Trick Ref Attributes for the [ExecutionConfiguration](#) object.*
- **virtual void print\_execution\_configuration ()=0**  
*Print the current Execution Configuration object to the console.*

- virtual void `wait_on_registration ()`  
*Waits on the registration of the `ExecutionConfiguration` object instances with the RTI.*
- virtual bool `wait_on_update ()`  
*Wait on an Execution Configuration update.*
- virtual bool `update_pending ()`  
*Check if an update is pending.*
- virtual void `clear_update_pending ()`  
*Clear the update pending flag.*
- virtual void `set_execution_control (ExecutionControlBase *exec_control)`  
*Get the reference to the associated `TrickHLA::ExecutionControlBase` object.*
- virtual `ExecutionControlBase * get_execution_control ()`  
*Get the reference to the associated `TrickHLA::ExecutionControlBase` object.*

## Data Fields

- const char \* `S_define_name`  
**Units:** –  
*Full path name in the `S_define` for this `ExecutionConfiguration` instance.*

## Protected Attributes

- bool `pending_update`  
**Units:** –  
*Pending update flag.*
- `ExecutionControlBase * execution_control`  
**Units:** –  
*Associates `TrickHLA::ExecutionConfigurationBase` class object instance.*

## Private Member Functions

- `ExecutionConfigurationBase (const ExecutionConfigurationBase &rhs)`  
*Copy constructor for `ExecutionConfigurationBase` class.*
- `ExecutionConfigurationBase & operator= (const ExecutionConfigurationBase &rhs)`  
*Assignment operator for `ExecutionConfigurationBase` class.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA__ExecutionConfigurationBase ()`

## Additional Inherited Members

### 7.13.1 Detailed Description

Definition at line 59 of file `ExecutionConfigurationBase.hh`.

### 7.13.2 Constructor & Destructor Documentation

### 7.13.2.1 ExecutionConfigurationBase() [1/3]

ExecutionConfigurationBase::ExecutionConfigurationBase ( )  
 Default constructor for the [TrickHLA ExecutionConfigurationBase](#) class.

**Trick Job Class:** *initialization*

Definition at line 78 of file ExecutionConfigurationBase.cpp.

References TrickHLA::Object::name, TrickHLA::Object::packing, and trick\_MM.

### 7.13.2.2 ExecutionConfigurationBase() [2/3]

ExecutionConfigurationBase::ExecutionConfigurationBase (   
     const char \* s\_define\_name )

Initialization constructor for the [TrickHLA ExecutionConfigurationBase](#) class.

Parameters

<code>s_define_name</code>	Full path name in the S_define for this <a href="#">ExecutionConfiguration</a> instance.
----------------------------	--

**Trick Job Class:** *initialization*

Definition at line 96 of file ExecutionConfigurationBase.cpp.

References TrickHLA::Object::name, TrickHLA::Object::packing, S\_define\_name, and trick\_MM.

### 7.13.2.3 ~ExecutionConfigurationBase()

ExecutionConfigurationBase::~ExecutionConfigurationBase ( ) [pure virtual]  
 Pure virtual destructor for the [TrickHLA ExecutionConfigurationBase](#) class.

**Trick Job Class:** *shutdown*

Definition at line 117 of file ExecutionConfigurationBase.cpp.

References S\_define\_name, and trick\_MM.

### 7.13.2.4 ExecutionConfigurationBase() [3/3]

TrickHLA::ExecutionConfigurationBase::ExecutionConfigurationBase (   
     const [ExecutionConfigurationBase](#) & rhs ) [private]

Copy constructor for [ExecutionConfigurationBase](#) class.

This constructor is private to prevent inadvertent copies.

## 7.13.3 Member Function Documentation

### 7.13.3.1 clear\_update\_pending()

virtual void TrickHLA::ExecutionConfigurationBase::clear\_update\_pending ( ) [inline], [virtual]  
 Clear the update pending flag.

Definition at line 145 of file ExecutionConfigurationBase.hh.

References pending\_update.

Referenced by DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), and IMSim::ExecutionControl::process\_execution\_control\_updates().

### 7.13.3.2 `configure()`

```
virtual void TrickHLA::ExecutionConfigurationBase::configure ( ) [pure virtual]
```

Configure the execution configuration object.

Implemented in [SpaceFOM::ExecutionConfiguration](#), and [TrickHLA::ExecutionConfiguration](#).

Referenced by [TrickHLA::ExecutionControlBase::initialize\(\)](#).

### 7.13.3.3 `configure_attributes()`

```
virtual void TrickHLA::ExecutionConfigurationBase::configure_attributes ( ) [pure virtual]
```

Sets up the attributes for this Execution Configuration object using default values. These can be overridden in the input file.

Implemented in [SpaceFOM::ExecutionConfiguration](#), and [TrickHLA::ExecutionConfiguration](#).

Referenced by [setup\(\)](#), and [TrickHLA::ExecutionControlBase::setup\(\)](#).

### 7.13.3.4 `get_execution_control()`

```
virtual ExecutionControlBase\* TrickHLA::ExecutionConfigurationBase::get_execution_control ( ) [inline], [virtual]
```

Get the reference to the associated [TrickHLA::ExecutionControlBase](#) object.

#### Returns

Pointer to the associated [TrickHLA::ExecutionControlBase](#) object.

Definition at line 156 of file [ExecutionConfigurationBase.hh](#).

References [execution\\_control](#).

### 7.13.3.5 `get_S_define_name()`

```
virtual const char* TrickHLA::ExecutionConfigurationBase::get_S_define_name ( ) [inline], [virtual]
```

Get the full path name in the S\_define to the [ExecutionConfiguration](#) object instance.

#### Returns

S\_define\_name Full path name in the S\_define for this [ExecutionConfiguration](#) instance.

Definition at line 106 of file [ExecutionConfigurationBase.hh](#).

References [S\\_define\\_name](#).

### 7.13.3.6 `operator=( )`

```
ExecutionConfigurationBase& TrickHLA::ExecutionConfigurationBase::operator= ( const ExecutionConfigurationBase & rhs ) [private]
```

Assignment operator for [ExecutionConfigurationBase](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.13.3.7 `pack()`

```
virtual void TrickHLA::ExecutionConfigurationBase::pack ( ) [pure virtual]
```

Pack the data before being sent.

Implements [TrickHLA::Packing](#).

Implemented in [SpaceFOM::ExecutionConfiguration](#), [IMSim::ExecutionConfiguration](#), [DIS::ExecutionConfiguration](#), [DSES::ExecutionConfiguration](#), and [TrickHLA::ExecutionConfiguration](#).

### 7.13.3.8 `print_execution_configuration()`

```
virtual void TrickHLA::ExecutionConfigurationBase::print_execution_configuration ( ) [pure virtual]
Print the current Execution Configuration object to the console.
```

Implemented in [SpaceFOM::ExecutionConfiguration](#), [IMSim::ExecutionConfiguration](#), [DIS::ExecutionConfiguration](#), [DSES::ExecutionConfiguration](#), and [TrickHLA::ExecutionConfiguration](#).

### 7.13.3.9 `reset_ownership_states()`

```
void ExecutionConfigurationBase::reset_ownership_states ( ) [virtual]
```

Resets the object and attribute ownership flags to locally owned and enabled the CONFIG\_TYPE\_INITIALIZE flag for each attribute.

Definition at line 184 of file ExecutionConfigurationBase.cpp.

References [TrickHLA::Object::attr\\_count](#), [TrickHLA::Object::attributes](#), [TrickHLA::CONFIG\\_INITIALIZE](#), [TrickHLA::Attribute::get\\_configuration\(\)](#), [TrickHLA::Attribute::mark\\_locally\\_owned\(\)](#), [TrickHLA::Attribute::set\\_configuration\(\)](#), and [TrickHLA::Object::set\\_create\\_HLA\\_instance\(\)](#).

Referenced by [DSES::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), [DIS::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), [IMSim::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), and [SpaceFOM::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#).

### 7.13.3.10 `reset_preferred_order()`

```
void ExecutionConfigurationBase::reset_preferred_order ( ) [virtual]
```

Resets the object and attribute preferred-order flags to Receive-Order.

Definition at line 171 of file ExecutionConfigurationBase.cpp.

References [TrickHLA::Object::any\\_attribute\\_timestamp\\_order](#), [TrickHLA::Object::attr\\_count](#), [TrickHLA::Object::attributes](#), [TrickHLA::Attribute::set\\_preferred\\_order\(\)](#), and [TrickHLA::TRANSPORT\\_RECEIVE\\_ORDER](#).

Referenced by [SpaceFOM::ExecutionConfiguration::configure\(\)](#), [DSES::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), [IMSim::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), and [SpaceFOM::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#).

### 7.13.3.11 `set_execution_control()`

```
virtual void TrickHLA::ExecutionConfigurationBase::set_execution_control (
    ExecutionControlBase * exec_control ) [inline], [virtual]
```

Get the reference to the associated [TrickHLA::ExecutionControlBase](#) object.

#### Parameters

<code>exec_control</code>	Pointer to the associated <a href="#">TrickHLA::ExecutionControlBase</a> object.
---------------------------	--

Definition at line 149 of file ExecutionConfigurationBase.hh.

References `execution_control`.

### 7.13.3.12 `set_master()`

```
void ExecutionConfigurationBase::set_master (
    bool is_master ) [virtual]
```

The Execution Configuration is published by the master federate and subscribed to by the non-master federates.

**Parameters**

<i>is_master</i>	True if the master, false otherwise.
------------------	--------------------------------------

Definition at line 212 of file ExecutionConfigurationBase.cpp.

References TrickHLA::Object::attr\_count, TrickHLA::Object::attributes, TrickHLA::Attribute::mark\_locally\_owned(), TrickHLA::Attribute::mark\_remotely\_owned(), TrickHLA::Object::set\_create\_HLA\_instance(), TrickHLA::Attribute::set\_publish(), and TrickHLA::Attribute::set\_subscribe().

Referenced by DSES::ExecutionControl::determine\_federation\_master(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), and TrickHLA::ExecutionControlBase::set\_master().

**7.13.3.13 set\_S\_define\_name()**

```
void ExecutionConfigurationBase::set_S_define_name (
    const char * new_name ) [virtual]
```

Set the full path name in the S\_define to the [ExecutionConfiguration](#) object instance.

**Parameters**

<i>new_name</i>	Full path name in the S_define for this <a href="#">ExecutionConfiguration</a> instance.
-----------------	--

**Trick Job Class: initialization**

Definition at line 154 of file ExecutionConfigurationBase.cpp.

References S\_define\_name, and trick\_MM.

**7.13.3.14 setup()**

```
void ExecutionConfigurationBase::setup (
    TrickHLA::ExecutionControlBase & exec_control ) [virtual]
```

Sets up the attributes for this Execution Configuration object using default values. These can be overridden in the input file.

**Parameters**

<i>exec_control</i>	Reference to the associated <a href="#">TrickHLA::ExecutionControlBase</a> object.
---------------------	--

**Assumptions and Limitations:**

- The [TrickHLA::ExecutionControlBase](#) class is actually an abstract class. Therefore, the actual object instance being passed in is an instantiable polymorphic child of the [TrickHLA::ExecutionControlBase](#) class.

**Trick Job Class: default\_data**

Definition at line 138 of file ExecutionConfigurationBase.cpp.

References configure\_attributes(), and execution\_control.

Referenced by TrickHLA::ExecutionControlBase::setup().

**7.13.3.15 setup\_ref\_attributes()**

```
virtual void TrickHLA::ExecutionConfigurationBase::setup_ref_attributes (
    Packing * packing_obj ) [pure virtual]
```

Setup the Trick Ref Attributes for the [ExecutionConfiguration](#) object.

**Parameters**

<i>packing_obj</i>	Associated packing object.
--------------------	----------------------------

Implemented in [SpaceFOM::ExecutionConfiguration](#), and [TrickHLA::ExecutionConfiguration](#).

**7.13.3.16 `unpack()`**

`virtual void TrickHLA::ExecutionConfigurationBase::unpack ( ) [pure virtual]`  
Unpack the received data. The default.

Implements [TrickHLA::Packing](#).

Implemented in [SpaceFOM::ExecutionConfiguration](#), [IMSim::ExecutionConfiguration](#), [DIS::ExecutionConfiguration](#), [DSES::ExecutionConfiguration](#), and [TrickHLA::ExecutionConfiguration](#).

**7.13.3.17 `update_pending()`**

`virtual bool TrickHLA::ExecutionConfigurationBase::update_pending ( ) [inline], [virtual]`  
Check if an update is pending.

**Returns**

True is an update is pending.

Definition at line 142 of file `ExecutionConfigurationBase.hh`.

References `pending_update`.

Referenced by `DSES::ExecutionControl::process_execution_control_updates()`, `SpaceFOM::ExecutionControl::process_execution_control_updates()`, `DIS::ExecutionControl::process_execution_control_updates()`, and `IMSim::ExecutionControl::process_execution_control_updates()`.

**7.13.3.18 `wait_on_registration()`**

`void ExecutionConfigurationBase::wait_on_registration ( ) [virtual]`  
Waits on the registration of the [ExecutionConfiguration](#) object instances with the RTI.

Calling this function will block until the execution configuration object instance in the Federation has been registered.

**Trick Job Class:** *initialization*

Definition at line 249 of file `ExecutionConfigurationBase.cpp`.

References `TrickHLA::Federate::check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `execution_control`, `TrickHLA::Object::get_federate()`, `TrickHLA::Object::get_FOM_name()`, `TrickHLA::Object::get_instance_handle()`, `TrickHLA::Object::get_name()`, `TrickHLA::Federate::is_execution_member()`, `TrickHLA::Object::is_instance_handle_valid()`, `TrickHLA::Object::is_required()`, `TrickHLA::ExecutionControlBase::should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, and `TrickHLA::StringUtilities::to_string()`.

Referenced by `SpaceFOM::ExecutionControl::late_joiner_hla_init_process()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

**7.13.3.19 `wait_on_update()`**

`bool ExecutionConfigurationBase::wait_on_update ( ) [virtual]`  
Wait on an Execution Configuration update.

**Returns**

True for successful wait.

Reimplemented in [SpaceFOM::ExecutionConfiguration](#), [IMSim::ExecutionConfiguration](#), [DIS::ExecutionConfiguration](#), and [DSES::ExecutionConfiguration](#).

Definition at line 345 of file `ExecutionConfigurationBase.cpp`.

References `TrickHLA::Object::any_remotely_owned_subscribed_init_attribute()`, `TrickHLA::Federate::check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `execution_control`, `TrickHLA::Object::get_federate()`, `TrickHLA::Object::is_changed()`, `TrickHLA::Federate::is_execution_member()`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::Object::receive_init_data()`, `TrickHLA::Federate::should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

## 7.13.4 Friends And Related Function Documentation

### 7.13.4.1 init\_attrTrickHLA\_\_ExecutionConfigurationBase

```
void init_attrTrickHLA__ExecutionConfigurationBase ( ) [friend]
```

### 7.13.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 66 of file `ExecutionConfigurationBase.hh`.

## 7.13.5 Field Documentation

### 7.13.5.1 execution\_control

```
ExecutionControlBase* TrickHLA::ExecutionConfigurationBase::execution_control [protected]
```

#### Units: -

Associates [TrickHLA::ExecutionConfigurationBase](#) class object instance.

Since this is an abstract class, the actual instance will be a concrete derived class instance (e.g. `SRFOM::ExecutionControl`).

Definition at line 164 of file `ExecutionConfigurationBase.hh`.

Referenced by `get_execution_control()`, `IMSim::ExecutionConfiguration::pack()`, `SpaceFOM::ExecutionConfiguration::pack()`, `DSES::ExecutionConfiguration::set_current_execution_mode()`, `DIS::ExecutionConfiguration::set_current_execution_mode()`, `IMSim::ExecutionConfiguration::set_current_execution_mode()`, `SpaceFOM::ExecutionConfiguration::set_current_execution_mode()`, `set_execution_control()`, `SpaceFOM::ExecutionConfiguration::set_least_common_time_step()`, `DIS::ExecutionConfiguration::set_next_execution_mode()`, `DSES::ExecutionConfiguration::set_next_execution_mode()`, `IMSim::ExecutionConfiguration::set_next_execution_mode()`, `SpaceFOM::ExecutionConfiguration::set_next_execution_mode()`, `DSES::ExecutionConfiguration::set_next_mode_cte_time()`, `DIS::ExecutionConfiguration::set_next_mode_cte_time()`, `IMSim::ExecutionConfiguration::set_next_mode_cte_time()`, `SpaceFOM::ExecutionConfiguration::set_next_mode_cte_time()`, `DSES::ExecutionConfiguration::set_next_mode_scenario_time()`, `DIS::ExecutionConfiguration::set_next_mode_scenario_time()`, `IMSim::ExecutionConfiguration::set_next_mode_scenario_time()`, `SpaceFOM::ExecutionConfiguration::set_next_mode_scenario_time()`, `DIS::ExecutionConfiguration::set_scenario_time_epoch()`, `DSES::ExecutionConfiguration::set_scenario_time_epoch()`, `IMSim::ExecutionConfiguration::set_scenario_time_epoch()`, `SpaceFOM::ExecutionConfiguration::set_scenario_time_epoch()`, `SpaceFOM::ExecutionConfiguration::setup()`, `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`, `IMSim::ExecutionConfiguration::unpack()`, `SpaceFOM::ExecutionConfiguration::unpack()`, `wait_on_registration()`, `wait_on_update()`, `DSES::unpack()`.

ExecutionConfiguration::wait\_on\_update(), DIS::ExecutionConfiguration::wait\_on\_update(), IMSim::ExecutionConfiguration::wait\_on\_update(), and SpaceFOM::ExecutionConfiguration::wait\_on\_update().

### 7.13.5.2 pending\_update

bool TrickHLA::ExecutionConfigurationBase::pending\_update [protected]

**Units:** –

Pending update flag.

Definition at line 162 of file ExecutionConfigurationBase.hh.

Referenced by clear\_update\_pending(), TrickHLA::ExecutionConfiguration::unpack(), DIS::ExecutionConfiguration::unpack(), DSES::ExecutionConfiguration::unpack(), IMSim::ExecutionConfiguration::unpack(), SpaceFOM::ExecutionConfiguration::unpack(), and update\_pending().

### 7.13.5.3 S\_define\_name

const char\* TrickHLA::ExecutionConfigurationBase::S\_define\_name

**Units:** –

Full path name in the S\_define for this [ExecutionConfiguration](#) instance.

Definition at line 72 of file ExecutionConfigurationBase.hh.

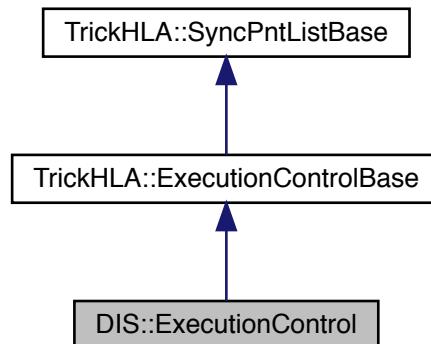
Referenced by TrickHLA::ExecutionConfiguration::configure\_attributes(), SpaceFOM::ExecutionConfiguration::configure\_attributes(), ExecutionConfigurationBase(), get\_S\_define\_name(), set\_S\_define\_name(), and ~ExecutionConfigurationBase().

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

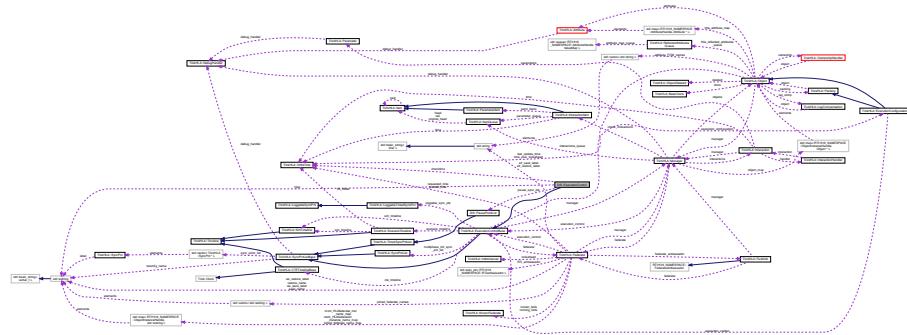
- [ExecutionConfigurationBase.hh](#)
- [ExecutionConfigurationBase.cpp](#)

## 7.14 DIS::ExecutionControl Class Reference

#include <ExecutionControl.hh>  
Inheritance diagram for DIS::ExecutionControl:



Collaboration diagram for DIS::ExecutionControl:



## Public Member Functions

- **ExecutionControl ()**  
*Default constructor for the [DIS ExecutionControl](#) class.*
- **virtual ~ExecutionControl ()**  
*Destructor for the [DIS ExecutionControl](#) class.*
- **virtual const std::wstring & get\_type ()**  
*Get the [ExecutionControl](#) type identification string.*
- **virtual void initialize (TrickHLA::Federate &fed)**  
*Execution Control initialization routine.*
- **virtual void join\_federation\_process ()**  
*Join federation execution process.*
- **virtual void pre\_multiphase\_init\_processes ()**  
*Process run before the multi-phase initialization begins.*
- **virtual void post\_multiphase\_init\_process ()**  
*Process run after the multi-phase initialization ends.*
- **virtual void shutdown ()**  
*Execution control specific shutdown process.*
- **virtual void setup\_object\_ref\_attributes ()**
- **virtual void setup\_interaction\_ref\_attributes ()**
- **virtual void setup\_object\_RTI\_handles ()**
- **virtual void setup\_interaction\_RTI\_handles ()**
- **virtual void add\_multiphase\_init\_sync\_points ()**
- **virtual void announce\_sync\_point (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador, std::wstring const &label, RTI1516\_USERDATA const &user\_supplied\_tag)**  
*The RTI has announced the existence of a synchronization point.*
- **virtual void clear\_multiphase\_init\_sync\_points ()**
- **virtual void publish ()**
- **virtual void unpublish ()**
- **virtual void subscribe ()**
- **virtual void unsubscribe ()**
- **virtual bool object\_instance\_name\_reservation\_failed (std::wstring const &obj\_instance\_name)**  
*The object instance name reservation failed for the given name.*
- **virtual void sync\_point\_registration\_succeeded (std::wstring const &label)**

- Callback from `TrickHLA::FedAmb` through `TrickHLA::Federate` for when registration of a synchronization point success. and is one of the sync-points created.*
- `virtual void sync_point_registration_failed (std::wstring const &label, bool not_unique)`  
*Callback from `TrickHLA::FedAmb` through `TrickHLA::Federate` for when registration of a synchronization point fails. and is one of the sync-points created.*
  - `virtual void join_federation_process ()`  
*Join federation execution process.*
  - `virtual void role_determination_process ()`  
*Determine the federate role in the federation execution.*
  - `virtual void early_joiner_hla_init_process ()`  
*Process to join the federation execution early in initialization.*
  - `virtual void late_joiner_hla_init_process ()`  
*Process to determine is a federate is joining late in or after initialization.*
  - `virtual void pre_multi_phase_init_processes ()`  
*Process run before the multi-phase initialization begins.*
  - `virtual void post_multi_phase_init_process ()`  
*Process run after the multi-phase initialization ends.*
  - `virtual void shutdown ()`  
*Execution control specific shutdown process.*
  - `void determine_federation_master ()`  
*Determine if this federate is the Master for the federation.*
  - `bool wait_on_init_sync_point ()`  
*Test to see if `ExecutionControl` needs to wait on initialization synchronization point.*
  - `virtual void set_next_execution_control_mode (TrickHLA::ExecutionControlEnum exec_control)`  
*Sets the next ExCO run mode.*
  - `virtual bool process_execution_control_updates ()`  
*Process changes from any received Execution Control Objects (ExCOs).*
  - `virtual bool check_mode_transition_request ()`  
*Check to see if a new MTR is valid.*
  - `virtual bool process_mode_interaction ()`  
*Process a new mode interaction.*
  - `virtual bool process_mode_transition_request ()`  
*Process a new Mode Transition Request (MTR).*
  - `virtual void clear_mode_values ()`  
*Clear the Mode Transition Request flag, the requested execution mode, and the current execution mode.*
  - `virtual bool run_mode_transition ()`  
*The run mode transition routine.*
  - `virtual void freeze_mode_announce ()`  
*Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.*
  - `virtual bool freeze_mode_transition ()`  
*The freeze mode transition routine.*
  - `virtual void check_freeze_exit ()`  
*Check for exit from freeze.*
  - `virtual void shutdown_mode_announce ()`  
*Announce to the federation execution that a shutdown is occurring.*
  - `virtual void shutdown_mode_transition ()`  
*The shutdown mode transition routine.*

- virtual void `check_freeze ()`  
*Check for exit from freeze.*
- virtual void `enter_freeze ()`  
*Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.*
- virtual virtual bool `check_freeze_exit ()`  
*Check for exit from freeze.*
- virtual void `exit_freeze ()`  
*Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.*
- virtual void `add_pause (TrickHLA::Int64Time *time, std::wstring const &label)`  
*Add pause time.*
- virtual bool `set_pending_mtr (MTREnum mtr_value)`
- virtual bool `is_mtr_valid (MTREnum mtr_value)`  
*Determine if the Mode Transition Request (MTR) is valid given the current mode.*
- virtual void `set_mode_request_from_mtr (MTREnum mtr_value)`  
*Translate MTR into a pending execution mode transition.*
- virtual void `set_time_padding (double t)`  
*Set the time-padding used to offset the go to run time.*
- virtual bool `is_save_and_restore_supported ()`
- virtual bool `is_save_initiated ()`  
*Checks if Save has been initiated by this `ExecutionControl` method.*

## Protected Member Functions

- `ExecutionConfiguration * get_execution_configuration ()`  
*Return the relevant `DIS::ExecutionConfiguration` object.*

## Protected Attributes

- `MTREnum pending_mtr`  
***Units:** –  
*Pending Mode Transition Requested.**
- `TrickHLA::Int64Time checktime`  
***Units:** –  
*For `DIS`: Checking time to pause**
- `PausePointList pause_sync_pts`  
***Units:** –  
*Synchronization points used for pausing the sim.**

## Static Protected Attributes

- static const std::wstring `type` = L"DIS"  
***Units:** –  
*`ExecutionControl` type string.**

## Private Member Functions

- `ExecutionControl (const ExecutionControl &rhs)`  
*Copy constructor for `ExecutionControl` class.*
- `ExecutionControl & operator= (const ExecutionControl &rhs)`  
*Assignment operator for `ExecutionControl` class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrDIS\\_\\_ExecutionControl \(\)](#)

## Additional Inherited Members

### 7.14.1 Detailed Description

Definition at line 55 of file DIS/ExecutionControl.hh.

### 7.14.2 Constructor & Destructor Documentation

#### 7.14.2.1 ExecutionControl() [1/2]

ExecutionControl::ExecutionControl ( )  
Default constructor for the [DIS ExecutionControl](#) class.

**Trick Job Class:** *initialization*

Definition at line 72 of file DIS/ExecutionControl.cpp.

#### 7.14.2.2 ~ExecutionControl()

ExecutionControl::~ExecutionControl ( ) [virtual]  
Destructor for the [DIS ExecutionControl](#) class.

**Trick Job Class:** *shutdown*

Definition at line 94 of file DIS/ExecutionControl.cpp.

References [clear\\_mode\\_values\(\)](#).

#### 7.14.2.3 ExecutionControl() [2/2]

DIS::ExecutionControl::ExecutionControl (   
     const [ExecutionControl](#) & rhs ) [private]  
Copy constructor for [ExecutionControl](#) class.  
This constructor is private to prevent inadvertent copies.

### 7.14.3 Member Function Documentation

#### 7.14.3.1 add\_multiphase\_init\_sync\_points()

void ExecutionControl::add\_multiphase\_init\_sync\_points ( ) [virtual]  
Add initialization synchronization points to regulate startup.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 393 of file DIS/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::add\\_sync\\_pnt\(\)](#), [DIS::INITIALIZE\\_SYNC\\_POINT](#), and [DIS::STARTUP\\_SYNCPOINT](#).

Referenced by [pre\\_multi\\_phase\\_init\\_processes\(\)](#).

### 7.14.3.2 add\_pause()

```
void ExecutionControl::add_pause (
    TrickHLA::Int64Time * time,
    std::wstring const & label ) [virtual]
```

Add pause time.

#### Parameters

<i>time</i>	Pause time.
<i>label</i>	Pause label (Synchronization point).

Definition at line 1480 of file DIS/ExecutionControl.cpp.

References TrickHLA::TimedSyncPntList::add\_sync\_pnt(), and pause\_sync\_pts.

Referenced by announce\_sync\_point().

### 7.14.3.3 announce\_sync\_point()

```
void ExecutionControl::announce_sync_point (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    std::wstring const & label,
    RTI1516_USERDATA const & user_supplied_tag ) [virtual]
```

The RTI has announced the existence of a synchronization point.

#### Parameters

<i>rti_ambassador</i>	Reference to the HLA RTI Ambassador instance.
<i>label</i>	Sync-point label.
<i>user_supplied_tag</i>	Use supplied tag.

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 402 of file DIS/ExecutionControl.cpp.

References TrickHLA::SyncPntListBase::achieve\_sync\_pnt(), add\_pause(), TrickHLA::SyncPntListBase::contains(), TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE←\_FEDERATE, TrickHLA::Int64Time::decode(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::SyncPntListBase←::mark\_announced(), TrickHLA::Int64Time::setTo(), TrickHLA::DebugHandler::should\_print(), TrickHLA::Execution←ControlBase::should\_print(), and THLA\_NEWLINE.

### 7.14.3.4 check\_freeze()

```
void ExecutionControl::check_freeze ( ) [virtual]
```

Check for exit from freeze.

Definition at line 1384 of file DIS/ExecutionControl.cpp.

### 7.14.3.5 check\_freeze\_exit() [1/2]

```
bool ExecutionControl::check_freeze_exit ( ) [virtual]
```

Check for exit from freeze.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 1437 of file DIS/ExecutionControl.cpp.

References TrickHLA::Federate::announce\_freeze, and TrickHLA::ExecutionControlBase::federate.

**7.14.3.6 check\_freeze\_exit() [2/2]**

```
virtual virtual bool DIS::ExecutionControl::check_freeze_exit ( ) [virtual]
Check for exit from freeze.
```

**Returns**

True if should exit from freeze.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

**7.14.3.7 check\_mode\_transition\_request()**

```
bool ExecutionControl::check_mode_transition_request ( ) [virtual]
Check to see if a new MTR is valid.
```

**Returns**

True if new MTR is valid.

Definition at line 746 of file DIS/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::ExecutionControlBase::is\\_mode\\_transition\\_requested\(\)](#), [is\\_mtr\\_valid\(\)](#), [DIS::mtr\\_enum\\_to\\_string\(\)](#), [pending\\_mtr](#), and [THLA\\_ENDL](#).

Referenced by [process\\_mode\\_transition\\_request\(\)](#).

**7.14.3.8 clear\_mode\_values()**

```
virtual void DIS::ExecutionControl::clear_mode_values ( ) [virtual]
Clear the Mode Transition Request flag, the requested execution mode, and the current execution mode.

Reimplemented from TrickHLA::ExecutionControlBase.
Referenced by ~ExecutionControl().
```

**7.14.3.9 clear\_multiphase\_init\_sync\_points()**

```
void ExecutionControl::clear_multiphase_init_sync_points ( ) [virtual]
Clear any remaining multiphase initialization synchronization points that have not been achieved and wait for the federation to be synchronized on it.
```

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 478 of file DIS/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), and [THLA\\_ENDL](#).

**7.14.3.10 determine\_federation\_master()**

```
void ExecutionControl::determine_federation_master ( )
Determine if this federate is the Master for the federation.
```

**Trick Job Class:** *initialization*

Definition at line 350 of file DIS/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::SyncPntListBase::register\\_all\\_sync\\_pnts\(\)](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), [THLA\\_NEWLINE](#), and [TrickHLA::SyncPntListBase::wait\\_for\\_all\\_announcements\(\)](#).

Referenced by `pre_multi_phase_init_processes()`.

#### 7.14.3.11 `early_joiner_hla_init_process()`

```
virtual void DIS::ExecutionControl::early_joiner_hla_init_process ( ) [virtual]
Process to join the federation execution early in initialization.
```

#### 7.14.3.12 `enter_freeze()`

```
void ExecutionControl::enter_freeze ( ) [virtual]
```

Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 1389 of file DIS/ExecutionControl.cpp.

References `TrickHLA::Federate::announce_freeze`, `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_SOUL`, `RCE_FEDERATE`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::freeze_delay_frames`, `TrickHLA::Federate::freeze_the_federation`, `TrickHLA::ExecutionControlBase::get_sim_time()`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::Federate::lookahead_time`, `TrickHLA::Federate::register_generic_sync_point()`, `TrickHLA::ExecutionControlBase::should_print()`, `THLA_NEWLINE`, and `TrickHLA::Federate::un_freeze()`.

#### 7.14.3.13 `exit_freeze()`

```
void ExecutionControl::exit_freeze ( ) [virtual]
```

Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 1447 of file DIS/ExecutionControl.cpp.

References `TrickHLA::TimedSyncPntList::achieve_all_sync_pnts()`, `TrickHLA::Federate::announce_freeze`, `checktime`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::freeze_the_federation`, `TrickHLA::Federate::get_RTI_ambassador()`, `pause_sync_pts`, and `THLA_NEWLINE`.

#### 7.14.3.14 `freeze_mode_announce()`

```
void ExecutionControl::freeze_mode_announce ( ) [virtual]
```

Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1290 of file DIS/ExecutionControl.cpp.

References `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::get_RTI_ambassador()`, `TrickHLA::ExecutionControlBase::is_master()`, and `TrickHLA::SyncPntListBase::register_sync_pnt()`.

Referenced by `process_execution_control_updates()`, and `process_mode_transition_request()`.

#### 7.14.3.15 `freeze_mode_transition()`

```
bool ExecutionControl::freeze_mode_transition ( ) [virtual]
```

The freeze mode transition routine.

##### Returns

Currently always returns False.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1301 of file DIS/ExecutionControl.cpp.

References `TrickHLA::SyncPnt::achieve_sync_point()`, `TrickHLA::ExecutionControlBase::current_execution_control_mode`, `TrickHLA::EXECUTION_CONTROL_FREEZE`, `DIS::EXECUTION_MODE_FREEZE`, `TrickHLA::ExecutionControlBase::federate`, `get_execution_configuration()`, `TrickHLA::Federate::get_RTI_ambassador()`, `TrickHLA::SyncPntListBase::get_sync_pnt()`, `DIS::ExecutionConfiguration::set_current_execution_mode()`, `THLA_ENDL`, `TrickHLA::SyncPnt::wait_for_announce()`, and `TrickHLA::SyncPnt::wait_for_synchronization()`.

#### 7.14.3.16 `get_execution_configuration()`

`ExecutionConfiguration * ExecutionControl::get_execution_configuration ()` [protected], [virtual]  
Return the relevant `DIS::ExecutionConfiguration` object.

##### Returns

Pointer to the relevant `DIS::ExecutionConfiguration` object.

Reimplemented from `TrickHLA::ExecutionControlBase`.

Definition at line 1487 of file `DIS/ExecutionControl.cpp`.

References `THLA_ENDL`.

Referenced by `freeze_mode_transition()`, `is_mtr_valid()`, `process_execution_control_updates()`, `process_mode_transition_request()`, `run_mode_transition()`, `set_next_execution_control_mode()`, and `shutdown_mode_announce()`.

#### 7.14.3.17 `get_type()`

`virtual const std::wstring& DIS::ExecutionControl::get_type ()` [inline], [virtual]  
Get the `ExecutionControl` type identification string.

##### Returns

A constant reference to the type identification string.

Implements `TrickHLA::ExecutionControlBase`.

Definition at line 77 of file `DIS/ExecutionControl.hh`.

References type.

#### 7.14.3.18 `initialize()`

`void ExecutionControl::initialize (`  
    `TrickHLA::Federate & fed )` [virtual]  
Execution Control initialization routine.

##### Parameters

<code>fed</code>	The associated <code>TrickHLA::Federate</code> .
------------------	--

This routine will set a lot of the data in the `TrickHLA::Federate` that is required for this execution control scheme. This should greatly simplify input files and reduce input file setting errors.

**Trick Job Class:** `initialization`

Definition at line 107 of file `DIS/ExecutionControl.cpp`.

References `TrickHLA::SyncPntListBase::add_sync_pnt()`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::execution_configuration`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::ExecutionControlBase::get_manager()`, `TrickHLA::Object::initialize()`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::ExecutionControlBase::is_master_preset()`, `TrickHLA::ExecutionControlBase::least_common_time_step`, `pre_multi_phase_init_processes()`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::Federate::time_constrained`, `TrickHLA::`

A::Federate::time\_management, TrickHLA::Federate::time\_regulating, and TrickHLA::ExecutionControlBase::use\_← preset\_master.

#### 7.14.3.19 is\_mtr\_valid()

```
bool ExecutionControl::is_mtr_valid (
    MTREnum mtr_value ) [virtual]
```

Determine if the Mode Transition Request (MTR) is valid given the current mode.

Returns

True if valid, false otherwise.

Parameters

<i>mtr_value</i>	Mode transition request.
------------------	--------------------------

Definition at line 578 of file DIS/ExecutionControl.cpp.

References DIS::ExecutionConfiguration::current\_execution\_mode, DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, get\_← \_execution\_configuration(), DIS::MTR\_GOTO\_FREEZE, DIS::MTR\_GOTO\_RUN, and DIS::MTR\_GOTO\_SHUTDOWN.

Referenced by check\_mode\_transition\_request(), and set\_pending\_mtr().

#### 7.14.3.20 is\_save\_and\_restore\_supported()

```
virtual bool DIS::ExecutionControl::is_save_and_restore_supported ( ) [inline], [virtual]
```

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 246 of file DIS/ExecutionControl.hh.

#### 7.14.3.21 is\_save\_initiated()

```
bool ExecutionControl::is_save_initiated ( ) [virtual]
```

Checks if Save has been initiated by this [ExecutionControl](#) method.

Returns

True if Save is initiated and synchronized with the federation, False if Save not supported.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 1502 of file DIS/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::federate, and TrickHLA::Federate::initiate\_save\_flag.

#### 7.14.3.22 join\_federation\_process() [1/2]

```
void ExecutionControl::join_federation_process ( ) [virtual]
```

Join federation execution process.

This routine implements the [DIS](#) Join Federation Process described in section 7.2 and figure 7-3.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 187 of file DIS/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::join\_federation\_process().

**7.14.3.23 join\_federation\_process() [2/2]**

```
virtual void DIS::ExecutionControl::join_federation_process ( ) [virtual]
```

Join federation execution process.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

**7.14.3.24 late\_joiner\_hla\_init\_process()**

```
virtual void DIS::ExecutionControl::late_joiner_hla_init_process ( ) [virtual]
```

Process to determine is a federate is joining late in or after initialization.

**7.14.3.25 object\_instance\_name\_reservation\_failed()**

```
virtual bool DIS::ExecutionControl::object_instance_name_reservation_failed ( std::wstring const & obj_instance_name ) [virtual]
```

The object instance name reservation failed for the given name.

**Returns**

True if [ExecutionConfiguration](#) object handled the failure.

**Parameters**

<i>obj_instance_name</i>	Object instance name.
--------------------------	-----------------------

Reimplemented from [TrickHLA::ExecutionControlBase](#).

**7.14.3.26 operator=()**

```
ExecutionControl& DIS::ExecutionControl::operator= ( const ExecutionControl & rhs ) [private]
```

Assignment operator for [ExecutionControl](#) class.

This assignment operator is private to prevent inadvertent copies.

**7.14.3.27 post\_multi\_phase\_init\_process() [1/2]**

```
void ExecutionControl::post_multi_phase_init_process ( ) [virtual]
```

Process run after the multi-phase initialization ends.

This routine implements the [DIS](#) post multi-phase initialization process.

**Trick Job Class:** *initialization*

Definition at line 321 of file [DIS/ExecutionControl.cpp](#).

References [TrickHLA::Federate::achieve\\_and\\_wait\\_for\\_synchronization\(\)](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::ExecutionControlBase::get\\_manager\(\)](#), [TrickHLA::Federate::load\\_and\\_print\\_running\\_federate\\_names\(\)](#), [TrickHLA::Federate::setup\\_time\\_management\(\)](#), and [DIS::STARTUP\\_SYNC\\_POINT](#).

**7.14.3.28 post\_multi\_phase\_init\_process() [2/2]**

```
virtual void DIS::ExecutionControl::post_multi_phase_init_process ( ) [virtual]
```

Process run after the multi-phase initialization ends.

### 7.14.3.29 pre\_multi\_phase\_init\_processes() [1/2]

```
void ExecutionControl::pre_multi_phase_init_processes ( ) [virtual]
```

Process run before the multi-phase initialization begins.

This routine implements the DIS pre multi-phase initialization process.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 201 of file DIS/ExecutionControl.cpp.

References TrickHLA::Federate::achieve\_and\_wait\_for\_synchronization(), add\_multiphase\_init\_sync\_points(), TrickHLA::Federate::create\_and\_join\_federation(), TrickHLA::Federate::create\_RTI\_ambassador\_and\_connect(), TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::Federate::destroy\_orphaned\_federation(), determine\_federation\_master(), TrickHLA::Federate::enable\_async\_delivery(), TrickHLA::ExecutionControlBase::execution\_configuration, TrickHLA::ExecutionControlBase::federate, TrickHLA::ExecutionControlBase::get\_manager(), TrickHLA::Federate::initialize\_MOM\_handles(), DIS::INITIALIZE\_SYNC\_POINT, TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::Manager::publish\_and\_subscribe(), TrickHLA::Federate::register\_generic\_sync\_point(), TrickHLA::Manager::register\_objects\_with\_RTI(), TrickHLA::Manager::reserve\_object\_names\_with\_RTI(), TrickHLA::ExecutionConfigurationBase::reset\_ownership\_states(), TrickHLA::Manager::set\_all\_object\_instance\_handles\_by\_name(), TrickHLA::Manager::setup\_all\_ref\_attributes(), TrickHLA::Manager::setup\_all\_RTI\_handles(), TrickHLA::Manager::setup\_preferred\_order\_with\_RTI(), TrickHLA::DebugHandler::should\_print(), DIS::STARTUP\_FREEZE\_SYNC\_POINT, THLA\_NEWLINE, TrickHLA::Federate::wait\_for\_required\_federates\_to\_join(), TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects(), and TrickHLA::Manager::wait\_on\_reservation\_of\_object\_names().

Referenced by [initialize\(\)](#).

### 7.14.3.30 pre\_multi\_phase\_init\_processes() [2/2]

```
virtual void DIS::ExecutionControl::pre_multi_phase_init_processes ( ) [virtual]
```

Process run before the multi-phase initialization begins.

Implements [TrickHLA::ExecutionControlBase](#).

### 7.14.3.31 process\_execution\_control\_updates()

```
bool ExecutionControl::process_execution_control_updates ( ) [virtual]
```

Process changes from any received Execution Control Objects (ExCOs).

**Returns**

True if mode change occurred.

**Assumptions and Limitations:**

- Called from the ExCO unpack routine.

**Trick Job Class:** *scheduled*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 889 of file DIS/ExecutionControl.cpp.

References TrickHLA::ExecutionConfigurationBase::clear\_update\_pending(), TrickHLA::ScenarioTimeline::compute\_simulation\_time(), TrickHLA::ExecutionControlBase::cte\_timeline, TrickHLA::ExecutionControlBase::current\_execution\_control\_mode, DIS::ExecutionConfiguration::current\_execution\_mode, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ExecutionControlBase::does\_cte\_timeline\_exist(), TrickHLA::execution\_control\_enum\_to\_string(), TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, DIS::execution\_mode\_enum\_to\_string(), DIS::EXECUTION\_MODE\_FREEZE, DIS::execution\_mode\_int16\_to\_enum(), DIS::EXECUTION\_MODE

\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, TrickHLA::ExecutionControlBase::federate, freeze\_mode\_announce(), TrickHLA::Timeline::get\_epoch(), get\_execution\_configuration(), TrickHLA::ScenarioTimeline::get\_sim\_offset(), TrickHLA::SimTimeline::get\_time(), TrickHLA::CTETimelineBase::get\_time(), TrickHLA::ScenarioTimeline::get\_time(), TrickHLA::ExecutionControlBase::is\_master(), DIS::ExecutionConfiguration::next\_execution\_mode, DIS::ExecutionConfiguration::next\_mode\_cte\_time, TrickHLA::ExecutionControlBase::next\_mode\_cte\_time, DIS::ExecutionConfiguration::next\_mode\_scenario\_time, TrickHLA::ExecutionControlBase::requested\_execution\_control\_mode, run\_mode\_transition(), TrickHLA::ExecutionControlBase::scenario\_freeze\_time, DIS::ExecutionConfiguration::scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Federate::should\_print(), TrickHLA::ExecutionControlBase::sim\_timeline, TrickHLA::ExecutionControlBase::simulation\_freeze\_time, THLA-ENDL, and TrickHLA::ExecutionConfigurationBase::update\_pending().

Referenced by run\_mode\_transition().

#### 7.14.3.32 process\_mode\_interaction()

```
virtual bool DIS::ExecutionControl::process_mode_interaction ( ) [inline], [virtual]
Process a new mode interaction.
```

##### Returns

True if new mode interaction is successfully processed.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 186 of file DIS/ExecutionControl.hh.

References process\_mode\_transition\_request().

#### 7.14.3.33 process\_mode\_transition\_request()

```
bool ExecutionControl::process_mode_transition_request ( ) [virtual]
Process a new Mode Transition Request (MTR).
```

##### Returns

True if new MTR is successfully processed.

Definition at line 778 of file DIS/ExecutionControl.cpp.

References check\_mode\_transition\_request(), TrickHLA::ExecutionControlBase::clear\_mode\_transition\_requested(), TrickHLA::ExecutionControlBase::cte\_timeline, TrickHLA::ExecutionControlBase::current\_execution\_control\_mode, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ExecutionControlBase::does\_cte\_timeline\_exist(), TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_RUN, TrickHLA::ExecutionControlBase::federate, freeze\_mode\_announce(), TrickHLA::Timeline::get\_epoch(), get\_execution\_configuration(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Federate::get\_requested\_time(), TrickHLA::ScenarioTimeline::get\_sim\_offset(), TrickHLA::SimTimeline::get\_time(), TrickHLA::CTETimelineBase::get\_time(), TrickHLA::ScenarioTimeline::get\_time(), DIS::MTR\_GOTO\_FREEZE, DIS::MTR\_GOTO\_RUN, DIS::MTR\_GOTO\_SHUTDOWN, DIS::ExecutionConfiguration::next\_mode\_cte\_time, DIS::ExecutionConfiguration::next\_mode\_scenario\_time, pending\_mtr, TrickHLA::ExecutionControlBase::scenario\_freeze\_time, DIS::ExecutionConfiguration::scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Object::send\_init\_data(), set\_mode\_request\_from\_mtr(), TrickHLA::Federate::should\_print(), shutdown\_mode\_announce(), TrickHLA::ExecutionControlBase::sim\_timeline, TrickHLA::ExecutionControlBase::simulation\_freeze\_time, and TrickHLA::ExecutionControlBase::time\_padding.

Referenced by process\_mode\_interaction().

#### 7.14.3.34 publish()

```
void ExecutionControl::publish ( ) [virtual]
```

Publish the [ExecutionControl](#) objects and interactions.  
 Implements [TrickHLA::ExecutionControlBase](#).  
 Definition at line 533 of file DIS/ExecutionControl.cpp.

#### 7.14.3.35 `role_determination_process()`

```
virtual void DIS::ExecutionControl::role_determination_process ( ) [virtual]
```

Determine the federate role in the federation execution.

#### 7.14.3.36 `run_mode_transition()`

```
bool ExecutionControl::run_mode_transition ( ) [virtual]
```

The run mode transition routine.

##### Returns

Currently always returns True.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1202 of file DIS/ExecutionControl.cpp.

References [TrickHLA::SyncPnt::achieve\\_sync\\_point\(\)](#), [TrickHLA::Federate::check\\_for\\_shutdown\\_with\\_termination\(\)](#), [TrickHLA::ExecutionControlBase::current\\_execution\\_control\\_mode](#), [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::ExecutionControlBase::does\\_cte\\_timeline\\_exist\(\)](#), [TrickHLA::EXECUTION\\_CONTROL\\_RUNNING](#), [DIS::EXECUTION\\_MODE\\_RUNNING](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::ExecutionControlBase::get\\_cte\\_time\(\)](#), [get\\_execution\\_configuration\(\)](#), [DIS::ExecutionConfiguration::get\\_next\\_mode\\_cte\\_time\(\)](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA::SyncPntListBase::get\\_sync\\_pnt\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [process\\_execution\\_control\\_updates\(\)](#), [TrickHLA::SyncPntListBase::register\\_sync\\_pnt\(\)](#), [TrickHLA::Object::send\\_init\\_data\(\)](#), [DIS::ExecutionConfiguration::set\\_current\\_execution\\_mode\(\)](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), [THLA\\_ENDL](#), [HLA\\_NEWLINE](#), [TrickHLA::SyncPnt::wait\\_for\\_announce\(\)](#), [TrickHLA::SyncPnt::wait\\_for\\_synchronization\(\)](#), and [DIS::ExecutionConfiguration::wait\\_on\\_update\(\)](#).

Referenced by [process\\_execution\\_control\\_updates\(\)](#).

#### 7.14.3.37 `set_mode_request_from_mtr()`

```
void ExecutionControl::set_mode_request_from_mtr (
    MTREnum mtr_value ) [virtual]
```

Translate MTR into a pending execution mode transition.

##### Parameters

<code>mtr_value</code>	MTR value for next execution mode.
------------------------	------------------------------------

Definition at line 600 of file DIS/ExecutionControl.cpp.

References [TrickHLA::EXECUTION\\_CONTROL\\_FREEZE](#), [TrickHLA::EXECUTION\\_CONTROL\\_INITIALIZING](#), [TrickHLA::EXECUTION\\_CONTROL\\_RUNNING](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::EXECUTION\\_CONTROL\\_UNINITIALIZED](#), [DIS::MTR\\_GOTO\\_FREEZE](#), [DIS::MTR\\_GOTO\\_RUN](#), [DIS::MTR\\_GOTO\\_SHUTDOWN](#), [DIS::MTR\\_INITIALIZING](#), [DIS::MTR\\_UNINITIALIZED](#), [pending\\_mtr](#), and [set\\_next\\_execution\\_control\\_mode\(\)](#).

Referenced by [process\\_mode\\_transition\\_request\(\)](#).

#### 7.14.3.38 set\_next\_execution\_control\_mode()

```
void ExecutionControl::set_next_execution_control_mode (
    TrickHLA::ExecutionControlEnum exec_control ) [virtual]
```

Sets the next ExCO run mode.

##### Parameters

exec_control	Next Execution configuration run mode.
--------------	--

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 637 of file DIS/ExecutionControl.cpp.

References TrickHLA::ScenarioTimeline::compute\_simulation\_time(), TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, DIS::EXECUTION\_MODE\_FREEZE, DIS::EXECUTION\_MODE\_INITIALIZING, DIS::EXECUTION\_MODE\_RUNNING, DIS::EXECUTION\_MODE\_SHUTDOWN, DIS::EXECUTION\_MODE\_UNINITIALIZED, TrickHLA::ExecutionControlBase::get\_cte\_time(), get\_execution\_configuration(), TrickHLA::ExecutionControlBase::get\_manager(), DIS::ExecutionConfiguration::get\_next\_mode\_cte\_time(), TrickHLA::ExecutionControlBase::get\_scenario\_time(), TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::ExecutionControlBase::next\_mode\_scenario\_time, TrickHLA::ExecutionControlBase::requested\_execution\_control\_mode, TrickHLA::ExecutionControlBase::scenario\_freeze\_time, TrickHLA::ExecutionControlBase::scenario\_timeline, DIS::ExecutionConfiguration::set\_next\_execution\_mode(), DIS::ExecutionConfiguration::set\_next\_mode\_cte\_time(), DIS::ExecutionConfiguration::set\_next\_mode\_scenario\_time(), DIS::ExecutionConfiguration::set\_scenario\_time\_epoch(), TrickHLA::ExecutionControlBase::should\_print(), TrickHLA::ExecutionControlBase::simulation\_freeze\_time, THLA-ENDL, and TrickHLA::ExecutionControlBase::time\_padding.

Referenced by set\_mode\_request\_from\_mtr(), and shutdown\_mode\_announce().

#### 7.14.3.39 set\_pending\_mtr()

```
bool ExecutionControl::set_pending_mtr (
    MTREnum mtr_value ) [virtual]
```

Definition at line 569 of file DIS/ExecutionControl.cpp.

References is\_mtr\_valid(), and pending\_mtr.

#### 7.14.3.40 set\_time\_padding()

```
virtual void DIS::ExecutionControl::set_time_padding (
    double t ) [virtual]
```

Set the time-padding used to offset the go to run time.

##### Parameters

t	Time in seconds to pad for time based mode transitions.
---	---

Reimplemented from [TrickHLA::ExecutionControlBase](#).

#### 7.14.3.41 setup\_interaction\_ref\_attributes()

```
virtual void DIS::ExecutionControl::setup_interaction_ref_attributes ( ) [virtual]
```

Setup the [ExecutionControl](#) interaction Trick ref ATTRIBUTES.

Implements [TrickHLA::ExecutionControlBase](#).

#### 7.14.3.42 setup\_interaction\_RTI\_handles()

```
void ExecutionControl::setup_interaction_RTI_handles ( ) [virtual]
```

Setup the [ExecutionControl](#) interaction HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 385 of file DIS/ExecutionControl.cpp.

#### 7.14.3.43 setup\_object\_ref\_attributes()

```
virtual void DIS::ExecutionControl::setup_object_ref_attributes ( ) [virtual]
```

Setup the [ExecutionControl](#) object Trick ref ATTRIBUTES.

Implements [TrickHLA::ExecutionControlBase](#).

#### 7.14.3.44 setup\_object\_RTI\_handles()

```
void ExecutionControl::setup_object_RTI_handles ( ) [virtual]
```

Setup the [ExecutionControl](#) objects HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 377 of file DIS/ExecutionControl.cpp.

#### 7.14.3.45 shutdown() [1/2]

```
void ExecutionControl::shutdown ( ) [virtual]
```

Execution control specific shutdown process.

**Trick Job Class:** *shutdown*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 342 of file DIS/ExecutionControl.cpp.

#### 7.14.3.46 shutdown() [2/2]

```
virtual void DIS::ExecutionControl::shutdown ( ) [virtual]
```

Execution control specific shutdown process.

Implements [TrickHLA::ExecutionControlBase](#).

#### 7.14.3.47 shutdown\_mode\_announce()

```
void ExecutionControl::shutdown_mode_announce ( ) [virtual]
```

Announce to the federation execution that a shutdown is occurring.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1335 of file DIS/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::clear\\_mode\\_transition\\_requested\(\)](#), [TrickHLA::ExecutionControlBase:::current\\_execution\\_control\\_mode](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::EXECUTION\\_CONTROL\\_UNINITIALIZED](#), [get\\_execution\\_configuration\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::Object::send\\_init\\_data\(\)](#), and [set\\_next\\_execution\\_control\\_mode\(\)](#).

Referenced by [process\\_mode\\_transition\\_request\(\)](#).

#### 7.14.3.48 shutdown\_mode\_transition()

```
void ExecutionControl::shutdown_mode_transition ( ) [virtual]
```

The shutdown mode transition routine.

**Trick Job Class:** `shutdown`

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1364 of file DIS/ExecutionControl.cpp.

References `TrickHLA::ExecutionControlBase::current_execution_control_mode`, `TrickHLA::EXECUTION_CONTROL`↔  
`_UNINITIALIZED`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::get_RTI_ambassador()`, `TrickHLA`↔  
`::ExecutionControlBase::is_master()`, and `TrickHLA::SyncPntListBase::register_sync_pnt()`.

#### 7.14.3.49 subscribe()

```
void ExecutionControl::subscribe ( ) [virtual]
```

Subscribe to the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 543 of file DIS/ExecutionControl.cpp.

#### 7.14.3.50 sync\_point\_registration\_failed()

```
void ExecutionControl::sync_point_registration_failed (
    std::wstring const & label,
    bool not_unique ) [virtual]
```

Callback from [TrickHLA::FedAmb](#) through [TrickHLA::Federate](#) for when registration of a synchronization point fails. and  
is one of the sync-points created.

##### Parameters

<i>label</i>	Sync-point label.
<i>not_unique</i>	True if not unique label.

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 504 of file DIS/ExecutionControl.cpp.

References `TrickHLA::ExecutionControlBase::federate`, `DIS::INITIALIZE_SYNC_POINT`, `TrickHLA::ExecutionControl`↔  
`Base::set_master()`, `TrickHLA::Federate::set_startup()`, `DIS::STARTUP_SYNC_POINT`, and `THLA_NEWLINE`.

#### 7.14.3.51 sync\_point\_registration\_succeeded()

```
void ExecutionControl::sync_point_registration_succeeded (
    std::wstring const & label ) [virtual]
```

Callback from [TrickHLA::FedAmb](#) through [TrickHLA::Federate](#) for when registration of a synchronization point success.  
and is one of the sync-points created.

##### Parameters

<i>label</i>	Sync-point label.
--------------	-------------------

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 491 of file DIS/ExecutionControl.cpp.

References `TrickHLA::ExecutionControlBase::federate`, `DIS::INITIALIZE_SYNC_POINT`, `TrickHLA::ExecutionControl`↔  
`Base::set_master()`, `TrickHLA::Federate::set_startup()`, and `DIS::STARTUP_SYNC_POINT`.

### 7.14.3.52 `unpublish()`

```
void ExecutionControl::unpublish ( ) [virtual]
Unpublish the ExecutionControl objects and interactions.
Implements TrickHLA::ExecutionControlBase.
Definition at line 538 of file DIS/ExecutionControl.cpp.
```

### 7.14.3.53 `unsubscribe()`

```
void ExecutionControl::unsubscribe ( ) [virtual]
Unsubscribe the ExecutionControl objects and interactions.
Implements TrickHLA::ExecutionControlBase.
Definition at line 548 of file DIS/ExecutionControl.cpp.
```

### 7.14.3.54 `wait_on_init_sync_point()`

```
bool DIS::ExecutionControl::wait_on_init_sync_point ( ) [inline], [virtual]
Test to see if ExecutionControl needs to wait on initialization synchronization point.
Most ExecutionControl approaches require that we wait for specific initialization synchronization points in specific orders. Currently, only the 'Simple' and 'DIS' scheme do not.
```

#### Returns

True if `ExecutionControl` needs to wait on the initialization synchronization points.

Reimplemented from `TrickHLA::ExecutionControlBase`.  
 Definition at line 171 of file DIS/ExecutionControl.hh.

## 7.14.4 Friends And Related Function Documentation

### 7.14.4.1 `init_attrDIS__ExecutionControl`

```
void init_attrDIS__ExecutionControl ( ) [friend]
```

### 7.14.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
Definition at line 62 of file DIS/ExecutionControl.hh.
```

## 7.14.5 Field Documentation

### 7.14.5.1 `checktime`

```
TrickHLA::Int64Time DIS::ExecutionControl::checktime [protected]
```

#### Units: –

For `DIS`: Checking time to pause

Definition at line 257 of file DIS/ExecutionControl.hh.

Referenced by `exit_freeze()`.

#### 7.14.5.2 pause\_sync\_pts

`PausePointList DIS::ExecutionControl::pause_sync_pts [protected]`

**Units:** –

Synchronization points used for pausing the sim.

Definition at line 258 of file `DIS/ExecutionControl.hh`.

Referenced by `add_pause()`, and `exit_freeze()`.

#### 7.14.5.3 pending\_mtr

`MTREnum DIS::ExecutionControl::pending_mtr [protected]`

**Units:** –

Pending Mode Transition Requested.

Definition at line 255 of file `DIS/ExecutionControl.hh`.

Referenced by `check_mode_transition_request()`, `process_mode_transition_request()`, `set_mode_request_from_mtr()`, and `set_pending_mtr()`.

#### 7.14.5.4 type

`const std::wstring DIS::ExecutionControl::type = L"DIS" [static], [protected]`

**Units:** –

`ExecutionControl` type string.

Definition at line 253 of file `DIS/ExecutionControl.hh`.

Referenced by `get_type()`.

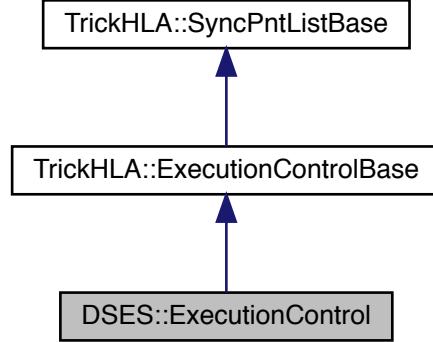
The documentation for this class was generated from the following files:

- [DIS/ExecutionControl.hh](#)
- [DIS/ExecutionControl.cpp](#)

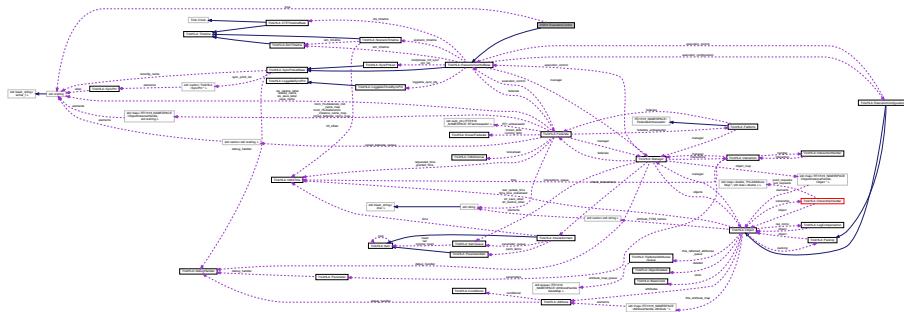
## 7.15 DSES::ExecutionControl Class Reference

```
#include <ExecutionControl.hh>
```

Inheritance diagram for DSES::ExecutionControl:



Collaboration diagram for DSES::ExecutionControl:



## Public Member Functions

- `ExecutionControl ()`  
*Default constructor for the `DSES ExecutionControl` class.*
- `virtual ~ExecutionControl ()`  
*Destructor for the `DSES ExecutionControl` class.*
- `virtual const std::wstring & get_type ()`  
*Get the `ExecutionControl` type identification string.*
- `virtual void initialize (TrickHLA::Federate &fed)`  
*Execution Control initialization routine.*
- `virtual void join_federation_process ()`  
*Join federation execution process.*
- `virtual void pre_multi_phase_init_processes ()`  
*Process run before the multi-phase initialization begins.*
- `virtual void post_multi_phase_init_process ()`  
*Process run after the multi-phase initialization ends.*

- `virtual void shutdown ()`  
*Execution control specific shutdown process.*
- `void determine_federation_master ()`  
*Determine if this federate is the Master for the federation.*
- `virtual void setup_object_ref_attributes ()`
- `virtual void setup_interaction_ref_attributes ()`
- `virtual void setup_object_RTI_handles ()`
- `virtual void setup_interaction_RTI_handles ()`
- `virtual void add_multiphase_init_sync_points ()`
- `virtual void announce_sync_point (RTI1516_NAMESPACE::RTIambassador &rti_ambassador, std::wstring const &label, RTI1516_USERDATA const &user_supplied_tag)`  
*The RTI has announced the existence of a synchronization point.*
- `void achieve_all_multiphase_init_sync_pnts (RTI1516_NAMESPACE::RTIambassador &rti_ambassador) throw ( RTI1516_NAMESPACE::SynchronizationPointLabelNotAnnounced, RTI1516_NAMESPACE::FederateNotExecutionMember, RTI1516_NAMESPACE::SaveInProgress, RTI1516_NAMESPACE::RestoreInProgress, RTI1516_NAMESPACE::NotConnected, RTI1516_NAMESPACE::RTIinternalError )`  
*Achieve all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not one of the predefined `ExecutionControl` synchronization points.*
- `void wait_for_all_multiphase_init_sync_pnts ()`  
*Wait for all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not one of the predefined `ExecutionControl` synchronization points.*
- `virtual void publish ()`
- `virtual void unpublish ()`
- `virtual void subscribe ()`
- `virtual void unsubscribe ()`
- `virtual bool object_instance_name_reservation_failed (std::wstring const &obj_instance_name)`  
*The object instance name reservation failed for the given name.*
- `virtual void set_next_execution_control_mode (TrickHLA::ExecutionControlEnum exec_control)`  
*Sets the next ExCO run mode.*
- `virtual bool process_execution_control_updates ()`  
*Process changes from any received Execution Control Objects (ExCOs).*
- `virtual bool check_mode_transition_request ()`  
*Check to see if a new MTR is valid.*
- `virtual bool process_mode_interaction ()`  
*Process a new mode interaction.*
- `virtual bool process_mode_transition_request ()`  
*Process a new Mode Transition Request (MTR).*
- `virtual void clear_mode_values ()`  
*Clear the Mode Transition Request flag, the requested execution mode, and the current execution mode.*
- `virtual bool run_mode_transition ()`  
*The run mode transition routine.*
- `virtual void freeze_mode_announce ()`  
*Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.*
- `virtual bool freeze_mode_transition ()`  
*The freeze mode transition routine.*
- `virtual void shutdown_mode_announce ()`  
*Announce to the federation execution that a shutdown is occurring.*
- `virtual void shutdown_mode_transition ()`

*The shutdown mode transition routine.*

- `virtual bool check_freeze_exit ()`  
*Check for exit from freeze.*
- `virtual bool set_pending_mtr (MTREnum mtr_value)`
- `virtual bool is_mtr_valid (MTREnum mtr_value)`

*Determine if the Mode Transition Request (MTR) is valid given the current mode.*

- `virtual void set_mode_request_from_mtr (MTREnum mtr_value)`  
*Translate MTR into a pending execution mode transition.*
- `virtual void set_time_padding (double t)`  
*Set the time-padding used to offset the go to run time.*

## Protected Member Functions

- `ExecutionConfiguration * get_execution_configuration ()`  
*Return the relevant `DSES::ExecutionConfiguration` object.*

## Protected Attributes

- `MTREnum pending_mtr`  
**Units:** –  
*Pending Mode Transition Requested.*

## Static Protected Attributes

- `static const std::wstring type = L"DSES"`  
**Units:** –  
*ExecutionControl type string.*

## Private Member Functions

- `ExecutionControl (const ExecutionControl &rhs)`  
*Copy constructor for `ExecutionControl` class.*
- `ExecutionControl & operator= (const ExecutionControl &rhs)`  
*Assignment operator for `ExecutionControl` class.*

## Friends

- `class InputProcessor`
- `void init_attrDSES__ExecutionControl ()`

## Additional Inherited Members

### 7.15.1 Detailed Description

Definition at line 53 of file DSES/ExecutionControl.hh.

### 7.15.2 Constructor & Destructor Documentation

### 7.15.2.1 ExecutionControl() [1/2]

```
ExecutionControl::ExecutionControl ( )
Default constructor for the DSES ExecutionControl class.
Trick Job Class: initialization
Definition at line 72 of file DSES/ExecutionControl.cpp.
```

### 7.15.2.2 ~ExecutionControl()

```
ExecutionControl::~ExecutionControl ( ) [virtual]
Destructor for the DSES ExecutionControl class.
Trick Job Class: shutdown
Definition at line 94 of file DSES/ExecutionControl.cpp.
References clear\_mode\_values\(\).
```

### 7.15.2.3 ExecutionControl() [2/2]

```
DSES::ExecutionControl::ExecutionControl (
    const ExecutionControl & rhs ) [private]
Copy constructor for ExecutionControl class.
This constructor is private to prevent inadvertent copies.
```

## 7.15.3 Member Function Documentation

### 7.15.3.1 achieve\_all\_multiphase\_init\_sync\_pnts()

```
void ExecutionControl::achieve_all_multiphase_init_sync_pnts (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador ) throw ( RTI1516_NAMESPACE::←
SynchronizationPointLabelNotAnnounced, RTI1516_NAMESPACE::FederateNotExecutionMember, RTI1516←
NAMESPACE::SaveInProgress, RTI1516_NAMESPACE::RestoreInProgress, RTI1516_NAMESPACE::NotConnected,
RTI1516_NAMESPACE::RTIinternalError)
Achieve all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not
one of the predefined ExecutionControl synchronization points.
```

#### Parameters

<code>rti_ambassador</code>	Reference to the HLA RTI Ambassador instance.
-----------------------------	---

#### **Trick Job Class:** *initialization*

Definition at line 393 of file DSES/ExecutionControl.cpp.  
 References [TrickHLA::SyncPnt::exists\(\)](#), [DSES::INITIALIZE\\_SYNC\\_POINT](#), [TrickHLA::SyncPnt::is\\_achieved\(\)](#), [TrickHLA::SyncPnt::label](#), [DSES::SIM\\_CONFIG\\_SYNC\\_POINT](#), and [DSES::STARTUP\\_SYNC\\_POINT](#).

### 7.15.3.2 add\_multiphase\_init\_sync\_points()

```
void ExecutionControl::add_multiphase_init_sync_points ( ) [virtual]
Add initialization synchronization points to regulate startup.
Trick Job Class: initialization
Reimplemented from TrickHLA::ExecutionControlBase.
Definition at line 347 of file DSES/ExecutionControl.cpp.
```

References TrickHLA::SyncPntListBase::add\_sync\_pnt(), DSES::INITIALIZE\_SYNC\_POINT, DSES::SIM\_CONFIG\_← SYNC\_POINT, and DSES::STARTUP\_SYNC\_POINT.  
 Referenced by pre\_multi\_phase\_init\_processes().

### 7.15.3.3 announce\_sync\_point()

```
void ExecutionControl::announce_sync_point (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    std::wstring const & label,
    RTI1516_USERDATA const & user_supplied_tag ) [virtual]
```

The RTI has announced the existence of a synchronization point.

#### Parameters

<i>rti_ambassador</i>	Reference to the HLA RTI Ambassador instance.
<i>label</i>	Sync-point label.
<i>user_supplied_tag</i>	Use supplied tag.

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 361 of file DSES/ExecutionControl.cpp.

References TrickHLA::SyncPntListBase::achieve\_sync\_pnt(), TrickHLA::SyncPntListBase::contains(), TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERA← TE, TrickHLA::SyncPntListBase::mark\_announced(), TrickHLA::DebugHandler::should\_print(), TrickHLA::Execution← ControlBase::should\_print(), and THLA\_NEWLINE.

### 7.15.3.4 check\_freeze\_exit()

```
bool ExecutionControl::check_freeze_exit ( ) [virtual]
```

Check for exit from freeze.

#### Returns

True if should exit from freeze.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 1338 of file DSES/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::current\_execution\_control\_mode, TrickHLA::ExecutionControlBase::execution\_configuration, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::ExecutionControlBase::federate, TrickHLA::ExecutionControlBase::get\_manager(), TrickHLA::Object::is\_attribute\_update\_requested(), TrickHLA::ExecutionControlBase::is\_master(), process\_execution\_control\_updates(), process\_mode\_transition← \_request(), TrickHLA::Object::receive\_init\_data(), TrickHLA::Manager::send\_requested\_data(), and TrickHLA::Federate::shutdown().

### 7.15.3.5 check\_mode\_transition\_request()

```
bool ExecutionControl::check_mode_transition_request ( ) [virtual]
```

Check to see if a new MTR is valid.

**Returns**

True if new MTR is valid.

Definition at line 749 of file DSES/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::ExecutionControlBase::is\\_mode\\_transition\\_requested\(\)](#), [is\\_mtr\\_valid\(\)](#), [DSES::mtr\\_enum\\_to\\_string\(\)](#), [pending\\_mtr](#), and [THLA\\_ENDL](#).

Referenced by [process\\_mode\\_transition\\_request\(\)](#).

**7.15.3.6 clear\_mode\_values()**

```
virtual void DSES::ExecutionControl::clear_mode_values ( ) [virtual]
```

Clear the Mode Transition Request flag, the requested execution mode, and the current execution mode.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Referenced by [~ExecutionControl\(\)](#).

**7.15.3.7 determine\_federation\_master()**

```
void ExecutionControl::determine_federation_master ( )
```

Determine if this federate is the Master for the federation.

**Trick Job Class:** *initialization*

Definition at line 534 of file DSES/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::ExecutionControlBase::execution\\_configuration](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\\_preset\(\)](#), [TrickHLA::Object::reserve\\_object\\_name\\_with\\_RTI\(\)](#), [TrickHLA::ExecutionConfigurationBase::set\\_master\(\)](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), [THLA\\_NEWLINE](#), and [TrickHLA::Object::wait\\_on\\_object\\_name\\_reservation\(\)](#).

Referenced by [pre\\_multi\\_phase\\_init\\_processes\(\)](#).

**7.15.3.8 freeze\_mode\_announce()**

```
void ExecutionControl::freeze_mode_announce ( ) [virtual]
```

Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1293 of file DSES/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), and [TrickHLA::SyncPntListBase::register\\_sync\\_pnt\(\)](#).

Referenced by [process\\_execution\\_control\\_updates\(\)](#), and [process\\_mode\\_transition\\_request\(\)](#).

**7.15.3.9 freeze\_mode\_transition()**

```
bool ExecutionControl::freeze_mode_transition ( ) [virtual]
```

The freeze mode transition routine.

**Returns**

Currently always returns False.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1304 of file DSES/ExecutionControl.cpp.

References [TrickHLA::SyncPnt::achieve\\_sync\\_point\(\)](#), [TrickHLA::ExecutionControlBase::current\\_execution\\_control\\_mode](#), [TrickHLA::EXECUTION\\_CONTROL\\_FREEZE](#), [DSES::EXECUTION\\_MODE\\_FREEZE](#), [TrickHLA::ExecutionControlBase::federate](#), [get\\_execution\\_configuration\(\)](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA::Sync](#)

PntListBase::get\_sync\_pnt(), DSES::ExecutionConfiguration::set\_current\_execution\_mode(), THLA\_ENDL, TrickHLA::SyncPnt::wait\_for\_announce(), and TrickHLA::SyncPnt::wait\_for\_synchronization().

#### 7.15.3.10 get\_execution\_configuration()

`ExecutionConfiguration * ExecutionControl::get_execution_configuration ( ) [protected], [virtual]`  
 Return the relevant [DSES::ExecutionConfiguration](#) object.

##### Returns

Pointer to the relevant [DSES::ExecutionConfiguration](#) object.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 1428 of file DSES/ExecutionControl.cpp.

References THLA\_ENDL.

Referenced by `freeze_mode_transition()`, `is_mtr_valid()`, `process_execution_control_updates()`, `process_mode_transition_request()`, `run_mode_transition()`, `set_next_execution_control_mode()`, and `shutdown_mode_announce()`.

#### 7.15.3.11 get\_type()

`virtual const std::wstring& DSES::ExecutionControl::get_type ( ) [inline], [virtual]`  
 Get the [ExecutionControl](#) type identification string.

##### Returns

A constant reference to the type identification string.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 75 of file DSES/ExecutionControl.hh.

References type.

#### 7.15.3.12 initialize()

`void ExecutionControl::initialize (`  
`TrickHLA::Federate & fed ) [virtual]`

Execution Control initialization routine.

##### Parameters

<code>fed</code>	The associated <a href="#">TrickHLA::Federate</a> .
------------------	---

This routine will set a lot of the data in the [TrickHLA::Federate](#) that is required for this execution control scheme. This should greatly simplify input files and reduce input file setting errors.

**Trick Job Class:** *initialization*

Definition at line 107 of file DSES/ExecutionControl.cpp.

References `TrickHLA::SyncPntListBase::add_sync_pnt()`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::execution_configuration`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::ExecutionControlBase::get_manager()`, `TrickHLA::Object::initialize()`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::ExecutionControlBase::is_master_preset()`, `TrickHLA::ExecutionControlBase::least_common_time_step`, `pre_multi_phase_init_processes()`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::Federate::time_constrained`, `TrickHLA::Federate::time_management`, `TrickHLA::Federate::time_regulating`, and `TrickHLA::ExecutionControlBase::use_preset_master`.

**7.15.3.13 is\_mtr\_valid()**

```
bool ExecutionControl::is_mtr_valid (
    MTREnum mtr_value ) [virtual]
```

Determine if the Mode Transition Request (MTR) is valid given the current mode.

**Returns**

True if valid, false otherwise.

**Parameters**

<i>mtr_value</i>	Mode transition request.
------------------	--------------------------

Definition at line 581 of file DSES/ExecutionControl.cpp.

References DSES::ExecutionConfiguration::current\_execution\_mode, DSES::EXECUTION\_MODE\_FREEZE, DSES::EXECUTION\_MODE\_INITIALIZING, DSES::EXECUTION\_MODE\_RUNNING, DSES::EXECUTION\_MODE\_SHUTDOWN, get\_execution\_configuration(), DSES::MTR\_GOTO\_FREEZE, DSES::MTR\_GOTO\_RUN, and DSES::MTR\_GOTO\_SHUTDOWN.

Referenced by check\_mode\_transition\_request(), and set\_pending\_mtr().

**7.15.3.14 join\_federation\_process()**

```
void ExecutionControl::join_federation_process ( ) [virtual]
```

Join federation execution process.

Setup the Trick Ref ATTRIBUTES for [ExecutionControl](#).

This routine implements the [DSES](#) Join Federation Process described in section 7.2 and figure 7-3.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 181 of file DSES/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::join\_federation\_process().

**7.15.3.15 object\_instance\_name\_reservation\_failed()**

```
virtual bool DSES::ExecutionControl::object_instance_name_reservation_failed (
    std::wstring const & obj_instance_name ) [virtual]
```

The object instance name reservation failed for the given name.

**Returns**

True if [ExecutionConfiguration](#) object handled the failure.

**Parameters**

<i>obj_instance_name</i>	Object instance name.
--------------------------	-----------------------

Reimplemented from [TrickHLA::ExecutionControlBase](#).

**7.15.3.16 operator=()**

```
ExecutionControl& DSES::ExecutionControl::operator= (
    const ExecutionControl & rhs ) [private]
```

Assignment operator for [ExecutionControl](#) class.

This assignment operator is private to prevent inadvertent copies.

#### 7.15.3.17 `post_multi_phase_init_process()`

```
void ExecutionControl::post_multi_phase_init_process ( ) [virtual]
```

Process run after the multi-phase initialization ends.

This routine implements the [DSES](#) post multi-phase initialization process.

**Trick Job Class:** *initialization*

Definition at line 308 of file [DSES/ExecutionControl.cpp](#).

References [TrickHLA::Federate::achieve\\_and\\_wait\\_for\\_synchronization\(\)](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::Federate::setup\\_time\\_management\(\)](#), and [DSES::STARTUP\\_SYNC\\_POINT](#).

#### 7.15.3.18 `pre_multi_phase_init_processes()`

```
void ExecutionControl::pre_multi_phase_init_processes ( ) [virtual]
```

Process run before the multi-phase initialization begins.

This routine implements the [DSES](#) pre multi-phase initialization process.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 195 of file [DSES/ExecutionControl.cpp](#).

References [TrickHLA::Federate::achieve\\_and\\_wait\\_for\\_synchronization\(\)](#), [add\\_multiphase\\_init\\_sync\\_points\(\)](#), [TrickHLA::Federate::create\\_and\\_join\\_federation\(\)](#), [TrickHLA::Federate::create\\_RTI\\_ambassador\\_and\\_connect\(\)](#), [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::Federate::destroy\\_orphaned\\_federation\(\)](#), [determine\\_federation\\_master\(\)](#), [TrickHLA::Federate::enable\\_async\\_delivery\(\)](#), [TrickHLA::ExecutionControlBase::execution\\_configuration](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::ExecutionControlBase::get\\_manager\(\)](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA::Federate::initialize\\_MOM\\_handles\(\)](#), [DSES::INITIALIZE\\_SYNC\\_POINT](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::Object::mark\\_required\(\)](#), [TrickHLA::Manager::publish\\_and\\_subscribe\(\)](#), [TrickHLA::ExecutionControlBase::receive\\_execution\\_configuration\(\)](#), [TrickHLA::SyncPntListBase::register\\_all\\_sync\\_pnts\(\)](#), [TrickHLA::Manager::register\\_objects\\_with\\_RTI\(\)](#), [TrickHLA::Manager::reserve\\_object\\_names\\_with\\_RTI\(\)](#), [TrickHLA::ExecutionConfigurationBase::reset\\_ownership\\_states\(\)](#), [TrickHLA::ExecutionConfigurationBase::reset\\_preferred\\_order\(\)](#), [TrickHLA::ExecutionControlBase::send\\_execution\\_configuration\(\)](#), [TrickHLA::Manager::setup\\_all\\_ref\\_attributes\(\)](#), [TrickHLA::Manager::setup\\_all\\_RTI\\_handles\(\)](#), [TrickHLA::Manager::setup\\_preferred\\_order\\_with\\_RTI\(\)](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), [DSES::SIM\\_CONFIG\\_SYNC\\_POINT](#), [THLA\\_NEWLINE](#), [TrickHLA::SyncPntListBase::wait\\_for\\_all\\_announcements\(\)](#), [TrickHLA::Federate::wait\\_for\\_required\\_federates\\_to\\_join\(\)](#), [TrickHLA::Manager::wait\\_on\\_registration\\_of\\_required\\_objects\(\)](#), and [TrickHLA::Manager::wait\\_on\\_reservation\\_of\\_object\\_names\(\)](#).

Referenced by [initialize\(\)](#).

#### 7.15.3.19 `process_execution_control_updates()`

```
bool ExecutionControl::process_execution_control_updates ( ) [virtual]
```

Process changes from any received Execution Control Objects (ExCOs).

##### Returns

True if mode change occurred.

## Assumptions and Limitations:

- Called from the ExCO unpack routine.

**Trick Job Class:** *scheduled*Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 892 of file DSES/ExecutionControl.cpp.

References `TrickHLA::ExecutionConfigurationBase::clear_update_pending()`, `TrickHLA::ScenarioTimeline::compute<=simulation_time()`, `TrickHLA::ExecutionControlBase::cte_timeline`, `TrickHLA::ExecutionControlBase::current<=execution_control_mode`, `DSES::ExecutionConfiguration::current_execution_mode`, `TrickHLA::DEBUG_LEVEL<=4_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::does_cte_timeline_exist()`, `TrickHLA::execution_control_enum_to_string()`, `TrickHLA::EXECUTION_CONTROL_FREEZE`, `TrickHLA::EXECUTION_CONTROL_INITIALIZING`, `TrickHLA::EXECUTION_CONTROL_RUNNING`, `TrickHLA::EXECUTION_CONTROL_SHUTDOWN`, `TrickHLA::EXECUTION_CONTROL_UNINITIALIZED`, `DSES::execution_mode_enum_to_string()`, `DSES::EXECUTION_MODE_FREEZE`, `DSES::execution_mode_int16_to_enum()`, `DSES::EXECUTION_MODE_RUNNING`, `DSES::EXECUTION_MODE_SHUTDOWN`, `TrickHLA::ExecutionControlBase::federate`, `freeze_mode<=announce()`, `TrickHLA::Timeline::get_epoch()`, `get_execution_configuration()`, `TrickHLA::ScenarioTimeline::get_sim<=offset()`, `TrickHLA::SimTimeline::get_time()`, `TrickHLA::CTETimelineBase::get_time()`, `TrickHLA::ScenarioTimeline::get_time()`, `TrickHLA::ExecutionControlBase::is_master()`, `DSES::ExecutionConfiguration::next_execution_mode`, `DSES::ExecutionConfiguration::next_mode_cte_time`, `TrickHLA::ExecutionControlBase::next_mode_cte_time`, `DSES::ExecutionConfiguration::next_mode_scenario_time`, `TrickHLA::ExecutionControlBase::requested_execution<=control_mode`, `run_mode_transition()`, `TrickHLA::ExecutionControlBase::scenario_freeze_time`, `DSES::ExecutionConfiguration::scenario_time_epoch`, `TrickHLA::ExecutionControlBase::scenario_timeline`, `TrickHLA::Federate::should_print()`, `TrickHLA::ExecutionControlBase::sim_timeline`, `TrickHLA::ExecutionControlBase::simulation_freeze<=time`, `THLA_ENDL`, and `TrickHLA::ExecutionConfigurationBase::update_pending()`.

Referenced by `check_freeze_exit()`, and `run_mode_transition()`.**7.15.3.20 process\_mode\_interaction()**

```
virtual bool DSES::ExecutionControl::process_mode_interaction ( ) [inline], [virtual]
Process a new mode interaction.
```

**Returns**

True if new mode interaction is successfully processed.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 161 of file DSES/ExecutionControl.hh.

References `process_mode_transition_request()`.**7.15.3.21 process\_mode\_transition\_request()**

```
bool ExecutionControl::process_mode_transition_request ( ) [virtual]
Process a new Mode Transition Request (MTR).
```

**Returns**

True if new MTR is successfully processed.

Definition at line 781 of file DSES/ExecutionControl.cpp.

References `check_mode_transition_request()`, `TrickHLA::ExecutionControlBase::clear_mode_transition_requested()`, `TrickHLA::ExecutionControlBase::cte_timeline`, `TrickHLA::ExecutionControlBase::current_execution_control_mode`, `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::does_cte_timeline_exist()`, `TrickHLA::EXECUTION_CONTROL_FREEZE`, `TrickHLA::EXECUTION_CONTROL_RUNNING`, `TrickHLA::ExecutionControlBase::federate`, `freeze_mode_announce()`, `TrickHLA::Timeline::get_epoch()`, `get_execution_configuration()`, `TrickHLA::Federate::get_granted_time()`, `TrickHLA::Federate::get_requested_time()`,

TrickHLA::ScenarioTimeline::get\_sim\_offset(), TrickHLA::SimTimeline::get\_time(), TrickHLA::CTETimelineBase::get\_time(), TrickHLA::ScenarioTimeline::get\_time(), DSES::MTR\_GOTO\_FREEZE, DSES::MTR\_GOTO\_RUN, DSES::MTR\_GOTO\_SHUTDOWN, DSES::ExecutionConfiguration::next\_mode\_cte\_time, DSES::ExecutionConfiguration::next\_mode\_scenario\_time, pending\_mtr, TrickHLA::ExecutionControlBase::scenario\_freeze\_time, DSES::ExecutionConfiguration::scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Object::send\_init\_data(), set\_mode\_request\_from\_mtr(), TrickHLA::Federate::should\_print(), shutdown\_mode\_announce(), TrickHLA::ExecutionControlBase::sim\_timeline, TrickHLA::ExecutionControlBase::simulation\_freeze\_time, and TrickHLA::ExecutionControlBase::time\_padding.

Referenced by check\_freeze\_exit(), and process\_mode\_interaction().

### 7.15.3.22 publish()

void ExecutionControl::publish ( ) [virtual]

Publish the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 474 of file DSES/ExecutionControl.cpp.

### 7.15.3.23 run\_mode\_transition()

bool ExecutionControl::run\_mode\_transition ( ) [virtual]

The run mode transition routine.

Returns

Currently always returns True.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1205 of file DSES/ExecutionControl.cpp.

References TrickHLA::SyncPnt::achieve\_sync\_point(), TrickHLA::Federate::check\_for\_shutdown\_with\_termination(), TrickHLA::ExecutionControlBase::current\_execution\_control\_mode, TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ExecutionControlBase::does\_cte\_timeline\_exist(), TrickHLA::EXECUTION\_CONTROL\_RUNNING, DSES::EXECUTION\_MODE\_RUNNING, TrickHLA::ExecutionControlBase::federate, TrickHLA::ExecutionControlBase::get\_cte\_time(), get\_execution\_configuration(), DSES::ExecutionConfiguration::get\_next\_mode\_cte\_time(), TrickHLA::Federate::get\_RTI\_ambassador(), TrickHLA::SyncPntListBase::get\_sync\_pnt(), TrickHLA::ExecutionControlBase::is\_master(), process\_execution\_control\_updates(), TrickHLA::SyncPntListBase::register\_sync\_pnt(), TrickHLA::Object::send\_init\_data(), DSES::ExecutionConfiguration::set\_current\_execution\_mode(), TrickHLA::DebugHandler::should\_print(), THLA\_EN\_DL, THLA\_NEWLINE, TrickHLA::SyncPnt::wait\_for\_announce(), TrickHLA::SyncPnt::wait\_for\_synchronization(), and DSES::ExecutionConfiguration::wait\_on\_update().

Referenced by process\_execution\_control\_updates().

### 7.15.3.24 set\_mode\_request\_from\_mtr()

void ExecutionControl::set\_mode\_request\_from\_mtr ( MTREnum mtr\_value ) [virtual]

Translate MTR into a pending execution mode transition.

Parameters

<i>mtr_value</i>	MTR value for next execution mode.
------------------	------------------------------------

Definition at line 603 of file DSES/ExecutionControl.cpp.

References TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, Trick→

HLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, DSES::MTR\_GOTO\_FREEZE, DSES::MTR\_GOTO\_RUN, DSES::MTR\_GOTO\_SHUTDOWN, DSES::MTR\_INITIALIZING, DSES::MTR\_UNINITIALIZED, pending\_mtr, and set\_next\_execution\_control\_mode().

Referenced by process\_mode\_transition\_request().

#### 7.15.3.25 set\_next\_execution\_control\_mode()

```
void ExecutionControl::set_next_execution_control_mode (
    TrickHLA::ExecutionControlEnum exec_control ) [virtual]
```

Sets the next ExCO run mode.

##### Parameters

exec_control	Next Execution configuration run mode.
--------------	--

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 640 of file DSES/ExecutionControl.cpp.

References TrickHLA::ScenarioTimeline::compute\_simulation\_time(), TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, DSES::EXECUTION\_MODE\_FREEZE, DSES::EXECUTION\_MODE\_INITIALIZING, DSES::EXECUTION\_MODE\_RUNNING, DSES::EXECUTION\_MODE\_SHUTDOWN, DSES::EXECUTION\_MODE\_UNINITIALIZED, TrickHLA::ExecutionControlBase::get\_cte\_time(), get\_execution\_configuration(), TrickHLA::ExecutionControlBase::get\_manager(), DSES::ExecutionConfiguration::get\_next\_mode\_cte\_time(), TrickHLA::ExecutionControlBase::get\_scenario\_time(), TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::ExecutionControlBase::next\_mode\_scenario\_time, TrickHLA::ExecutionControlBase::requested\_execution\_control\_mode, TrickHLA::ExecutionControlBase::scenario\_freeze\_time, TrickHLA::ExecutionControlBase::scenario\_timeline, DSES::ExecutionConfiguration::set\_next\_execution\_mode(), DSES::ExecutionConfiguration::set\_next\_mode\_cte\_time(), DSES::ExecutionConfiguration::set\_next\_mode\_scenario\_time(), DSES::ExecutionConfiguration::set\_scenario\_time\_epoch(), TrickHLA::ExecutionControlBase::should\_print(), TrickHLA::ExecutionControlBase::simulation\_freeze\_time, THLA\_ENDL, and TrickHLA::ExecutionControlBase::time\_padding.

Referenced by set\_mode\_request\_from\_mtr(), and shutdown\_mode\_announce().

#### 7.15.3.26 set\_pending\_mtr()

```
bool ExecutionControl::set_pending_mtr (
    MTREnum mtr_value ) [virtual]
```

Definition at line 572 of file DSES/ExecutionControl.cpp.

References is\_mtr\_valid(), and pending\_mtr.

#### 7.15.3.27 set\_time\_padding()

```
virtual void DSES::ExecutionControl::set_time_padding (
    double t ) [virtual]
```

Set the time-padding used to offset the go to run time.

##### Parameters

t	Time in seconds to pad for time based mode transitions.
---	---

Reimplemented from [TrickHLA::ExecutionControlBase](#).

#### 7.15.3.28 `setup_interaction_ref_attributes()`

`virtual void DSES::ExecutionControl::setup_interaction_ref_attributes ( ) [virtual]`

Setup the [ExecutionControl](#) interaction Trick ref ATTRIBUTES.

Implements [TrickHLA::ExecutionControlBase](#).

#### 7.15.3.29 `setup_interaction_RTI_handles()`

`void ExecutionControl::setup_interaction_RTI_handles ( ) [virtual]`

Setup the [ExecutionControl](#) interaction HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 339 of file DSES/ExecutionControl.cpp.

#### 7.15.3.30 `setup_object_ref_attributes()`

`virtual void DSES::ExecutionControl::setup_object_ref_attributes ( ) [virtual]`

Setup the [ExecutionControl](#) object Trick ref ATTRIBUTES.

Implements [TrickHLA::ExecutionControlBase](#).

#### 7.15.3.31 `setup_object_RTI_handles()`

`void ExecutionControl::setup_object_RTI_handles ( ) [virtual]`

Setup the [ExecutionControl](#) objects HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 331 of file DSES/ExecutionControl.cpp.

#### 7.15.3.32 `shutdown()`

`void ExecutionControl::shutdown ( ) [virtual]`

Execution control specific shutdown process.

**Trick Job Class:** *shutdown*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 323 of file DSES/ExecutionControl.cpp.

#### 7.15.3.33 `shutdown_mode_announce()`

`void ExecutionControl::shutdown_mode_announce ( ) [virtual]`

Announce to the federation execution that a shutdown is occurring.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1379 of file DSES/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::clear\\_mode\\_transition\\_requested\(\)](#), [TrickHLA::ExecutionControlBase::current\\_execution\\_control\\_mode](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::EXECUTION\\_CONTROL\\_UNINITIALIZED](#), [get\\_execution\\_configuration\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::Object::send\\_init\\_data\(\)](#), and [set\\_next\\_execution\\_control\\_mode\(\)](#).

Referenced by [process\\_mode\\_transition\\_request\(\)](#).

### 7.15.3.34 shutdown\_mode\_transition()

```
void ExecutionControl::shutdown_mode_transition ( ) [virtual]
```

The shutdown mode transition routine.

**Trick Job Class:** *shutdown*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1408 of file DSES/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::current\\_execution\\_control\\_mode](#), [TrickHLA::EXECUTION\\_CONTROL](#)↔  
\_UNINITIALIZED, [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA](#)↔  
::ExecutionControlBase::is\_master(), and [TrickHLA::SyncPntListBase::register\\_sync\\_pnt\(\)](#).

### 7.15.3.35 subscribe()

```
void ExecutionControl::subscribe ( ) [virtual]
```

Subscribe to the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 484 of file DSES/ExecutionControl.cpp.

### 7.15.3.36 unpublish()

```
void ExecutionControl::unpublish ( ) [virtual]
```

Unpublish the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 479 of file DSES/ExecutionControl.cpp.

### 7.15.3.37 unsubscribe()

```
void ExecutionControl::unsubscribe ( ) [virtual]
```

Unsubscribe the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 489 of file DSES/ExecutionControl.cpp.

### 7.15.3.38 wait\_for\_all\_multiphase\_init\_sync\_pnts()

```
void ExecutionControl::wait_for_all_multiphase_init_sync_pnts ( )
```

Wait for all the user defined multi-phase initialization synchronization points if they are not already achieved and are not one of the predefined [ExecutionControl](#) synchronization points.

**Trick Job Class:** *initialization*

Definition at line 423 of file DSES/ExecutionControl.cpp.

References [TrickHLA::Federate::check\\_for\\_shutdown\\_with\\_termination\(\)](#), [TrickHLA::SyncPnt::exists\(\)](#), [TrickHLA](#)↔  
ExecutionControlBase::federate, [DSES::INITIALIZE\\_SYNC\\_POINT](#), [TrickHLA::SyncPnt::is\\_achieved\(\)](#), [TrickHLA](#)↔  
Federate::is\_execution\_member(), [TrickHLA::SyncPnt::label](#), [DSES::SIM\\_CONFIG\\_SYNC\\_POINT](#), [DSES::STARTUP\\_SYNC\\_POINT](#), [TrickHLA::SyncPntListBase::sync\\_point\\_list](#), and [THLA\\_ENDL](#).

## 7.15.4 Friends And Related Function Documentation

#### 7.15.4.1 `init_attrDSES__ExecutionControl`

```
void init_attrDSES__ExecutionControl ( )  [friend]
```

#### 7.15.4.2 `InputProcessor`

```
friend class InputProcessor  [friend]  
Definition at line 60 of file DSES/ExecutionControl.hh.
```

### 7.15.5 Field Documentation

#### 7.15.5.1 `pending_mtr`

```
MTREnum DSES::ExecutionControl::pending_mtr  [protected]
```

**Units:** –

Pending Mode Transition Requested.

Definition at line 205 of file DSES/ExecutionControl.hh.

Referenced by `check_mode_transition_request()`, `process_mode_transition_request()`, `set_mode_request_from_mtr()`, and `set_pending_mtr()`.

#### 7.15.5.2 `type`

```
const std::wstring DSES::ExecutionControl::type = L"DSES"  [static],  [protected]
```

**Units:** –

`ExecutionControl` type string.

Definition at line 203 of file DSES/ExecutionControl.hh.

Referenced by `get_type()`.

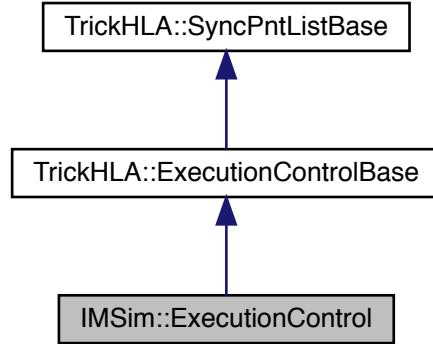
The documentation for this class was generated from the following files:

- [DSES/ExecutionControl.hh](#)
- [DSES/ExecutionControl.cpp](#)

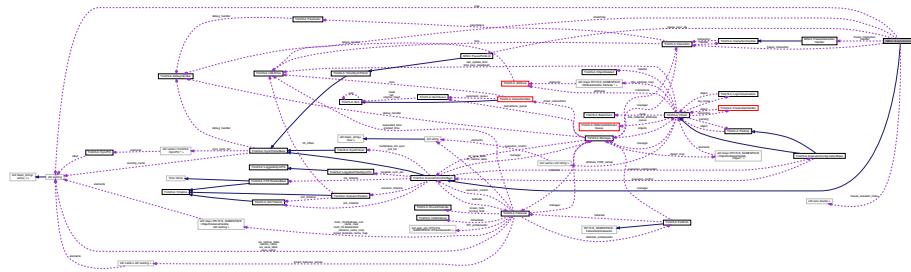
## 7.16 `IMSim::ExecutionControl` Class Reference

```
#include <ExecutionControl.hh>
```

Inheritance diagram for IMSim::ExecutionControl:



Collaboration diagram for IMSim::ExecutionControl:



## Public Member Functions

- `ExecutionControl ()`  
*Default constructor for the [IMSim ExecutionControl](#) class.*
- `virtual ~ExecutionControl ()`  
*Destructor for the [IMSim ExecutionControl](#) class.*
- `virtual const std::wstring & get_type ()`  
*Get the [ExecutionControl](#) type identification string.*
- `virtual void initialize (TrickHLA::Federate &federate)`  
*Execution Control initialization routine.*
- `virtual void join_federation_process ()`  
*Join federation execution process.*
- `virtual void pre_multi_phase_init_processes ()`  
*Process run before the multi-phase initialization begins.*
- `virtual void post_multi_phase_init_process ()`  
*Process run after the multi-phase initialization ends.*
- `virtual void shutdown ()`

*Execution control specific shutdown process.*

- `TrickHLA::FederateJoinEnum determine_if_late_joining_or_restoring_federate ()`

*Determine if this federate is late in joining the federation or is to restore itself. This call blocks until it has determined if the federate is late or not or when its been cleared to restore.*

- `virtual void setup_object_ref_attributes ()`
- `virtual void setup_interaction_ref_attributes ()`
- `virtual void setup_object_RTI_handles ()`
- `virtual void setup_interaction_RTI_handles ()`
- `virtual void add_multiphase_init_sync_points ()`
- `virtual void announce_sync_point (RTI1516_NAMESPACE::RTIambassador &rti_ambassador, std::wstring const &label, RTI1516_USERDATA const &user_supplied_tag)`

*The RTI has announced the existence of a synchronization point.*

- `void achieve_all_multiphase_init_sync_pnts (RTI1516_NAMESPACE::RTIambassador &rti_ambassador) throw ( RTI1516_NAMESPACE::SynchronizationPointLabelNotAnnounced, RTI1516_NAMESPACE::FederateNotExecutionMember, RTI1516_NAMESPACE::SaveInProgress, RTI1516_NAMESPACE::RestoreInProgress, RTI1516_NAMESPACE::NotConnected, RTI1516_NAMESPACE::RTIinternalError )`

*Achieve all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not one of the predefined `ExecutionControl` synchronization points.*

- `void wait_for_all_multiphase_init_sync_pnts ()`

*Wait for all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not one of the predefined `ExecutionControl` synchronization points.*

- `virtual void publish ()`
- `virtual void unpublish ()`
- `virtual void subscribe ()`
- `virtual void unsubscribe ()`
- `virtual bool mark_synchronized (std::wstring const &label)`

*Mark the given synchronization point as synchronized in the federation.*

- `virtual void receive_interaction (RTI1516_NAMESPACE::InteractionClassHandle const &theInteraction, RTI1516_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516_USERDATA const &theUserSuppliedTag, RTI1516_NAMESPACE::LogicalTime const &theTime, bool received_as_TSO)`

*Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.*

- `virtual void send_mode_transition_interaction (TrickHLA::ModeTransitionEnum requested_mode)`

*Send a mode transition request to the Master federate.*

- `virtual bool process_mode_interaction ()`

*Process a new mode interaction.*

- `virtual void set_next_execution_control_mode (TrickHLA::ExecutionControlEnum exec_control)`

*Sets the next `ExecutionControl` run mode.*

- `virtual bool process_execution_control_updates ()`

*Process changes from any received Execution Control Objects (ExCOs).*

- `virtual bool check_mode_transition_request ()`

*Check to see if a new MTR is valid.*

- `virtual bool process_mode_interaction ()`

*Process a new mode interaction.*

- `virtual bool process_mode_transition_request ()`

*Process a new Mode Transition Request (MTR).*

- `virtual void clear_mode_values ()`

*Clear the Mode Transition Request flag, the requested execution mode, and the current execution mode.*

- `virtual bool run_mode_transition ()`

*The run mode transition routine.*

- virtual void `freeze_mode_announce` ()  
*Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.*
- virtual bool `freeze_mode_transition` ()  
*The freeze mode transition routine.*
- virtual void `shutdown_mode_announce` ()  
*Announce to the federation execution that a shutdown is occurring.*
- virtual void `shutdown_mode_transition` ()  
*The shutdown mode transition routine.*
- virtual void `enter_freeze` ()  
*Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.*
- virtual virtual bool `check_freeze_exit` ()  
*Check for exit from freeze.*
- virtual void `exit_freeze` ()  
*Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.*
- virtual void `un_freeze` ()  
*Routine to handle `ExecutionControl` specific action needed to un-freeze.*
- virtual void `check_pause` (const double `check_pause_delta`)  
*Check if we hit a pause sync point and need to go to freeze.*
- void `check_pause_at_init` (const double `check_pause_delta`)  
*Checking if we started in freeze.*
- virtual bool `set_pending_mtr` (MTREnum `mtr_value`)  
*Determine if the Mode Transition Request (MTR) is valid given the current mode.*
- virtual void `set_mode_request_from_mtr` (MTREnum `mtr_value`)  
*Translate MTR into a pending execution mode transition.*
- virtual void `start_federation_save_at_scenario_time` (double `freeze_scenario_time`, const char \*`file_name`)  
*Start the Federation save at the specified scenario time.*
- virtual void `add_freeze_scenario_time` (double `t`)  
*Adds a freeze interaction time into freeze scenario time collection.*
- virtual void `trigger_freeze_interaction` (double &`freeze_scenario_time`)  
*Trigger a FREEZE interaction from the `FreezeInteractionHandler` and updated the supplied time with the time computed by the `FreezeInteractionHandler`.*
- virtual bool `check_freeze_time` ()  
*Checks for a freeze interaction time from the freeze sim time collection.*
- virtual bool `check_scenario_freeze_time` ()  
*Checks for scenario freeze times.*
- virtual void `check_pause` (const double `check_pause_delta`)  
*Check if we hit a pause sync point and need to go to freeze.*
- virtual void `check_pause_at_init` (const double `check_pause_delta`)  
*Checking if we started in freeze.*
- virtual void `add_pause` (TrickHLA::Int64Time \*`time`, std::wstring const &`label`)  
*Add pause time.*
- virtual void `clear_pause` (std::wstring const &`label`)  
*Clear a pause time by label.*
- virtual void `set_time_padding` (double `t`)  
*Set the time-padding used to offset the go to run time.*
- virtual bool `is_save_and_restore_supported` ()

- virtual bool `is_save_initiated ()`  
*Checks if Save has been initiated by this `ExecutionControl` method.*
- virtual bool `perform_save ()`  
*Federates that did not announce the save, perform a save.*
- void `convert_loggable_sync_pts ()`  
*Converts HLA sync points into something Trick can save in a checkpoint.*
- void `reinstate_logged_sync_pts ()`  
*Converts checkpointed sync points into HLA sync points.*

## Protected Member Functions

- `ExecutionConfiguration * get_execution_configuration ()`  
*Return the relevant `IMSim::ExecutionConfiguration` object.*

## Protected Attributes

- `MTREnum pending_mtr`  
**Units:** –  
*Pending Mode Transition Requested.*
- `int freeze_inter_count`  
**Data I/O:** \*\*  
*Number of `TrickHLA` Freeze Interactions.*
- `TrickHLA::Interaction * freeze_interaction`  
**Data I/O:** \*\*  
*Interaction to FREEZE the sim at a specified time. `MTRInteractionHandler` `mtr_interaction_handler`; //< **Units:** – SRFOM MTR interaction handler.*
- `IMSim::FreezeInteractionHandler freeze_interaction_handler`  
**Units:** –  
*Freeze interaction handler.*
- `FreezeTimeSet freeze_scenario_times`  
**Data I/O:** \*\*  
*collection of scenario times when we must enter FREEZE mode*
- `TrickHLA::Int64Time checktime`  
**Units:** –  
*For `DIS`: Checking time to pause*
- `PausePointList pause_sync_pts`  
**Units:** –  
*Synchronization points used for pausing the sim.*

## Static Protected Attributes

- static const std::wstring `type` = L"IMSim"  
**Units:** –  
*`ExecutionControl` type string.*

## Private Member Functions

- `ExecutionControl (const ExecutionControl &rhs)`  
*Copy constructor for `ExecutionControl` class.*
- `ExecutionControl & operator= (const ExecutionControl &rhs)`  
*Assignment operator for `ExecutionControl` class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrIMSim\\_\\_ExecutionControl \(\)](#)

## Additional Inherited Members

### 7.16.1 Detailed Description

Definition at line 59 of file IMSim/ExecutionControl.hh.

### 7.16.2 Constructor & Destructor Documentation

#### 7.16.2.1 ExecutionControl() [1/2]

`ExecutionControl::ExecutionControl ()`  
Default constructor for the [IMSim ExecutionControl](#) class.

**Trick Job Class:** *initialization*

Definition at line 89 of file IMSim/ExecutionControl.cpp.

#### 7.16.2.2 ~ExecutionControl()

`ExecutionControl::~ExecutionControl () [virtual]`  
Destructor for the [IMSim ExecutionControl](#) class.

**Trick Job Class:** *shutdown*

Definition at line 100 of file IMSim/ExecutionControl.cpp.

References `clear_mode_values()`, `freeze_inter_count`, `freeze_interaction`, `TrickHLA::Interaction::get_handler()`, and `TrickHLA::Interaction::set_handler()`.

#### 7.16.2.3 ExecutionControl() [2/2]

`IMSim::ExecutionControl::ExecutionControl (`  
    `const ExecutionControl & rhs ) [private]`

Copy constructor for [ExecutionControl](#) class.

This constructor is private to prevent inadvertent copies.

### 7.16.3 Member Function Documentation

#### 7.16.3.1 achieve\_all\_multiphase\_init\_sync\_pnts()

```
void ExecutionControl::achieve_all_multiphase_init_sync_pnts (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador ) throw ( RTI1516_NAMESPACE::-
    SynchronizationPointLabelNotAnnounced, RTI1516_NAMESPACE::FederateNotExecutionMember, RTI1516_-
    NAMESPACE::SaveInProgress, RTI1516_NAMESPACE::RestoreInProgress, RTI1516_NAMESPACE::NotConnected,
    RTI1516_NAMESPACE::RTIinternalError)
```

Achieve all the user defined multi-phase initialization synchronization points if they are not already achieved and are not one of the predefined [ExecutionControl](#) synchronization points.

## Parameters

<code>rti_ambassador</code>	Reference to the HLA RTI Ambassador instance.
-----------------------------	---

**Trick Job Class:** *initialization*

Definition at line 1160 of file IMSim/ExecutionControl.cpp.

References `TrickHLA::SyncPnt::exists()`, `IMSim::INITIALIZE_SYNC_POINT`, `TrickHLA::SyncPnt::is_achieved()`, `TrickHLA::SyncPnt::label`, `IMSim::SIM_CONFIG_SYNC_POINT`, and `IMSim::STARTUP_SYNC_POINT`.

**7.16.3.2 add\_freeze\_scenario\_time()**

```
void ExecutionControl::add_freeze_scenario_time (
    double t ) [virtual]
```

Adds a freeze interaction time into freeze scenario time collection.

## Parameters

<code>t</code>	Scenario time to freeze the simulation in seconds.
----------------	--

Definition at line 2529 of file IMSim/ExecutionControl.cpp.

References `TrickHLA::Federate::announce_save`, `TrickHLA::ExecutionControlBase::federate`, `freeze_scenario_times`, `TrickHLA::ExecutionControlBase::get_manager()`, and `TrickHLA::ExecutionControlBase::get_scenario_time()`.

**7.16.3.3 add\_multiphase\_init\_sync\_points()**

```
void ExecutionControl::add_multiphase_init_sync_points ( ) [virtual]
```

Add initialization synchronization points to regulate startup.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 1044 of file IMSim/ExecutionControl.cpp.

References `TrickHLA::SyncPntListBase::add_sync_pnt()`, `IMSim::INITIALIZE_SYNC_POINT`, `IMSim::SIM_CONFIG_SYNC_POINT`, and `IMSim::STARTUP_SYNC_POINT`.

Referenced by `pre_multi_phase_init_processes()`.

**7.16.3.4 add\_pause()**

```
void ExecutionControl::add_pause (
    TrickHLA::Int64Time * time,
    std::wstring const & label ) [virtual]
```

Add pause time.

## Parameters

<code>time</code>	Pause time.
<code>label</code>	Pause label (Synchronization point).

Definition at line 2474 of file IMSim/ExecutionControl.cpp.

References `TrickHLA::TimedSyncPntList::add_sync_pnt()`, and `pause_sync_pts`.

Referenced by `announce_sync_point()`.

### 7.16.3.5 announce\_sync\_point()

```
void ExecutionControl::announce_sync_point (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    std::wstring const & label,
    RTI1516_USERDATA const & user_supplied_tag ) [virtual]
```

The RTI has announced the existence of a synchronization point.

#### Parameters

<i>rti_ambassador</i>	Reference to the HLA RTI Ambassador instance.
<i>label</i>	Sync-point label.
<i>user_supplied_tag</i>	Use supplied tag.

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 1058 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::achieve\\_sync\\_pnt\(\)](#), [add\\_pause\(\)](#), [TrickHLA::SyncPntListBase::contains\(\)](#), [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [TrickHLA::Int64Time::decode\(\)](#), [IMSim::FEDRUN\\_SYNC\\_POINT](#), [IMSim::FEDSAVE\\_SYNC\\_POINT](#), [TrickHLA::Manager::get\\_execution\\_control\(\)](#), [TrickHLA::ExecutionControlBase::get\\_manager\(\)](#), [TrickHLA::Int64Time::getDoubleTime\(\)](#), [TrickHLA::ExecutionControlBase::init\\_complete\\_sp\\_exists](#), [IMSim::INIT\\_COMPLETE\\_SYNC\\_POINT](#), [TrickHLA::SyncPntListBase::mark\\_announced\(\)](#), [TrickHLA::Int64Time::setTo\(\)](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), [TrickHLA::ExecutionControlBase::should\\_print\(\)](#), and [THLA\\_NEWLINE](#).

### 7.16.3.6 check\_freeze\_exit()

```
bool ExecutionControl::check_freeze_exit ( ) [virtual]
```

Check for exit from freeze.

#### Returns

True if should exit from freeze.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2287 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::TimedSyncPntList::achieve\\_all\\_sync\\_pnts\(\)](#), [TrickHLA::Federate::announce\\_freeze](#), [IMSim::PausePointList::check\\_state\(\)](#), [checktime](#), [TrickHLA::DEBUG\\_LEVEL\\_4\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [pause\\_sync\\_pts](#), [TrickHLA::Federate::set\\_restart\(\)](#), [TrickHLA::Federate::set\\_restart\\_cfg\(\)](#), [IMSim::PausePointList::should\\_exit\(\)](#), [TrickHLA::ExecutionControlBase::should\\_print\(\)](#), [IMSim::PausePointList::should\\_reconfig\(\)](#), [IMSim::PausePointList::should\\_restart\(\)](#), [IMSim::PausePointList::should\\_run\(\)](#), [THLA\\_NEWLINE](#), and [TrickHLA::Federate::un\\_freeze\(\)](#).

### 7.16.3.7 check\_freeze\_time()

```
bool ExecutionControl::check_freeze_time ( ) [virtual]
```

Checks for a freeze interaction time from the freeze sim time collection.

#### Returns

True if freeze time found; False otherwise.

If found, clears the element, registers the FEDSAVE\_v2 sync point with the RTI if we are the master federate and returns true. Otherwise, when a freeze interaction time was not found, false is returned.

Definition at line 2566 of file IMSim/ExecutionControl.cpp.

References `TrickHLA::Federate::announce_freeze`, `TrickHLA::Federate::announce_save`, `check_scenario_freeze<time()`, `TrickHLA::ExecutionControlBase::federate`, `IMSim::FEDSAVE_SYNC_POINT`, and `TrickHLA::Federate<::register_generic_sync_point()`.

#### 7.16.3.8 `check_mode_transition_request()`

```
bool ExecutionControl::check_mode_transition_request ( ) [virtual]
```

Check to see if a new MTR is valid.

##### Returns

True if new MTR is valid.

Definition at line 1617 of file `IMSim/ExecutionControl.cpp`.

References `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::ExecutionControlBase::is_mode_transition<requested()`, `is_mtr_valid()`, `IMSim::mtr_enum_to_string()`, `pending_mtr`, and `THLA_ENDL`.

Referenced by `process_mode_transition_request()`.

#### 7.16.3.9 `check_pause()` [1/2]

```
void ExecutionControl::check_pause (
    const double check_pause_delta ) [virtual]
```

Check if we hit a pause sync point and need to go to freeze.

##### Parameters

<code>check_pause_delta</code>	Check pause job delta time in seconds.
--------------------------------	--

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2420 of file `IMSim/ExecutionControl.cpp`.

References `TrickHLA::TimedSyncPntList::check_sync_pnts()`, `checktime`, `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::freeze<the_federation`, `TrickHLA::ExecutionControlBase::get_manager()`, `TrickHLA::ExecutionControlBase::get_sim_time()`, `TrickHLA::Manager::is_late_joining_federate()`, `pause_sync_pts`, `TrickHLA::Federate::requested_time`, `TrickHLA::Int64Time::setTo()`, `TrickHLA::ExecutionControlBase::should_print()`, and `THLA_NEWLINE`.

#### 7.16.3.10 `check_pause()` [2/2]

```
virtual void IMSim::ExecutionControl::check_pause (
    const double check_pause_delta ) [virtual]
```

Check if we hit a pause sync point and need to go to freeze.

##### Parameters

<code>check_pause_delta</code>	Check pause job delta time in seconds.
--------------------------------	--

Reimplemented from [TrickHLA::ExecutionControlBase](#).

#### 7.16.3.11 `check_pause_at_init()` [1/2]

```
void ExecutionControl::check_pause_at_init (
```

```
const double check_pause_delta ) [virtual]
Checking if we started in freeze.
```

#### Parameters

<code>check_pause_delta</code>	Check pause job delta time in seconds.
--------------------------------	--

Note that we could just have one `check_pause` routine and 2 instances of it in the `S_define` file (one would be an initialization job and one would be a logging job). But early Trick 10 versions cannot distinguish between multiple instances when setting job cycle, so having this `check_pause_at_init` routine solves that problem. **Trick Job Class: initialization**

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2464 of file `IMSim/ExecutionControl.cpp`.

References `TrickHLA::ExecutionControlBase::check_pause_at_init()`, `TrickHLA::Manager::get_execution_control()`, and `TrickHLA::ExecutionControlBase::get_manager()`.

#### 7.16.3.12 `check_pause_at_init()` [2/2]

```
virtual void IMSim::ExecutionControl::check_pause_at_init (
    const double check_pause_delta ) [virtual]
```

Checking if we started in freeze.

#### Parameters

<code>check_pause_delta</code>	Check pause job delta time in seconds.
--------------------------------	--

Reimplemented from [TrickHLA::ExecutionControlBase](#).

#### 7.16.3.13 `check_scenario_freeze_time()`

```
bool ExecutionControl::check_scenario_freeze_time ( ) [virtual]
Checks for scenario freeze times.
```

#### Returns

True is time to go to freeze; False otherwise.

Definition at line 2584 of file `IMSim/ExecutionControl.cpp`.

References `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `TrickHLA::ExecutionControlBase::federate`, `freeze_scenario_times`, `TrickHLA::Federate::freeze_the_federation`, `TrickHLA::ExecutionControlBase::get_scenario_time()`, `TrickHLA::ExecutionControlBase::get_sim_time()`, `TrickHLA::Int64Time::getDoubleTime()`, `TrickHLA::Federate::granted_time`, `TrickHLA::ExecutionControlBase::should_print()`, `THLA_ENDL`, and `TrickHLA::Federate::time_management`.

Referenced by `check_freeze_time()`.

#### 7.16.3.14 `clear_mode_values()`

```
virtual void IMSim::ExecutionControl::clear_mode_values ( ) [virtual]
Clear the Mode Transition Request flag, the requested execution mode, and the current execution mode.

Reimplemented from TrickHLA::ExecutionControlBase.
```

Referenced by `~ExecutionControl()`.

### 7.16.3.15 clear\_pause()

```
void ExecutionControl::clear_pause (
    std::wstring const & label )  [virtual]
```

Clear a pause time by label.

#### Parameters

<i>label</i>	Pause label (Synchronization point).
--------------	--------------------------------------

Definition at line 2481 of file IMSim/ExecutionControl.cpp.

References IMSim::PausePointList::clear\_sync\_pnt(), and pause\_sync\_pts.

Referenced by mark\_synchronized().

### 7.16.3.16 convert\_loggable\_sync\_pts()

```
void ExecutionControl::convert_loggable_sync_pts ( )  [virtual]
```

Converts HLA sync points into something Trick can save in a checkpoint.

**Trick Job Class:** *checkpoint*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2709 of file IMSim/ExecutionControl.cpp.

References TrickHLA::TimedSyncPntList::convert\_sync\_pts(), TrickHLA::ExecutionControlBase::loggable\_sync\_pts, TrickHLA::ExecutionControlBase::logged\_sync\_pts\_count, pause\_sync\_pts, and THLA\_NEWLINE.

### 7.16.3.17 determine\_if\_late\_joining\_or\_restoring\_federate()

```
FederateJoinEnum ExecutionControl::determine_if_late_joining_or_restoring_federate ( )
```

Determine if this federate is late in joining the federation or is to restore itself. This call blocks until it has determined if the federate is late or not or when its been cleared to restore.

#### Returns

Initialization Federate State:

- 0 – normal execution (neither late joiner nor federate restore),
- 1 – late joiner,
- 2 – federate restore.

**Trick Job Class:** *initialization*

Definition at line 751 of file IMSim/ExecutionControl.cpp.

References TrickHLA::Federate::check\_for\_shutdown\_with\_termination(), TrickHLA::SyncPntListBase::contains(), TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE<=MANAGER, TrickHLA::ExecutionControlBase::does\_init\_complete\_sync\_point\_exist(), TrickHLA::ExecutionControlBase::federate, TrickHLA::FEDERATE\_JOIN\_LATE, TrickHLA::FEDERATE\_JOIN\_NOMINAL, TrickHLA::FEDERATE\_JOIN\_RESTORING, TrickHLA::ExecutionControlBase::get\_manager(), TrickHLA::Federate::has\_restore\_been\_announced(), TrickHLA::Federate::is\_execution\_member(), TrickHLA::ExecutionControlBase::is\_late\_joiner(), TrickHLA::Federate::is\_start\_to\_restore(), TrickHLA::ExecutionControlBase::late\_joiner, TrickHLA::ExecutionControlBase::late\_joiner\_determined, TrickHLA::Manager::restore\_determined, TrickHLA::Manager::restore\_federate, TrickHLA::DebugHandler::should\_print(), IMSim::SIM\_CONFIG\_SYNC\_POINT, THLA\_ENDL, and THLA\_NEWLINE.

Referenced by pre\_multi\_phase\_init\_processes().

### 7.16.3.18 enter\_freeze()

```
void ExecutionControl::enter_freeze ( )  [virtual]
```

Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2255 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::Federate::announce\\_freeze](#), [TrickHLA::DEBUG\\_LEVEL\\_4\\_TRACE](#), [TrickHLA::DEBUG\\_SOUL](#), [RCE\\_FEDERATE](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::Federate::freeze\\_the\\_federation](#), [TrickHLA::ExecutionControlBase::get\\_sim\\_time\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::ExecutionControlBase::should\\_print\(\)](#), [THLA\\_NEWLINE](#), [trigger\\_freeze\\_interaction\(\)](#), and [TrickHLA::Federate::un\\_freeze\(\)](#).

#### 7.16.3.19 exit\_freeze()

```
void ExecutionControl::exit_freeze ( ) [virtual]
```

Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2351 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::TimedSyncPntList::achieve\\_all\\_sync\\_pnts\(\)](#), [TrickHLA::Federate::announce\\_freeze](#), [TrickHLA::TimedSyncPntList::check\\_sync\\_pnts\(\)](#), [checktime](#), [TrickHLA::ExecutionControlBase::federate](#), [IMSim::FEDRUN\\_SYNC\\_POINT](#), [TrickHLA::Federate::freeze\\_the\\_federation](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA::Federate::is\\_execution\\_member\(\)](#), [pause\\_sync\\_pts](#), [TrickHLA::Federate::register\\_generic\\_sync\\_point\(\)](#), [THLA\\_ENDL](#), and [THLA\\_NEWLINE](#).

#### 7.16.3.20 freeze\_mode\_announce()

```
void ExecutionControl::freeze_mode_announce ( ) [virtual]
```

Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2161 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), and [TrickHLA::SyncPntListBase::register\\_sync\\_pnt\(\)](#).

Referenced by [process\\_execution\\_control\\_updates\(\)](#), and [process\\_mode\\_transition\\_request\(\)](#).

#### 7.16.3.21 freeze\_mode\_transition()

```
bool ExecutionControl::freeze_mode_transition ( ) [virtual]
```

The freeze mode transition routine.

Returns

Currently always returns False.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2172 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::SyncPnt::achieve\\_sync\\_point\(\)](#), [TrickHLA::ExecutionControlBase::current\\_execution\\_control\\_mode](#), [TrickHLA::EXECUTION\\_CONTROL\\_FREEZE](#), [IMSim::EXECUTION\\_MODE\\_FREEZE](#), [TrickHLA::ExecutionControlBase::federate](#), [get\\_execution\\_configuration\(\)](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA::SyncPntListBase::get\\_sync\\_pnt\(\)](#), [IMSim::ExecutionConfiguration::set\\_current\\_execution\\_mode\(\)](#), [THLA\\_ENDL](#), [TrickHLA::SyncPnt::wait\\_for\\_announce\(\)](#), and [TrickHLA::SyncPnt::wait\\_for\\_synchronization\(\)](#).

#### 7.16.3.22 get\_execution\_configuration()

```
ExecutionConfiguration * ExecutionControl::get_execution_configuration ( ) [protected], [virtual]
```

Return the relevant [IMSim::ExecutionConfiguration](#) object.

**Returns**

Pointer to the relevant [IMSim::ExecutionConfiguration](#) object.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2487 of file [IMSim/ExecutionControl.cpp](#).

References [THLA\\_ENDIAN](#).

Referenced by [freeze\\_mode\\_transition\(\)](#), [is\\_mtr\\_valid\(\)](#), [process\\_execution\\_control\\_updates\(\)](#), [process\\_mode\\_transition\\_request\(\)](#), [run\\_mode\\_transition\(\)](#), [set\\_next\\_execution\\_control\\_mode\(\)](#), and [shutdown\\_mode\\_announce\(\)](#).

**7.16.3.23 get\_type()**

```
virtual const std::wstring& IMSim::ExecutionControl::get_type ( ) [inline], [virtual]
```

Get the [ExecutionControl](#) type identification string.

**Returns**

A constant reference to the type identification string.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 81 of file [IMSim/ExecutionControl.hh](#).

References [type](#).

**7.16.3.24 initialize()**

```
void ExecutionControl::initialize (
    TrickHLA::Federate & federate ) [virtual]
```

Execution Control initialization routine.

**Parameters**

<a href="#">federate</a>	The associated <a href="#">TrickHLA::Federate</a> .
--------------------------	---

This routine will set a lot of the data in the [TrickHLA::Federate](#) that is required for this execution control scheme. This should greatly simplify input files and reduce input file setting errors.

**Trick Job Class:** *initialization*

Definition at line 128 of file [IMSim/ExecutionControl.cpp](#).

References [TrickHLA::SyncPntListBase::add\\_sync\\_pnt\(\)](#), [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::ExecutionControlBase::execution\\_configuration](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::ExecutionControlBase::get\\_manager\(\)](#), [TrickHLA::Object::initialize\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\\_preset\(\)](#), [TrickHLA::ExecutionControlBase::least\\_common\\_time\\_step](#), [pre\\_multi\\_phase\\_init\\_processes\(\)](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), [THLA\\_ENDIAN](#), [THLA\\_NEWLINE](#), [TrickHLA::Federate::time\\_constrained](#), [TrickHLA::Federate::time\\_management](#), [TrickHLA::Federate::time\\_regulating](#), and [TrickHLA::ExecutionControlBase::use\\_preset\\_master](#).

**7.16.3.25 is\_mtr\_valid()**

```
bool ExecutionControl::is_mtr_valid (
    MTREnum mtr_value ) [virtual]
```

Determine if the Mode Transition Request (MTR) is valid given the current mode.

**Returns**

True if valid, false otherwise.

**Parameters**

<i>mtr_value</i>	Mode transition request.
------------------	--------------------------

Definition at line 1449 of file IMSim/ExecutionControl.cpp.

References IMSim::ExecutionConfiguration::current\_execution\_mode, IMSim::EXECUTION\_MODE\_FREEZE, IMSim::EXECUTION\_MODE\_INITIALIZING, IMSim::EXECUTION\_MODE\_RUNNING, IMSim::EXECUTION\_MODE\_SHUTDOWN, get\_execution\_configuration(), IMSim::MTR\_GOTO\_FREEZE, IMSim::MTR\_GOTO\_RUN, and IMSim::MTR\_GOTO\_SHUTDOWN.

Referenced by check\_mode\_transition\_request(), and set\_pending\_mtr().

**7.16.3.26 is\_save\_and\_restore\_supported()**

virtual bool IMSim::ExecutionControl::is\_save\_and\_restore\_supported ( ) [inline], [virtual]

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 300 of file IMSim/ExecutionControl.hh.

**7.16.3.27 is\_save\_initiated()**

bool ExecutionControl::is\_save\_initiated ( ) [virtual]

Checks if Save has been initiated by this [ExecutionControl](#) method.

**Returns**

True if Save is initiated and synchronized with the federation, False if Save not supported.

This routine will block on the FEDSAVE\_SYNC\_POINT synchronization point until it is achieved and Save is initiated.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2657 of file IMSim/ExecutionControl.cpp.

References TrickHLA::TimedSyncPntList::achieve\_all\_sync\_pnts(), TrickHLA::Federate::announce\_save, check\_time, TrickHLA::ExecutionControlBase::federate, IMSim::FEDSAVE\_SYNC\_POINT, TrickHLA::Federate::get\_RTI\_ambassador(), TrickHLA::Federate::initiate\_save\_flag, TrickHLA::Federate::is\_execution\_member(), pause\_sync\_pts, TrickHLA::Federate::register\_generic\_sync\_point(), TrickHLA::Federate::save\_completed, and THLA\_ENDL.

**7.16.3.28 join\_federation\_process()**

void ExecutionControl::join\_federation\_process ( ) [virtual]

Join federation execution process.

This routine implements the [IMSim](#) Join Federation Process described in section 7.2 and figure 7-3.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 201 of file IMSim/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::join\_federation\_process().

**7.16.3.29 mark\_synchronized()**

```
bool ExecutionControl::mark_synchronized (
    std::wstring const & label ) [virtual]
```

Mark the given synchronization point as synchronized in the federation.

**Returns**

True if synchronization point label is valid.

**Parameters**

<i>label</i>	The synchronization point label.
--------------	----------------------------------

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 1337 of file IMSim/ExecutionControl.cpp.

References TrickHLA::Federate::announce\_save, clear\_pause(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::D←EBUG\_SOURCE\_FEDERATE, TrickHLA::ExecutionControlBase::federate, IMSim::FEDRUN\_SYNC\_POINT, IMSim←::FEDSAVE\_SYNC\_POINT, TrickHLA::ExecutionControlBase::get\_sim\_time(), TrickHLA::Federate::initiate\_save\_flag, TrickHLA::ExecutionControlBase::should\_print(), THLA\_NEWLINE, and TrickHLA::Federate::un\_freeze().

**7.16.3.30 operator=()**

```
ExecutionControl& IMSim::ExecutionControl::operator= (
    const ExecutionControl & rhs ) [private]
```

Assignment operator for [ExecutionControl](#) class.

This assignment operator is private to prevent inadvertent copies.

**7.16.3.31 perform\_save()**

```
bool ExecutionControl::perform_save ( ) [inline], [virtual]
```

Federates that did not announce the save, perform a save.

**Returns**

True if Save can proceed, False if not.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 307 of file IMSim/ExecutionControl.hh.

**7.16.3.32 post\_multi\_phase\_init\_process()**

```
void ExecutionControl::post_multi_phase_init_process ( ) [virtual]
```

Process run after the multi-phase initialization ends.

This routine implements the [IMSim](#) post multi-phase initialization process.

**Trick Job Class:** *initialization*

Definition at line 846 of file IMSim/ExecutionControl.cpp.

References TrickHLA::Federate::achieve\_and\_wait\_for\_synchronization(), TrickHLA::ExecutionControlBase::federate, TrickHLA::ExecutionControlBase::get\_manager(), IMSim::INIT\_COMPLETE\_SYNC\_POINT, TrickHLA::Execution←ControlBase::is\_master(), TrickHLA::Federate::load\_and\_print\_running\_federate\_names(), TrickHLA::Federate←::register\_generic\_sync\_point(), TrickHLA::Federate::setup\_time\_management(), IMSim::STARTUP\_SYNC\_POINT, and TrickHLA::Federate::time\_advance\_request\_to\_GALT().

**7.16.3.33 pre\_multi\_phase\_init\_processes()**

```
void ExecutionControl::pre_multi_phase_init_processes ( ) [virtual]
```

Process run before the multi-phase initialization begins.

This routine implements the [IMSim](#) pre multi-phase initialization process.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 215 of file IMSim/ExecutionControl.cpp.

References TrickHLA::Federate::achieve\_and\_wait\_for\_synchronization(), add\_multiphase\_init\_sync\_points(), TrickHLA::SyncPntListBase::add\_sync\_pnt(), TrickHLA::Federate::check\_HLA\_save\_directory(), TrickHLA::Federate::copy\_running\_feds\_into\_known\_feds(), TrickHLA::Federate::create\_and\_join\_federation(), TrickHLA::Federate::create\_RTIAmbassador\_and\_connect(), TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::Federate::destroy\_orphaned\_federation(), determine\_if\_late\_joining\_or\_restoring\_federate(), TrickHLA::Federate::enable\_async\_delivery(), TrickHLA::ExecutionControlBase::execution\_configuration, TrickHLA::ExecutionControlBase::federate, TrickHLA::FEDERATE\_JOIN\_RESTORING, TrickHLA::Federate::get\_joined\_federate\_handles(), TrickHLA::ExecutionControlBase::get\_manager(), TrickHLA::Object::get\_name(), TrickHLA::Federate::get\_RTIAmbassador(), TrickHLA::Federate::has\_restore\_request\_failed(), TrickHLA::Federate::inform\_RTI\_of\_restore\_completion(), TrickHLA::Federate::initialize\_MOM\_handles(), IMSim::INITIALIZE\_SYNC\_POINT, TrickHLA::Federate::initiate\_restore\_announce(), TrickHLA::Federate::is\_federation\_created\_by\_federate(), TrickHLA::ExecutionControlBase::is\_late\_joiner(), TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::ExecutionControlBase::is\_master\_preset(), TrickHLA::Federate::is\_time\_management\_enabled(), TrickHLA::Object::mark\_required(), TrickHLA::Manager::publish\_and\_subscribe(), TrickHLA::Manager::pull\_ownership\_upon\_rejoin(), TrickHLA::Federate::read\_running\_feds\_file(), TrickHLA::ExecutionControlBase::receive\_execution\_configuration(), TrickHLA::SyncPntListBase::register\_all\_sync\_pnts(), TrickHLA::Federate::register\_generic\_sync\_point(), TrickHLA::Manager::register\_objects\_with\_RTI(), TrickHLA::SyncPntListBase::register\_sync\_pnt(), TrickHLA::Manager::request\_data\_update(), TrickHLA::Object::reserve\_object\_name\_with\_RTI(), TrickHLA::ExecutionConfigurationBase::reset\_ownership\_states(), TrickHLA::ExecutionConfigurationBase::reset\_preferred\_order(), TrickHLA::Federate::restart\_checkpoint(), TrickHLA::Manager::restart\_initialization(), TrickHLA::Federate::restore\_checkpoint(), TrickHLA::Federate::restore\_federate\_handles\_from\_MOM(), TrickHLA::ExecutionControlBase::send\_execution\_configuration(), TrickHLA::Federate::set\_federate\_has\_begun\_execution(), TrickHLA::ExecutionConfigurationBase::set\_master(), TrickHLA::ExecutionControlBase::set\_master(), TrickHLA::Federate::set\_restore\_is\_imminent(), TrickHLA::Manager::setup\_all\_ref\_attributes(), TrickHLA::Manager::setup\_all\_RTI\_handles(), TrickHLA::Manager::setup\_preferred\_order\_with\_RTI(), TrickHLA::DebugHandler::should\_print(), TrickHLA::ExecutionControlBase::should\_print(), IMSim::SIM\_CONFIG\_SYNC\_POINT, IMSim::STARTUP\_FREEZE\_SYNC\_POINT, IMSim::STARTUP\_SYNC\_POINT, TrickHLA::Object::subscribe\_to\_object\_attributes(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::SyncPntListBase::wait\_for\_all\_announcements(), TrickHLA::Federate::wait\_for\_federation\_restore\_begun(), TrickHLA::Federate::wait\_for\_federation\_restore\_failed\_callback\_to\_complete(), TrickHLA::Federate::wait\_for\_federation\_restore\_to\_complete(), TrickHLA::Federate::wait\_for\_required\_federates\_to\_join(), TrickHLA::Federate::wait\_for\_restore\_request\_callback(), TrickHLA::ExecutionControlBase::wait\_for\_sync\_point\_announce(), TrickHLA::Manager::wait\_on\_discovery\_of\_objects(), TrickHLA::Object::wait\_on\_object\_name\_reservation(), TrickHLA::ExecutionConfigurationBase::wait\_on\_registration(), TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects(), TrickHLA::Manager::wait\_on\_reservation\_of\_object\_names(), TrickHLA::ExecutionConfigurationBase::wait\_on\_update(), and TrickHLA::Federate::wait\_until\_federation\_is\_ready\_to\_restore().

Referenced by initialize().

#### 7.16.3.34 process\_execution\_control\_updates()

```
bool ExecutionControl::process_execution_control_updates ( ) [virtual]
```

Process changes from any received Execution Control Objects (ExCOs).

##### Returns

True if mode change occurred.

##### Assumptions and Limitations:

- Called from the ExCO unpack routine.

##### Trick Job Class: *scheduled*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1760 of file IMSim/ExecutionControl.cpp.

References `TrickHLA::ExecutionConfigurationBase::clear_update_pending()`, `TrickHLA::ScenarioTimeline::compute<=simulation_time()`, `TrickHLA::ExecutionControlBase::cte_timeline`, `TrickHLA::ExecutionControlBase::current<=execution_control_mode`, `IMSim::ExecutionConfiguration::current_execution_mode`, `TrickHLA::DEBUG_LEVEL<=4_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::does_cte_timeline_exist()`, `TrickHLA::execution_control_enum_to_string()`, `TrickHLA::EXECUTION_CONTROL_FREEZE`, `TrickHLA::EXECUTION_CONTROL_INITIALIZING`, `TrickHLA::EXECUTION_CONTROL_RUNNING`, `TrickHLA::EXECUTION_CONTROL_SHUTDOWN`, `TrickHLA::EXECUTION_CONTROL_UNINITIALIZED`, `IMSim::execution_mode_enum_to_string()`, `IMSim::EXECUTION_MODE_FREEZE`, `IMSim::execution_mode_int16_to_enum()`, `IMSim::EXECUTION_MODE_RUNNING`, `IMSim::EXECUTION_MODE_SHUTDOWN`, `TrickHLA::ExecutionControlBase::federate`, `freeze_mode<=announce()`, `TrickHLA::Timeline::get_epoch()`, `get_execution_configuration()`, `TrickHLA::ScenarioTimeline::get_sim<=offset()`, `TrickHLA::SimTimeline::get_time()`, `TrickHLA::CTETimelineBase::get_time()`, `TrickHLA::ScenarioTimeline::get_time()`, `TrickHLA::ExecutionControlBase::is_master()`, `IMSim::ExecutionConfiguration::next_execution_mode`, `IMSim::ExecutionConfiguration::next_mode_cte_time`, `TrickHLA::ExecutionControlBase::next_mode_cte_time`, `IMSim::ExecutionConfiguration::next_mode_scenario_time`, `TrickHLA::ExecutionControlBase::requested_execution<=control_mode`, `run_mode_transition()`, `TrickHLA::ExecutionControlBase::scenario_freeze_time`, `IMSim::ExecutionConfiguration::scenario_time_epoch`, `TrickHLA::ExecutionControlBase::scenario_timeline`, `TrickHLA::Federate::should_print()`, `TrickHLA::ExecutionControlBase::sim_timeline`, `TrickHLA::ExecutionControlBase::simulation_freeze<_time`, `THLA_ENDL`, and `TrickHLA::ExecutionConfigurationBase::update_pending()`.

Referenced by `run_mode_transition()`.

#### 7.16.3.35 process\_mode\_interaction() [1/2]

`virtual bool IMSim::ExecutionControl::process_mode_interaction () [virtual]`  
Process a new mode interaction.

##### Returns

True if new mode interaction is successfully processed.

Implements [TrickHLA::ExecutionControlBase](#).

#### 7.16.3.36 process\_mode\_interaction() [2/2]

`virtual bool IMSim::ExecutionControl::process_mode_interaction () [inline], [virtual]`  
Process a new mode interaction.

##### Returns

True if new mode interaction is successfully processed.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 191 of file IMSim/ExecutionControl.hh.

References `process_mode_transition_request()`.

#### 7.16.3.37 process\_mode\_transition\_request()

`bool ExecutionControl::process_mode_transition_request () [virtual]`  
Process a new Mode Transition Request (MTR).

**Returns**

True if new MTR is successfully processed.

Definition at line 1649 of file IMSim/ExecutionControl.cpp.

References `check_mode_transition_request()`, `TrickHLA::ExecutionControlBase::clear_mode_transition_requested()`, `TrickHLA::ExecutionControlBase::cte_timeline`, `TrickHLA::ExecutionControlBase::current_execution_control_mode`, `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::does_cte_timeline_exist()`, `TrickHLA::EXECUTION_CONTROL_FREEZE`, `TrickHLA::EXECUTION_CONTROL_RUNNING`, `TrickHLA::ExecutionControlBase::federate`, `freeze_mode_announce()`, `TrickHLA::Timeline::get_epoch()`, `get_execution_configuration()`, `TrickHLA::Federate::get_granted_time()`, `TrickHLA::Federate::get_requested_time()`, `TrickHLA::ScenarioTimeline::get_sim_offset()`, `TrickHLA::SimTimeline::get_time()`, `TrickHLA::CTETimelineBase::get_time()`, `TrickHLA::ScenarioTimeline::get_time()`, `IMSim::MTR_GOTO_FREEZE`, `IMSim::MTR_GOTO_RUN`, `IMSim::MTR_GOTO_SHUTDOWN`, `IMSim::ExecutionConfiguration::next_mode_cte_time`, `IMSim::ExecutionConfiguration::next_mode_scenario_time`, `pending_mtr`, `TrickHLA::ExecutionControlBase::scenario_freeze_time`, `IMSim::ExecutionConfiguration::scenario_time_epoch`, `TrickHLA::ExecutionControlBase::scenario_timeline`, `TrickHLA::Object::send_init_data()`, `set_mode_request_from_mtr()`, `TrickHLA::Federate::should_print()`, `shutdown_mode_announce()`, `TrickHLA::ExecutionControlBase::sim_timeline`, `TrickHLA::ExecutionControlBase::simulation_freeze_time`, and `TrickHLA::ExecutionControlBase::time_padding`.

Referenced by `process_mode_interaction()`.

**7.16.3.38 publish()**

```
void ExecutionControl::publish ( ) [virtual]
```

Publish the `ExecutionControl` objects and interactions.

Implements `TrickHLA::ExecutionControlBase`.

Definition at line 1241 of file IMSim/ExecutionControl.cpp.

References `freeze_inter_count`, `freeze_interaction`, and `TrickHLA::Interaction::publish_interaction()`.

**7.16.3.39 receive\_interaction()**

```
void ExecutionControl::receive_interaction (
    RTI1516_NAMESPACE::InteractionClassHandle const & theInteraction,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_NAMESPACE::UserData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    bool received_as_TSO ) [virtual]
```

Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.

**Parameters**

<code>theInteraction</code>	Interaction handle.
<code>theParameterValues</code>	Parameter values.
<code>theUserSuppliedTag</code>	Users tag.
<code>theTime</code>	HLA time for the interaction.
<code>received_as_TSO</code>	True if interaction was received by RTI as TSO.

**Trick Job Class: *scheduled***

Implements `TrickHLA::ExecutionControlBase`.

Definition at line 1377 of file IMSim/ExecutionControl.cpp.

References `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::Interaction::extract_data()`, `freeze_inter_count`, `freeze_interaction`, `TrickHLA::Int64Time::getDoubleTime()`, `TrickHLA::Interaction::process_interaction()`, `TrickHLA::Int64Time::setTo()`, `TrickHLA::Debug`

Handler::should\_print(), THLA\_NEWLINE, and TrickHLA::TRICKHLA\_MANAGER\_BUILTIN\_FREEZE\_INTERACTION.

#### 7.16.3.40 reinstate\_logged\_sync\_pts()

void ExecutionControl::reinstate\_logged\_sync\_pts ( ) [virtual]

Converts checkpointed sync points into HLA sync points.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2728 of file IMSim/ExecutionControl.cpp.

References checktime, TrickHLA::ExecutionControlBase::loggable\_sync\_pts, TrickHLA::ExecutionControlBase::logged\_sync\_pts\_count, pause\_sync\_pts, TrickHLA::SyncPntListBase::reset(), and TrickHLA::Int64Time::setTo().

#### 7.16.3.41 run\_mode\_transition()

bool ExecutionControl::run\_mode\_transition ( ) [virtual]

The run mode transition routine.

##### Returns

Currently always returns True.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2073 of file IMSim/ExecutionControl.cpp.

References TrickHLA::SyncPnt::achieve\_sync\_point(), TrickHLA::Federate::check\_for\_shutdown\_with\_termination(), TrickHLA::ExecutionControlBase::current\_execution\_control\_mode, TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ExecutionControlBase::does\_cte\_timeline\_exist(), TrickHLA::EXECUTION\_CONTROL\_RUNNING, IMSim::EXECUTION\_MODE::RUNNING, TrickHLA::ExecutionControlBase::federate, TrickHLA::ExecutionControlBase::get\_cte\_time(), get\_execution\_configuration(), IMSim::ExecutionConfiguration::get\_next\_mode\_cte\_time(), TrickHLA::Federate::get\_RTI\_ambassador(), TrickHLA::SyncPntListBase::get\_sync\_pnt(), TrickHLA::ExecutionControlBase::is\_master(), process\_execution\_control\_updates(), TrickHLA::SyncPntListBase::register\_sync\_pnt(), TrickHLA::Object::send\_init\_data(), IMSim::ExecutionConfiguration::set\_current\_execution\_mode(), TrickHLA::DebugHandler::should\_print(), THLA\_EN::DL, THLA\_NEWLINE, TrickHLA::SyncPnt::wait\_for\_announce(), TrickHLA::SyncPnt::wait\_for\_synchronization(), and IMSim::ExecutionConfiguration::wait\_on\_update().

Referenced by process\_execution\_control\_updates().

#### 7.16.3.42 send\_mode\_transition\_interaction()

virtual void IMSim::ExecutionControl::send\_mode\_transition\_interaction (   
     [TrickHLA::ModeTransitionEnum](#) requested\_mode ) [virtual]

Send a mode transition request to the Master federate.

##### Parameters

<i>requested_mode</i>	Requested mode.
-----------------------	-----------------

Implements [TrickHLA::ExecutionControlBase](#).

#### 7.16.3.43 set\_mode\_request\_from\_mtr()

void ExecutionControl::set\_mode\_request\_from\_mtr (   
     [MTREnum](#) mtr\_value ) [virtual]

Translate MTR into a pending execution mode transition.

## Parameters

<i>mtr_value</i>	MTR value for next execution mode.
------------------	------------------------------------

Definition at line 1471 of file IMSim/ExecutionControl.cpp.

References TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, IMSim::MTR\_GOTO\_FREEZE, IMSim::MTR\_GOTO\_RUN, IMSim::MTR\_GOTO\_SHUTDOWN, IMSim::MTR\_INITIALIZING, IMSim::MTR\_UNINITIALIZED, pending\_mtr, and set\_next\_execution\_control\_mode().

Referenced by process\_mode\_transition\_request().

#### 7.16.3.44 set\_next\_execution\_control\_mode()

```
void ExecutionControl::set_next_execution_control_mode (
    TrickHLA::ExecutionControlEnum exec_control ) [virtual]
```

Sets the next [ExecutionControl](#) run mode.

## Parameters

<i>exec_control</i>	Next <a href="#">ExecutionControl</a> run mode.
---------------------	---

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1508 of file IMSim/ExecutionControl.cpp.

References TrickHLA::ScenarioTimeline::compute\_simulation\_time(), TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_INITIALIZING, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, IMSim::EXECUTION\_MODE\_FREEZE, IMSim::EXECUTION\_MODE\_INITIALIZING, IMSim::EXECUTION\_MODE\_RUNNING, IMSim::EXECUTION\_MODE\_SHUTDOWN, IMSim::EXECUTION\_MODE\_UNINITIALIZED, TrickHLA::ExecutionControlBase::get\_cte\_time(), get\_execution\_configuration(), TrickHLA::ExecutionControlBase::get\_manager(), IMSim::ExecutionConfiguration::get\_next\_mode\_cte\_time(), TrickHLA::ExecutionControlBase::get\_scenario\_time(), TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::ExecutionControlBase::next\_mode\_scenario\_time, TrickHLA::ExecutionControlBase::requested\_execution\_control\_mode, TrickHLA::ExecutionControlBase::scenario\_freeze\_time, TrickHLA::ExecutionControlBase::scenario\_timeline, IMSim::ExecutionConfiguration::set\_next\_execution\_mode(), IMSim::ExecutionConfiguration::set\_next\_mode\_cte\_time(), IMSim::ExecutionConfiguration::set\_next\_mode\_scenario\_time(), IMSim::ExecutionConfiguration::set\_scenario\_time\_epoch(), TrickHLA::ExecutionControlBase::should\_print(), TrickHLA::ExecutionControlBase::simulation\_freeze\_time, THLA\_ENDL, and TrickHLA::ExecutionControlBase::time\_padding.

Referenced by set\_mode\_request\_from\_mtr(), and shutdown\_mode\_announce().

#### 7.16.3.45 set\_pending\_mtr()

```
bool ExecutionControl::set_pending_mtr (
    MTREnum mtr_value ) [virtual]
```

Definition at line 1440 of file IMSim/ExecutionControl.cpp.

References is\_mtr\_valid(), and pending\_mtr.

#### 7.16.3.46 set\_time\_padding()

```
virtual void IMSim::ExecutionControl::set_time_padding (
    double t ) [virtual]
```

Set the time-padding used to offset the go to run time.

**Parameters**

<i>t</i>	Time in seconds to pad for time based mode transitions.
----------	---

Reimplemented from [TrickHLA::ExecutionControlBase](#).

#### 7.16.3.47 `setup_interaction_ref_attributes()`

`void ExecutionControl::setup_interaction_ref_attributes ( ) [virtual]`

Setup the [ExecutionControl](#) interaction Trick ref ATTRIBUTES.

This routine is used to perform and inline build of the Trick ref ATTRIBUTES for the [IMSim](#) freeze interaction. This is used by federates, other than the Master, to request mode transitions.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 902 of file IMSim/ExecutionControl.cpp.

References `attrTrickHLA_FreezelInteractionHandler`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_LEVEL_9_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ENCODING_LOGICAL_TIME`, `freeze_inter_count`, `freeze_interaction`, `IMSim::FreezelInteractionHandler::get_address_of_interaction_time()`, `TrickHLA::Parameter::get_FOM_name()`, `TrickHLA::Interaction::get_FOM_name()`, `TrickHLA::ExecutionControlBase::get_manager()`, `TrickHLA::Interaction::get_parameter_count()`, `TrickHLA::Parameter::initialize()`, `TrickHLA::Interaction::initialize()`, `TrickHLA::Parameter::set_debug_level()`, `TrickHLA::Parameter::set_encoding()`, `TrickHLA::Parameter::set_FOM_name()`, `TrickHLA::Interaction::set_FOM_name()`, `TrickHLA::Interaction::set_handler()`, `TrickHLA::Interaction::set_parameter_count()`, `TrickHLA::Interaction::set_parameters()`, `TrickHLA::Interaction::set_publish()`, `TrickHLA::Interaction::set_subscribe()`, `TrickHLA::DebugHandler::should_print()`, and `THLA_NEWLINE`.

#### 7.16.3.48 `setup_interaction_RTI_handles()`

`void ExecutionControl::setup_interaction_RTI_handles ( ) [virtual]`

Setup the [ExecutionControl](#) interaction HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1036 of file IMSim/ExecutionControl.cpp.

#### 7.16.3.49 `setup_object_ref_attributes()`

`void ExecutionControl::setup_object_ref_attributes ( ) [virtual]`

Setup the [ExecutionControl](#) object Trick ref ATTRIBUTES.

This routine is used to perform and inline build of the Trick ref ATTRIBUTES for the [IMSim](#) ExecutionControl.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 889 of file IMSim/ExecutionControl.cpp.

#### 7.16.3.50 `setup_object_RTI_handles()`

`void ExecutionControl::setup_object_RTI_handles ( ) [virtual]`

Setup the [ExecutionControl](#) objects HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1028 of file IMSim/ExecutionControl.cpp.

#### 7.16.3.51 shutdown()

void ExecutionControl::shutdown ( ) [virtual]  
 Execution control specific shutdown process.

**Trick Job Class:** *shutdown*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 878 of file IMSim/ExecutionControl.cpp.

#### 7.16.3.52 shutdown\_mode\_announce()

void ExecutionControl::shutdown\_mode\_announce ( ) [virtual]  
 Announce to the federation execution that a shutdown is occurring.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2206 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::clear\\_mode\\_transition\\_requested\(\)](#), [TrickHLA::ExecutionControlBase::current\\_execution\\_control\\_mode](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::EXECUTION\\_CONTROL\\_UNINITIALIZED](#), [get\\_execution\\_configuration\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::Object::send\\_init\\_data\(\)](#), and [set\\_next\\_execution\\_control\\_mode\(\)](#).  
 Referenced by [process\\_mode\\_transition\\_request\(\)](#).

#### 7.16.3.53 shutdown\_mode\_transition()

void ExecutionControl::shutdown\_mode\_transition ( ) [virtual]  
 The shutdown mode transition routine.

**Trick Job Class:** *shutdown*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2235 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::current\\_execution\\_control\\_mode](#), [TrickHLA::EXECUTION\\_CONTROL\\_UNINITIALIZED](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), and [TrickHLA::SyncPntListBase::register\\_sync\\_pnt\(\)](#).

#### 7.16.3.54 start\_federation\_save\_at\_scenario\_time()

```
void ExecutionControl::start_federation_save_at_scenario_time (
    double freeze_scenario_time,
    const char * file_name ) [virtual]
```

Start the Federation save at the specified scenario time.

##### Parameters

<i>freeze_scenario_time</i>	Scenario time to freeze.
<i>file_name</i>	Checkpoint file name.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2502 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::ExecutionControlBase::federate](#), [freeze\\_interaction](#), [TrickHLA::Interaction::get\\_handler\(\)](#), [TrickHLA::ExecutionControlBase::get\\_manager\(\)](#), [TrickHLA::Manager::initiate\\_federation\\_save\(\)](#), [TrickHLA::Manager::initiate\\_federation\\_save\(\)](#).

A::Federate::set\_announce\_save(), TrickHLA::DebugHandler::should\_print(), THLA\_NEWLINE, and trigger\_freeze\_interaction().

#### 7.16.3.55 subscribe()

void ExecutionControl::subscribe ( ) [virtual]  
 Subscribe to the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1285 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::execution\\_configuration](#), [freeze\\_inter\\_count](#), [freeze\\_interaction](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::Interaction::subscribe\\_to\\_interaction\(\)](#), and [TrickHLA::Object::subscribe\\_to\\_object\\_attributes\(\)](#).

#### 7.16.3.56 trigger\_freeze\_interaction()

void ExecutionControl::trigger\_freeze\_interaction ( double & *freeze\_scenario\_time* ) [virtual]

Trigger a FREEZE interaction from the [FreezeInteractionHandler](#) and updated the supplied time with the time computed by the [FreezeInteractionHandler](#).

##### Parameters

<i>freeze_scenario_time</i>	Scenario freeze time.
-----------------------------	-----------------------

Definition at line 2545 of file IMSim/ExecutionControl.cpp.

References [freeze\\_interaction](#), [TrickHLA::Interaction::get\\_handler\(\)](#), [TrickHLA::ExecutionControlBase::is\\_lateJoiner\(\)](#), and [IMSim::FreezeInteractionHandler::send\\_scenario\\_freeze\\_interaction\(\)](#).

Referenced by [enter\\_freeze\(\)](#), and [start\\_federation\\_save\\_at\\_scenario\\_time\(\)](#).

#### 7.16.3.57 un\_freeze()

void ExecutionControl::un\_freeze ( ) [inline], [virtual]

Routine to handle [ExecutionControl](#) specific action needed to un-freeze.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 224 of file IMSim/ExecutionControl.hh.

#### 7.16.3.58 unpublish()

void ExecutionControl::unpublish ( ) [virtual]

Unpublish the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1251 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::execution\\_configuration](#), [freeze\\_inter\\_count](#), [freeze\\_interaction](#), [TrickHLA::Interaction::get\\_class\\_handle\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::Object::unpublish\\_all\\_object\\_attributes\(\)](#), and [TrickHLA::Interaction::unpublish\\_interaction\(\)](#).

#### 7.16.3.59 unsubscribe()

void ExecutionControl::unsubscribe ( ) [virtual]

Unsubscribe the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1301 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::execution\\_configuration](#), [freeze\\_inter\\_count](#), [freeze\\_interaction](#), [TrickHLA::Interaction::get\\_class\\_handle\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::Object::unsubscribe\\_all\\_object\\_attributes\(\)](#), and [TrickHLA::Interaction::unsubscribe\\_from\\_interaction\(\)](#).

#### 7.16.3.60 wait\_for\_all\_multiphase\_init\_sync\_pnts()

```
void ExecutionControl::wait_for_all_multiphase_init_sync_pnts ( )
```

Wait for all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not one of the predefined [ExecutionControl](#) synchronization points.

**Trick Job Class:** *initialization*

Definition at line 1190 of file IMSim/ExecutionControl.cpp.

References [TrickHLA::Federate::check\\_for\\_shutdown\\_with\\_termination\(\)](#), [TrickHLA::SyncPnt::exists\(\)](#), [TrickHLA::ExecutionControlBase::federate](#), [IMSim::INITIALIZE\\_SYNC\\_POINT](#), [TrickHLA::SyncPnt::is\\_achieved\(\)](#), [TrickHLA::Federate::is\\_execution\\_member\(\)](#), [TrickHLA::SyncPnt::label](#), [IMSim::SIM\\_CONFIG\\_SYNC\\_POINT](#), [IMSim::STARTUP\\_SYNC\\_POINT](#), [TrickHLA::SyncPntListBase::sync\\_point\\_list](#), and [THLA\\_ENDL](#).

### 7.16.4 Friends And Related Function Documentation

#### 7.16.4.1 init\_attrIMSim\_\_ExecutionControl

```
void init_attrIMSim__ExecutionControl ( ) [friend]
```

#### 7.16.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 66 of file IMSim/ExecutionControl.hh.

### 7.16.5 Field Documentation

#### 7.16.5.1 checktime

```
TrickHLA::Int64Time IMSim::ExecutionControl::checktime [protected]
```

**Units:** –

For [DIS](#): Checking time to pause

Definition at line 324 of file IMSim/ExecutionControl.hh.

Referenced by [check\\_freeze\\_exit\(\)](#), [check\\_pause\(\)](#), [exit\\_freeze\(\)](#), [is\\_save\\_initiated\(\)](#), and [reinstate\\_logged\\_sync\\_pts\(\)](#).

#### 7.16.5.2 freeze\_inter\_count

```
int IMSim::ExecutionControl::freeze_inter_count [protected]
```

**Data I/O:** \*\*

Number of [TrickHLA](#) Freeze Interactions.

Definition at line 318 of file IMSim/ExecutionControl.hh.

Referenced by [publish\(\)](#), [receive\\_interaction\(\)](#), [setup\\_interaction\\_ref\\_attributes\(\)](#), [subscribe\(\)](#), [unpublish\(\)](#), [unsubscribe\(\)](#), and [~ExecutionControl\(\)](#).

### 7.16.5.3 `freeze_interaction`

`TrickHLA::Interaction* IMSim::ExecutionControl::freeze_interaction` [protected]

**Data I/O:** \*\*

Interaction to FREEZE the sim at a specified time. MTRInteractionHandler mtr\_interaction\_handler; //< **Units:** – SRFOM MTR interaction handler.

Definition at line 319 of file IMSim/ExecutionControl.hh.

Referenced by `publish()`, `receive_interaction()`, `setup_interaction_ref_attributes()`, `start_federation_save_at_scenario_time()`, `subscribe()`, `trigger_freeze_interaction()`, `unpublish()`, `unsubscribe()`, and `~ExecutionControl()`.

### 7.16.5.4 `freeze_interaction_handler`

`IMSim::FreezeInteractionHandler IMSim::ExecutionControl::freeze_interaction_handler` [protected]

**Units:** –

Freeze interaction handler.

Definition at line 320 of file IMSim/ExecutionControl.hh.

### 7.16.5.5 `freeze_scenario_times`

`FreezeTimeSet IMSim::ExecutionControl::freeze_scenario_times` [protected]

**Data I/O:** \*\*

collection of scenario times when we must enter FREEZE mode

Definition at line 322 of file IMSim/ExecutionControl.hh.

Referenced by `add_freeze_scenario_time()`, and `check_scenario_freeze_time()`.

### 7.16.5.6 `pause_sync_pts`

`PausePointList IMSim::ExecutionControl::pause_sync_pts` [protected]

**Units:** –

Synchronization points used for pausing the sim.

Definition at line 325 of file IMSim/ExecutionControl.hh.

Referenced by `add_pause()`, `check_freeze_exit()`, `check_pause()`, `clear_pause()`, `convert_loggable_sync_pts()`, `exit_freeze()`, `is_save_initiated()`, and `reinstate_logged_sync_pts()`.

### 7.16.5.7 `pending_mtr`

`MTREnum IMSim::ExecutionControl::pending_mtr` [protected]

**Units:** –

Pending Mode Transition Requested.

Definition at line 316 of file IMSim/ExecutionControl.hh.

Referenced by `check_mode_transition_request()`, `process_mode_transition_request()`, `set_mode_request_from_mtr()`, and `set_pending_mtr()`.

### 7.16.5.8 `type`

`const std::wstring IMSim::ExecutionControl::type = L"IMSim"` [static], [protected]

**Units:** –

`ExecutionControl` type string.

Definition at line 314 of file IMSim/ExecutionControl.hh.

Referenced by `get_type()`.

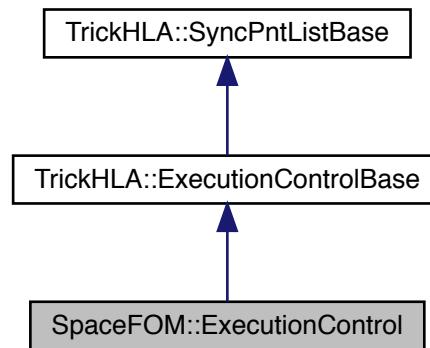
The documentation for this class was generated from the following files:

- [IMSim/ExecutionControl.hh](#)
- [IMSim/ExecutionControl.cpp](#)

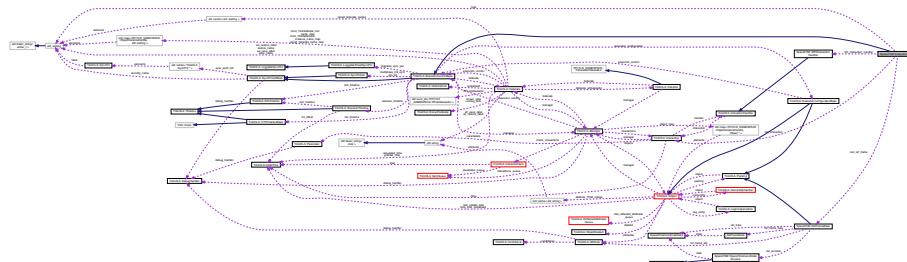
## 7.17 SpaceFOM::ExecutionControl Class Reference

#include <ExecutionControl.hh>

Inheritance diagram for SpaceFOM::ExecutionControl:



Collaboration diagram for SpaceFOM::ExecutionControl:



### Public Member Functions

- [ExecutionControl \(\)](#)  
*Default constructor for the [SpaceFOM ExecutionControl](#) class.*
- [ExecutionControl \(ExecutionConfiguration &exec\\_config\)](#)  
*Initialization constructor for the [ExecutionControl](#) class.*
- [virtual ~ExecutionControl \(\)](#)  
*Destructor for the [SpaceFOM ExecutionControl](#) class.*
- [virtual const std::wstring & get\\_type \(\)](#)  
*Get the [ExecutionControl](#) type identification string.*

- virtual void `initialize ()`  
*Execution Control initialization routine.*
- virtual void `join_federation_process ()`  
*Join federation execution process.*
- virtual void `pre_multi_phase_init_processes ()`  
*Process run before the multi-phase initialization begins.*
- virtual void `post_multi_phase_init_processes ()`  
*Process run after the multi-phase initialization ends.*
- virtual void `shutdown ()`  
*Execution control specific shutdown process.*
- virtual void `role_determination_process ()`  
*Determine the federate role in the federation execution.*
- virtual void `early_joiner_hla_init_process ()`  
*Process to join the federation execution early in initialization.*
- virtual void `late_joiner_hla_init_process ()`  
*Process to determine a federate is joining late in or after initialization.*
- virtual void `setup_object_ref_attributes ()`  
*Setup the `ExecutionControl` object Trick ref ATTRIBUTES.*
- virtual void `setup_interaction_ref_attributes ()`  
*Setup the `ExecutionControl` interaction Trick ref ATTRIBUTES.*
- virtual void `setup_object_RTI_handles ()`  
*Setup the `ExecutionControl` objects HLA RTI handles.*
- virtual void `setup_interaction_RTI_handles ()`  
*Setup the `ExecutionControl` interaction HLA RTI handles.*
- virtual void `add_initialization_sync_points ()`
- virtual void `announce_sync_point (RTI1516_NAMESPACE::RTIambassador &rti_ambassador, std::wstring const &label, RTI1516_USERDATA const &user_supplied_tag)`  
*The RTI has announced the existence of a synchronization point.*
- virtual void `publish ()`  
*Publish the `ExecutionControl` objects and interactions.*
- virtual void `unpublish ()`  
*Unpublish the `ExecutionControl` objects and interactions.*
- virtual void `subscribe ()`  
*Subscribe to the `ExecutionControl` objects and interactions.*
- virtual void `unsubscribe ()`  
*Unsubscribe the `ExecutionControl` objects and interactions.*
- virtual void `receive_interaction (RTI1516_NAMESPACE::InteractionClassHandle const &theInteraction, RTI1516_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516_USERDATA const &theUserSuppliedTag, RTI1516_NAMESPACE::LogicalTime const &theTime, bool received_as_TSO)`  
*Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.*
- virtual void `send_mode_transition_interaction (TrickHLA::ModeTransitionEnum requested_mode)`  
*Send a mode transition request to the Master federate.*
- virtual bool `process_mode_interaction ()`  
*Process a new mode interaction.*
- virtual void `set_next_execution_control_mode (TrickHLA::ExecutionControlEnum exec_control)`  
*Sets the next `ExecutionControl` run mode.*
- virtual bool `process_execution_control_updates ()`

- Process changes from any received Execution Control Objects (ExCOs).*
- `virtual void epoch_and_root_frame_discovery_process ()`

*Process to determine the federation execution epoch and root reference frame.*
  - `void wait_on_root_frame_discovered_synchronization ()`

*Waits on synchronization of the root\_frame\_discovered synchronization point.*
  - `void send_MTR_interaction (MTREnum requested_mode)`

*Send a mode transition request to the Master federate.*
  - `virtual bool check_mode_transition_request ()`

*Check to see if a new MTR is valid.*
  - `virtual bool process_mode_transition_request ()`

*Process a new Mode Transition Request (MTR).*
  - `void send_init_root_ref_frame ()`

*Send the root reference frame initialization data.*
  - `void receive_init_root_ref_frame ()`

*Wait to receive the root reference frame initialization data.*
  - `void send_root_ref_frame ()`

*Send the root reference frame data.*
  - `void receive_root_ref_frame ()`

*Wait to receive the the root reference frame data.*
  - `virtual bool run_mode_transition ()`

*The run mode transition routine.*
  - `virtual void freeze_mode_announce ()`

*Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.*
  - `virtual bool freeze_mode_transition ()`

*The freeze mode transition routine.*
  - `virtual void shutdown_mode_announce ()`

*Announce to the federation execution that a shutdown is occurring.*
  - `virtual void shutdown_mode_transition ()`

*The shutdown mode transition routine.*
  - `virtual bool check_for_shutdown ()`

*Checks to see if shutdown has been commanded.*
  - `virtual bool check_for_shutdown_with_termination ()`

*Checks to see if shutdown has been commanded and, if so, terminates the simulation.*
  - `virtual void freeze_init ()`

*Routine to handle going from run to freeze.*
  - `virtual void enter_freeze ()`

*Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.*
  - `virtual bool check_freeze_exit ()`

*Check for exit from freeze.*
  - `virtual void exit_freeze ()`

*Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.*
  - `virtual bool set_pending_mtr (MTREnum mtr_value)`
  - `virtual bool is_mtr_valid (MTREnum mtr_value)`

*Determine if the Mode Transition Request (MTR) is valid given the current mode.*
  - `virtual void set_mode_request_from_mtr (MTREnum mtr_value)`

*Translate MTR into a pending execution mode transition.*
  - `bool is_pacing () const`

- `bool is_root_frame_publisher () const`

*Query if this is the Pacing federate.*
- `virtual void start_federation_save_at_scenario_time (double freeze_scenario_time, const char *file_name)`

*Start the Federation save at the specified scenario time.*
- `virtual void set_least_common_time_step (int64_t lcts)`

*Set the least common time step in microseconds for the federation.*
- `virtual void set_time_padding (double t)`

*Set the time-padding used to offset the go to run time.*

## Data Fields

- `bool pacing`

**Units:** –  
*Is true when this federate is the "pacing".*
- `bool root_frame_pub`

**Units:** –  
*Is true when this federate is the "root reference frame publisher" federate for the Multiphase initialization process.*
- `RefFrameBase * root_ref_frame`

**Units:** –  
*Reference to the root reference frame object instance.*

## Protected Member Functions

- `ExecutionConfiguration * get_execution_configuration ()`

*Return the relevant `SpaceFOM::ExecutionConfiguration` object.*

## Protected Attributes

- `MTREnum pending_mtr`

**Units:** –  
*Pending Mode Transition Requested.*
- `TrickHLA::Interaction * mtr_interaction`

**Units:** –  
*SpaceFOM Mode Transition Request (MTR) interaction.*
- `MTRInteractionHandler mtr_interaction_handler`

**Units:** –  
*SpaceFOM MTR interaction handler.*

## Static Protected Attributes

- `static const std::wstring type = L"SpaceFOM"`

**Units:** –  
*ExecutionControl type string.*

## Private Member Functions

- `ExecutionControl (const ExecutionControl &rhs)`

*Copy constructor for `ExecutionControl` class.*
- `ExecutionControl & operator= (const ExecutionControl &rhs)`

*Assignment operator for `ExecutionControl` class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrSpaceFOM\\_\\_ExecutionControl \(\)](#)

### 7.17.1 Detailed Description

Definition at line 56 of file SpaceFOM/ExecutionControl.hh.

### 7.17.2 Constructor & Destructor Documentation

#### 7.17.2.1 ExecutionControl() [1/3]

`ExecutionControl::ExecutionControl ( )`  
Default constructor for the [SpaceFOM ExecutionControl](#) class.

**Trick Job Class:** *initialization*

Definition at line 91 of file SpaceFOM/ExecutionControl.cpp.

#### 7.17.2.2 ExecutionControl() [2/3]

`ExecutionControl::ExecutionControl (`  
    [ExecutionConfiguration & exec\\_config](#) ) [explicit]  
Initialization constructor for the [ExecutionControl](#) class.

##### Parameters

<code>exec_config</code>	The associated <a href="#">ExecutionControl</a> class instance.
--------------------------	---

**Trick Job Class:** *initialization*

Definition at line 105 of file SpaceFOM/ExecutionControl.cpp.

#### 7.17.2.3 ~ExecutionControl()

`ExecutionControl::~ExecutionControl ( ) [virtual]`  
Destructor for the [SpaceFOM ExecutionControl](#) class.  
**Trick Job Class:** *shutdown*  
Definition at line 121 of file SpaceFOM/ExecutionControl.cpp.  
References [TrickHLA::ExecutionControlBase::clear\\_mode\\_values\(\)](#).

#### 7.17.2.4 ExecutionControl() [3/3]

`SpaceFOM::ExecutionControl::ExecutionControl (`  
    [const ExecutionControl & rhs](#) ) [private]  
Copy constructor for [ExecutionControl](#) class.  
This constructor is private to prevent inadvertent copies.

### 7.17.3 Member Function Documentation

### 7.17.3.1 add\_initialization\_sync\_points()

```
void ExecutionControl::add_initialization_sync_points ( ) [virtual]
Add initialization synchronization points to regulate startup.
```

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 361 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::add\\_multiphase\\_init\\_sync\\_points\(\)](#), [TrickHLA::SyncPntListBase::add\\_sync\\_pnt\(\)](#), [SpaceFOM::INIT\\_COMPLETED\\_SYNC\\_POINT](#), [SpaceFOM::INIT\\_STARTED\\_SYNC\\_POINT](#), [SpaceFOM::OBJECTS\\_DISCOVERED\\_SYNC\\_POINT](#), and [SpaceFOM::ROOT\\_FRAME\\_DISCOVERED\\_SYNC\\_POINT](#).

Referenced by [pre\\_multi\\_phase\\_init\\_processes\(\)](#).

### 7.17.3.2 announce\_sync\_point()

```
void ExecutionControl::announce_sync_point (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    std::wstring const & label,
    RTI1516_USERDATA const & user_supplied_tag ) [virtual]
```

The RTI has announced the existence of a synchronization point.

#### Parameters

<i>rti_ambassador</i>	Reference to the HLA RTI Ambassador instance.
<i>label</i>	Sync-point label.
<i>user_supplied_tag</i>	Use supplied tag.

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 376 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::achieve\\_sync\\_pnt\(\)](#), [TrickHLA::SyncPntListBase::contains\(\)](#), [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [TrickHLA::ExecutionControlBase::init\\_complete\\_sp\\_exists](#), [SpaceFOM::INIT\\_COMPLETED\\_SYNC\\_POINT](#), [TrickHLA::SyncPntListBase::mark\\_announced\(\)](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), [TrickHLA::ExecutionControlBase::should\\_print\(\)](#), and [THLA\\_NEWLINE](#).

### 7.17.3.3 check\_for\_shutdown()

```
bool ExecutionControl::check_for_shutdown ( ) [virtual]
```

Checks to see if shutdown has been commanded.

Returns

True if shutdown has been announced, else False.

**Trick Job Class:** *shutdown*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2125 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::DEBUG\\_LEVEL\\_FULL\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [TrickHLA::SyncPntListBase::is\\_sync\\_pnt\\_announced\(\)](#), [TrickHLA::ExecutionControlBase::should\\_print\(\)](#), and [THLA\\_NEWLINE](#).

Referenced by [check\\_for\\_shutdown\\_with\\_termination\(\)](#).

### 7.17.3.4 check\_for\_shutdown\_with\_termination()

```
bool ExecutionControl::check_for_shutdown_with_termination ( ) [virtual]
```

Checks to see if shutdown has been commanded and, if so, terminates the simulation.

**Returns**

False if shutdown has NOT been announced.

NOTE: If a shutdown has been announced, this routine calls the Trick `exec_terminate()` function. So, for shutdown, it should never return. **Trick Job Class:** `shutdown`

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2146 of file SpaceFOM/ExecutionControl.cpp.

References `check_for_shutdown()`, `TrickHLA::DEBUG_LEVEL_FULL_TRACE`, `TrickHLA::DEBUG_SOURCE_FED_ERATE`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::get_federate_name()`, `TrickHLA::Federate::get_federation_name()`, `TrickHLA::ExecutionControlBase::manager`, `TrickHLA::ExecutionControlBase::should_print()`, `TrickHLA::Manager::shutdown()`, `THLA_ENDL`, and `THLA_NEWLINE`.

**7.17.3.5 check\_freeze\_exit()**

```
bool ExecutionControl::check_freeze_exit ( ) [virtual]
Check for exit from freeze.
```

**Returns**

True if should exit from freeze.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2254 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::ExecutionControlBase::current_execution_control_mode`, `TrickHLA::ExecutionControlBase::execution_configuration`, `TrickHLA::EXECUTION_CONTROL_SHUTDOWN`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::get_freeze_announced()`, `TrickHLA::Object::is_attribute_update_requested()`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::ExecutionControlBase::manager`, `process_execution_control_updates()`, `process_mode_transition_request()`, `TrickHLA::Object::receive_init_data()`, `TrickHLA::Manager::send_requested_data()`, and `TrickHLA::Federate::shutdown()`.

**7.17.3.6 check\_mode\_transition\_request()**

```
bool ExecutionControl::check_mode_transition_request ( ) [virtual]
Check to see if a new MTR is valid.
```

**Returns**

True if new MTR is valid.

Definition at line 1441 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::ExecutionControlBase::is_mode_transition_requested()`, `is_mtr_valid()`, `SpaceFOM::mtr_enum_to_string()`, `pending_mtr`, and `THLA_ENDL`.

Referenced by `post_multi_phase_init_processes()`, and `process_mode_transition_request()`.

**7.17.3.7 early\_joiner\_hla\_init\_process()**

```
void ExecutionControl::early_joiner_hla_init_process ( ) [virtual]
Process to join the federation execution early in initialization.
```

This routine implements the [SpaceFOM](#) Master and Early Joiner HLA initialization process described in section 7.2 and figure 7-5.

**Trick Job Class:** `initialization`

Definition at line 689 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::SyncPntListBase::achieve\_and\_wait\_for\_synchronization(), TrickHLA::ExecutionControlBase::federate, get\_execution\_configuration(), TrickHLA::Federate::get\_RTI\_ambassador(), SpaceFOM::INIT\_STARTE\_D\_SYNC\_POINT, TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::ExecutionControlBase::manager, SpaceFOM::OBJECTS\_DISCOVERED\_SYNC\_POINT, TrickHLA::Manager::publish\_and\_subscribe(), TrickHLA::Manager::register\_objects\_with\_RTI(), TrickHLA::Object::reserve\_object\_name\_with\_RTI(), TrickHLA::Manager::reserve\_object\_names\_with\_RTI(), SpaceFOM::ROOT\_FRAME\_DISCOVERED\_SYNC\_POINT, TrickHLA::Manager::setup\_all\_RTI\_handles(), TrickHLA::SyncPntListBase::wait\_for\_announcement(), TrickHLA::Object::wait\_on\_object\_name\_reservation(), TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects(), and TrickHLA::Manager::wait\_on\_reservation\_of\_object\_names().

Referenced by pre\_multi\_phase\_init\_processes().

#### 7.17.3.8 enter\_freeze()

void ExecutionControl::enter\_freeze ( ) [virtual]

Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2200 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::ExecutionControlBase::federate, freeze\_mode::announce(), get\_execution\_configuration(), TrickHLA::Federate::get\_freeze\_announced(), TrickHLA::Federate::get\_freeze\_pending(), TrickHLA::ExecutionControlBase::get\_requested\_execution\_control\_mode(), TrickHLA::ExecutionControlBase::get\_simulation\_freeze\_time(), TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::Object::send\_init\_data(), set\_next\_execution\_control\_mode(), THLA\_NEWLINE, and TrickHLA::Federate::unfreeze().

#### 7.17.3.9 epoch\_and\_root\_frame\_discovery\_process()

void ExecutionControl::epoch\_and\_root\_frame\_discovery\_process ( ) [virtual]

Process to determine the federation execution epoch and root reference frame.

This routine implements the [SpaceFOM](#) Epoch and Root Reference Frame Discovery initialization process described in section 7.2 and figure 7-6.

**Trick Job Class:** *initialization*

Definition at line 2345 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::Timeline::get\_epoch(), get\_execution\_configuration(), SpaceFOM::RefFrameBase::get\_name(), SpaceFOM::ExecutionConfiguration::get\_scenario\_time\_epoch(), TrickHLA::ExecutionControlBase::is\_master(), is\_root\_frame\_publisher(), process\_execution\_control\_updates(), receive\_init\_root\_ref\_frame(), root\_ref\_frame, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Object::send\_init\_data(), send\_init\_root\_ref\_frame(), TrickHLA::Timeline::set\_epoch(), SpaceFOM::ExecutionConfiguration::set\_root\_frame\_name(), SpaceFOM::ExecutionConfiguration::set\_scenario\_time\_epoch(), wait\_on\_root\_frame\_discovered\_synchronization(), and SpaceFOM::ExecutionConfiguration::wait\_on\_update().

Referenced by pre\_multi\_phase\_init\_processes().

#### 7.17.3.10 exit\_freeze()

void ExecutionControl::exit\_freeze ( ) [virtual]

Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2300 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::EXECUTION\_CONTROL\_RUNNING, get\_execution\_configuration(), TrickHLA::ExecutionControlBase::is\_master(), run\_mode\_transition(), TrickHLA::Object::send\_init\_data(), set\_next\_execution\_control\_mode(), and the\_clock.

### 7.17.3.11 `freeze_init()`

```
void ExecutionControl::freeze_init ( ) [virtual]
Routine to handle going from run to freeze.
```

**Assumptions and Limitations:**

- Currently only used with [SpaceFOM](#) initialization schemes. **Trick Job Class:** `freeze_init`

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2178 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::EXECUTION_CONTROL_FREEZE`, `TrickHLA::EXECUTION_CONTROL_INITIALIZING`, `TrickHLA::ExecutionControlBase::federate`, `freeze_mode_transition()`, `TrickHLA::ExecutionControlBase::get_current_execution_control_mode()`, `TrickHLA::ExecutionControlBase::is_lateJoiner()`, `TrickHLA::ExecutionControlBase::set_current_execution_control_mode()`, and `TrickHLA::Federate::set_freeze_announced()`.

### 7.17.3.12 `freeze_mode_announce()`

```
void ExecutionControl::freeze_mode_announce ( ) [virtual]
Announce the pending freeze mode transition with an 'mtr_freeze' sync-point.
```

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2028 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::get_RTI_ambassador()`, `TrickHLA::ExecutionControlBase::is_master()`, and `TrickHLA::SyncPntListBase::register_sync_pnt()`.

Referenced by `enter_freeze()`, `post_multi_phase_init_processes()`, `process_execution_control_updates()`, and `process_mode_transition_request()`.

### 7.17.3.13 `freeze_mode_transition()`

```
bool ExecutionControl::freeze_mode_transition ( ) [virtual]
The freeze mode transition routine.
```

**Returns**

Currently always returns False.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2039 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::SyncPnt::achieve_sync_point()`, `TrickHLA::ExecutionControlBase::current_execution_control_mode`, `TrickHLA::EXECUTION_CONTROL_FREEZE`, `SpaceFOM::EXECUTION_MODE_FREEZE`, `TrickHLA::ExecutionControlBase::federate`, `get_execution_configuration()`, `TrickHLA::Federate::get_RTI_ambassador()`, `TrickHLA::SyncPntListBase::get_sync_pnt()`, `SpaceFOM::ExecutionConfiguration::set_current_execution_mode()`, `THLA-ENDL`, `TrickHLA::SyncPnt::wait_for_announce()`, and `TrickHLA::SyncPnt::wait_for_synchronization()`.

Referenced by `freeze_init()`.

### 7.17.3.14 `get_execution_configuration()`

```
ExecutionConfiguration * ExecutionControl::get_execution_configuration ( ) [protected], [virtual]
Return the relevant SpaceFOM::ExecutionConfiguration object.
```

**Returns**

Pointer to the relevant [SpaceFOM::ExecutionConfiguration](#) object.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2324 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::execution\\_configuration](#), and [THLA\\_ENDL](#).

Referenced by [earlyJoinerHlaInitProcess\(\)](#), [enterFreeze\(\)](#), [epochAndRootFrameDiscoveryProcess\(\)](#), [exitFreeze\(\)](#), [freezeModeTransition\(\)](#), [isMtrValid\(\)](#), [lateJoinerHlaInitProcess\(\)](#), [postMultiPhaseInitProcesses\(\)](#), [preMultiPhaseInitProcesses\(\)](#), [processExecutionControlUpdates\(\)](#), [processModeTransitionRequest\(\)](#), [runModeTransition\(\)](#), [setNextExecutionControlMode\(\)](#), and [setupObjectRTIHandles\(\)](#).

**7.17.3.15 get\_type()**

```
virtual const std::wstring& SpaceFOM::ExecutionControl::get_type ( ) [inline], [virtual]
```

Get the [ExecutionControl](#) type identification string.

**Returns**

A constant reference to the type identification string.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 93 of file SpaceFOM/ExecutionControl.hh.

References type.

**7.17.3.16 initialize()**

```
void ExecutionControl::initialize ( ) [virtual]
```

Execution Control initialization routine.

This routine will set a lot of the data in the [TrickHLA::Federate](#) that is required for this execution control scheme. This should greatly simplify input files and reduce input file setting errors.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 134 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::add\\_sync\\_pnt\(\)](#), [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\\_preset\(\)](#), [is\\_pacing\(\)](#), [TrickHLA::ExecutionControlBase::least\\_common\\_time\\_step](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), [THLA\\_ENDL](#), [THLA\\_NEWLINE](#), [TrickHLA::Federate::time\\_constrained](#), [TrickHLA::Federate::time\\_management](#), [TrickHLA::Federate::time\\_regulating](#), and [TrickHLA::ExecutionControlBase::use\\_preset\\_master](#).

**7.17.3.17 is\_mtr\_valid()**

```
bool ExecutionControl::is_mtr_valid (
    MTREnum mtr_value ) [virtual]
```

Determine if the Mode Transition Request (MTR) is valid given the current mode.

**Returns**

True if valid, false otherwise.

**Parameters**

<i>mtr_value</i>	Mode transition request.
------------------	--------------------------

Definition at line 1273 of file SpaceFOM/ExecutionControl.cpp.

References SpaceFOM::ExecutionConfiguration::current\_execution\_mode, SpaceFOM::EXECUTION\_MODE\_FREEZE, SpaceFOM::EXECUTION\_MODE\_INITIALIZING, SpaceFOM::EXECUTION\_MODE\_RUNNING, SpaceFOM::EXECUTION\_MODE\_SHUTDOWN, get\_execution\_configuration(), SpaceFOM::MTR\_GOTO\_FREEZE, SpaceFOM::MTR\_GOTO\_RUN, and SpaceFOM::MTR\_GOTO\_SHUTDOWN.

Referenced by check\_mode\_transition\_request(), SpaceFOM::MTRInteractionHandler::receive\_interaction(), and set\_pending\_mtr().

#### 7.17.3.18 is\_pacing()

```
bool SpaceFOM::ExecutionControl::is_pacing ( ) const [inline]
```

Query if this is the Pacing federate.

##### Returns

True if there is the Pacing federate; False otherwise.

Definition at line 248 of file SpaceFOM/ExecutionControl.hh.

References pacing.

Referenced by initialize().

#### 7.17.3.19 is\_root\_frame\_publisher()

```
bool SpaceFOM::ExecutionControl::is_root_frame_publisher ( ) const [inline]
```

Query if this is the Root Reference Frame Publisher federate.

##### Returns

True if there is the Root Reference Frame Publisher federate; False otherwise.

Definition at line 251 of file SpaceFOM/ExecutionControl.hh.

References root\_frame\_pub.

Referenced by epoch\_and\_root\_frame\_discovery\_process(), pre\_multi\_phase\_init\_processes(), receive\_root\_ref\_frame(), and send\_root\_ref\_frame().

#### 7.17.3.20 join\_federation\_process()

```
void ExecutionControl::join_federation_process ( ) [virtual]
```

Join federation execution process.

This routine implements the [SpaceFOM](#) Join Federation Process described in section 7.2 and figure 7-3.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 552 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::join\_federation\_process().

Referenced by pre\_multi\_phase\_init\_processes().

#### 7.17.3.21 late\_joiner\_hla\_init\_process()

```
void ExecutionControl::late_joiner_hla_init_process ( ) [virtual]
```

Process to determine is a federate is joining late in or after initialization.

This routine implements the [SpaceFOM](#) Late Joiner initialization process described in section 7.2 and figure 7-9.

**Trick Job Class:** *initialization*

Definition at line 764 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::cte\_timeline, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ExecutionControlBase::does\_cte\_timeline\_exist(), TrickHLA::ExecutionControlBase::federate, TrickHLA::Timeline::get\_epoch(), get\_execution\_configuration(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Object::get\_name(), TrickHLA::Federate::get\_requested\_time(), SpaceFOM::ExecutionConfiguration::get\_scenario\_time\_epoch(), TrickHLA::ScenarioTimeline::get\_sim\_offset(), TrickHLA::SimTimeline::get\_time(), TrickHLA::CTETimelineBase::get\_time(), TrickHLA::ScenarioTimeline::get\_time(), TrickHLA::ExecutionControlBase::manager, SpaceFOM::ExecutionConfiguration::next\_execution\_mode, process\_execution\_control\_updates(), TrickHLA::Manager::publish\_and\_subscribe(), TrickHLA::Manager::register\_objects\_with\_RTI(), TrickHLA::Manager::request\_data\_update(), TrickHLA::Manager::reserve\_object\_names\_with\_RTI(), SpaceFOM::ExecutionConfiguration::scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Timeline::set\_epoch(), TrickHLA::ExecutionControlBase::set\_requested\_execution\_control\_mode(), TrickHLA::Manager::setup\_all\_RTI\_handles(), TrickHLA::Federate::should\_print(), TrickHLA::ExecutionControlBase::sim\_timeline, TrickHLA::Object::subscribe\_to\_object\_attributes(), TrickHLA::ExecutionConfigurationBase::wait\_on\_registration(), TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects(), TrickHLA::Manager::wait\_on\_reservation\_of\_object\_names(), and SpaceFOM::ExecutionConfiguration::wait\_on\_update().

Referenced by pre\_multi\_phase\_init\_processes().

### 7.17.3.22 operator=( )

```
ExecutionControl& SpaceFOM::ExecutionControl::operator= (
    const ExecutionControl & rhs ) [private]
```

Assignment operator for [ExecutionControl](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.17.3.23 post\_multi\_phase\_init\_processes()

```
void ExecutionControl::post_multi_phase_init_processes ( ) [virtual]
```

Process run after the multi-phase initialization ends.

This routine implements the [SpaceFOM](#) post multi-phase initialization process described in section 7.2 and figures 7-8 and 7-9.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1038 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::SyncPntListBase::achieve\_and\_wait\_for\_synchronization(), check\_mode\_transition\_request(), TrickHLA::ExecutionControlBase::cte\_timeline, TrickHLA::ExecutionControlBase::current\_execution\_control\_mode, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ExecutionControlBase::does\_cte\_timeline\_exist(), TrickHLA::execution\_control\_enum\_to\_string(), TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::ExecutionControlBase::federate, freeze\_mode\_announce(), TrickHLA::Timeline::get\_epoch(), get\_execution\_configuration(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Federate::get\_requested\_time(), TrickHLA::Federate::get\_RTI\_ambassador(), TrickHLA::ScenarioTimeline::get\_sim\_offset(), TrickHLA::SimTimeline::get\_time(), TrickHLA::CTETimelineBase::get\_time(), TrickHLA::ScenarioTimeline::get\_time(), SpaceFOM::INIT\_COMPLETE\_SYNC\_POINT, SpaceFOM::INIT\_STARTED\_SYNC\_POINT, TrickHLA::ExecutionControlBase::is\_lateJoiner(), TrickHLA::ExecutionControlBase::is\_master(), pending\_mtr, SpaceFOM::ExecutionConfiguration::print\_execution\_configuration(), process\_execution\_control\_updates(), TrickHLA::SyncPntListBase::register\_sync\_pnt(), TrickHLA::ExecutionControlBase::requested\_execution\_control\_mode, TrickHLA::Federate::restore\_orig\_MOM\_auto\_provide\_setting(), run\_mode\_transition(), SpaceFOM::ExecutionConfiguration::scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Object::send\_init\_data(), set\_mode\_request\_from\_mtr(), set\_next\_execution\_control\_mode(), TrickHLA::ScenarioTimeline::set\_sim\_offset(), TrickHLA::Federate::setup\_time\_management(), TrickHLA::Federate::should\_print(), shutdown\_mode\_announce(), TrickHLA::ExecutionControlBase::sim\_timeline, THLA\_NEWLINE, TrickHLA::Federate::time\_advance\_request\_to\_GALT\_LCTS\_multiple(), and SpaceFOM::ExecutionConfiguration::wait\_on\_update().

### 7.17.3.24 pre\_multi\_phase\_init\_processes()

```
void ExecutionControl::pre_multi_phase_init_processes ( ) [virtual]
Process run before the multi-phase initialization begins.
```

This routine implements the [SpaceFOM](#) pre multi-phase initialization process described in section 7.2 figure 7-2.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 854 of file SpaceFOM/ExecutionControl.cpp.

References `add_initialization_sync_points()`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUGLEV` ↔ `EL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `earlyJoinerHlaInitProcess()`, `epochAndRootFrame` ↔ `_discovery_process()`, `TrickHLA::EXECUTION_CONTROL_INITIALIZING`, `SpaceFOM::EXECUTION_MODE_INIT` ↔ `INITIALIZING`, `TrickHLA::ExecutionControlBase::federate`, `getExecutionConfiguration()`, `TrickHLA::Object::getFOM` ↔ `name()`, `SpaceFOM::ExecutionConfiguration::getLeastCommonTimeStep()`, `TrickHLA::Federate::getLookahead()`, `TrickHLA::Int64Interval::getTimeInMicros()`, `TrickHLA::ExecutionControlBase::isMaster()`, `isRootFramePublisher()`, `joinFederationProcess()`, `TrickHLA::ExecutionControlBase::lateJoiner`, `lateJoinerHlaInitProcess()`, `SpaceFOM` ↔ `M::ExecutionConfiguration::leastCommonTimeStep`, `TrickHLA::ExecutionControlBase::leastCommonTimeStep`, `TrickHLA::ExecutionControlBase::manager`, `TrickHLA::Object::markRequired()`, `TrickHLA::ExecutionConfiguration` ↔ `Base::resetOwnershipStates()`, `TrickHLA::ExecutionConfigurationBase::resetPreferredOrder()`, `roleDetermination` ↔ `_process()`, `rootRefFrame`, `TrickHLA::ExecutionControlBase::setCurrentExecutionControlMode()`, `SpaceFOM` ↔ `ExecutionConfiguration::setCurrentExecutionMode()`, `TrickHLA::ExecutionConfigurationBase::setMaster()`, `set` ↔ `nextExecutionControlMode()`, `TrickHLA::ExecutionControlBase::setRequestedExecutionControlMode()`, `TrickHLA::Manager::setupAllRefAttributes()`, `TrickHLA::DebugHandler::shouldPrint()`, `THLA_ENDL`, `THLA_NEWLINE`, and `TrickHLA::ExecutionControlBase::timePadding`.

### 7.17.3.25 process\_execution\_control\_updates()

```
bool ExecutionControl::process_execution_control_updates ( ) [virtual]
```

Process changes from any received Execution Control Objects (ExCOs).

**Returns**

True if mode change occurred.

**Assumptions and Limitations:**

- Called from the ExCO unpack routine.

**Trick Job Class:** *scheduled*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1589 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::ExecutionConfigurationBase::clearUpdatePending()`, `TrickHLA::ScenarioTimeline::compute` ↔ `_simulationTime()`, `TrickHLA::ExecutionControlBase::cteTimeline`, `TrickHLA::ExecutionControlBase::current` ↔ `executionControlMode`, `SpaceFOM::ExecutionConfiguration::currentExecutionMode`, `TrickHLA::DEBUGLEV` ↔ `EL_4_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::doesCteTimelineExist()`, `TrickHLA::executionControlEnumToString()`, `TrickHLA::EXECUTION_CONTROL_FREEZE`, `TrickHLA::EXECUTION_CONTROL_INITIALIZING`, `TrickHLA::EXECUTION_CONTROL_RUNNING`, `TrickHLA::EXECUTION_CONTROL_SHUTDOWN`, `TrickHLA::EXECUTION_CONTROL_UNINITIALIZED`, `SpaceFOM::executionModeEnumToString()`, `SpaceFOM::EXECUTION_MODE_FREEZE`, `SpaceFOM::executionModeInt16ToString()`, `SpaceFOM::EXECUTION_MODE_RUNNING`, `SpaceFOM::EXECUTION_MODE_SHUTDOWN`, `TrickHLA::ExecutionControlBase::federate`, `freezeModeAnnounce()`, `SpaceFOM::fromExecutionControlEnum()`, `TrickHLA::Timeline::getEpoch()`, `getExecutionConfiguration()`, `TrickHLA::ScenarioTimeline::getSimOffset()`, `TrickHLA::SimTimeline::getTime()`, `TrickHLA::CTETimelineBase::getTime()`, `TrickHLA::ScenarioTimeline::getTime()`, `TrickHLA::ExecutionControlBase::isMaster()`, `SpaceFOM::ExecutionConfiguration::nextExecutionMode`, `SpaceFOM::ExecutionConfiguration::nextModeCteTime`, `TrickHLA::ExecutionControlBase::nextModeCteTime`, `SpaceFOM::ExecutionConfiguration::nextModeScenarioTime`, `TrickHLA::ExecutionControlBase::requestedExecutionControlMode`, `runMode` ↔

transition(), TrickHLA::ExecutionControlBase::scenario\_freeze\_time, SpaceFOM::ExecutionConfiguration::scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Federate::should\_print(), TrickHLA::ExecutionControlBase::sim\_timeline, TrickHLA::ExecutionControlBase::simulation\_freeze\_time, THLA\_ENDL, and TrickHLA::ExecutionConfigurationBase::update\_pending().

Referenced by check\_freeze\_exit(), epoch\_and\_root\_frame\_discovery\_process(), late\_joiner\_hla\_init\_process(), post\_multi\_phase\_init\_processes(), and run\_mode\_transition().

#### 7.17.3.26 process\_mode\_interaction()

bool ExecutionControl::process\_mode\_interaction ( ) [virtual]

Process a new mode interaction.

Returns

True if new mode interaction is successfully processed.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1473 of file SpaceFOM/ExecutionControl.cpp.

References process\_mode\_transition\_request().

#### 7.17.3.27 process\_mode\_transition\_request()

bool ExecutionControl::process\_mode\_transition\_request ( ) [virtual]

Process a new Mode Transition Request (MTR).

Returns

True if new MTR is successfully processed.

Definition at line 1478 of file SpaceFOM/ExecutionControl.cpp.

References check\_mode\_transition\_request(), TrickHLA::ExecutionControlBase::clear\_mode\_transition\_requested(), TrickHLA::ExecutionControlBase::cte\_timeline, TrickHLA::ExecutionControlBase::current\_execution\_control\_mode, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ExecutionControlBase::does\_cte\_timeline\_exist(), TrickHLA::EXECUTION\_CONTROL\_FREEZE, TrickHLA::EXECUTION\_CONTROL\_RUNNING, TrickHLA::ExecutionControlBase::federate, freeze\_mode\_announce(), TrickHLA::Timeline::get\_epoch(), get\_execution\_configuration(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Federate::get\_requested\_time(), TrickHLA::ScenarioTimeline::get\_sim\_offset(), TrickHLA::SimTimeline::get\_time(), TrickHLA::CTETimelineBase::get\_time(), TrickHLA::ScenarioTimeline::get\_time(), SpaceFOM::MTR\_GOTO\_FREEZE, SpaceFOM::MTR\_GOTO\_RUN, SpaceFOM::MTR\_GOTO\_SHUTDOWN, SpaceFOM::ExecutionConfiguration::next\_mode\_cte\_time, SpaceFOM::ExecutionConfiguration::next\_mode\_scenario\_time, pending\_mtr, TrickHLA::ExecutionControlBase::scenario\_freeze\_time, SpaceFOM::ExecutionConfiguration::scenario\_time\_epoch, TrickHLA::ExecutionControlBase::scenario\_timeline, TrickHLA::Object::send\_init\_data(), set\_mode\_request\_from\_mtr(), TrickHLA::Federate::should\_print(), shutdown\_mode\_announce(), TrickHLA::ExecutionControlBase::sim\_timeline, TrickHLA::ExecutionControlBase::simulation\_freeze\_time, and TrickHLA::ExecutionControlBase::time\_padding.

Referenced by check\_freeze\_exit(), and process\_mode\_interaction().

#### 7.17.3.28 publish()

void ExecutionControl::publish ( ) [virtual]

Publish the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 423 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::execution\_configuration, TrickHLA::ExecutionControlBase::is\_master(), mtr\_interaction, TrickHLA::Interaction::publish\_interaction(), and TrickHLA::Object::publish\_object\_attributes().

### 7.17.3.29 receive\_init\_root\_ref\_frame()

```
void ExecutionControl::receive_init_root_ref_frame ( )
```

Wait to receive the root reference frame initialization data.

**Trick Job Class:** *initialization*

Definition at line 2488 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, SpaceFOM::RefFrameBase::get\_name(), TrickHLA::Manager::is\_late\_joining\_federate(), TrickHLA::ExecutionControlBase::manager, receive\_root\_ref\_frame(), root\_ref\_frame, TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

Referenced by epoch\_and\_root\_frame\_discovery\_process().

### 7.17.3.30 receive\_interaction()

```
void ExecutionControl::receive_interaction (
    RTI1516_NAMESPACE::InteractionClassHandle const & theInteraction,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_NAMESPACE::Userdata const & theUserSuppliedTag,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    bool received_as_TSO ) [virtual]
```

Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.

#### Parameters

<i>theInteraction</i>	Interaction handle.
<i>theParameterValues</i>	Parameter values.
<i>theUserSuppliedTag</i>	Users tag.
<i>theTime</i>	HLA time for the interaction.
<i>received_as_TSO</i>	True if interaction was received by RTI as TSO.

**Trick Job Class:** *scheduled*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 489 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::Interaction::extract\_data(), TrickHLA::Interaction::get\_class\_handle(), TrickHLA::Interaction::get\_parameter\_count(), TrickHLA::Interaction::get\_parameters(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Interaction::is\_subscribe(), mtr\_interaction, TrickHLA::Interaction::process\_interaction(), TrickHLA::Int64Time::setTo(), TrickHLA::DebugHandler::should\_print(), THLA\_NEWLINE, and TrickHLA::TRICKHLA\_MANAGER\_BUILTIN\_MTR\_INTERACTION.

### 7.17.3.31 receive\_root\_ref\_frame()

```
void ExecutionControl::receive_root_ref_frame ( )
```

Wait to receive the the root reference frame data.

**Trick Job Class:** *initialization*

Definition at line 2508 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::Object::any\_remotely\_owned\_subscribed\_init\_attribute(), TrickHLA::Federate::check\_for\_shutdown\_with\_termination(), TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ExecutionControlBase::federate, SpaceFOM::RefFrameBase::get\_object(), TrickHLA::Object::is\_changed(), TrickHLA::Federate::is\_execution\_member(), is\_root\_frame\_publisher(), TrickHLA::Object::receive\_init\_data(), root\_ref\_frame, TrickHLA::DebugHandler::should\_print(), THLA\_ENDL, and THLA\_NEWLINE.

Referenced by `receive_init_root_ref_frame()`.

#### 7.17.3.32 `role_determination_process()`

```
void ExecutionControl::role_determination_process ( ) [virtual]
```

Determine the federate role in the federation execution.

This routine implements the [SpaceFOM](#) Role Determination Process described in section 7.2 and figure 7-4.

**Trick Job Class:** *initialization*

Definition at line 567 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::Federate::backup_auto_provide_setting_from_MOM_then_disable()`, `TrickHLA::Federate::check_for_shutdown_with_termination()`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::does_init_complete_sync_point_exist()`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::get_joined_federate_handles()`, `TrickHLA::Federate::get_RTI_ambassador()`, `TrickHLA::SyncPntListBase::get_sync_pnt()`, `SpaceFOM::INIT_STARTED_SYNC_POINT`, `TrickHLA::Federate::initialize_MOM_handles()`, `TrickHLA::SyncPnt::is_announced()`, `TrickHLA::Federate::is_execution_member()`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::ExecutionControlBase::late_joiner`, `TrickHLA::ExecutionControlBase::late_joiner_determined`, `TrickHLA::ExecutionControlBase::multiphase_init_sync_pnt_list`, `SpaceFOM::OBJECTS_DISCOVERED_SYNC_POINT`, `TrickHLA::SyncPntListBase::register_all_sync_pnts()`, `TrickHLA::SyncPntListBase::register_sync_pnt()`, `SpaceFOM::ROOT_FRAME_DISCOVERED_SYNC_POINT`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::Federate::wait_for_required_federates_to_join()`, `TrickHLA::ExecutionControlBase::wait_sleep`, and `TrickHLA::ExecutionControlBase::wait_timeout`. Referenced by `pre_multi_phase_init_processes()`.

#### 7.17.3.33 `run_mode_transition()`

```
bool ExecutionControl::run_mode_transition ( ) [virtual]
```

The run mode transition routine.

**Returns**

Currently always returns True.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1940 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::SyncPnt::achieve_sync_point()`, `TrickHLA::Federate::check_for_shutdown_with_termination()`, `TrickHLA::ExecutionControlBase::current_execution_control_mode`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::does_cte_timeline_exist()`, `TrickHLA::EXECUTION_CONTROL_RUNNING`, `SpaceFOM::EXECUTION_MODE_RUNNING`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::ExecutionControlBase::get_cte_time()`, `get_execution_configuration()`, `SpaceFOM::ExecutionConfiguration::get_next_mode_cte_time()`, `TrickHLA::Federate::get_RTI_ambassador()`, `TrickHLA::SyncPntListBase::get_sync_pnt()`, `TrickHLA::ExecutionControlBase::is_master()`, `process_execution_control_updates()`, `TrickHLA::SyncPntListBase::register_sync_pnt()`, `TrickHLA::Object::send_init_data()`, `SpaceFOM::ExecutionConfiguration::set_current_execution_mode()`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::SyncPnt::wait_for_announce()`, `TrickHLA::SyncPnt::wait_for_synchronization()`, and `SpaceFOM::ExecutionConfiguration::wait_on_update()`. Referenced by `exit_freeze()`, `post_multi_phase_init_processes()`, and `process_execution_control_updates()`.

#### 7.17.3.34 `send_init_root_ref_frame()`

```
void ExecutionControl::send_init_root_ref_frame ( )
```

Send the root reference frame initialization data.

**Trick Job Class:** *initialization*

Definition at line 2419 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBU←G_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::execution_configuration`, `TrickHLA::Object::get_name()`, `TrickHLA::Manager::is_late_joining_federate()`, `TrickHLA::ExecutionControlBase::manager`, `send_root_ref_frame()`, `TrickHLA::DebugHandler::should_print()`, and `THLA_NEWLINE`.  
 Referenced by `epoch_and_root_frame_discovery_process()`.

#### 7.17.3.35 send\_mode\_transition\_interaction()

```
void ExecutionControl::send_mode_transition_interaction (
    TrickHLA::ModeTransitionEnum requested_mode ) [virtual]
```

Send a mode transition request to the Master federate.

##### Parameters

<code>requested_mode</code>	Requested mode.
-----------------------------	-----------------

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1923 of file SpaceFOM/ExecutionControl.cpp.

References `SpaceFOM::from_mode_transition_enum()`, and `send_MTR_interaction()`.

#### 7.17.3.36 send\_MTR\_interaction()

```
void ExecutionControl::send_MTR_interaction (
    SpaceFOM::MTREnum requested_mode )
```

Send a mode transition request to the Master federate.

##### Parameters

<code>requested_mode</code>	Requested mode.
-----------------------------	-----------------

##### Trick Job Class: *scheduled*

Definition at line 1933 of file SpaceFOM/ExecutionControl.cpp.

References `mtr_interaction_handler`, and `SpaceFOM::MTRInteractionHandler::send_interaction()`.

Referenced by `send_mode_transition_interaction()`.

#### 7.17.3.37 send\_root\_ref\_frame()

```
void ExecutionControl::send_root_ref_frame ( )
```

Send the root reference frame data.

##### Trick Job Class: *initialization*

Definition at line 2441 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::Object::any_locally_owned_published_init_attribute()`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `SpaceFOM::RefFrameBase::get_object()`, `is_root_frame_publisher()`, `root_ref_frame`, `TrickHLA::Object::send_init_data()`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `send_init_root_ref_frame()`.

#### 7.17.3.38 set\_least\_common\_time\_step()

```
void ExecutionControl::set_least_common_time_step (
```

```
int64_t lcts ) [virtual]
```

Set the least common time step in microseconds for the federation.

#### Parameters

<i>lcts</i>	Least Common Time Step time in microseconds.
-------------	--

WARNING: Only the Master federate should ever set this.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2599 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::execution\\_configuration](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::ExecutionControlBase::least\\_common\\_time\\_step](#), [SpaceFOM::ExecutionConfiguration::set\\_least\\_common\\_time\\_step\(\)](#), and [THLA\\_ENDL](#).

#### 7.17.3.39 set\_mode\_request\_from\_mtr()

```
void ExecutionControl::set_mode_request_from_mtr (
    MTREnum mtr_value ) [virtual]
```

Translate MTR into a pending execution mode transition.

#### Parameters

<i>mtr_value</i>	MTR value for next execution mode.
------------------	------------------------------------

Definition at line 1295 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::EXECUTION\\_CONTROL\\_FREEZE](#), [TrickHLA::EXECUTION\\_CONTROL\\_INITIALIZING](#), [TrickHLA::EXECUTION\\_CONTROL\\_RUNNING](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::EXECUTION\\_CONTROL\\_UNINITIALIZED](#), [SpaceFOM::MTR\\_GOTO\\_FREEZE](#), [SpaceFOM::MTR\\_GOTO\\_RUN](#), [SpaceFOM::MTR\\_GOTO\\_SHUTDOWN](#), [SpaceFOM::MTR\\_INITIALIZING](#), [SpaceFOM::MTR\\_UNINITIALIZED](#), [pending\\_mtr](#), and [set\\_next\\_execution\\_control\\_mode\(\)](#).

Referenced by [post\\_multi\\_phase\\_init\\_processes\(\)](#), and [process\\_mode\\_transition\\_request\(\)](#).

#### 7.17.3.40 set\_next\_execution\_control\_mode()

```
void ExecutionControl::set_next_execution_control_mode (
    TrickHLA::ExecutionControlEnum exec_control ) [virtual]
```

Sets the next [ExecutionControl](#) run mode.

#### Parameters

<i>exec_control</i>	Next <a href="#">ExecutionControl</a> run mode.
---------------------	---

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1332 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::ScenarioTimeline::compute\\_simulation\\_time\(\)](#), [TrickHLA::DEBUG\\_LEVEL\\_1\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::EXECUTION\\_CONTROL\\_FREEZE](#), [TrickHLA::EXECUTION\\_CONTROL\\_INITIALIZING](#), [TrickHLA::EXECUTION\\_CONTROL\\_RUNNING](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::EXECUTION\\_CONTROL\\_UNINITIALIZED](#), [SpaceFOM::EXECUTION\\_MODE\\_FREEZE](#), [SpaceFOM::EXECUTION\\_MODE\\_INITIALIZING](#), [SpaceFOM::EXECUTION\\_MODE\\_RUNNING](#), [SpaceFOM::EXECUTION\\_MODE\\_SHUTDOWN](#), [SpaceFOM::EXECUTION\\_MODE\\_UNINITIALIZED](#), [TrickHLA::ExecutionControlBase::get\\_cte\\_time\(\)](#), [get\\_execution\\_configuration\(\)](#), [TrickHLA::ExecutionControlBase::get\\_manager\(\)](#), [SpaceFOM::ExecutionConfiguration::get\\_next\\_mode\\_cte\\_time\(\)](#), [TrickHLA::ExecutionControlBase::get\\_scenario\\_time\(\)](#), [TrickHLA::ExecutionControlBase::get\\_time\(\)](#).

HLA::ExecutionControlBase::is\_master(), TrickHLA::ExecutionControlBase::next\_mode\_scenario\_time, TrickHLA::ExecutionControlBase::requested\_execution\_control\_mode, TrickHLA::ExecutionControlBase::scenario\_freeze\_time, TrickHLA::ExecutionControlBase::scenario\_timeline, SpaceFOM::ExecutionConfiguration::set\_next\_execution\_mode(), SpaceFOM::ExecutionConfiguration::set\_next\_mode\_cte\_time(), SpaceFOM::ExecutionConfiguration::set\_next\_mode\_scenario\_time(), SpaceFOM::ExecutionConfiguration::set\_scenario\_time\_epoch(), TrickHLA::ExecutionControlBase::should\_print(), TrickHLA::ExecutionControlBase::simulation\_freeze\_time, THLA\_ENDL, and TrickHLA::ExecutionControlBase::time\_padding.  
Referenced by enter\_freeze(), exit\_freeze(), post\_multi\_phase\_init\_processes(), pre\_multi\_phase\_init\_processes(), set\_mode\_request\_from\_mtr(), and shutdown\_mode\_announce().

#### 7.17.3.41 set\_pending\_mtr()

```
bool ExecutionControl::set_pending_mtr (
    MTREnum mtr_value ) [virtual]
```

Definition at line 1264 of file SpaceFOM/ExecutionControl.cpp.

References is\_mtr\_valid(), and pending\_mtr.

Referenced by SpaceFOM::MTRInteractionHandler::receive\_interaction().

#### 7.17.3.42 set\_time\_padding()

```
void ExecutionControl::set_time_padding (
    double t ) [virtual]
```

Set the time-padding used to offset the go to run time.

##### Parameters

<i>t</i>	Time in seconds to pad for time based mode transitions.
----------	---

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 2621 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::least\_common\_time\_step, THLA\_ENDL, THLA\_NEWLINE, and TrickHLA::ExecutionControlBase::time\_padding.

#### 7.17.3.43 setup\_interaction\_ref\_attributes()

```
void ExecutionControl::setup_interaction_ref_attributes ( ) [virtual]
```

Setup the [ExecutionControl](#) interaction Trick ref ATTRIBUTES.

This routine is used to perform and inline build of the Trick ref ATTRIBUTES for the [SpaceFOM](#) Mode Transition Request (MTR) interaction. This is used by federates, other than the Master, to request mode transitions.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 210 of file SpaceFOM/ExecutionControl.cpp.

References attrSpaceFOM\_\_MTRInteractionHandler, TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_G\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_LEVEL\_9\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ENCODING\_LITTLE\_ENDIAN, SpaceFOM::MTRInteractionHandler::get\_address\_of\_interaction\_mode(), TrickHLA::Parameter::get\_FOM\_name(), TrickHLA::Interaction::get\_FOM\_name(), TrickHLA::Interaction::get\_parameter\_count(), TrickHLA::Parameter::initialize(), TrickHLA::Interaction::initialize(), TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::ExecutionControlBase::manager, mtr\_interaction, mtr\_interaction\_handler, TrickHLA::Parameter::set\_debug\_level(), TrickHLA::Parameter::set\_encoding(), TrickHLA::Parameter::set\_FOM\_name(), TrickHLA::Interaction::set\_FOM\_name(), TrickHLA::Interaction::set\_handler(), SpaceFOM::MTRInteractionHandler::set\_name(), TrickHLA::Interaction::set\_parameter\_count(), TrickHLA::Interaction::set\_parameters(), TrickHLA::Interaction::set\_publish(), TrickHLA::Interaction::set\_subscribe(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

#### 7.17.3.44 setup\_interaction\_RTI\_handles()

void ExecutionControl::setup\_interaction\_RTI\_handles ( ) [virtual]

Setup the [ExecutionControl](#) interaction HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 348 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::manager](#), [mtr\\_interaction](#), and [TrickHLA::Manager::setup\\_interaction\\_RTI\\_handles\(\)](#).

#### 7.17.3.45 setup\_object\_ref\_attributes()

void ExecutionControl::setup\_object\_ref\_attributes ( ) [virtual]

Setup the [ExecutionControl](#) object Trick ref ATTRIBUTES.

This routine is used to perform and inline build of the Trick ref ATTRIBUTES for the [SpaceFOM](#) ExecutionControl.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 197 of file SpaceFOM/ExecutionControl.cpp.

#### 7.17.3.46 setup\_object\_RTI\_handles()

void ExecutionControl::setup\_object\_RTI\_handles ( ) [virtual]

Setup the [ExecutionControl](#) objects HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 330 of file SpaceFOM/ExecutionControl.cpp.

References [get\\_execution\\_configuration\(\)](#), [TrickHLA::ExecutionControlBase::manager](#), [TrickHLA::Manager::setup\\_object\\_RTI\\_handles\(\)](#), and [THLA\\_ENDL](#).

#### 7.17.3.47 shutdown()

void ExecutionControl::shutdown ( ) [virtual]

Execution control specific shutdown process.

**Trick Job Class:** *shutdown*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 1233 of file SpaceFOM/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::current\\_execution\\_control\\_mode](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::EXECUTION\\_CONTROL\\_SHUTDOWN](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::ExecutionControlBase::get\\_time\\_padding\(\)](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::ExecutionControlBase::should\\_print\(\)](#), [shutdown\\_mode\\_announce\(\)](#), [shutdown\\_mode\\_transition\(\)](#), and [THLA\\_NEWLINE](#).

#### 7.17.3.48 shutdown\_mode\_announce()

void ExecutionControl::shutdown\_mode\_announce ( ) [virtual]

Announce to the federation execution that a shutdown is occurring.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2073 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::clear\_mode\_transition\_requested(), TrickHLA::ExecutionControlBase::current\_execution\_control\_mode, TrickHLA::ExecutionControlBase::execution\_configuration, TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, TrickHLA::ExecutionControlBase::is\_master(), TrickHLA::Object::send\_init\_data(), and set\_next\_execution\_control\_mode().

Referenced by post\_multi\_phase\_init\_processes(), process\_mode\_transition\_request(), and shutdown().

#### 7.17.3.49 shutdown\_mode\_transition()

```
void ExecutionControl::shutdown_mode_transition ( ) [virtual]
```

The shutdown mode transition routine.

**Trick Job Class:** *shutdown*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2102 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::current\_execution\_control\_mode, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, TrickHLA::ExecutionControlBase::federate, TrickHLA::Federate::get\_RTI\_ambassador(), TrickHLA::ExecutionControlBase::is\_master(), and TrickHLA::SyncPntListBase::register\_sync\_pnt().

Referenced by shutdown().

#### 7.17.3.50 start\_federation\_save\_at\_scenario\_time()

```
void ExecutionControl::start_federation_save_at_scenario_time (
    double freeze_scenario_time,
    const char * file_name ) [virtual]
```

Start the Federation save at the specified scenario time.

##### Parameters

<i>freeze_scenario_time</i>	Scenario time to freeze.
<i>file_name</i>	Checkpoint file name.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 2584 of file SpaceFOM/ExecutionControl.cpp.

References THLA\_ENDL.

#### 7.17.3.51 subscribe()

```
void ExecutionControl::subscribe ( ) [virtual]
```

Subscribe to the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 454 of file SpaceFOM/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::execution\_configuration, TrickHLA::ExecutionControlBase::is\_master(), mtr\_interaction, TrickHLA::Interaction::subscribe\_to\_interaction(), and TrickHLA::Object::subscribe\_to\_object\_attributes().

#### 7.17.3.52 unpublish()

```
void ExecutionControl::unpublish ( ) [virtual]
```

Unpublish the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 437 of file SpaceFOM/ExecutionControl.cpp.

References `TrickHLA::ExecutionControlBase::execution_configuration`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::Interaction::is_publish()`, `mtr_interaction`, `TrickHLA::Object::unpublish_all_object_attributes()`, and `TrickHLA::Interaction::unpublish_interaction()`.

#### 7.17.3.53 `unsubscribe()`

`void ExecutionControl::unsubscribe ( ) [virtual]`

Unsubscribe the `ExecutionControl` objects and interactions.

Implements `TrickHLA::ExecutionControlBase`.

Definition at line 469 of file `SpaceFOM/ExecutionControl.cpp`.

References `TrickHLA::ExecutionControlBase::execution_configuration`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::Interaction::is_subscribe()`, `mtr_interaction`, `TrickHLA::Object::unsubscribe_all_object_attributes()`, and `TrickHLA::Interaction::unsubscribe_from_interaction()`.

#### 7.17.3.54 `wait_on_root_frame_discovered_synchronization()`

`void ExecutionControl::wait_on_root_frame_discovered_synchronization ( )`

Waits on synchronization of the `root_frame_discovered` synchronization point.

Calling this function will block until the all early joiner federates have discovered the root reference frame and achieved the `root_frame_discovered` synchronization point. **Trick Job Class:** *initialization*

Definition at line 1909 of file `SpaceFOM/ExecutionControl.cpp`.

References `TrickHLA::SyncPntListBase::achieve_and_wait_for_synchronization()`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::get_RTI_ambassador()`, `SpaceFOM::ROOT_FRAME_DISCOVERED_SYNC_POINT`, `TrickHLA::DebugHandler::should_print()`, and `THLA_NEWLINE`.

Referenced by `epoch_and_root_frame_discovery_process()`.

### 7.17.4 Friends And Related Function Documentation

#### 7.17.4.1 `init_attrSpaceFOM__ExecutionControl`

`void init_attrSpaceFOM__ExecutionControl ( ) [friend]`

#### 7.17.4.2 `InputProcessor`

`friend class InputProcessor [friend]`

Definition at line 63 of file `SpaceFOM/ExecutionControl.hh`.

### 7.17.5 Field Documentation

#### 7.17.5.1 `mtr_interaction`

`TrickHLA::Interaction* SpaceFOM::ExecutionControl::mtr_interaction [protected]`

**Units:** –

`SpaceFOM` Mode Transition Request (MTR) interaction.

Definition at line 276 of file `SpaceFOM/ExecutionControl.hh`.

Referenced by `publish()`, `receive_interaction()`, `setup_interaction_ref_attributes()`, `setup_interaction_RTI_handles()`, `subscribe()`, `unpublish()`, and `unsubscribe()`.

### 7.17.5.2 mtr\_interaction\_handler

`MTRInteractionHandler` SpaceFOM::ExecutionControl::mtr\_interaction\_handler [protected]

**Units:** –

`SpaceFOM` MTR interaction handler.

Definition at line 277 of file SpaceFOM/ExecutionControl.hh.

Referenced by `send_MTR_interaction()`, and `setup_interaction_ref_attributes()`.

### 7.17.5.3 pacing

`bool` SpaceFOM::ExecutionControl::pacing

**Units:** –

Is true when this federate is the "pacing".

(default: false)

Definition at line 70 of file SpaceFOM/ExecutionControl.hh.

Referenced by `is_pacing()`.

### 7.17.5.4 pending\_mtr

`MTREnum` SpaceFOM::ExecutionControl::pending\_mtr [protected]

**Units:** –

Pending Mode Transition Requested.

Definition at line 274 of file SpaceFOM/ExecutionControl.hh.

Referenced by `check_mode_transition_request()`, `post_multi_phase_init_processes()`, `process_mode_transition_request()`, `set_mode_request_from_mtr()`, and `set_pending_mtr()`.

### 7.17.5.5 root\_frame\_pub

`bool` SpaceFOM::ExecutionControl::root\_frame\_pub

**Units:** –

Is true when this federate is the "root reference frame publisher" federate for the Multiphase initialization process.

(default: false)

Definition at line 72 of file SpaceFOM/ExecutionControl.hh.

Referenced by `is_root_frame_publisher()`.

### 7.17.5.6 root\_ref\_frame

`RefFrameBase*` SpaceFOM::ExecutionControl::root\_ref\_frame

**Units:** –

Reference to the root reference frame object instance.

Definition at line 77 of file SpaceFOM/ExecutionControl.hh.

Referenced by `SpaceFOM::RefFrameBase::default_data()`, `epoch_and_root_frame_discovery_process()`, `pre_multi_phase_init_processes()`, `receive_init_root_ref_frame()`, `receive_root_ref_frame()`, and `send_root_ref_frame()`.

### 7.17.5.7 type

`const std::wstring` SpaceFOM::ExecutionControl::type = L"SpaceFOM" [static], [protected]

**Units:** –**ExecutionControl** type string.

Definition at line 272 of file SpaceFOM/ExecutionControl.hh.

Referenced by `get_type()`.

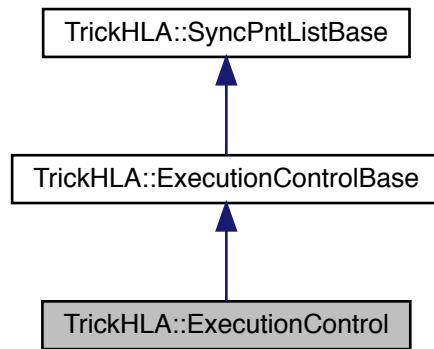
The documentation for this class was generated from the following files:

- [SpaceFOM/ExecutionControl.hh](#)
- [SpaceFOM/ExecutionControl.cpp](#)

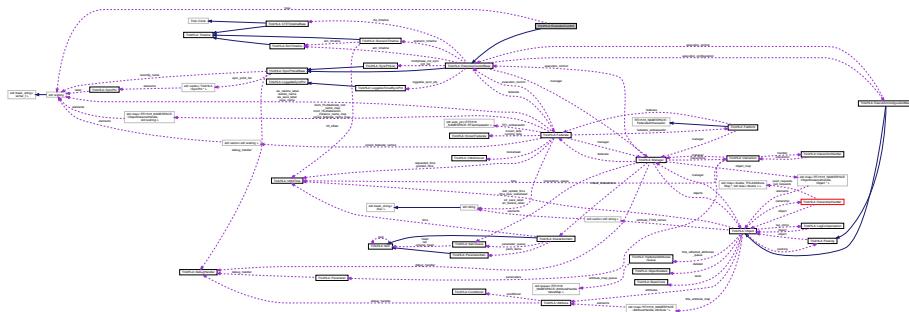
## 7.18 TrickHLA::ExecutionControl Class Reference

```
#include <ExecutionControl.hh>
```

Inheritance diagram for TrickHLA::ExecutionControl:



Collaboration diagram for TrickHLA::ExecutionControl:



### Public Member Functions

- `ExecutionControl ()`  
Default constructor for the `TrickHLA ExecutionControl` class.
- `ExecutionControl (ExecutionConfiguration &exec_config)`

*Initialization constructor for the `ExecutionControl` class.*

- virtual `~ExecutionControl ()`

*Destructor for the `TrickHLA ExecutionControl` class.*

- virtual const std::wstring & `get_type ()`

*Get the `ExecutionControl` type identification string.*

- virtual void `initialize ()`

*Execution Control initialization routine.*

- virtual void `join_federation_process ()`

*Join federation execution process.*

- virtual void `pre_multi_phase_init_processes ()`

*Process run before the multi-phase initialization begins.*

- virtual void `post_multi_phase_init_processes ()`

*Process run after the multi-phase initialization ends.*

- virtual void `shutdown ()`

*Execution control specific shutdown process.*

- virtual void `setup_object_ref_attributes ()`

- virtual void `setup_interaction_ref_attributes ()`

- virtual void `setup_object_RTI_handles ()`

- virtual void `setup_interaction_RTI_handles ()`

- virtual void `add_initialization_sync_points ()`

- virtual void `add_multiphase_init_sync_points ()`

- virtual void `announce_sync_point (RTI1516_NAMESPACE::RTIambassador &rti_ambassador, std::wstring const &label, RTI1516_USERDATA const &user_supplied_tag)`

*The RTI has announced the existence of a synchronization point.*

- virtual void `clear_multiphase_init_sync_points ()`

- virtual void `publish ()`

- virtual void `unpublish ()`

- virtual void `subscribe ()`

- virtual void `unsubscribe ()`

- bool `wait_on_init_data ()`

*Test to see if `ExecutionControl` needs to wait on initialization data.*

- bool `wait_on_init_sync_point ()`

*Test to see if `ExecutionControl` needs to wait on initialization synchronization point.*

- virtual void `receive_interaction (RTI1516_NAMESPACE::InteractionClassHandle const &theInteraction, RTI1516_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516_USERDATA const &theUserSuppliedTag, RTI1516_NAMESPACE::LogicalTime const &theTime, bool received_as_TSO)`

*Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.*

- virtual void `send_mode_transition_interaction (TrickHLA::ModeTransitionEnum requested_mode)`

*Send a mode transition request to the Master federate.*

- virtual bool `process_mode_interaction ()`

*Process a new mode interaction.*

- virtual void `set_next_execution_control_mode (TrickHLA::ExecutionControlEnum exec_control)`

*Sets the next `ExecutionControl` run mode.*

- virtual bool `process_execution_control_updates ()`

*Process changes from any received Execution Control Objects (ExCOs).*

- virtual bool `run_mode_transition ()`

*The run mode transition routine.*

- virtual void `freeze_mode_announce ()`

*Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.*

- `virtual bool freeze_mode_transition ()`

*The freeze mode transition routine.*

- `virtual void shutdown_mode_announce ()`

*Announce to the federation execution that a shutdown is occurring.*

- `virtual void shutdown_mode_transition ()`

*The shutdown mode transition routine.*

- `virtual void set_least_common_time_step (int64_t lcts)`

*Set the least common time step in microseconds for the federation.*

- `virtual void set_time_padding (double t)`

*Set the time-padding used to offset the go to run time.*

- `virtual void start_federation_save_at_scenario_time (double freeze_scenario_time, const char *file_name)`

*Start the Federation save at the specified scenario time.*

## Protected Member Functions

- `ExecutionConfiguration * get_execution_configuration ()`

*Return the relevant `TrickHLA::ExecutionConfiguration` object.*

## Static Protected Attributes

- `static const std::wstring type = L"Simple"`

*Units:* –

*ExecutionControl* type string.

## Private Member Functions

- `ExecutionControl (const ExecutionControl &rhs)`

*Copy constructor for `ExecutionControl` class.*

- `ExecutionControl & operator= (const ExecutionControl &rhs)`

*Assignment operator for `ExecutionControl` class.*

## Friends

- `class InputProcessor`
- `void init_attrTrickHLA__ExecutionControl ()`

## Additional Inherited Members

### 7.18.1 Detailed Description

Definition at line 51 of file TrickHLA/ExecutionControl.hh.

### 7.18.2 Constructor & Destructor Documentation

### 7.18.2.1 ExecutionControl() [1/3]

```
ExecutionControl::ExecutionControl ( )  
Default constructor for the TrickHLA ExecutionControl class.  
Trick Job Class: initialization  
Definition at line 76 of file TrickHLA/ExecutionControl.cpp.
```

### 7.18.2.2 ExecutionControl() [2/3]

```
ExecutionControl::ExecutionControl (   
    ExecutionConfiguration & exec_config ) [explicit]  
Initialization constructor for the ExecutionControl class.
```

#### Parameters

<code>exec_config</code>	The associated <a href="#">ExecutionControl</a> class instance.
--------------------------	---

#### **Trick Job Class:** *initialization*

Definition at line 84 of file TrickHLA/ExecutionControl.cpp.

### 7.18.2.3 ~ExecutionControl()

```
ExecutionControl::~ExecutionControl ( ) [virtual]  
Destructor for the TrickHLA ExecutionControl class.  
Trick Job Class: shutdown  
Definition at line 94 of file TrickHLA/ExecutionControl.cpp.  
References TrickHLA::ExecutionControlBase::clear\_mode\_values\(\).
```

### 7.18.2.4 ExecutionControl() [3/3]

```
TrickHLA::ExecutionControl::ExecutionControl (   
    const ExecutionControl & rhs ) [private]  
Copy constructor for ExecutionControl class.  
This constructor is private to prevent inadvertent copies.
```

## 7.18.3 Member Function Documentation

### 7.18.3.1 add\_initialization\_sync\_points()

```
void ExecutionControl::add_initialization_sync_points ( ) [virtual]  
Add initialization synchronization points to regulate startup.  
Trick Job Class: initialization  
Implements TrickHLA::ExecutionControlBase.  
Definition at line 288 of file TrickHLA/ExecutionControl.cpp.  
References add\_multiphase\_init\_sync\_points\(\).
```

### 7.18.3.2 add\_multiphase\_init\_sync\_points()

```
void ExecutionControl::add_multiphase_init_sync_points ( ) [virtual]
```

Add initialization synchronization points to regulate startup.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 303 of file TrickHLA/ExecutionControl.cpp.

Referenced by `add_initialization_sync_points()`.

#### 7.18.3.3 `announce_sync_point()`

```
void ExecutionControl::announce_sync_point (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    std::wstring const & label,
    RTI1516_USERDATA const & user_supplied_tag ) [virtual]
```

The RTI has announced the existence of a synchronization point.

**Parameters**

<i>rti_ambassador</i>	Reference to the HLA RTI Ambassador instance.
<i>label</i>	Sync-point label.
<i>user_supplied_tag</i>	Use supplied tag.

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 308 of file TrickHLA/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::announce\\_sync\\_point\(\)](#).

#### 7.18.3.4 `clear_multiphase_init_sync_points()`

```
void ExecutionControl::clear_multiphase_init_sync_points ( ) [virtual]
```

Clear any remaining multiphase initialization synchronization points that have not been achieved and wait for the federation to be synchronized on it.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 325 of file TrickHLA/ExecutionControl.cpp.

References [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), and [THLA\\_ENDL](#).

#### 7.18.3.5 `freeze_mode_announce()`

```
void ExecutionControl::freeze_mode_announce ( ) [virtual]
```

Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 405 of file TrickHLA/ExecutionControl.cpp.

#### 7.18.3.6 `freeze_mode_transition()`

```
bool ExecutionControl::freeze_mode_transition ( ) [virtual]
```

The freeze mode transition routine.

**Returns**

Currently always returns False.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 411 of file TrickHLA/ExecutionControl.cpp.

#### 7.18.3.7 `get_execution_configuration()`

```
ExecutionConfiguration * ExecutionControl::get_execution_configuration ( ) [protected], [virtual]
```

Return the relevant [TrickHLA::ExecutionConfiguration](#) object.

**Returns**

Pointer to the relevant [TrickHLA::ExecutionConfiguration](#) object.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 432 of file TrickHLA/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::execution\\_configuration](#), and [THLA\\_ENDL](#).

Referenced by [setup\\_object\\_RTI\\_handles\(\)](#).

#### 7.18.3.8 `get_type()`

```
virtual const std::wstring& TrickHLA::ExecutionControl::get_type ( ) [inline], [virtual]
```

Get the [ExecutionControl](#) type identification string.

**Returns**

A constant reference to the type identification string.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 76 of file TrickHLA/ExecutionControl.hh.

References [type](#).

#### 7.18.3.9 `initialize()`

```
void ExecutionControl::initialize ( ) [virtual]
```

Execution Control initialization routine.

This routine will set a lot of the data in the [TrickHLA::Federate](#) that is required for this execution control scheme. This should greatly simplify input files and reduce input file setting errors.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 107 of file TrickHLA/ExecutionControl.cpp.

References [TrickHLA::Federate::enable\\_known\\_feds](#), [TrickHLA::ExecutionControlBase::federate](#), [TrickHLA::Federate::known\\_feds\\_count](#), and [TrickHLA::ExecutionControlBase::use\\_preset\\_master](#).

#### 7.18.3.10 `join_federation_process()`

```
void ExecutionControl::join_federation_process ( ) [virtual]
```

Join federation execution process.

This routine implements the [TrickHLA](#) Join Federation Process described.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 125 of file TrickHLA/ExecutionControl.cpp.

References [TrickHLA::ExecutionControlBase::join\\_federation\\_process\(\)](#).

### 7.18.3.11 operator=( )

```
ExecutionControl& TrickHLA::ExecutionControl::operator= (
    const ExecutionControl & rhs ) [private]
```

Assignment operator for `ExecutionControl` class.

This assignment operator is private to prevent inadvertent copies.

### 7.18.3.12 post\_multi\_phase\_init\_processes()

```
void ExecutionControl::post_multi_phase_init_processes ( ) [virtual]
```

Process run after the multi-phase initialization ends.

This routine implements the `TrickHLA` post multi-phase initialization process.

**Trick Job Class:** *initialization*

Implements `TrickHLA::ExecutionControlBase`.

Definition at line 220 of file `TrickHLA/ExecutionControl.cpp`.

References `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::Federate::setup_time_management()`, and `TrickHLA::Federate::time_advance_request_to_GALT()`.

### 7.18.3.13 pre\_multi\_phase\_init\_processes()

```
void ExecutionControl::pre_multi_phase_init_processes ( ) [virtual]
```

Process run before the multi-phase initialization begins.

This routine implements the `TrickHLA` pre multi-phase initialization process.

**Trick Job Class:** *initialization*

Implements `TrickHLA::ExecutionControlBase`.

Definition at line 139 of file `TrickHLA/ExecutionControl.cpp`.

References `TrickHLA::Federate::create_and_join_federation()`, `TrickHLA::Federate::create_RTI_ambassador_and_connect()`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_G_SOURCE_MANAGER`, `TrickHLA::Federate::destroy_orphaned_federation()`, `TrickHLA::Federate::enable_async_delivery()`, `TrickHLA::ExecutionControlBase::execution_configuration`, `TrickHLA::ExecutionControlBase::federate`, `TrickHLA::ExecutionControlBase::get_manager()`, `TrickHLA::Federate::initialize_MOM_handles()`, `TrickHLA::Federate::is_federation_created_by_federate()`, `TrickHLA::ExecutionControlBase::is_master()`, `TrickHLA::ExecutionControlBase::is_master_preset()`, `TrickHLA::Manager::publish_and_subscribe()`, `TrickHLA::Manager::register_objects_with_RTI()`, `TrickHLA::Manager::reserve_object_names_with_RTI()`, `TrickHLA::ExecutionConfigurationBase::set_master()`, `TrickHLA::ExecutionControlBase::set_master()`, `TrickHLA::Manager::setup_all_ref_attributes()`, `TrickHLA::Manager::setup_all_RTI_handles()`, `TrickHLA::Manager::setup_preferred_order_with_RTI()`, `TrickHLA::DebugHandler::should_print()`, `THLA_NEWLINE`, and `TrickHLA::Manager::wait_on_reservation_of_object_names()`.

### 7.18.3.14 process\_execution\_control\_updates()

```
bool ExecutionControl::process_execution_control_updates ( ) [virtual]
```

Process changes from any received Execution Control Objects (ExCOs).

**Returns**

True if mode change occurred.

**Assumptions and Limitations:**

- Called from the ExCO unpack routine.

**Trick Job Class:** *scheduled*

Implements `TrickHLA::ExecutionControlBase`.

Definition at line 392 of file `TrickHLA/ExecutionControl.cpp`.

**7.18.3.15 process\_mode\_interaction()**

```
virtual bool TrickHLA::ExecutionControl::process_mode_interaction ( ) [inline], [virtual]
Process a new mode interaction.
```

**Returns**

True if new mode interaction is successfully processed.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 159 of file TrickHLA/ExecutionControl.hh.

**7.18.3.16 publish()**

```
void ExecutionControl::publish ( ) [virtual]
```

Publish the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 338 of file TrickHLA/ExecutionControl.cpp.

**7.18.3.17 receive\_interaction()**

```
void ExecutionControl::receive_interaction (
    RTI1516_NAMESPACE::InteractionClassHandle const & theInteraction,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_NAMESPACE::UserData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    bool received_as_TSO ) [virtual]
```

Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.

**Parameters**

<i>theInteraction</i>	Interaction handle.
<i>theParameterValues</i>	Parameter values.
<i>theUserSuppliedTag</i>	Users tag.
<i>theTime</i>	HLA time for the interaction.
<i>received_as_TSO</i>	True if interaction was received by RTI as TSO.

**Trick Job Class: *scheduled***

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 361 of file TrickHLA/ExecutionControl.cpp.

**7.18.3.18 run\_mode\_transition()**

```
bool ExecutionControl::run_mode_transition ( ) [virtual]
```

The run mode transition routine.

**Returns**

Currently always returns True.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 399 of file TrickHLA/ExecutionControl.cpp.

### 7.18.3.19 send\_mode\_transition\_interaction()

```
void ExecutionControl::send_mode_transition_interaction (
    TrickHLA::ModeTransitionEnum requested_mode ) [virtual]
```

Send a mode transition request to the Master federate.

#### Parameters

<i>requested_mode</i>	Requested mode.
-----------------------	-----------------

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 373 of file [TrickHLA/ExecutionControl.cpp](#).

### 7.18.3.20 set\_least\_common\_time\_step()

```
void ExecutionControl::set_least_common_time_step (
    int64_t lcts ) [virtual]
```

Set the least common time step in microseconds for the federation.

#### Parameters

<i>lcts</i>	Least Common Time Step time in microseconds.
-------------	--

WARNING: Only the Master federate should ever set this.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 450 of file [TrickHLA/ExecutionControl.cpp](#).

References [TrickHLA::ExecutionControlBase::execution\\_configuration](#), [TrickHLA::ExecutionControlBase::is\\_master\(\)](#), [TrickHLA::ExecutionControlBase::least\\_common\\_time\\_step](#), and [THLA\\_ENDL](#).

### 7.18.3.21 set\_next\_execution\_control\_mode()

```
void ExecutionControl::set_next_execution_control_mode (
    TrickHLA::ExecutionControlEnum exec_control ) [virtual]
```

Sets the next [ExecutionControl](#) run mode.

#### Parameters

<i>exec_control</i>	Next <a href="#">ExecutionControl</a> run mode.
---------------------	---

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 379 of file [TrickHLA/ExecutionControl.cpp](#).

### 7.18.3.22 set\_time\_padding()

```
void ExecutionControl::set_time_padding (
    double t ) [virtual]
```

Set the time-padding used to offset the go to run time.

#### Parameters

<i>t</i>	Time in seconds to pad for time based mode transitions.
----------	---

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 471 of file TrickHLA/ExecutionControl.cpp.

References TrickHLA::ExecutionControlBase::least\_common\_time\_step, THLA\_ENDL, THLA\_NEWLINE, and TrickHLA::ExecutionControlBase::time\_padding.

#### 7.18.3.23 setup\_interaction\_ref\_attributes()

```
void ExecutionControl::setup_interaction_ref_attributes ( ) [virtual]
```

Setup the [ExecutionControl](#) interaction Trick ref ATTRIBUTES.

This routine is used to perform and inline build of the Trick ref ATTRIBUTES for any mode transition interactions. This implementation does not have any.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 254 of file TrickHLA/ExecutionControl.cpp.

#### 7.18.3.24 setup\_interaction\_RTI\_handles()

```
void ExecutionControl::setup_interaction_RTI_handles ( ) [virtual]
```

Setup the [ExecutionControl](#) interaction HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 280 of file TrickHLA/ExecutionControl.cpp.

#### 7.18.3.25 setup\_object\_ref\_attributes()

```
void ExecutionControl::setup_object_ref_attributes ( ) [virtual]
```

Setup the [ExecutionControl](#) object Trick ref ATTRIBUTES.

This routine is used to perform and inline build of the Trick ref ATTRIBUTES.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 242 of file TrickHLA/ExecutionControl.cpp.

#### 7.18.3.26 setup\_object\_RTI\_handles()

```
void ExecutionControl::setup_object_RTI_handles ( ) [virtual]
```

Setup the [ExecutionControl](#) objects HLA RTI handles.

**Trick Job Class:** *initialization*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 262 of file TrickHLA/ExecutionControl.cpp.

References `get_execution_configuration()`, [TrickHLA::ExecutionControlBase::manager](#), [TrickHLA::Manager::setup\\_object\\_RTI\\_handles\(\)](#), and `THLA_ENDL`.

#### 7.18.3.27 shutdown()

```
void ExecutionControl::shutdown ( ) [virtual]
```

Execution control specific shutdown process.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 231 of file TrickHLA/ExecutionControl.cpp.

### 7.18.3.28 shutdown\_mode\_announce()

```
void ExecutionControl::shutdown_mode_announce ( ) [virtual]
```

Announce to the federation execution that a shutdown is occurring.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 417 of file TrickHLA/ExecutionControl.cpp.

### 7.18.3.29 shutdown\_mode\_transition()

```
void ExecutionControl::shutdown_mode_transition ( ) [virtual]
```

The shutdown mode transition routine.

**Trick Job Class:** *shutdown*

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 426 of file TrickHLA/ExecutionControl.cpp.

### 7.18.3.30 start\_federation\_save\_at\_scenario\_time()

```
void ExecutionControl::start_federation_save_at_scenario_time (
    double freeze_scenario_time,
    const char * file_name ) [virtual]
```

Start the Federation save at the specified scenario time.

#### Parameters

<i>freeze_scenario_time</i>	Scenario time to freeze.
<i>file_name</i>	Checkpoint file name.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 508 of file TrickHLA/ExecutionControl.cpp.

References THLA\_ENDL.

### 7.18.3.31 subscribe()

```
void ExecutionControl::subscribe ( ) [virtual]
```

Subscribe to the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 348 of file TrickHLA/ExecutionControl.cpp.

### 7.18.3.32 unpublish()

```
void ExecutionControl::unpublish ( ) [virtual]
```

Unpublish the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 343 of file TrickHLA/ExecutionControl.cpp.

### 7.18.3.33 unsubscribe()

```
void ExecutionControl::unsubscribe ( ) [virtual]
```

Unsubscribe the [ExecutionControl](#) objects and interactions.

Implements [TrickHLA::ExecutionControlBase](#).

Definition at line 353 of file TrickHLA/ExecutionControl.cpp.

#### 7.18.3.34 wait\_on\_init\_data()

```
bool TrickHLA::ExecutionControl::wait_on_init_data ( ) [inline], [virtual]
```

Test to see if [ExecutionControl](#) needs to wait on initialization data.

Most [ExecutionControl](#) approaches require that we wait for the required initialization data. Currently, only the 'Simple' scheme does not.

##### Returns

True if [ExecutionControl](#) needs to wait on the initialization data.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 129 of file TrickHLA/ExecutionControl.hh.

#### 7.18.3.35 wait\_on\_init\_sync\_point()

```
bool TrickHLA::ExecutionControl::wait_on_init_sync_point ( ) [inline], [virtual]
```

Test to see if [ExecutionControl](#) needs to wait on initialization synchronization point.

Most [ExecutionControl](#) approaches require that we wait for specific initialization synchronization points in specific orders. Currently, only the 'Simple' and 'DIS' scheme do not.

##### Returns

True if [ExecutionControl](#) needs to wait on the initialization synchronization points.

Reimplemented from [TrickHLA::ExecutionControlBase](#).

Definition at line 136 of file TrickHLA/ExecutionControl.hh.

### 7.18.4 Friends And Related Function Documentation

#### 7.18.4.1 init\_attrTrickHLA\_\_ExecutionControl

```
void init_attrTrickHLA__ExecutionControl ( ) [friend]
```

#### 7.18.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 58 of file TrickHLA/ExecutionControl.hh.

### 7.18.5 Field Documentation

#### 7.18.5.1 type

```
const std::wstring TrickHLA::ExecutionControl::type = L"Simple" [static], [protected]
```

##### Units: -

[ExecutionControl](#) type string.

Definition at line 200 of file TrickHLA/ExecutionControl.hh.

Referenced by [get\\_type\(\)](#).

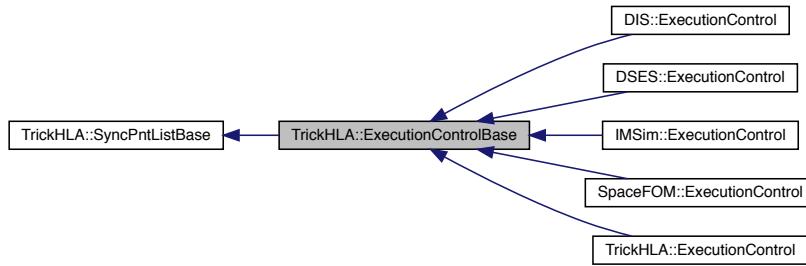
The documentation for this class was generated from the following files:

- [TrickHLA/ExecutionControl.hh](#)
- [TrickHLA/ExecutionControl.cpp](#)

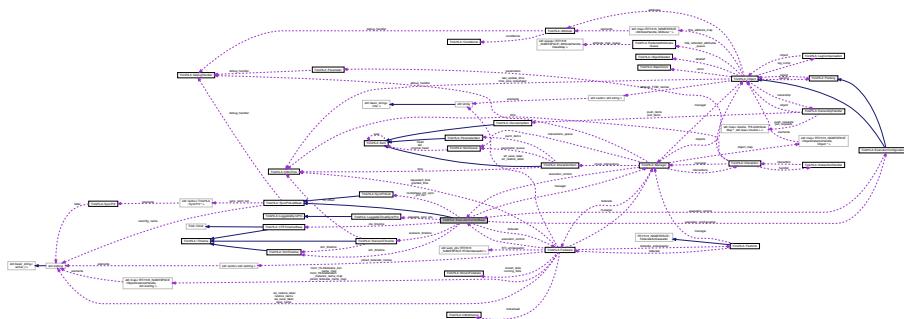
## 7.19 TrickHLA::ExecutionControlBase Class Reference

```
#include <ExecutionControlBase.hh>
```

Inheritance diagram for TrickHLA::ExecutionControlBase:



Collaboration diagram for TrickHLA::ExecutionControlBase:



### Public Member Functions

- [ExecutionControlBase \(\)](#)  
*Default constructor for the `ExecutionControlBase` class.*
- [ExecutionControlBase \(ExecutionConfigurationBase &exec\\_config\)](#)  
*Initialization constructor for the `ExecutionControlBase` class.*
- [virtual ~ExecutionControlBase \(\)=0](#)  
*Destructor for the `ExecutionControlBase` class.*
- [virtual void setup \(TrickHLA::Federate &federate, TrickHLA::Manager &manager, TrickHLA::ExecutionConfigurationBase &exec\\_config\)](#)  
*Setup the federate wide references in the `ExecutionControl` class instance.*
- [virtual void setup \(TrickHLA::Federate &federate, TrickHLA::Manager &manager\)](#)  
*Setup the federate wide references in the `ExecutionControl` class instance.*
- [virtual void initialize \(\)](#)  
*Initialize the `TrickHLA::ExecutionControlBase` object instance.*

- virtual void [join\\_federation\\_process](#) ()
 

*Join federation execution process.*
- virtual void [pre\\_multi\\_phase\\_init\\_processes](#) ()=0
 

*Processes run before the multi-phase initialization begins.*
- virtual void [post\\_multi\\_phase\\_init\\_processes](#) ()=0
 

*Processes run after the multi-phase initialization ends.*
- virtual void [shutdown](#) ()=0
 

*Execution control specific shutdown process.*
- virtual const std::wstring & [get\\_type](#) ()=0
- virtual void [setup\\_object\\_ref\\_attributes](#) ()=0
- virtual void [setup\\_interaction\\_ref\\_attributes](#) ()=0
- virtual void [setup\\_object\\_RTI\\_handles](#) ()=0
- virtual void [setup\\_interaction\\_RTI\\_handles](#) ()=0
- virtual bool [object\\_instance\\_name\\_reservation\\_succeeded](#) (std::wstring const &obj\_instance\_name)
 

*The object instance name reservation succeeded for the given name.*
- virtual bool [object\\_instance\\_name\\_reservation\\_failed](#) (std::wstring const &obj\_instance\_name)
 

*The object instance name reservation failed for the given name.*
- virtual void [register\\_objects\\_with\\_RTI](#) ()
 

*Add a [TrickHLA::Object](#) to the manager object map.*
- virtual void [add\\_object\\_to\\_map](#) (Object \*object)
 

*Add a [TrickHLA::Object](#) to the manager object map.*
- virtual void [register\\_interactions\\_with\\_RTI](#) ()
 

*Add a [TrickHLA::Interaction](#) to the manager interaction map.*
- virtual void [wait\\_for\\_sync\\_point\\_announce](#) (std::wstring const &sync\_pnt\_label)
 

*Wait for a specified synchronization point label to be announced.*
- virtual void [add\\_initialization\\_sync\\_points](#) ()=0
- virtual void [add\\_multiphase\\_init\\_sync\\_points](#) ()
 

*Clear any remaining multiphase initialization synchronization points that have not been achieved and wait for the federation to be synchronized on it.*
- void [achieve\\_all\\_multiphase\\_init\\_sync\\_pnts](#) (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador) throw ( RTI1516\_NAMESPACE::SynchronizationPointLabelNotAnnounced, RTI1516\_NAMESPACE::FederateNotExecutionMember, RTI1516\_NAMESPACE::SaveInProgress, RTI1516\_NAMESPACE::RestoreInProgress, RTI1516\_NAMESPACE::NotConnected, RTI1516\_NAMESPACE::RTIinternalError )
 

*Achieve all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not one of the predefined [ExecutionControl](#) synchronization points.*
- void [wait\\_for\\_all\\_multiphase\\_init\\_sync\\_pnts](#) ()
 

*Wait for all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not one of the predefined [ExecutionControl](#) synchronization points.*
- virtual void [publish](#) ()=0
- virtual void [unpublish](#) ()=0
- virtual void [subscribe](#) ()=0
- virtual void [unsubscribe](#) ()=0
- virtual void [send\\_execution\\_configuration](#) ()
 

*Send the [ExecutionConfiguration](#) data if we are the master federate.*
- virtual void [receive\\_execution\\_configuration](#) ()
 

*Receive the [ExecutionConfiguration](#) data from the master federate.*
- virtual void [send\\_requested\\_data](#) (double current\_time, double job\_cycle\_time)
 

*Send the attribute value requested data to the remote federates.*
- virtual void [receive\\_cyclic\\_data](#) (double current\_time)
 

*Handle the received cyclic data.*

- virtual void `provide_attribute_update` (RTI1516\_NAMESPACE::ObjectInstanceHandle const &theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes)
 

*Requesting an attribute value update for the given object instance and attributes.*
- virtual `Object * get_trickhla_object` (std::wstring const &obj\_instance\_name)
 

*Gets the `TrickHLA Object` for the specified RTI `Object` Instance Name.*
- virtual `Object * get_unregistered_object` (RTI1516\_NAMESPACE::ObjectClassHandle const &theObjectClass, std::wstring const &theObjectName)
 

*Returns the first object that matches the specified Object-Class, object instance name, and is not registered, i.e. the instance ID == 0.*
- virtual `Object * get_unregistered_remote_object` (RTI1516\_NAMESPACE::ObjectClassHandle const &theObjectClass)
 

*Returns the first object that is remotely owned, has the same Object-Class, is not registered, and does not have an `Object` Instance Name associated with it.*
- virtual bool `mark_object_as_deleted_from_federation` (RTI1516\_NAMESPACE::ObjectInstanceHandle const &instance\_id)
 

*Identifies the object as deleted from the RTI.*
- virtual void `process_deleted_objects` ()
 

*Scheduled method used as a callback to identify if any objects were deleted from the RTI.*
- virtual bool `wait_on_init_data` ()
 

*Test to see if `ExecutionControl` needs to wait on initialization data.*
- virtual bool `wait_on_init_sync_point` ()
 

*Test to see if `ExecutionControl` needs to wait on initialization synchronization point.*
- virtual void `receive_interaction` (RTI1516\_NAMESPACE::InteractionClassHandle const &theInteraction, RTI1516\_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516\_NAMESPACE::Userdata const &theUserSuppliedTag, RTI1516\_NAMESPACE::LogicalTime const &theTime, bool received\_as\_TSO)=0
 

*Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.*
- virtual void `send_mode_transition_interaction` (ModeTransitionEnum requested\_mode)=0
 

*Send a mode transition request to the Master federate.*
- virtual bool `process_mode_interaction` ()=0
 

*Process a new mode interaction.*
- virtual void `set_next_execution_control_mode` (ExecutionControlEnum exec\_control)=0
 

*Sets the next `ExecutionControl` run mode.*
- virtual bool `process_execution_control_updates` ()=0
 

*Process changes from any received Execution Control Objects (ExCOs).*
- void `set_scenario_timeline` (ScenarioTimeline \*timeline)
 

*Set the Scenario Timeline.*
- bool `does_scenario_timeline_exist` () const
 

*Check to see if the Scenario Timeline exists.*
- bool `does_sim_timeline_exist` () const
 

*Check to see if the Simulation Timeline exists.*
- bool `does_cte_timeline_exist` () const
 

*Check to see if the CTE Timeline exists.*
- double `get_sim_time` ()
 

*Get the current simulation time from Simulation Timeline.*
- double `get_cte_time` ()
 

*Get the current Central Timing Equipment time from CTE Timeline.*
- double `get_scenario_time` ()
 

*Get the current scenario time from Scenario Timeline.*

- double `convert_scenario_time_to_sim_time` (double scenario\_time)  
*Convert the a given scenario time into simulation time.*
- double `convert_sim_time_to_scenario_time` (double sim\_time)  
*Convert the a given simulation time into scenario time.*
- virtual void `clear_mode_values` ()  
*Clear the Mode Transition Request flag, the requested execution mode, and the current execution mode.*
- virtual bool `run_mode_transition` ()=0  
*The run mode transition routine.*
- virtual void `freeze_mode_announce` ()=0  
*Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.*
- virtual bool `freeze_mode_transition` ()=0  
*The freeze mode transition routine.*
- virtual void `shutdown_mode_announce` ()=0  
*Announce to the federation execution that a shutdown is occurring.*
- virtual void `shutdown_mode_transition` ()=0  
*The shutdown mode transition routine.*
- virtual bool `check_for_shutdown` ()  
*Checks to see if shutdown has been commanded.*
- virtual bool `check_for_shutdown_with_termination` ()  
*Checks to see if shutdown has been commanded and, if so, terminates the simulation.*
- virtual bool `is_initializing` ()  
*Is the federate execution in initialization.*
- virtual bool `is_running` ()  
*Is the federate execution in initialization.*
- virtual bool `is_in_freeze` ()  
*Is the federate execution in initialization.*
- virtual bool `is_in_restart` ()  
*Is the federate execution in initialization.*
- virtual bool `is_in_reconfig` ()  
*Is the federate execution in initialization.*
- virtual bool `is_shutdown` ()  
*Is the federate execution in initialization.*
- virtual void `freeze_init` ()  
*Routine to handle going from run to freeze.*
- virtual void `enter_freeze` ()  
*Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.*
- virtual bool `check_freeze_exit` ()  
*Check for exit from freeze.*
- virtual void `exit_freeze` ()  
*Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.*
- virtual void `un_freeze` ()  
*Routine to handle `ExecutionControl` specific action needed to un-freeze.*
- virtual void `check_pause` (const double check\_pause\_delta)  
*Check if we hit a pause sync point and need to go to freeze.*
- virtual void `check_pause_at_init` (const double check\_pause\_delta)  
*Checking if we started in freeze.*
- virtual void `set_mode_transition_requested` ()

- virtual void `clear_mode_transition_requested ()`

*Set the mode transition requested flag.*
- virtual bool `is_mode_transition_requested ()`

*Clear the mode transition requested flag.*
- virtual bool `is_master_preset () const`

*Determine if a mode transition has been requested.*
- virtual bool `is_master () const`

*Query if there is a preset Master.*
- virtual void `set_master (bool master_flag)`

*Set this as the Master federate.*
- virtual bool `is_master () const`

*Query if this is the Master federate.*
- virtual bool `is_lateJoiner ()`

*Determine if this federate is a late joining federate.*
- virtual bool `is_lateJoiner_determined ()`

*Check if we have determine if this federate is a late joining federate.*
- virtual `ExecutionControlEnum get_requested_execution_control_mode ()`

*Get the currently requested execution mode.*
- virtual void `set_requested_execution_control_mode (ExecutionControlEnum mode)`

*Set the currently requested execution mode.*
- virtual void `set_requested_execution_control_mode (int16_t mode)`

*Set the currently requested execution mode.*
- virtual `ExecutionControlEnum get_current_execution_control_mode ()`

*Get the current execution mode.*
- virtual void `set_current_execution_control_mode (ExecutionControlEnum mode)`

*Set the current execution control mode.*
- virtual void `start_federation_save_at_scenario_time (double freeze_scenario_time, const char *file_name)=0`

*Start the Federation save at the specified scenario time.*
- virtual void `setup_checkpoint ()`

*Setup the checkpoint data structures.*
- virtual void `set_federate (TrickHLA::Federate *fed)`

*Set the reference to the associated `TrickHLA::Federate`.*
- virtual `TrickHLA::Federate * get_federate ()`

*Get the reference to the associated `TrickHLA::Federate`.*
- virtual `TrickHLA::Manager * get_manager ()`

*Get the reference to the associated `TrickHLA::Manager`.*
- virtual void `set_execution_configuration (ExecutionConfigurationBase *exec_config)`

*Get the reference to the associated `TrickHLA::ExecutionConfigurationBase` object.*
- virtual `ExecutionConfigurationBase * get_execution_configuration ()`

*Get the reference to the associated `TrickHLA::ExecutionConfigurationBase` object.*
- virtual void `remove_execution_configuration ()`

*Remove the `ExecutionConfiguration` instance from the federation execution.*
- virtual bool `is_execution_configuration_used ()`

*Test is an execution configuration object is used.*
- virtual bool `does_init_complete_sync_point_exist () const`

*Query if the 'initialization\_completed' sync-point exists.*
- virtual void `set_least_common_time_step (int64_t lcts)`

*Set the least common time step in microseconds for the federation.*

- virtual int64\_t [get\\_least\\_common\\_time\\_step \(\)](#)  
*Get the value of the least common time step.*
- virtual void [set\\_time\\_padding \(double t\)](#)  
*Set the time-padding used to offset the go to run time.*
- virtual double [get\\_time\\_padding \(\)](#)  
*Get the time-padding used to offset the go to run time.*
- virtual double [get\\_simulation\\_freeze\\_time \(\)](#)  
*Get the Federation Execution simulation time for freeze.*
- virtual void [set\\_simulation\\_freeze\\_time \(double freeze\\_time\)](#)  
*Set the Federation Execution simulation time for freeze.*
- virtual double [get\\_scenario\\_freeze\\_time \(\)](#)  
*Get the Federation Execution scenario time for freeze.*
- virtual void [set\\_scenario\\_freeze\\_time \(double freeze\\_time\)](#)  
*Set the Federation Execution scenario time for freeze.*
- virtual bool [is\\_save\\_and\\_restore\\_supported \(\)](#)
- virtual bool [is\\_save\\_initiated \(\)](#)  
*Checks if Save has been initiated by this [ExecutionControl](#) method.*
- virtual bool [perform\\_save \(\)](#)  
*Federates that did not announce the save, perform a save.*
- virtual void [convert\\_loggable\\_sync\\_pts \(\)](#)  
*Converts HLA sync points into something Trick can save in a checkpoint.*
- virtual void [reinstate\\_logged\\_sync\\_pts \(\)](#)  
*Converts checkpointed sync points into HLA sync points.*
- virtual void [set\\_wait\\_sleep \(unsigned int t\)](#)  
*Set the time interval in microseconds a sleep loop timeout.*
- virtual void [set\\_wait\\_timeout \(unsigned int t\)](#)  
*Set the time interval in microseconds a wait loop timeout.*
- bool [should\\_print \(const DebugLevelEnum &level, const DebugSourceEnum &code\) const](#)  
*Determine if the verbose debug comments should be printed to the console.*

## Data Fields

- ScenarioTimeline \* [scenario\\_timeline](#)  
**Data I/O:** \*\*  
*The scenario timeline.*
- SimTimeline \* [sim\\_timeline](#)  
**Data I/O:** \*\*  
*The simulation timeline.*
- CTETimelineBase \* [cte\\_timeline](#)  
**Data I/O:** \*\*  
*The Central Timing Equipment (CTE) timeline.*
- bool [use\\_preset\\_master](#)  
**Units:** –  
*Set to true to force the use of the preset value for the "master" flag.*
- bool [master](#)  
**Units:** –  
*Is true when this federate is the "master" federate for the Multiphase initialization process.*
- char \* [multiphase\\_init\\_sync\\_points](#)  
**Units:** –  
*Comma-separated list of multi-phase initialization sync-points.*

## Protected Attributes

- double `time_padding`

**Units:** s  
*Time in seconds to add to the go-to-run time.*
- int64\_t `least_common_time_step`

**Units:** –  
*A 64 bit integer time that represents microseconds for the least common value of all the time step values in the federation execution (LCTS).*
- `ExecutionConfigurationBase * execution_configuration`

**Units:** –  
*Associates `TrickHLA::ExecutionConfigurationBase` class object instance.*
- `SyncPntList multiphase_init_sync_pnt_list`

**Units:** –  
*Synchronization points used for multi-phase initialization control.*
- bool `init_complete_sp_exists`

**Units:** –  
*Internal flag, for Initialization Complete Sync-Point exists.*
- bool `mode_transition_requested`

**Units:** –  
*Flag to indicate a mode transition has been requested.*
- `ExecutionControlEnum requested_execution_control_mode`

**Units:** –  
*The latest mode transition requested.*
- `ExecutionControlEnum current_execution_control_mode`

**Units:** –  
*Current SRFOM federate current execution mode.*
- double `next_mode_scenario_time`

**Units:** s  
*Scenario time for mode transition.*
- double `next_mode_cte_time`

**Units:** s  
*CTE time for next managed mode transition.*
- double `simulation_freeze_time`

**Units:** s  
*Trick simulation time for freeze.*
- double `scenario_freeze_time`

**Units:** s  
*Federation execution scenario time for freeze.*
- bool `late_joiner`

**Units:** –  
*Flag that this federate is a late joiner.*
- bool `late_joiner_determined`

**Units:** –  
*Flag for late joiner determination.*
- unsigned int `wait_sleep`

**Units:** us  
*Wait loop sleep times.*
- unsigned int `wait_timeout`

**Units:** us  
*Wait loop timeout.*

- `TrickHLA::Federate * federate`  
`Data I/O: **`  
`Associated federate.`
- `TrickHLA::Manager * manager`  
`Data I/O: **`  
`Associated manager.`
- `size_t logged_sync_pts_count`  
`Units: -`  
`number of logged sync pts`
- `LoggableTimedSyncPnt * loggable_sync_pts`  
`Units: -`  
`converted Sync Point data that gets checkpointed`

## Private Member Functions

- `ExecutionControlBase (const ExecutionControlBase &rhs)`  
`Copy constructor for ExecutionControlBase class.`
- `ExecutionControlBase & operator= (const ExecutionControlBase &rhs)`  
`Assignment operator for ExecutionControlBase class.`

## Friends

- class `InputProcessor`
- class `Manager`
- void `init_attrTrickHLA__ExecutionControlBase ()`

## Additional Inherited Members

### 7.19.1 Detailed Description

Definition at line 69 of file `ExecutionControlBase.hh`.

### 7.19.2 Constructor & Destructor Documentation

#### 7.19.2.1 `ExecutionControlBase()` [1/3]

`ExecutionControlBase::ExecutionControlBase ( )`  
Default constructor for the `ExecutionControlBase` class.

**Trick Job Class:** *initialization*

Definition at line 68 of file `ExecutionControlBase.cpp`.

#### 7.19.2.2 `ExecutionControlBase()` [2/3]

`ExecutionControlBase::ExecutionControlBase (`  
`ExecutionConfigurationBase & exec_config )`  
Initialization constructor for the `ExecutionControlBase` class.

#### Parameters

<code>exec_config</code>	The associated <code>ExecutionConfigurationBase</code> class instance.
--------------------------	--

**Trick Job Class:** *initialization*

Definition at line 102 of file ExecutionControlBase.cpp.

**7.19.2.3 ~ExecutionControlBase()**

ExecutionControlBase::~ExecutionControlBase ( ) [pure virtual]

Destructor for the [ExecutionControlBase](#) class.**Trick Job Class:** *shutdown*

Definition at line 137 of file ExecutionControlBase.cpp.

References [TrickHLA::LoggableSyncPnt::clear\(\)](#), [clear\\_mode\\_values\(\)](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [loggable\\_sync\\_pts](#), [logged\\_sync\\_pts\\_count](#), [multiphase\\_init\\_sync\\_points](#), [should\\_print\(\)](#), and [THLA\\_NEWLINE](#).**7.19.2.4 ExecutionControlBase() [3/3]**TrickHLA::ExecutionControlBase::ExecutionControlBase (   
     const [ExecutionControlBase](#) & rhs ) [private]Copy constructor for [ExecutionControlBase](#) class.

This constructor is private to prevent inadvertent copies.

**7.19.3 Member Function Documentation****7.19.3.1 achieve\_all\_multiphase\_init\_sync\_pnts()**

```
void ExecutionControlBase::achieve_all_multiphase_init_sync_pnts (   
     RTI1516_NAMESPACE::RTIambassador & rti_ambassador ) throw ( RTI1516_NAMESPACE::←   
 SynchronizationPointLabelNotAnnounced, RTI1516_NAMESPACE::FederateNotExecutionMember, RTI1516←   
 NAMESPACE::SaveInProgress, RTI1516_NAMESPACE::RestoreInProgress, RTI1516_NAMESPACE::NotConnected,   
 RTI1516_NAMESPACE::RTIinternalError )  
Achieve all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not one of the predefined ExecutionControl synchronization points.
```

**Parameters**

<i>rti_ambassador</i>	Reference to the HLA RTI Ambassador instance.
-----------------------	---

**Trick Job Class:** *initialization*

Definition at line 601 of file ExecutionControlBase.cpp.

Referenced by [clear\\_multiphase\\_init\\_sync\\_points\(\)](#).**7.19.3.2 add\_initialization\_sync\_points()**

virtual void TrickHLA::ExecutionControlBase::add\_initialization\_sync\_points ( ) [pure virtual]

Add initialization synchronization points to regulate startup.

Implemented in [SpaceFOM::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).**7.19.3.3 add\_multiphase\_init\_sync\_points()**

void ExecutionControlBase::add\_multiphase\_init\_sync\_points ( ) [virtual]

Add user defined multiphase initialization synchronization points to regulate the multiphase initialization process.

**Trick Job Class:** *initialization*

Reimplemented in [IMSim::ExecutionControl](#), [DSES::ExecutionControl](#), [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).  
Definition at line 494 of file ExecutionControlBase.cpp.

References [TrickHLA::SyncPntListBase::add\\_sync\\_pnt\(\)](#), [multiphase\\_init\\_sync\\_pnt\\_list](#), [multiphase\\_init\\_sync\\_points](#), [TrickHLA::StringUtilities::to\\_wstring\(\)](#), and [TrickHLA::StringUtilities::tokenize\(\)](#).  
Referenced by [SpaceFOM::ExecutionControl::add\\_initialization\\_sync\\_points\(\)](#).

#### 7.19.3.4 add\_object\_to\_map()

```
void ExecutionControlBase::add_object_to_map (
    Object * object ) [virtual]
```

Add a [TrickHLA::Object](#) to the manager object map.

**Parameters**

<i>object</i>	<a href="#">TrickHLA::Object</a> to add to the manager object map.
---------------	--

**Trick Job Class:** *initialization*

Definition at line 420 of file ExecutionControlBase.cpp.  
References [TrickHLA::Manager::add\\_object\\_to\\_map\(\)](#), and [manager](#).  
Referenced by [register\\_objects\\_with\\_RTI\(\)](#).

#### 7.19.3.5 check\_for\_shutdown()

```
bool ExecutionControlBase::check_for_shutdown ( ) [virtual]
```

Checks to see if shutdown has been commanded.

**Returns**

True if shutdown has been announced, else False.

**Trick Job Class:** *shutdown*

Reimplemented in [SpaceFOM::ExecutionControl](#).  
Definition at line 965 of file ExecutionControlBase.cpp.  
Referenced by [TrickHLA::Federate::check\\_shutdown\(\)](#).

#### 7.19.3.6 check\_for\_shutdown\_with\_termination()

```
bool ExecutionControlBase::check_for_shutdown_with_termination ( ) [virtual]
```

Checks to see if shutdown has been commanded and, if so, terminates the simulation.

**Returns**

False if shutdown has NOT been announced.

NOTE: If a shutdown has been announced, this routine calls the Trick exec\_terminate() function. So, for shutdown, it should never return. **Trick Job Class:** *shutdown*

Reimplemented in [SpaceFOM::ExecutionControl](#).  
Definition at line 975 of file ExecutionControlBase.cpp.  
Referenced by [TrickHLA::Federate::check\\_for\\_shutdown\\_with\\_termination\(\)](#).

### 7.19.3.7 `check_freeze_exit()`

```
bool ExecutionControlBase::check_freeze_exit ( ) [virtual]
Check for exit from freeze.
```

#### Returns

True if should exit from freeze.

Reimplemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [DIS::ExecutionControl](#), and [DSES::ExecutionControl](#).

Definition at line 997 of file ExecutionControlBase.cpp.

Referenced by [TrickHLA::Federate::check\\_freeze\(\)](#).

### 7.19.3.8 `check_pause()`

```
void ExecutionControlBase::check_pause (
    const double check_pause_delta ) [virtual]
```

Check if we hit a pause sync point and need to go to freeze.

#### Parameters

<i>check_pause_delta</i>	Check pause job delta time in seconds.
--------------------------	--

Reimplemented in [IMSim::ExecutionControl](#), and [IMSim::ExecutionControl](#).

Definition at line 1007 of file ExecutionControlBase.cpp.

Referenced by [check\\_pause\\_at\\_init\(\)](#).

### 7.19.3.9 `check_pause_at_init()`

```
void ExecutionControlBase::check_pause_at_init (
    const double check_pause_delta ) [virtual]
```

Checking if we started in freeze.

#### Parameters

<i>check_pause_delta</i>	Check pause job delta time in seconds.
--------------------------	--

Reimplemented in [IMSim::ExecutionControl](#), and [IMSim::ExecutionControl](#).

Definition at line 1012 of file ExecutionControlBase.cpp.

References [check\\_pause\(\)](#), [federate](#), [is\\_master\(\)](#), and [TrickHLA::Federate::set\\_freeze\\_announced\(\)](#).

Referenced by [IMSim::ExecutionControl::check\\_pause\\_at\\_init\(\)](#).

### 7.19.3.10 `clear_mode_transition_requested()`

```
virtual void TrickHLA::ExecutionControlBase::clear_mode_transition_requested ( ) [inline], [virtual]
Clear the mode transition requested flag.
```

Definition at line 438 of file ExecutionControlBase.hh.

References [mode\\_transition\\_requested](#).

Referenced by [DSES::ExecutionControl::process\\_mode\\_transition\\_request\(\)](#), [SpaceFOM::ExecutionControl::process\\_mode\\_transition\\_request\(\)](#), [DIS::ExecutionControl::process\\_mode\\_transition\\_request\(\)](#), [IMSim::ExecutionControl::process\\_mode\\_transition\\_request\(\)](#), [DSES::ExecutionControl::shutdown\\_mode\\_announce\(\)](#), [DIS::ExecutionControl::shutdown\\_mode\\_announce\(\)](#).

Control::shutdown\_mode\_announce(), IMSim::ExecutionControl::shutdown\_mode\_announce(), and SpaceFOM::ExecutionControl::shutdown\_mode\_announce().

#### 7.19.3.11 clear\_mode\_values()

```
void ExecutionControlBase::clear_mode_values ( ) [virtual]
```

Clear the Mode Transition Request flag, the requested execution mode, and the current execution mode.

Reimplemented in [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), and [DSES::ExecutionControl](#).

Definition at line 954 of file ExecutionControlBase.cpp.

References current\_execution\_control\_mode, TrickHLA::EXECUTION\_CONTROL\_UNINITIALIZED, mode\_transition\_requested, and requested\_execution\_control\_mode.

Referenced by TrickHLA::ExecutionControl::~ExecutionControl(), SpaceFOM::ExecutionControl::~ExecutionControl(), and ~ExecutionControlBase().

#### 7.19.3.12 clear\_multiphase\_init\_sync\_points()

```
void ExecutionControlBase::clear_multiphase_init_sync_points ( ) [virtual]
```

Clear any remaining multiphase initialization synchronization points that have not been achieved and wait for the federation to be synchronized on it.

**Trick Job Class:** *initialization*

Reimplemented in [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Definition at line 517 of file ExecutionControlBase.cpp.

References achieve\_all\_multiphase\_init\_sync\_pnts(), TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_G\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, TrickHLA::DEBUG\_SOURCE\_MANAGER, federate, TrickHLA::Federate::get\_RTI\_ambassador(), TrickHLA::Manager::is\_late\_joining\_federate(), manager, TrickHLA::SyncPntListBase::print\_sync\_pnts(), RTI1516\_EXCEPTION, TrickHLA::DebugHandler::should\_print(), should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), and wait\_for\_all\_multiphase\_init\_sync\_pnts().

Referenced by TrickHLA::Manager::clear\_init\_sync\_points().

#### 7.19.3.13 convert\_loggable\_sync\_pts()

```
virtual void TrickHLA::ExecutionControlBase::convert_loggable_sync_pts ( ) [inline], [virtual]
```

Converts HLA sync points into something Trick can save in a checkpoint.

Reimplemented in [IMSim::ExecutionControl](#).

Definition at line 557 of file ExecutionControlBase.hh.

Referenced by TrickHLA::Federate::convert\_sync\_pts().

#### 7.19.3.14 convert\_scenario\_time\_to\_sim\_time()

```
double TrickHLA::ExecutionControlBase::convert_scenario_time_to_sim_time (
    double scenario_time ) [inline]
```

Convert the a given scenario time into simulation time.

##### Returns

Corresponding simulation time in seconds.

##### Parameters

<code>scenario_time</code>	Scenario time to convert.
----------------------------	---------------------------

Definition at line 333 of file ExecutionControlBase.hh.  
 References `get_scenario_time()`, and `get_sim_time()`.

#### 7.19.3.15 `convert_sim_time_to_scenario_time()`

```
double TrickHLA::ExecutionControlBase::convert_sim_time_to_scenario_time (
    double sim_time ) [inline]
```

Convert the a given simulation time into scenario time.

##### Returns

Corresponding scenario time in seconds.

##### Parameters

<code>sim_time</code>	Simulation time to convert.
-----------------------	-----------------------------

Definition at line 341 of file ExecutionControlBase.hh.  
 References `get_scenario_time()`, and `get_sim_time()`.  
 Referenced by `TrickHLA::Manager::start_federation_save_at_sim_time()`.

#### 7.19.3.16 `does_cte_timeline_exist()`

```
bool TrickHLA::ExecutionControlBase::does_cte_timeline_exist ( ) const [inline]
```

Check to see if the CTE Timeline exists.

##### Returns

True if it exists, False otherwise.

Definition at line 316 of file ExecutionControlBase.hh.

References `cte_timeline`.

Referenced by `TrickHLA::Packing::get_cte_time()`, `TrickHLA::LagCompensation::get_cte_time()`, `TrickHLA::InteractionHandler::get_cte_time()`, `TrickHLA::OwnershipHandler::get_cte_time()`, `get_cte_time()`, `initialize()`, `SpaceFOM::ExecutionControl::late_joinder_hla_init_process()`, `SpaceFOM::ExecutionControl::post_multi_phase_init_processes()`, `DSES::ExecutionControl::process_execution_control_updates()`, `SpaceFOM::ExecutionControl::process_execution_control_updates()`, `DIS::ExecutionControl::process_execution_control_updates()`, `IMSim::ExecutionControl::process_execution_control_updates()`, `DSES::ExecutionControl::process_mode_transition_request()`, `SpaceFOM::ExecutionControl::process_mode_transition_request()`, `IMSim::ExecutionControl::process_mode_transition_request()`, `DIS::ExecutionControl::process_mode_transition_request()`, `SpaceFOM::MTRInteractionHandler::receive_interaction()`, `DSES::ExecutionControl::run_mode_transition()`, `DIS::ExecutionControl::run_mode_transition()`, `IMSim::ExecutionControl::run_mode_transition()`, `SpaceFOM::ExecutionControl::run_mode_transition()`, and `SpaceFOM::MTRInteractionHandler::send_interaction()`.

#### 7.19.3.17 `does_init_complete_sync_point_exist()`

```
virtual bool TrickHLA::ExecutionControlBase::does_init_complete_sync_point_exist ( ) const [inline], [virtual]
```

Query if the 'initialization\_completed' sync-point exists.

**Returns**

True if sync-point exists; False otherwise.

Definition at line 516 of file ExecutionControlBase.hh.

References init\_complete\_sp\_exists.

Referenced by [IMSim::ExecutionControl::determine\\_if\\_late\\_joining\\_or\\_restoring\\_federate\(\)](#), and [SpaceFOM::ExecutionControl::role\\_determination\\_process\(\)](#).

**7.19.3.18 does\_scenario\_timeline\_exist()**

```
bool TrickHLA::ExecutionControlBase::does_scenario_timeline_exist ( ) const [inline]
```

Check to see if the Scenario [Timeline](#) exists.

**Returns**

True if it exists, False otherwise.

Definition at line 308 of file ExecutionControlBase.hh.

References scenario\_timeline.

Referenced by [get\\_scenario\\_time\(\)](#), and [initialize\(\)](#).

**7.19.3.19 does\_sim\_timeline\_exist()**

```
bool TrickHLA::ExecutionControlBase::does_sim_timeline_exist ( ) const [inline]
```

Check to see if the Simulation [Timeline](#) exists.

**Returns**

True if it exists, False otherwise.

Definition at line 312 of file ExecutionControlBase.hh.

References sim\_timeline.

Referenced by [get\\_sim\\_time\(\)](#).

**7.19.3.20 enter\_freeze()**

```
void ExecutionControlBase::enter_freeze ( ) [virtual]
```

Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.

Reimplemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), and [DIS::ExecutionControl](#).

Definition at line 985 of file ExecutionControlBase.cpp.

References [TrickHLA::DEBUG\\_LEVEL\\_4\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [federate](#), [TrickHLA::Federate::get\\_freeze\\_announced\(\)](#), [TrickHLA::Federate::get\\_freeze\\_pending\(\)](#), [should\\_print\(\)](#), and [THLA\\_NEWLINE](#). Referenced by [TrickHLA::Federate::enter\\_freeze\(\)](#).

**7.19.3.21 exit\_freeze()**

```
void ExecutionControlBase::exit_freeze ( ) [virtual]
```

Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.

Reimplemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), and [DIS::ExecutionControl](#).

Definition at line 1002 of file ExecutionControlBase.cpp.

Referenced by [TrickHLA::Federate::exit\\_freeze\(\)](#).

### 7.19.3.22 `freeze_init()`

```
void ExecutionControlBase::freeze_init ( ) [virtual]
```

Routine to handle going from run to freeze.

Reimplemented in [SpaceFOM::ExecutionControl](#).

Definition at line 980 of file `ExecutionControlBase.cpp`.

Referenced by `TrickHLA::Federate::freeze_init()`.

### 7.19.3.23 `freeze_mode_announce()`

```
virtual void TrickHLA::ExecutionControlBase::freeze_mode_announce ( ) [pure virtual]
```

Announce the pending freeze mode transition with an 'mtr\_freeze' sync-point.

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [TrickHLA::ExecutionControl](#), and [DSES::ExecutionControl](#).

### 7.19.3.24 `freeze_mode_transition()`

```
virtual bool TrickHLA::ExecutionControlBase::freeze_mode_transition ( ) [pure virtual]
```

The freeze mode transition routine.

#### Returns

Currently always returns False.

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [TrickHLA::ExecutionControl](#), and [DSES::ExecutionControl](#).

### 7.19.3.25 `get_cte_time()`

```
double ExecutionControlBase::get_cte_time ( )
```

Get the current Central Timing Equipment time from CTE [Timeline](#).

#### Returns

The current CTE time in seconds.

Definition at line 945 of file `ExecutionControlBase.cpp`.

References `cte_timeline`, `does_cte_timeline_exist()`, and `TrickHLA::CTETimelineBase::get_time()`.

Referenced by `TrickHLA::Packing::get_cte_time()`, `TrickHLA::LagCompensation::get_cte_time()`, `TrickHLA::InteractionHandler::get_cte_time()`, `TrickHLA::OwnershipHandler::get_cte_time()`, `DSES::ExecutionControl::run_mode_transition()`, `DIS::ExecutionControl::run_mode_transition()`, `IMSim::ExecutionControl::run_mode_transition()`, `SpaceFOM::ExecutionControl::run_mode_transition()`, `DSES::ExecutionControl::set_next_execution_control_mode()`, `SpaceFOM::ExecutionControl::set_next_execution_control_mode()`, `DIS::ExecutionControl::set_next_execution_control_mode()`, and `IMSim::ExecutionControl::set_next_execution_control_mode()`.

### 7.19.3.26 `get_current_execution_control_mode()`

```
virtual ExecutionControlEnum TrickHLA::ExecutionControlBase::get_current_execution_control_mode ( ) [inline], [virtual]
```

Get the current execution mode.

**Returns**

The current execution mode.

Definition at line 472 of file ExecutionControlBase.hh.

References current\_execution\_control\_mode.

Referenced by SpaceFOM::ExecutionControl::freeze\_init().

**7.19.3.27 get\_execution\_configuration()**

```
virtual ExecutionConfigurationBase* TrickHLA::ExecutionControlBase::get_execution_configuration ( ) [inline], [virtual]
```

Get the reference to the associated [TrickHLA::ExecutionConfigurationBase](#) object.

**Returns**

Pointer to the associated [TrickHLA::ExecutionConfigurationBase](#) object.

Reimplemented in [IMSim::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [DIS::ExecutionControl](#), [DSES::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Definition at line 505 of file ExecutionControlBase.hh.

References execution\_configuration.

Referenced by TrickHLA::Manager::get\_execution\_configuration(), get\_trickhla\_object(), and get\_unregistered\_object().

**7.19.3.28 get\_federate()**

```
virtual TrickHLA::Federate* TrickHLA::ExecutionControlBase::get_federate ( ) [inline], [virtual]
```

Get the reference to the associated [TrickHLA::Federate](#).

**Returns**

Pointer to the associated [TrickHLA::Federate](#).

Definition at line 493 of file ExecutionControlBase.hh.

References federate.

Referenced by TrickHLA::Packing::get\_cte\_time(), TrickHLA::LagCompensation::get\_cte\_time(), TrickHLA::OwnershipHandler::get\_cte\_time(), TrickHLA::Packing::get\_scenario\_time(), TrickHLA::LagCompensation::get\_scenario\_time(), TrickHLA::OwnershipHandler::get\_scenario\_time(), and join\_federation\_process().

**7.19.3.29 get\_least\_common\_time\_step()**

```
virtual int64_t TrickHLA::ExecutionControlBase::get_least_common_time_step ( ) [inline], [virtual]
```

Get the value of the least common time step.

**Returns**

The value of the least common time step.

Definition at line 524 of file ExecutionControlBase.hh.

References least\_common\_time\_step.

Referenced by TrickHLA::Federate::time\_advance\_request\_to\_GALT\_LCTS\_multiple().

**7.19.3.30 get\_manager()**

```
virtual TrickHLA::Manager* TrickHLA::ExecutionControlBase::get_manager ( ) [inline], [virtual]
```

Get the reference to the associated [TrickHLA::Manager](#).

**Returns**

Pointer to the associated [TrickHLA::Manager](#).

Definition at line 496 of file ExecutionControlBase.hh.

References manager.

Referenced by `IMSim::ExecutionControl::add_freeze_scenario_time()`, `IMSim::ExecutionControl::announce_sync_point()`, `DSES::ExecutionControl::check_freeze_exit()`, `IMSim::ExecutionControl::check_pause()`, `IMSim::ExecutionControl::check_pause_at_init()`, `IMSim::ExecutionControl::determine_if_late_joining_or_restoring_federate()`, `DSES::ExecutionControl::initialize()`, `DIS::ExecutionControl::initialize()`, `IMSim::ExecutionControl::initialize()`, `DIS::ExecutionControl::post_multi_phase_init_process()`, `IMSim::ExecutionControl::post_multi_phase_init_process()`, `TrickHLA::ExecutionControl::pre_multi_phase_init_processes()`, `DSES::ExecutionControl::pre_multi_phase_init_processes()`, `DIS::ExecutionControl::pre_multi_phase_init_processes()`, `IMSim::ExecutionControl::pre_multi_phase_init_processes()`, `DSES::ExecutionControl::set_next_execution_control_mode()`, `SpaceFOM::ExecutionControl::set_next_execution_control_mode()`, `DIS::ExecutionControl::set_next_execution_control_mode()`, `IMSim::ExecutionControl::set_next_execution_control_mode()`, `IMSim::ExecutionControl::setup_interaction_ref_attributes()`, and `IMSim::ExecutionControl::start_federation_save_at_scenario_time()`.

**7.19.3.31 get\_requested\_execution\_control\_mode()**

```
virtual ExecutionControlEnum TrickHLA::ExecutionControlBase::get_requested_execution_control_mode
( ) [inline], [virtual]
```

Get the currently requested execution mode.

**Returns**

The currently requested execution mode.

Definition at line 463 of file ExecutionControlBase.hh.

References `requested_execution_control_mode`.

Referenced by `SpaceFOM::ExecutionControl::enter_freeze()`.

**7.19.3.32 get\_scenario\_freeze\_time()**

```
virtual double TrickHLA::ExecutionControlBase::get_scenario_freeze_time ( ) [inline], [virtual]
```

Get the Federation Execution scenario time for freeze.

**Returns**

Scenario time in seconds for the Federation Execution to go to freeze.

Definition at line 539 of file ExecutionControlBase.hh.

References `scenario_freeze_time`.

**7.19.3.33 get\_scenario\_time()**

```
double ExecutionControlBase::get_scenario_time ( )
```

Get the current scenario time from Scenario [Timeline](#).

**Returns**

The current scenario time in seconds.

Definition at line 927 of file ExecutionControlBase.cpp.

References `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `does_scenario_timeline_exist()`, `get_sim_time()`, `TrickHLA::ScenarioTimeline::get_time()`, `scenario_timeline`, `should_print()`, and `THLA_ENDL`.

Referenced by IMSim::ExecutionControl::add\_freeze\_scenario\_time(), IMSim::ExecutionControl::check\_scenario\_freeze\_time(), convert\_scenario\_time\_to\_sim\_time(), convert\_sim\_time\_to\_scenario\_time(), TrickHLA::Packing::get\_scenario\_time(), TrickHLA::LagCompensation::get\_scenario\_time(), TrickHLA::InteractionHandler::get\_scenario\_time(), TrickHLA::OwnershipHandler::get\_scenario\_time(), DSES::ExecutionControl::set\_next\_execution\_control\_mode(), SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionControl::set\_next\_execution\_control\_mode(), and IMSim::ExecutionControl::set\_next\_execution\_control\_mode().

#### 7.19.3.34 get\_sim\_time()

```
double ExecutionControlBase::get_sim_time ( )
```

Get the current simulation time from Simulation [Timeline](#).

##### Returns

The current simulation time in seconds.

Definition at line 910 of file ExecutionControlBase.cpp.

References TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, does\_sim\_timeline\_exist(), TrickHLA::SimTimeline::get\_time(), should\_print(), sim\_timeline, and THLA\_ENDL.

Referenced by IMSim::ExecutionControl::check\_pause(), IMSim::ExecutionControl::check\_scenario\_freeze\_time(), convert\_scenario\_time\_to\_sim\_time(), convert\_sim\_time\_to\_scenario\_time(), DIS::ExecutionControl::enter\_freeze(), IMSim::ExecutionControl::enter\_freeze(), TrickHLA::Federate::enter\_freeze(), get\_scenario\_time(), TrickHLA::InteractionHandler::get\_sim\_time(), IMSim::ExecutionControl::mark\_synchronized(), TrickHLA::Object::pull\_ownership(), and TrickHLA::Object::push\_ownership().

#### 7.19.3.35 get\_simulation\_freeze\_time()

```
virtual double TrickHLA::ExecutionControlBase::get_simulation_freeze_time ( ) [inline], [virtual]
```

Get the Federation Execution simulation time for freeze.

##### Returns

Simulation time in seconds for the Federation Execution to go to freeze.

Definition at line 533 of file ExecutionControlBase.hh.

References simulation\_freeze\_time.

Referenced by SpaceFOM::ExecutionControl::enter\_freeze().

#### 7.19.3.36 get\_time\_padding()

```
virtual double TrickHLA::ExecutionControlBase::get_time_padding ( ) [inline], [virtual]
```

Get the time-padding used to offset the go to run time.

##### Returns

Time in seconds to pad for time based mode transitions.

Definition at line 530 of file ExecutionControlBase.hh.

References time\_padding.

Referenced by SpaceFOM::ExecutionControl::shutdown().

#### 7.19.3.37 get\_trickhla\_object()

```
Object * ExecutionControlBase::get_trickhla_object (
    std::wstring const & obj_instance_name ) [virtual]
```

Gets the [TrickHLA Object](#) for the specified RTI [Object](#) Instance Name.

## Returns

TrickHLA Object.

## Parameters

<code>obj_instance_name</code>	<code>Object</code> instance name.
--------------------------------	------------------------------------

**Trick Job Class:** *scheduled*

Definition at line 803 of file ExecutionControlBase.cpp.

References `execution_configuration`, `get_execution_configuration()`, and `TrickHLA::StringUtilities::to_wstring()`.

Referenced by `TrickHLA::Manager::get_trickhla_object()`.

**7.19.3.38 `get_type()`**

```
virtual const std::wstring& TrickHLA::ExecutionControlBase::get_type ( ) [pure virtual]
```

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [TrickHLA::ExecutionControl](#), and [DSES::ExecutionControl](#).

**7.19.3.39 `get_unregistered_object()`**

```
Object * ExecutionControlBase::get_unregistered_object (
    RTI1516_NAMESPACE::ObjectClassHandle const & theObjectClass,
    std::wstring const & theObjectName ) [virtual]
```

Returns the first object that matches the specified Object-Class, object instance name, and is not registered, i.e. the instance ID == 0.

## Returns

TrickHLA Object

## Parameters

<code>theObjectClass</code>	RTI <code>Object</code> class type.
<code>theObjectName</code>	<code>Object</code> instance name.

**Trick Job Class:** *scheduled*

Definition at line 825 of file ExecutionControlBase.cpp.

References `execution_configuration`, `TrickHLA::Object::get_class_handle()`, `get_execution_configuration()`, `TrickHLA::Object::is_instance_handle_valid()`, and `TrickHLA::StringUtilities::to_wstring()`.

Referenced by `TrickHLA::Manager::get_unregistered_object()`.

**7.19.3.40 `get_unregistered_remote_object()`**

```
Object * ExecutionControlBase::get_unregistered_remote_object (
    RTI1516_NAMESPACE::ObjectClassHandle const & theObjectClass ) [virtual]
```

Returns the first object that is remotely owned, has the same Object-Class, is not registered, and does not have an `Object` Instance Name associated with it.

**Returns**

The associated [TrickHLA::Object](#) instance; otherwise NULL.

**Parameters**

<i>theObjectClass</i>	RTI <a href="#">Object</a> class type.
-----------------------	--

**Trick Job Class: *scheduled***

Definition at line 854 of file ExecutionControlBase.cpp.

References `execution_configuration`, `TrickHLA::Object::get_class_handle()`, `TrickHLA::Object::get_name()`, `TrickHLA::Object::is_create_HLA_instance()`, `TrickHLA::Object::is_instance_handle_valid()`, and `TrickHLA::Object::is_name_required()`.

Referenced by `TrickHLA::Manager::get_unregistered_remote_object()`.

**7.19.3.41 `initialize()`**

```
void ExecutionControlBase::initialize ( ) [virtual]
```

Initialize the [TrickHLA::ExecutionControlBase](#) object instance.

**Trick Job Class: *initialization***

Reimplemented in [SpaceFOM::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Definition at line 232 of file ExecutionControlBase.cpp.

References `TrickHLA::CTETimelineBase::clock_init()`, `TrickHLA::ExecutionConfigurationBase::configure()`, `cte_timeline`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `TrickHLA::def_scenario_timeline`, `does_cte_timeline_exist()`, `does_scenario_timeline_exist()`, `execution_configuration`, `TrickHLA::Manager::get_debug_handler()`, `is_execution_configuration_used()`, `is_master_preset()`, `manager`, `scenario_timeline`, `TrickHLA::SyncPntListBase::set_debug_level()`, `set_master()`, `should_print()`, and `THLA_NEWLINE`.

Referenced by `TrickHLA::Federate::initialize()`.

**7.19.3.42 `is_execution_configuration_used()`**

```
virtual bool TrickHLA::ExecutionControlBase::is_execution_configuration_used ( ) [inline], [virtual]
```

Test is an execution configuration object is used.

**Returns**

True if an execution configuration object is used.

Definition at line 513 of file ExecutionControlBase.hh.

References `execution_configuration`.

Referenced by `initialize()`, and `TrickHLA::Manager::is_execution_configuration_used()`.

**7.19.3.43 `is_in_freeze()`**

```
virtual bool TrickHLA::ExecutionControlBase::is_in_freeze ( ) [inline], [virtual]
```

Is the federate execution in initialization.

**Returns**

True if federate is initializing, false otherwise.

Definition at line 385 of file ExecutionControlBase.hh.

References `current_execution_control_mode`, and `TrickHLA::EXECUTION_CONTROL_FREEZE`.

#### 7.19.3.44 `is_in_reconfig()`

```
virtual bool TrickHLA::ExecutionControlBase::is_in_reconfig ( ) [inline], [virtual]
Is the federate execution in initialization.
```

##### Returns

True if federate is initializing, false otherwise.

Definition at line 397 of file ExecutionControlBase.hh.

References `current_execution_control_mode`, and `TrickHLA::EXECUTION_CONTROL_RECONFIG`.

#### 7.19.3.45 `is_in_restart()`

```
virtual bool TrickHLA::ExecutionControlBase::is_in_restart ( ) [inline], [virtual]
Is the federate execution in initialization.
```

##### Returns

True if federate is initializing, false otherwise.

Definition at line 391 of file ExecutionControlBase.hh.

References `current_execution_control_mode`, and `TrickHLA::EXECUTION_CONTROL_RESTART`.

#### 7.19.3.46 `is_initializing()`

```
virtual bool TrickHLA::ExecutionControlBase::is_initializing ( ) [inline], [virtual]
Is the federate execution in initialization.
```

##### Returns

True if federate is initializing, false otherwise.

Definition at line 373 of file ExecutionControlBase.hh.

References `current_execution_control_mode`, and `TrickHLA::EXECUTION_CONTROL_INITIALIZING`.

#### 7.19.3.47 `is_late_joiner()`

```
virtual bool TrickHLA::ExecutionControlBase::is_late_joiner ( ) [inline], [virtual]
Determine if this federate is a late joining federate.
```

##### Returns

True if this is a late joining federate.

Definition at line 455 of file ExecutionControlBase.hh.

References `late_joiner`.

Referenced by `IMSim::ExecutionControl::determine_if_late_joining_or_restoring_federate()`, `SpaceFOM::ExecutionControl::freeze_init()`, `TrickHLA::Manager::is_late_joining_federate()`, `SpaceFOM::ExecutionControl::post_multi_phase_init_processes()`, `IMSim::ExecutionControl::pre_multi_phase_init_processes()`, `TrickHLA::Federate::time_advance_request_to_GALT()`, `TrickHLA::Federate::time_advance_request_to_GALT_LCTS_multiple()`, and `IMSim::ExecutionControl::trigger_freeze_interaction()`.

**7.19.3.48 is\_late\_joine\_determined()**

```
virtual bool TrickHLA::ExecutionControlBase::is_late_joine_determined ( ) [inline], [virtual]
Check if we have determine if this federate is a late joining federate.
```

**Returns**

True if late joining status is determined.

Definition at line 458 of file ExecutionControlBase.hh.

References late\_joine\_determined.

**7.19.3.49 is\_master()**

```
virtual bool TrickHLA::ExecutionControlBase::is_master ( ) const [inline], [virtual]
Query if this is the Master federate.
```

**Returns**

True if there is the Master; False otherwise.

Definition at line 452 of file ExecutionControlBase.hh.

References master.

Referenced by DSES::ExecutionControl::check\_freeze\_exit(), SpaceFOM::ExecutionControl::check\_freeze\_exit(), DSES::ExecutionControl::check\_mode\_transition\_request(), DIS::ExecutionControl::check\_mode\_transition\_request(), SpaceFOM::ExecutionControl::check\_mode\_transition\_request(), IMSim::ExecutionControl::check\_mode\_transition\_request(), check\_pause\_at\_init(), DSES::ExecutionControl::determine\_federation\_master(), DIS::ExecutionControl::determine\_federation\_master(), SpaceFOM::ExecutionControl::early\_joine\_hla\_init\_process(), DIS::ExecutionControl::enter\_freeze(), IMSim::ExecutionControl::enter\_freeze(), SpaceFOM::ExecutionControl::enter\_freeze(), SpaceFOM::ExecutionControl::epoch\_and\_root\_frame\_discovery\_process(), SpaceFOM::ExecutionControl::exit\_freeze(), DSES::ExecutionControl::freeze\_mode\_announce(), DIS::ExecutionControl::freeze\_mode\_announce(), IMSim::ExecutionControl::freeze\_mode\_announce(), SpaceFOM::ExecutionControl::freeze\_mode\_announce(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), IMSim::ExecutionControl::post\_multi\_phase\_init\_process(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::publish(), receive\_execution\_configuration(), TrickHLA::Manager::restart\_initialization(), TrickHLA::Federate::restore\_checkpoint(), SpaceFOM::ExecutionControl::role\_determination\_process(), DSES::ExecutionControl::run\_mode\_transition(), DIS::ExecutionControl::run\_mode\_transition(), IMSim::ExecutionControl::run\_mode\_transition(), SpaceFOM::ExecutionControl::run\_mode\_transition(), send\_execution\_configuration(), DSES::ExecutionConfiguration::set\_current\_execution\_mode(), DIS::ExecutionConfiguration::set\_current\_execution\_mode(), IMSim::ExecutionConfiguration::set\_current\_execution\_mode(), SpaceFOM::ExecutionConfiguration::set\_current\_execution\_mode(), SpaceFOM::ExecutionConfiguration::set\_least\_common\_time\_step(), TrickHLA::ExecutionControl::set\_least\_common\_time\_step(), SpaceFOM::ExecutionControl::set\_least\_common\_time\_step(), DSES::ExecutionControl::set\_next\_execution\_control\_mode(), SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionControl::set\_next\_execution\_control\_mode(), IMSim::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionConfiguration::set\_next\_execution\_mode(), DSES::ExecutionConfiguration::set\_next\_execution\_mode(), IMSim::ExecutionConfiguration::set\_next\_execution\_mode(), SpaceFOM::ExecutionConfiguration::set\_next\_execution\_mode(), DSES::ExecutionConfiguration::set\_next\_mode\_cte\_time(), DIS::ExecutionConfiguration::set\_next\_mode\_cte\_time(), IMSim::ExecutionConfiguration::set\_next\_mode\_cte\_time(), SpaceFOM::ExecutionConfiguration::set\_next\_mode\_cte\_time(), DSES::ExecutionConfiguration::set\_next\_mode\_cte\_time(), DIS::ExecutionConfiguration::set\_next\_mode\_scenario\_time(), IMSim::ExecutionConfiguration::set\_next\_mode\_scenario\_time()

```
::set_next_mode_scenario_time(), SpaceFOM::ExecutionConfiguration::set_next_mode_scenario_time(), DSES::ExecutionConfiguration::set_scenario_time_epoch(), DIS::ExecutionConfiguration::set_scenario_time_epoch(), IMSim::ExecutionConfiguration::set_scenario_time_epoch(), SpaceFOM::ExecutionConfiguration::set_scenario_time_epoch(), SpaceFOM::ExecutionControl::setup_interaction_ref_attributes(), SpaceFOM::ExecutionConfiguration::setup_ref_attributes(), SpaceFOM::ExecutionControl::shutdown(), DSES::ExecutionControl::shutdown_mode_announce(), DIS::ExecutionControl::shutdown_mode_announce(), IMSim::ExecutionControl::shutdown_mode_announce(), SpaceFOM::ExecutionControl::shutdown_mode_announce(), DSES::ExecutionControl::shutdown_mode_transition(), DIS::ExecutionControl::shutdown_mode_transition(), IMSim::ExecutionControl::shutdown_mode_transition(), SpaceFOM::ExecutionControl::shutdown_mode_transition(), SpaceFOM::ExecutionControl::subscribe(), IMSim::ExecutionControl::subscribe(), TrickHLA::Federate::time_advance_request_to_GALT(), TrickHLA::Federate::time_advance_request_to_GALT_LCTS_multiple(), SpaceFOM::ExecutionControl::unpublish(), IMSim::ExecutionControl::unpublish(), SpaceFOM::ExecutionControl::unsubscribe(), IMSim::ExecutionControl::unsubscribe(), TrickHLA::ExecutionConfigurationBase::wait_on_update(), DIS::ExecutionConfiguration::wait_on_update(), DSES::ExecutionConfiguration::wait_on_update(), IMSim::ExecutionConfiguration::wait_on_update(), and SpaceFOM::ExecutionConfiguration::wait_on_update().
```

#### 7.19.3.50 is\_master\_preset()

```
virtual bool TrickHLA::ExecutionControlBase::is_master_preset () const [inline], [virtual]
Query if there is a preset Master.
```

##### Returns

True if there is a preset Master; False otherwise.

Definition at line 446 of file ExecutionControlBase.hh.

References use\_preset\_master.

Referenced by DSES::ExecutionControl::determine\_federation\_master(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), initialize(), object\_instance\_name\_reservation\_succeeded(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), and set\_master().

#### 7.19.3.51 is\_mode\_transition\_requested()

```
virtual bool TrickHLA::ExecutionControlBase::is_mode_transition_requested () [inline], [virtual]
Determine if a mode transition has been requested.
```

##### Returns

mode\_change\_requested True if a mode transition has been requested.

Definition at line 441 of file ExecutionControlBase.hh.

References mode\_transition\_requested.

Referenced by DSES::ExecutionControl::check\_mode\_transition\_request(), DIS::ExecutionControl::check\_mode\_transition\_request(), SpaceFOM::ExecutionControl::check\_mode\_transition\_request(), and IMSim::ExecutionControl::check\_mode\_transition\_request().

#### 7.19.3.52 is\_running()

```
virtual bool TrickHLA::ExecutionControlBase::is_running () [inline], [virtual]
Is the federate execution in initialization.
```

**Returns**

True if federate is initializing, false otherwise.

Definition at line 379 of file ExecutionControlBase.hh.

References current\_execution\_control\_mode, and TrickHLA::EXECUTION\_CONTROL\_RUNNING.

**7.19.3.53 is\_save\_and\_restore\_supported()**

```
virtual bool TrickHLA::ExecutionControlBase::is_save_and_restore_supported () [inline], [virtual]
Reimplemented in IMSim::ExecutionControl, and DIS::ExecutionControl.
```

Definition at line 548 of file ExecutionControlBase.hh.

Referenced by TrickHLA::Federate::is\_HLA\_save\_and\_restore\_supported().

**7.19.3.54 is\_save\_initiated()**

```
virtual bool TrickHLA::ExecutionControlBase::is_save_initiated () [inline], [virtual]
```

Checks if Save has been initiated by this [ExecutionControl](#) method.

**Returns**

True if Save is initiated and synchronized with the federation, False if Save not supported.

Reimplemented in [IMSim::ExecutionControl](#), and [DIS::ExecutionControl](#).

Definition at line 552 of file ExecutionControlBase.hh.

Referenced by TrickHLA::Federate::setup\_checkpoint().

**7.19.3.55 is\_shutdown()**

```
virtual bool TrickHLA::ExecutionControlBase::is_shutdown () [inline], [virtual]
```

Is the federate execution in initialization.

**Returns**

True if federate is initializing, false otherwise.

Definition at line 403 of file ExecutionControlBase.hh.

References current\_execution\_control\_mode, and TrickHLA::EXECUTION\_CONTROL\_SHUTDOWN.

**7.19.3.56 join\_federation\_process()**

```
void ExecutionControlBase::join_federation_process () [virtual]
```

Join federation execution process.

**Trick Job Class: initialization**

Reimplemented in [DIS::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [DSES::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Definition at line 276 of file ExecutionControlBase.cpp.

References TrickHLA::Federate::check\_for\_shutdown\_with\_termination(), TrickHLA::Federate::create\_and\_join\_federation(), TrickHLA::Federate::create\_RTI\_ambassador\_and\_connect(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::Federate::destroy\_orphaned\_federation(), TrickHLA::Federate::enable\_async\_delivery(), federate, get\_federate(), TrickHLA::Federate::should\_print(), and THLA\_NEWLINE.

Referenced by TrickHLA::ExecutionControl::join\_federation\_process(), DSES::ExecutionControl::join\_federation\_process(), DIS::ExecutionControl::join\_federation\_process(), IMSim::ExecutionControl::join\_federation\_process(), and SpaceFOM::ExecutionControl::join\_federation\_process().

### 7.19.3.57 `mark_object_as_deleted_from_federation()`

```
bool ExecutionControlBase::mark_object_as_deleted_from_federation (
    RTI1516_NAMESPACE::ObjectInstanceHandle const & instance_id ) [virtual]
```

Identifies the object as deleted from the RTI.

#### Parameters

<code>instance_id</code>	HLA object instance handle.
--------------------------	-----------------------------

Definition at line 874 of file ExecutionControlBase.cpp.

References TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, execution\_configuration, TrickHLA::Object::get\_instance\_handle(), TrickHLA::Object::get\_name(), TrickHLA::Object::remove\_object\_instance(), TrickHLA::DebugHandler::should\_print(), THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

Referenced by TrickHLA::Manager::mark\_object\_as\_deleted\_from\_federation().

### 7.19.3.58 `object_instance_name_reservation_failed()`

```
bool ExecutionControlBase::object_instance_name_reservation_failed (
    std::wstring const & obj_instance_name ) [virtual]
```

The object instance name reservation failed for the given name.

#### Returns

True if [ExecutionConfiguration](#) object handled the failure.

#### Parameters

<code>obj_instance_name</code>	<a href="#">Object</a> instance name.
--------------------------------	---------------------------------------

#### Trick Job Class: *initialization*

Reimplemented in [DSES::ExecutionControl](#), and [DIS::ExecutionControl](#).

Definition at line 556 of file DIS/ExecutionControl.cpp.

Referenced by TrickHLA::Manager::object\_instance\_name\_reservation\_failed().

### 7.19.3.59 `object_instance_name_reservation_succeeded()`

```
bool ExecutionControlBase::object_instance_name_reservation_succeeded (
    std::wstring const & obj_instance_name ) [virtual]
```

The object instance name reservation succeeded for the given name.

#### Returns

True if [ExecutionConfiguration](#) object name matched the object instance name.

#### Parameters

<code>obj_instance_name</code>	<a href="#">Object</a> instance name.
--------------------------------	---------------------------------------

#### Trick Job Class: *initialization*

Definition at line 310 of file ExecutionControlBase.cpp.

References TrickHLA::SyncPntListBase::debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBU←G\_SOURCE\_MANAGER, execution\_configuration, TrickHLA::Object::get\_name(), is\_master\_preset(), set\_master(), TrickHLA::Object::set\_name\_registered(), TrickHLA::DebugHandler::should\_print(), THLA\_NEWLINE, and TrickHLA←::StringUtilities::to\_wstring().

Referenced by TrickHLA::Manager::object\_instance\_name\_reservation\_succeeded().

#### 7.19.3.60 operator=()

```
ExecutionControlBase& TrickHLA::ExecutionControlBase::operator= (
    const ExecutionControlBase & rhs ) [private]
```

Assignment operator for [ExecutionControlBase](#) class.

This assignment operator is private to prevent inadvertent copies.

#### 7.19.3.61 perform\_save()

```
virtual bool TrickHLA::ExecutionControlBase::perform_save ( ) [inline], [virtual]
```

Federates that did not announce the save, perform a save.

Returns

True if Save can proceed, False if not.

Reimplemented in [IMSim::ExecutionControl](#).

Definition at line 555 of file ExecutionControlBase.hh.

Referenced by TrickHLA::Federate::perform\_checkpoint().

#### 7.19.3.62 post\_multi\_phase\_init\_processes()

```
virtual void TrickHLA::ExecutionControlBase::post_multi_phase_init_processes ( ) [pure virtual]
```

Processes run after the multi-phase initialization ends.

Implemented in [SpaceFOM::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by TrickHLA::Federate::post\_multiphase\_initialization().

#### 7.19.3.63 pre\_multi\_phase\_init\_processes()

```
virtual void TrickHLA::ExecutionControlBase::pre_multi_phase_init_processes ( ) [pure virtual]
```

Processes run before the multi-phase initialization begins.

Implemented in [DIS::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [DSES::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by TrickHLA::Federate::pre\_multiphase\_initialization().

#### 7.19.3.64 process\_deleted\_objects()

```
void ExecutionControlBase::process_deleted_objects ( ) [virtual]
```

Scheduled method used as a callback to identify if any objects were deleted from the RTI.

**Trick Job Class:** *logging*

Definition at line 899 of file ExecutionControlBase.cpp.

References execution\_configuration, TrickHLA::Object::is\_object\_deleted\_from\_RTI(), and TrickHLA::Object::process←\_deleted\_object().

Referenced by TrickHLA::Manager::process\_deleted\_objects().

### 7.19.3.65 process\_execution\_control\_updates()

```
virtual bool TrickHLA::ExecutionControlBase::process_execution_control_updates ( ) [pure virtual]
Process changes from any received Execution Control Objects (ExCOs).
```

#### Returns

True if mode transition occurred.

Implemented in [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [TrickHLA::ExecutionControl](#), and [DSES::ExecutionControl](#).

Referenced by [receive\\_cyclic\\_data\(\)](#).

### 7.19.3.66 process\_mode\_interaction()

```
virtual bool TrickHLA::ExecutionControlBase::process_mode_interaction ( ) [pure virtual]
Process a new mode interaction.
```

#### Returns

True if new mode interaction is successfully processed.

Implemented in [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [IMSim::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [DSES::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by [TrickHLA::Manager::process\\_interactions\(\)](#).

### 7.19.3.67 provide\_attribute\_update()

```
void ExecutionControlBase::provide_attribute_update (
    RTI1516_NAMESPACE::ObjectInstanceHandle const & theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes ) [virtual]
```

Requesting an attribute value update for the given object instance and attributes.

#### Parameters

<i>theObject</i>	HLA object instance handle.
<i>theAttributes</i>	HLA attribute handle set.

#### Trick Job Class: *scheduled*

Definition at line 788 of file [ExecutionControlBase.cpp](#).

References [execution\\_configuration](#), [TrickHLA::Object::get\\_instance\\_handle\(\)](#), and [TrickHLA::Object::provide\\_attribute\\_update\(\)](#).

Referenced by [TrickHLA::Manager::provide\\_attribute\\_update\(\)](#).

### 7.19.3.68 publish()

```
virtual void TrickHLA::ExecutionControlBase::publish ( ) [pure virtual]
```

Publish the [ExecutionControl](#) objects and interactions.

Implemented in [IMSim::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [DSES::ExecutionControl](#), [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by [TrickHLA::Manager::publish\(\)](#).

### 7.19.3.69 receive\_cyclic\_data()

```
void ExecutionControlBase::receive_cyclic_data (
    double current_time ) [virtual]
```

Handle the received cyclic data.

#### Parameters

<i>current_time</i>	Current time.
---------------------	---------------

#### Trick Job Class: *scheduled*

Definition at line 770 of file ExecutionControlBase.cpp.

References `execution_configuration`, `process_execution_control_updates()`, and `TrickHLA::Object::receive_init_data()`.

Referenced by `TrickHLA::Manager::receive_cyclic_data()`.

### 7.19.3.70 receive\_execution\_configuration()

```
void ExecutionControlBase::receive_execution_configuration ( ) [virtual]
```

Receive the [ExecutionConfiguration](#) data from the master federate.

#### Trick Job Class: *initialization*

Definition at line 675 of file ExecutionControlBase.cpp.

References `TrickHLA::Object::any_remotely_owned_subscribed_init_attribute()`, `TrickHLA::Federate::check_for_shutdown_with_termination()`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `execution_configuration`, `federate`, `TrickHLA::Object::is_changed()`, `TrickHLA::Federate::is_execution_member()`, `is_master()`, `TrickHLA::Object::receive_init_data()`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `DSES::ExecutionControl::pre_multi_phase_init_processes()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

### 7.19.3.71 receive\_interaction()

```
virtual void TrickHLA::ExecutionControlBase::receive_interaction (
    RTI1516_NAMESPACE::InteractionClassHandle const & theInteraction,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_NAMESPACE::UserData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    bool received_as_TSO ) [pure virtual]
```

Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.

#### Parameters

<i>theInteraction</i>	Interaction handle.
<i>theParameterValues</i>	Parameter values.
<i>theUserSuppliedTag</i>	Users tag.
<i>theTime</i>	HLA time for the interaction.
<i>received_as_TSO</i>	True if interaction was received by RTI as TSO.

Implemented in [IMSim::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by `TrickHLA::Manager::receive_interaction()`.

### 7.19.3.72 register\_interactions\_with\_RTI()

```
virtual void TrickHLA::ExecutionControlBase::register_interactions_with_RTI ( ) [inline], [virtual]
```

Setup the [ExecutionControl](#) interactions HLA RTI handles.

Definition at line 169 of file [ExecutionControlBase.hh](#).

### 7.19.3.73 register\_objects\_with\_RTI()

```
void ExecutionControlBase::register_objects_with_RTI ( ) [virtual]
```

Setup the [ExecutionControl](#) objects HLA RTI handles.

**Trick Job Class:** *initialization*

Definition at line 397 of file [ExecutionControlBase.cpp](#).

References [add\\_object\\_to\\_map\(\)](#), [TrickHLA::SyncPntListBase::debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRA←CE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [execution\\_configuration](#), [TrickHLA::Object::register\\_object\\_with\\_RTI\(\)](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), and [THLA\\_NEWLINE](#).

Referenced by [TrickHLA::Manager::register\\_objects\\_with\\_RTI\(\)](#).

### 7.19.3.74 reinstate\_logged\_sync\_pts()

```
virtual void TrickHLA::ExecutionControlBase::reinstate_logged_sync_pts ( ) [inline], [virtual]
```

Converts checkpointed sync points into HLA sync points.

Reimplemented in [IMSim::ExecutionControl](#).

Definition at line 559 of file [ExecutionControlBase.hh](#).

Referenced by [TrickHLA::Federate::reinstate\\_logged\\_sync\\_pts\(\)](#).

### 7.19.3.75 remove\_execution\_configuration()

```
void ExecutionControlBase::remove_execution_configuration ( ) [virtual]
```

Remove the [ExecutionConfiguration](#) instance from the federation execution.

**Trick Job Class:** *shutdown*

Definition at line 1055 of file [ExecutionControlBase.cpp](#).

References [execution\\_configuration](#), and [TrickHLA::Object::remove\(\)](#).

Referenced by [TrickHLA::Federate::shutdown\(\)](#).

### 7.19.3.76 run\_mode\_transition()

```
virtual bool TrickHLA::ExecutionControlBase::run_mode_transition ( ) [pure virtual]
```

The run mode transition routine.

**Returns**

Currently always returns True.

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [TrickHLA::ExecutionControl](#), and [DSES::ExecutionControl](#).

### 7.19.3.77 send\_execution\_configuration()

```
void ExecutionControlBase::send_execution_configuration ( ) [virtual]
```

Send the [ExecutionConfiguration](#) data if we are the master federate.

**Trick Job Class:** *initialization*

Definition at line 631 of file [ExecutionControlBase.cpp](#).

References `TrickHLA::Object::any_locally_owned_published_init_attribute()`, `TrickHLA::SyncPntListBase::debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `execution_configuration`, `is_master()`, `TrickHLA::Object::send_init_data()`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `DSES::ExecutionControl::pre_multi_phase_init_processes()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

#### 7.19.3.78 send\_mode\_transition\_interaction()

```
virtual void TrickHLA::ExecutionControlBase::send_mode_transition_interaction (
    ModeTransitionEnum requested_mode ) [pure virtual]
```

Send a mode transition request to the Master federate.

##### Parameters

<code>requested_mode</code>	Requested mode.
-----------------------------	-----------------

Implemented in [IMSim::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

#### 7.19.3.79 send\_requested\_data()

```
void ExecutionControlBase::send_requested_data (
    double current_time,
    double job_cycle_time ) [virtual]
```

Send the attribute value requested data to the remote federates.

##### Parameters

<code>current_time</code>	Current time.
<code>job_cycle_time</code>	Cycle time for this send requested data call.

##### Trick Job Class: *scheduled*

Definition at line 755 of file `ExecutionControlBase.cpp`.

References `execution_configuration`, and `TrickHLA::Object::send_requested_data()`.

Referenced by `TrickHLA::Manager::send_requested_data()`.

#### 7.19.3.80 set\_current\_execution\_control\_mode()

```
virtual void TrickHLA::ExecutionControlBase::set_current_execution_control_mode (
    ExecutionControlEnum mode ) [inline], [virtual]
```

Set the current execution control mode.

##### Parameters

<code>mode</code>	The current execution control mode.
-------------------	-------------------------------------

Definition at line 475 of file `ExecutionControlBase.hh`.

References `current_execution_control_mode`.

Referenced by `SpaceFOM::ExecutionControl::freeze_init()`, and `SpaceFOM::ExecutionControl::pre_multi_phase_init_processes()`.

### 7.19.3.81 set\_execution\_configuration()

```
virtual void TrickHLA::ExecutionControlBase::set_execution_configuration (
    ExecutionConfigurationBase * exec_config ) [inline], [virtual]
```

Get the reference to the associated [TrickHLA::ExecutionConfigurationBase](#) object.

#### Parameters

<i>exec_config</i>	Pointer to the associated <a href="#">TrickHLA::ExecutionConfigurationBase</a> object.
--------------------	--

Definition at line 499 of file ExecutionControlBase.hh.

References [execution\\_configuration](#).

Referenced by [TrickHLA::Manager::set\\_execution\\_configuration\(\)](#).

### 7.19.3.82 set\_federate()

```
virtual void TrickHLA::ExecutionControlBase::set_federate (
    TrickHLA::Federate * fed ) [inline], [virtual]
```

Set the reference to the associated [TrickHLA::Federate](#).

#### Parameters

<i>fed</i>	Associated <a href="#">TrickHLA::Federate</a> .
------------	---

Definition at line 490 of file ExecutionControlBase.hh.

References [federate](#).

### 7.19.3.83 set\_least\_common\_time\_step()

```
void ExecutionControlBase::set_least_common_time_step (
    int64_t lcts ) [virtual]
```

Set the least common time step in microseconds for the federation.

#### Parameters

<i>lcts</i>	Least Common Time Step time in microseconds.
-------------	--

WARNING: Only the Master federate should ever set this.

Reimplemented in [SpaceFOM::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Definition at line 1067 of file ExecutionControlBase.cpp.

References [is\\_master\(\)](#), and [least\\_common\\_time\\_step](#).

### 7.19.3.84 set\_master()

```
void ExecutionControlBase::set_master (
    bool master_flag ) [virtual]
```

Set this as the Master federate.

#### Parameters

<i>master_flag</i>	True for a Master federate; False otherwise.
--------------------	--

Definition at line 1024 of file ExecutionControlBase.cpp.

References `execution_configuration`, `is_master_preset()`, `master`, and `TrickHLA::ExecutionConfigurationBase::set_master()`.

Referenced by `initialize()`, `object_instance_name_reservation_succeeded()`, `TrickHLA::ExecutionControl::pre_multi_phase_init_processes()`, `IMSim::ExecutionControl::pre_multi_phase_init_processes()`, `TrickHLA::Manager::restart_initialization()`, `DIS::ExecutionControl::sync_point_registration_failed()`, and `DIS::ExecutionControl::sync_point_registration_succeeded()`.

#### 7.19.3.85 `set_mode_transition_requested()`

`virtual void TrickHLA::ExecutionControlBase::set_mode_transition_requested ( ) [inline], [virtual]`  
Set the mode transition requested flag.

Definition at line 436 of file ExecutionControlBase.hh.

References `mode_transition_requested`.

Referenced by `SpaceFOM::MTRInteractionHandler::receive_interaction()`.

#### 7.19.3.86 `set_next_execution_control_mode()`

`virtual void TrickHLA::ExecutionControlBase::set_next_execution_control_mode ( ExecutionControlEnum exec_control ) [pure virtual]`

Sets the next [ExecutionControl](#) run mode.

##### Parameters

<code>exec_control</code>	Next <a href="#">ExecutionControl</a> run mode.
---------------------------	---

Implemented in [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [TrickHLA::ExecutionControl](#), and [DSES::ExecutionControl](#).

#### 7.19.3.87 `set_requested_execution_control_mode()` [1/2]

`virtual void TrickHLA::ExecutionControlBase::set_requested_execution_control_mode ( ExecutionControlEnum mode ) [inline], [virtual]`

Set the currently requested execution mode.

##### Parameters

<code>mode</code>	The requested execution mode.
-------------------	-------------------------------

Definition at line 466 of file ExecutionControlBase.hh.

References `requested_execution_control_mode`.

Referenced by `SpaceFOM::ExecutionControl::late_joiner_hla_init_process()`, and `SpaceFOM::ExecutionControl::pre_multi_phase_init_processes()`.

#### 7.19.3.88 `set_requested_execution_control_mode()` [2/2]

`virtual void TrickHLA::ExecutionControlBase::set_requested_execution_control_mode ( int16_t mode ) [inline], [virtual]`

Set the currently requested execution mode.

**Parameters**

<i>mode</i>	The requested execution mode.
-------------	-------------------------------

Definition at line 469 of file ExecutionControlBase.hh.

References TrickHLA::execution\_control\_int16\_to\_enum(), and requested\_execution\_control\_mode.

**7.19.3.89 set\_scenario\_freeze\_time()**

```
virtual void TrickHLA::ExecutionControlBase::set_scenario_freeze_time (
    double freeze_time ) [inline], [virtual]
```

Set the Federation Execution scenario time for freeze.

**Parameters**

<i>freeze_time</i>	Scenario time in seconds for the Federation Execution to go to freeze.
--------------------	--

Definition at line 542 of file ExecutionControlBase.hh.

References scenario\_freeze\_time.

**7.19.3.90 set\_scenario\_timeline()**

```
void TrickHLA::ExecutionControlBase::set_scenario_timeline (
    ScenarioTimeline * timeline ) [inline]
```

Set the Scenario [Timeline](#).

**Parameters**

<i>timeline</i>	Scenario timeline.
-----------------	--------------------

Definition at line 300 of file ExecutionControlBase.hh.

**7.19.3.91 set\_simulation\_freeze\_time()**

```
virtual void TrickHLA::ExecutionControlBase::set_simulation_freeze_time (
    double freeze_time ) [inline], [virtual]
```

Set the Federation Execution simulation time for freeze.

**Parameters**

<i>freeze_time</i>	Simulation time in seconds for the Federation Execution to go to freeze.
--------------------	--

Definition at line 536 of file ExecutionControlBase.hh.

References simulation\_freeze\_time.

**7.19.3.92 set\_time\_padding()**

```
void ExecutionControlBase::set_time_padding (
    double t ) [virtual]
```

Set the time-padding used to offset the go to run time.

**Parameters**

<i>t</i>	Time in seconds to pad for time based mode transitions.
----------	---

Reimplemented in [IMSim::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [DIS::ExecutionControl](#), [DSES::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Definition at line 1078 of file `ExecutionControlBase.cpp`.

References `time_padding`.

**7.19.3.93 set\_wait\_sleep()**

```
virtual void TrickHLA::ExecutionControlBase::set_wait_sleep (
    unsigned int t ) [inline], [virtual]
```

Set the time interval in microseconds a sleep loop timeout.

**Parameters**

<i>t</i>	Time in microseconds a sleep loop timeout.
----------	--

Definition at line 563 of file `ExecutionControlBase.hh`.

References `wait_sleep`.

**7.19.3.94 set\_wait\_timeout()**

```
virtual void TrickHLA::ExecutionControlBase::set_wait_timeout (
    unsigned int t ) [inline], [virtual]
```

Set the time interval in microseconds a wait loop timeout.

**Parameters**

<i>t</i>	Time in microseconds a wait loop timeout.
----------	---

Definition at line 566 of file `ExecutionControlBase.hh`.

References `wait_timeout`.

**7.19.3.95 setup() [1/2]**

```
void ExecutionControlBase::setup (
    TrickHLA::Federate & federate,
    TrickHLA::Manager & manager ) [virtual]
```

Setup the federate wide references in the [ExecutionControl](#) class instance.

**Parameters**

<i>federate</i>	Associated federate manager class instance.
<i>manager</i>	Associated federate manager class instance.

#### Assumptions and Limitations:

- This assumes that the [TrickHLA::ExecutionConfigurationBase](#) instance is set elsewhere.

#### Trick Job Class: *default\_data*

Definition at line 205 of file [ExecutionControlBase.cpp](#).

References [TrickHLA::ExecutionConfigurationBase::configure\\_attributes\(\)](#), [execution\\_configuration](#), [federate](#), [manager](#), and [TrickHLA::ExecutionConfigurationBase::setup\(\)](#).

#### 7.19.3.96 `setup()` [2/2]

```
void ExecutionControlBase::setup (
    TrickHLA::Federate & federate,
    TrickHLA::Manager & manager,
    TrickHLA::ExecutionConfigurationBase & exec_config ) [virtual]
```

Setup the federate wide references in the [ExecutionControl](#) class instance.

#### Parameters

<code>federate</code>	Associated federate manager class instance.
<code>manager</code>	Associated federate manager class instance.
<code>exec_config</code>	Associated Execution Configuration <a href="#">Object</a> (ExCO).

#### Assumptions and Limitations:

- The [TrickHLA::ExecutionConfigurationBase](#) class is actually an abstract class. Therefore, the actual object instance being passed in is an instantiable polymorphic child of the [TrickHLA::ExecutionConfigurationBase](#) class.

#### Trick Job Class: *default\_data*

Definition at line 174 of file [ExecutionControlBase.cpp](#).

References [TrickHLA::ExecutionConfigurationBase::configure\\_attributes\(\)](#), [execution\\_configuration](#), [federate](#), [manager](#), and [TrickHLA::ExecutionConfigurationBase::setup\(\)](#).

Referenced by [TrickHLA::Federate::setup\(\)](#).

#### 7.19.3.97 `setup_checkpoint()`

```
void ExecutionControlBase::setup_checkpoint ( ) [virtual]
```

Setup the checkpoint data structures.

Definition at line 1037 of file [ExecutionControlBase.cpp](#).

References [execution\\_configuration](#), [TrickHLA::Object::is\\_instance\\_handle\\_valid\(\)](#), [TrickHLA::Object::mark\\_required\(\)](#), and [TrickHLA::Object::setup\\_ownership\\_transfer\\_checkpointed\\_data\(\)](#).

Referenced by [TrickHLA::Manager::setup\\_checkpoint\(\)](#).

#### 7.19.3.98 `setup_interaction_ref_attributes()`

```
virtual void TrickHLA::ExecutionControlBase::setup_interaction_ref_attributes ( ) [pure virtual]
```

Setup the [ExecutionControl](#) interaction Trick ref ATTRIBUTES.

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DSES::ExecutionControl](#), [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by [TrickHLA::Manager::setup\\_interaction\\_ref\\_attributes\(\)](#).

**7.19.3.99 setup\_interaction\_RTI\_handles()**

```
virtual void TrickHLA::ExecutionControlBase::setup_interaction_RTI_handles ( ) [pure virtual]
Setup the ExecutionControl interaction HLA RTI handles.
```

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DSES::ExecutionControl](#), [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by [TrickHLA::Manager::setup\\_all\\_RTI\\_handles\(\)](#).

**7.19.3.100 setup\_object\_ref\_attributes()**

```
virtual void TrickHLA::ExecutionControlBase::setup_object_ref_attributes ( ) [pure virtual]
Setup the ExecutionControl object Trick ref ATTRIBUTES.
```

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DSES::ExecutionControl](#), [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

**7.19.3.101 setup\_object\_RTI\_handles()**

```
virtual void TrickHLA::ExecutionControlBase::setup_object_RTI_handles ( ) [pure virtual]
Setup the ExecutionControl objects HLA RTI handles.
```

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DSES::ExecutionControl](#), [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by [TrickHLA::Manager::setup\\_all\\_RTI\\_handles\(\)](#).

**7.19.3.102 should\_print()**

```
bool ExecutionControlBase::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const
```

Determine if the verbose debug comments should be printed to the console.

**Returns**

Returns true if the requested message should print level.

**Parameters**

<i>level</i>	Debug level of the incoming message.
<i>code</i>	Source code association of the incoming messages.

Definition at line 1086 of file [ExecutionControlBase.cpp](#).

References [federate](#), and [TrickHLA::Federate::should\\_print\(\)](#).

Referenced by [DIS::ExecutionControl::announce\\_sync\\_point\(\)](#), [DSES::ExecutionControl::announce\\_sync\\_point\(\)](#), [IMSim::ExecutionControl::announce\\_sync\\_point\(\)](#), [SpaceFOM::ExecutionControl::announce\\_sync\\_point\(\)](#), [SpaceFO](#) M::ExecutionControl::check\_for\_shutdown(), [SpaceFOM::ExecutionControl::check\\_for\\_shutdown\\_with\\_termination\(\)](#), [IMSim::ExecutionControl::check\\_freeze\\_exit\(\)](#), [IMSim::ExecutionControl::check\\_pause\(\)](#), [IMSim::ExecutionControl::check\\_scenario\\_freeze\\_time\(\)](#), [clear\\_multiphase\\_init\\_sync\\_points\(\)](#), [DIS::ExecutionControl::enter\\_freeze\(\)](#), [IMSim::ExecutionControl::enter\\_freeze\(\)](#), [enter\\_freeze\(\)](#), [get\\_scenario\\_time\(\)](#), [get\\_sim\\_time\(\)](#), [initialize\(\)](#), [IMSim::ExecutionControl::mark\\_synchronized\(\)](#), [IMSim::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), [DSES::ExecutionControl::set\\_next\\_execution\\_control\\_mode\(\)](#), [SpaceFOM::ExecutionControl::set\\_next\\_execution\\_control\\_mode\(\)](#), [DIS::ExecutionControl::set\\_next\\_execution\\_control\\_mode\(\)](#), [IMSim::ExecutionControl::set\\_next\\_execution\\_control\\_mode\(\)](#), [SpaceFOM::ExecutionControl::shutdown\(\)](#), [TrickHLA::ExecutionConfigurationBase::wait\\_on\\_registration\(\)](#), and [~ExecutionControlBase\(\)](#).

### 7.19.3.103 shutdown()

virtual void TrickHLA::ExecutionControlBase::shutdown ( ) [pure virtual]

Execution control specific shutdown process.

Implemented in [DIS::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [DSES::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by [TrickHLA::Federate::shutdown\(\)](#).

### 7.19.3.104 shutdown\_mode\_announce()

virtual void TrickHLA::ExecutionControlBase::shutdown\_mode\_announce ( ) [pure virtual]

Announce to the federation execution that a shutdown is occurring.

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [TrickHLA::ExecutionControl](#), and [DSES::ExecutionControl](#).

### 7.19.3.105 shutdown\_mode\_transition()

virtual void TrickHLA::ExecutionControlBase::shutdown\_mode\_transition ( ) [pure virtual]

The shutdown mode transition routine.

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DIS::ExecutionControl](#), [TrickHLA::ExecutionControl](#), and [DSES::ExecutionControl](#).

### 7.19.3.106 start\_federation\_save\_at\_scenario\_time()

virtual void TrickHLA::ExecutionControlBase::start\_federation\_save\_at\_scenario\_time ( double *freeze\_scenario\_time*, const char \* *file\_name* ) [pure virtual]

Start the Federation save at the specified scenario time.

#### Parameters

<i>freeze_scenario_time</i>	Scenario time to freeze.
<i>file_name</i>	Checkpoint file name.

Implemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by [TrickHLA::Manager::start\\_federation\\_save\\_at\\_scenario\\_time\(\)](#).

### 7.19.3.107 subscribe()

virtual void TrickHLA::ExecutionControlBase::subscribe ( ) [pure virtual]

Subscribe to the [ExecutionControl](#) objects and interactions.

Implemented in [IMSim::ExecutionControl](#), [SpaceFOM::ExecutionControl](#), [DSES::ExecutionControl](#), [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Referenced by [TrickHLA::Manager::subscribe\(\)](#).

### 7.19.3.108 un\_freeze()

virtual void TrickHLA::ExecutionControlBase::un\_freeze ( ) [inline], [virtual]

Routine to handle [ExecutionControl](#) specific action needed to un-freeze.  
 Reimplemented in [IMSim::ExecutionControl](#).  
 Definition at line 423 of file ExecutionControlBase.hh.  
 Referenced by [TrickHLA::Federate::un\\_freeze\(\)](#).

#### 7.19.3.109 [unpublish\(\)](#)

```
virtual void TrickHLA::ExecutionControlBase::unpublish ( ) [pure virtual]
Unpublish the ExecutionControl objects and interactions.
Implemented in IMSim::ExecutionControl, SpaceFOM::ExecutionControl, DSES::ExecutionControl, DIS::ExecutionControl, and TrickHLA::ExecutionControl.  

Referenced by TrickHLA::Manager::unpublish\(\).
```

#### 7.19.3.110 [unsubscribe\(\)](#)

```
virtual void TrickHLA::ExecutionControlBase::unsubscribe ( ) [pure virtual]
Unsubscribe the ExecutionControl objects and interactions.
Implemented in IMSim::ExecutionControl, SpaceFOM::ExecutionControl, DSES::ExecutionControl, DIS::ExecutionControl, and TrickHLA::ExecutionControl.  

Referenced by TrickHLA::Manager::unsubscribe\(\).
```

#### 7.19.3.111 [wait\\_for\\_all\\_multiphase\\_init\\_sync\\_pnts\(\)](#)

```
void ExecutionControlBase::wait_for_all_multiphase_init_sync_pnts ( )
Wait for all the user defined mulit-phase initialization synchronization points if they are not already achieved and are not one of the predefined ExecutionControl synchronization points.
Trick Job Class: initialization
Definition at line 619 of file ExecutionControlBase.cpp.
References federate, multiphase_init_sync_pnt_list, and TrickHLA::SyncPntListBase::wait\_for\_list\_synchronization\(\).  

Referenced by clear\_multiphase\_init\_sync\_points\(\).
```

#### 7.19.3.112 [wait\\_for\\_sync\\_point\\_announce\(\)](#)

```
void ExecutionControlBase::wait_for_sync_point_announce (
    std::wstring const & sync_pnt_label ) [virtual]
Wait for a specified synchronization point label to be announced.
```

##### Parameters

<i>sync_pnt_label</i>	Synchronization label to wait for.
-----------------------	------------------------------------

Definition at line 428 of file ExecutionControlBase.cpp.  
 References [TrickHLA::Federate::check\\_for\\_shutdown\\_with\\_termination\(\)](#), [TrickHLA::SyncPnt::exists\(\)](#), [federate](#), [TrickHLA::SyncPnt::get\\_label\(\)](#), [TrickHLA::SyncPnt::is\\_announced\(\)](#), [TrickHLA::Federate::is\\_execution\\_member\(\)](#), [TrickHLA::SyncPntListBase::sync\\_point\\_list](#), [THLA\\_ENDL](#), and [TrickHLA::StringUtilities::to\\_string\(\)](#).  
 Referenced by [IMSim::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#).

#### 7.19.3.113 [wait\\_on\\_init\\_data\(\)](#)

```
virtual bool TrickHLA::ExecutionControlBase::wait_on_init_data ( ) [inline], [virtual]
```

Test to see if [ExecutionControl](#) needs to wait on initialization data.

Most [ExecutionControl](#) approaches require that we wait for the required initialization data. Currently, only the 'Simple' scheme does not.

#### Returns

True if [ExecutionControl](#) needs to wait on the initialization data.

Reimplemented in [TrickHLA::ExecutionControl](#).

Definition at line 257 of file `ExecutionControlBase.hh`.

Referenced by [TrickHLA::Manager::receive\\_init\\_data\(\)](#).

#### 7.19.3.114 `wait_on_init_sync_point()`

```
virtual bool TrickHLA::ExecutionControlBase::wait_on_init_sync_point ( ) [inline], [virtual]
```

Test to see if [ExecutionControl](#) needs to wait on initialization synchronization point.

Most [ExecutionControl](#) approaches require that we wait for specific initialization synchronization points in specific orders. Currently, only the 'Simple' and 'DIS' scheme do not.

#### Returns

True if [ExecutionControl](#) needs to wait on the initialization synchronization points.

Reimplemented in [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Definition at line 264 of file `ExecutionControlBase.hh`.

### 7.19.4 Friends And Related Function Documentation

#### 7.19.4.1 `init_attrTrickHLA__ExecutionControlBase`

```
void init_attrTrickHLA__ExecutionControlBase ( ) [friend]
```

#### 7.19.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 76 of file `ExecutionControlBase.hh`.

#### 7.19.4.3 `Manager`

```
friend class Manager [friend]
```

Definition at line 82 of file `ExecutionControlBase.hh`.

### 7.19.5 Field Documentation

#### 7.19.5.1 `cte_timeline`

```
CTETimelineBase* TrickHLA::ExecutionControlBase::cte_timeline
```

##### **Data I/O: \*\***

The Central Timing Equipment (CTE) timeline.

Definition at line 88 of file `ExecutionControlBase.hh`.

Referenced by `does_cte_timeline_exist()`, `get_cte_time()`, `initialize()`, `SpaceFOM::ExecutionControl::lateJoiner_hla_init_process()`, `SpaceFOM::ExecutionControl::post_multi_phase_init_processes()`, `DSES::ExecutionControl::process_execution_control_updates()`, `SpaceFOM::ExecutionControl::process_execution_control_updates()`, `DIS::ExecutionControl::process_execution_control_updates()`, `IMSim::ExecutionControl::process_execution_control_updates()`, `DS::ExecutionControl::process_mode_transition_request()`, `SpaceFOM::ExecutionControl::process_mode_transition_request()`, `DIS::ExecutionControl::process_mode_transition_request()`, and `IMSim::ExecutionControl::process_mode_transition_request()`.

### 7.19.5.2 current\_execution\_control\_mode

`ExecutionControlEnum TrickHLA::ExecutionControlBase::current_execution_control_mode [protected]`

**Units:** –

Current SRFOM federate current execution mode.

Definition at line 605 of file `ExecutionControlBase.hh`.

Referenced by `DSES::ExecutionControl::check_freeze_exit()`, `SpaceFOM::ExecutionControl::check_freeze_exit()`, `clear_mode_values()`, `DSES::ExecutionControl::freeze_mode_transition()`, `DIS::ExecutionControl::freeze_mode_transition()`, `IMSim::ExecutionControl::freeze_mode_transition()`, `SpaceFOM::ExecutionControl::freeze_mode_transition()`, `get_current_execution_control_mode()`, `is_in_freeze()`, `is_in_reconfig()`, `is_in_restart()`, `is_initializing()`, `is_running()`, `is_shutdown()`, `SpaceFOM::ExecutionControl::post_multi_phase_init_processes()`, `DSES::ExecutionControl::process_execution_control_updates()`, `SpaceFOM::ExecutionControl::process_execution_control_updates()`, `DIS::ExecutionControl::process_execution_control_updates()`, `IMSim::ExecutionControl::process_execution_control_updates()`, `DSES::ExecutionControl::process_mode_transition_request()`, `SpaceFOM::ExecutionControl::process_mode_transition_request()`, `DIS::ExecutionControl::process_mode_transition_request()`, `IMSim::ExecutionControl::process_mode_transition_request()`, `DSES::ExecutionControl::run_mode_transition()`, `DIS::ExecutionControl::run_mode_transition()`, `IMSim::ExecutionControl::run_mode_transition()`, `SpaceFOM::ExecutionControl::run_mode_transition()`, `set_current_execution_control_mode()`, `SpaceFOM::ExecutionControl::shutdown()`, `DSES::ExecutionControl::shutdown_mode_announce()`, `DIS::ExecutionControl::shutdown_mode_announce()`, `IMSim::ExecutionControl::shutdown_mode_announce()`, `SpaceFOM::ExecutionControl::shutdown_mode_announce()`, `DS::ExecutionControl::shutdown_mode_transition()`, `DIS::ExecutionControl::shutdown_mode_transition()`, `IMSim::ExecutionControl::shutdown_mode_transition()`, and `SpaceFOM::ExecutionControl::shutdown_mode_transition()`.

### 7.19.5.3 execution\_configuration

`ExecutionConfigurationBase* TrickHLA::ExecutionControlBase::execution_configuration [protected]`

**Units:** –

Associates `TrickHLA::ExecutionConfigurationBase` class object instance.

Since this is an abstract class, the actual instance will be a concrete derived class instance (e.g. `SRFOM::ExecutionControl`).

Definition at line 589 of file `ExecutionControlBase.hh`.

Referenced by `DSES::ExecutionControl::check_freeze_exit()`, `SpaceFOM::ExecutionControl::check_freeze_exit()`, `DSES::ExecutionControl::determine_federation_master()`, `TrickHLA::ExecutionControl::get_execution_configuration()`, `SpaceFOM::ExecutionControl::get_execution_configuration()`, `get_execution_configuration()`, `get_trickhla_object()`, `get_unregisterd_object()`, `get_unregisterd_remote_object()`, `DSES::ExecutionControl::initialize()`, `DIS::ExecutionControl::initialize()`, `IMSim::ExecutionControl::initialize()`, `initialize()`, `is_execution_configuration_used()`, `mark_object_as_deleted_from_federation()`, `object_instance_name_reservation_succeeded()`, `TrickHLA::ExecutionControl::pre_multi_phase_init_processes()`, `DSES::ExecutionControl::pre_multi_phase_init_processes()`, `DIS::ExecutionControl::pre_multi_phase_init_processes()`, `IMSim::ExecutionControl::pre_multi_phase_init_processes()`, `process_deleted_objects()`, `provide_attribute_update()`, `SpaceFOM::ExecutionControl::publish()`, `receive_cyclic_data()`, `receive_execution_configuration()`, `register_objects_with_RTI()`, `remove_execution_configuration()`, `send_execution_configuration()`, `SpaceFOM::ExecutionControl::send_init_root_ref_frame()`, `send_requested_data()`, `set_execution_configuration()`, `TrickHLA::ExecutionControl::set_least_common_time_step()`, `SpaceFOM::ExecutionControl::set_least_common_time_step()`, `set_master()`, `setup()`, `setup_checkpoint()`, `SpaceFOM::ExecutionControl::shutdown()`.

mode\_announce(), SpaceFOM::ExecutionControl::subscribe(), IMSim::ExecutionControl::subscribe(), SpaceFOM::ExecutionControl::unpublish(), IMSim::ExecutionControl::unpublish(), SpaceFOM::ExecutionControl::unsubscribe(), and IMSim::ExecutionControl::unsubscribe().

#### 7.19.5.4 federate

`TrickHLA::Federate* TrickHLA::ExecutionControlBase::federate [protected]`

**Data I/O: \*\***

Associated federate.

Definition at line 620 of file ExecutionControlBase.hh.

Referenced by IMSim::ExecutionControl::add\_freeze\_scenario\_time(), SpaceFOM::ExecutionControl::check\_for\_shutdown\_with\_termination(), DSES::ExecutionControl::check\_freeze\_exit(), DIS::ExecutionControl::check\_freeze\_exit(), IMSim::ExecutionControl::check\_freeze\_exit(), SpaceFOM::ExecutionControl::check\_freeze\_exit(), IMSim::ExecutionControl::check\_freeze\_time(), IMSim::ExecutionControl::check\_pause(), check\_pause\_at\_init(), IMSim::ExecutionControl::check\_scenario\_freeze\_time(), clear\_multiphase\_init\_sync\_points(), DIS::ExecutionControl::determine\_federation\_master(), IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate(), SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), DIS::ExecutionControl::enter\_freeze(), IMSim::ExecutionControl::enter\_freeze(), SpaceFOM::ExecutionControl::enter\_freeze(), enter\_freeze(), DIS::ExecutionControl::exit\_freeze(), IMSim::ExecutionControl::exit\_freeze(), SpaceFOM::ExecutionControl::freeze\_init(), DSES::ExecutionControl::freeze\_mode\_announce(), DIS::ExecutionControl::freeze\_mode\_announce(), IMSim::ExecutionControl::freeze\_mode\_announce(), SpaceFOM::ExecutionControl::freeze\_mode\_announce(), DSES::ExecutionControl::freeze\_mode\_transition(), DIS::ExecutionControl::freeze\_mode\_transition(), IMSim::ExecutionControl::freeze\_mode\_transition(), SpaceFOM::ExecutionControl::freeze\_mode\_transition(), get\_federate(), TrickHLA::ExecutionControl::initialize(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), DIS::ExecutionControl::is\_save\_initiated(), IMSim::ExecutionControl::is\_save\_initiated(), join\_federation\_process(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), IMSim::ExecutionControl::mark\_synchronized(), DSES::ExecutionControl::post\_multi\_phase\_init\_process(), DIS::ExecutionControl::post\_multi\_phase\_init\_process(), IMSim::ExecutionControl::post\_multi\_phase\_init\_process(), IMSim::ExecutionControl::post\_multi\_phase\_init\_processes(), TrickHLA::ExecutionControl::post\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), IMSim::ExecutionControl::process\_mode\_transition\_request(), receive\_execution\_configuration(), SpaceFOM::ExecutionControl::receive\_root\_ref\_frame(), SpaceFOM::ExecutionControl::role\_determination\_process(), DSES::ExecutionControl::run\_mode\_transition(), DIS::ExecutionControl::run\_mode\_transition(), IMSim::ExecutionControl::run\_mode\_transition(), SpaceFOM::ExecutionControl::run\_mode\_transition(), set\_federate(), setup(), should\_print(), SpaceFOM::ExecutionControl::shutdown(), DSES::ExecutionControl::shutdown\_mode\_transition(), DIS::ExecutionControl::shutdown\_mode\_transition(), IMSim::ExecutionControl::shutdown\_mode\_transition(), SpaceFOM::ExecutionControl::shutdown\_mode\_transition(), IMSim::ExecutionControl::start\_federation\_save\_at\_scenario\_time(), DIS::ExecutionControl::sync\_point\_registration\_failed(), DIS::ExecutionControl::sync\_point\_registration\_succeeded(), DSES::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), wait\_for\_all\_multiphase\_init\_sync\_pnts(), wait\_for\_sync\_point\_announce(), and SpaceFOM::ExecutionControl::wait\_on\_root\_frame\_discovered synchronization().

#### 7.19.5.5 init\_complete\_sp\_exists

`bool TrickHLA::ExecutionControlBase::init_complete_sp_exists [protected]`

**Units:** –

Internal flag, for Initialization Complete Sync-Point exists.

(default: false)

Definition at line 598 of file ExecutionControlBase.hh.

Referenced by IMSim::ExecutionControl::announce\_sync\_point(), SpaceFOM::ExecutionControl::announce\_sync\_point(), and does\_init\_complete\_sync\_point\_exist().

**7.19.5.6 late\_joiner**

```
bool TrickHLA::ExecutionControlBase::late_joiner [protected]
```

**Units:** –

Flag that this federate is a late joiner.

Definition at line 613 of file ExecutionControlBase.hh.

Referenced by IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate(), is\_late\_joiner(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), and SpaceFOM::ExecutionControl::role\_determination\_process().

**7.19.5.7 late\_joiner\_determined**

```
bool TrickHLA::ExecutionControlBase::late_joiner_determined [protected]
```

**Units:** –

Flag for late joiner determination.

Definition at line 614 of file ExecutionControlBase.hh.

Referenced by IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate(), is\_late\_joiner\_determined(), and SpaceFOM::ExecutionControl::role\_determination\_process().

**7.19.5.8 least\_common\_time\_step**

```
int64_t TrickHLA::ExecutionControlBase::least_common_time_step [protected]
```

**Units:** –

A 64 bit integer time that represents microseconds for the least common value of all the time step values in the federation execution (LCTS).

This value is set by the Master **Federate** and does not change during the federation execution. This is used in the computation to find the next HLA Logical Time Boundary (HLTB) available to all federates in the federation execution. The basic equation is  $HLTB = (\text{floor}(GALT/LCTS) + 1) * LCTS$ , where GALT is the greatest available logical time. This is used to synchronize the federates in a federation execution to be on a common logical time boundary.

Definition at line 577 of file ExecutionControlBase.hh.

Referenced by get\_least\_common\_time\_step(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), TrickHLA::ExecutionControl::set\_least\_common\_time\_step(), SpaceFOM::ExecutionControl::set\_least\_common\_time\_step(), set\_least\_common\_time\_step(), TrickHLA::ExecutionControl::set\_time\_padding(), and SpaceFOM::ExecutionControl::set\_time\_padding().

**7.19.5.9 loggable\_sync\_pts**

```
LoggableTimedSyncPnt* TrickHLA::ExecutionControlBase::loggable_sync_pts [protected]
```

**Units:** –

converted Sync Point data that gets checkpointed

Definition at line 624 of file ExecutionControlBase.hh.

Referenced by IMSim::ExecutionControl::convert\_loggable\_sync\_pts(), IMSim::ExecutionControl::reinstate\_logged\_sync\_pts(), and ~ExecutionControlBase().

### 7.19.5.10 logged\_sync\_pts\_count

`size_t TrickHLA::ExecutionControlBase::logged_sync_pts_count [protected]`

**Units:** –

number of logged sync pts

Definition at line 623 of file `ExecutionControlBase.hh`.

Referenced by `IMSim::ExecutionControl::convert_loggable_sync_pts()`, `IMSim::ExecutionControl::reinstate_logged_sync_pts()`, and `~ExecutionControlBase()`.

### 7.19.5.11 manager

`TrickHLA::Manager* TrickHLA::ExecutionControlBase::manager [protected]`

**Data I/O:** \*\*

Associated manager.

Definition at line 621 of file `ExecutionControlBase.hh`.

Referenced by `add_object_to_map()`, `SpaceFOM::ExecutionControl::check_for_shutdown_with_termination()`, `SpaceFOM::ExecutionControl::check_freeze_exit()`, `clear_multiphase_init_sync_points()`, `SpaceFOM::ExecutionControl::earlyJoinerHlaInitProcess()`, `get_manager()`, `initialize()`, `SpaceFOM::ExecutionControl::lateJoinerHlaInitProcess()`, `SpaceFOM::ExecutionControl::preMultiPhaseInitProcesses()`, `SpaceFOM::ExecutionControl::receiveInitRootRefFrame()`, `SpaceFOM::ExecutionControl::sendInitRootRefFrame()`, `setup()`, `SpaceFOM::ExecutionControl::setupInteractionRefAttributes()`, `SpaceFOM::ExecutionControl::setupInteractionRTIHandles()`, `TrickHLA::ExecutionControl::setupObjectRTIHandles()`, and `SpaceFOM::ExecutionControl::setupObjectRTIHandles()`.

### 7.19.5.12 master

`bool TrickHLA::ExecutionControlBase::master`

**Units:** –

Is true when this federate is the "master" federate for the Multiphase initialization process.

(default: false)

Definition at line 94 of file `ExecutionControlBase.hh`.

Referenced by `is_master()`, and `set_master()`.

### 7.19.5.13 mode\_transition\_requested

`bool TrickHLA::ExecutionControlBase::mode_transition_requested [protected]`

**Units:** –

Flag to indicate a mode transition has been requested.

Definition at line 603 of file `ExecutionControlBase.hh`.

Referenced by `clearModeTransitionRequested()`, `clearModeValues()`, `isModeTransitionRequested()`, and `setModeTransitionRequested()`.

### 7.19.5.14 multiphase\_init\_sync\_pnt\_list

`SyncPntList TrickHLA::ExecutionControlBase::multiphase_init_sync_pnt_list [protected]`

**Units:** –

Synchronization points used for multi-phase initialization control.

Definition at line 595 of file `ExecutionControlBase.hh`.

Referenced by `addMultiphaseInitSyncPoints()`, `SpaceFOM::ExecutionControl::roleDeterminationProcess()`, and `waitForAllMultiphaseInitSyncPnts()`.

### 7.19.5.15 multiphase\_init\_sync\_points

```
char* TrickHLA::ExecutionControlBase::multiphase_init_sync_points
```

**Units:** –

Comma-separated list of multi-phase initialization sync-points.

Definition at line 99 of file ExecutionControlBase.hh.

Referenced by add\_multiphase\_init\_sync\_points(), and ~ExecutionControlBase().

### 7.19.5.16 next\_mode\_cte\_time

```
double TrickHLA::ExecutionControlBase::next_mode_cte_time [protected]
```

**Units:** s

CTE time for next managed mode transition.

Definition at line 608 of file ExecutionControlBase.hh.

Referenced by DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), and IMSim::ExecutionControl::process\_execution\_control\_updates().

### 7.19.5.17 next\_mode\_scenario\_time

```
double TrickHLA::ExecutionControlBase::next_mode_scenario_time [protected]
```

**Units:** s

Scenario time for mode transition.

Definition at line 607 of file ExecutionControlBase.hh.

Referenced by DSES::ExecutionControl::set\_next\_execution\_control\_mode(), SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionControl::set\_next\_execution\_control\_mode(), and IMSim::ExecutionControl::set\_next\_execution\_control\_mode().

### 7.19.5.18 requested\_execution\_control\_mode

```
ExecutionControlEnum TrickHLA::ExecutionControlBase::requested_execution_control_mode [protected]
```

**Units:** –

The latest mode transition requested.

Definition at line 604 of file ExecutionControlBase.hh.

Referenced by clear\_mode\_values(), get\_requested\_execution\_control\_mode(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::set\_next\_execution\_control\_mode(), SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionControl::set\_next\_execution\_control\_mode(), IMSim::ExecutionControl::set\_next\_execution\_control\_mode(), and set\_requested\_execution\_control\_mode().

### 7.19.5.19 scenario\_freeze\_time

```
double TrickHLA::ExecutionControlBase::scenario_freeze_time [protected]
```

**Units:** s

Federation execution scenario time for freeze.

Definition at line 611 of file ExecutionControlBase.hh.

Referenced by get\_scenario\_freeze\_time(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), IMSim::ExecutionControl::process\_mode\_transition\_request(), DSES::ExecutionControl::set\_next\_execution\_control\_mode(), SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionControl::set\_next\_execution\_control\_mode(), IMSim::ExecutionControl::set\_next\_execution\_control\_mode(), and set\_scenario\_freeze\_time().

#### 7.19.5.20 scenario\_timeline

`ScenarioTimeline* TrickHLA::ExecutionControlBase::scenario_timeline`

**Data I/O:** \*\*

The scenario timeline.

Definition at line 86 of file ExecutionControlBase.hh.

Referenced by does\_scenario\_timeline\_exist(), SpaceFOM::ExecutionControl::epoch\_and\_root\_frame\_discovery\_process(), get\_scenario\_time(), initialize(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), IMSim::ExecutionConfiguration::pack(), SpaceFOM::ExecutionConfiguration::pack(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), IMSim::ExecutionControl::process\_mode\_transition\_request(), DSES::ExecutionControl::set\_next\_execution\_control\_mode(), SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionControl::set\_next\_execution\_control\_mode(), IMSim::ExecutionControl::set\_next\_execution\_control\_mode(), IMSim::ExecutionConfiguration::unpack(), and SpaceFOM::ExecutionConfiguration::unpack().

#### 7.19.5.21 sim\_timeline

`SimTimeline* TrickHLA::ExecutionControlBase::sim_timeline`

**Data I/O:** \*\*

The simulation timeline.

Definition at line 87 of file ExecutionControlBase.hh.

Referenced by does\_sim\_timeline\_exist(), get\_sim\_time(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), and IMSim::ExecutionControl::process\_mode\_transition\_request().

#### 7.19.5.22 simulation\_freeze\_time

`double TrickHLA::ExecutionControlBase::simulation_freeze_time [protected]`

**Units:** s

Trick simulation time for freeze.

Definition at line 610 of file ExecutionControlBase.hh.

Referenced by get\_simulation\_freeze\_time(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), IMSim::ExecutionControl::process\_mode\_transition\_request(), DSES::

ExecutionControl::set\_next\_execution\_control\_mode(), SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionControl::set\_next\_execution\_control\_mode(), IMSim::ExecutionControl::set\_next\_execution\_control\_mode(), and set\_simulation\_freeze\_time().

#### 7.19.5.23 time\_padding

```
double TrickHLA::ExecutionControlBase::time_padding [protected]
```

**Units:** s

Time in seconds to add to the go-to-run time.

Definition at line 575 of file ExecutionControlBase.hh.

Referenced by get\_time\_padding(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), IMSim::ExecutionControl::process\_mode\_transition\_request(), DSES::ExecutionControl::set\_next\_execution\_control\_mode(), SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionControl::set\_next\_execution\_control\_mode(), IMSim::ExecutionControl::set\_next\_execution\_control\_mode(), TrickHLA::ExecutionControl::set\_time\_padding(), SpaceFOM::ExecutionControl::set\_time\_padding(), and set\_time\_padding().

#### 7.19.5.24 use\_preset\_master

```
bool TrickHLA::ExecutionControlBase::use_preset_master
```

**Units:** –

Set to true to force the use of the preset value for the "master" flag.

(default: false)

Definition at line 91 of file ExecutionControlBase.hh.

Referenced by TrickHLA::ExecutionControl::initialize(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), and is\_master\_preset().

#### 7.19.5.25 wait\_sleep

```
unsigned int TrickHLA::ExecutionControlBase::wait_sleep [protected]
```

**Units:** us

Wait loop sleep times.

Definition at line 616 of file ExecutionControlBase.hh.

Referenced by SpaceFOM::ExecutionControl::role\_determination\_process(), and set\_wait\_sleep().

#### 7.19.5.26 wait\_timeout

```
unsigned int TrickHLA::ExecutionControlBase::wait_timeout [protected]
```

**Units:** us

Wait loop timeout.

Definition at line 617 of file ExecutionControlBase.hh.

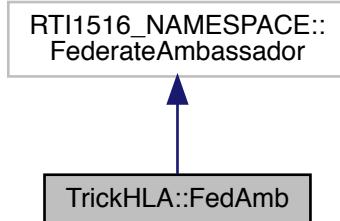
Referenced by SpaceFOM::ExecutionControl::role\_determination\_process(), and set\_wait\_timeout().

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

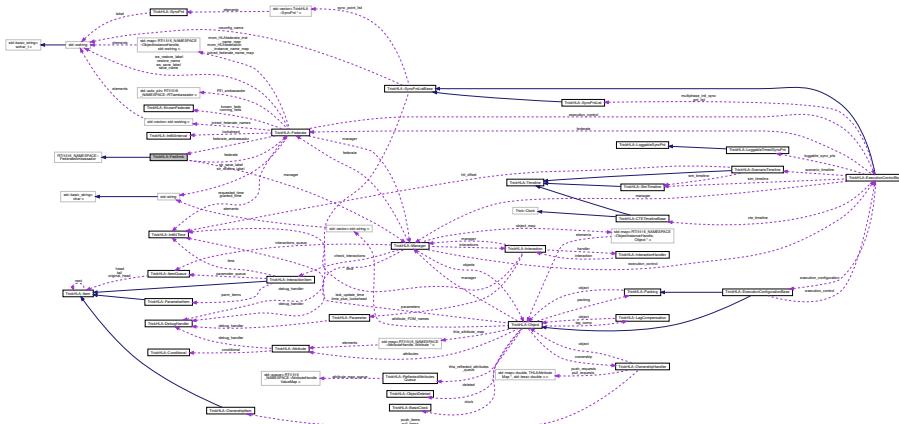
- [ExecutionControlBase.hh](#)
- [DIS/ExecutionControl.cpp](#)
- [ExecutionControlBase.cpp](#)

## 7.20 TrickHLA::FedAmb Class Reference

```
#include <FedAmb.hh>
Inheritance diagram for TrickHLA::FedAmb:
```



Collaboration diagram for TrickHLA::FedAmb:



### Public Member Functions

- `FedAmb ()`  
*Default constructor for the `TrickHLA FedAmb` class.*
- `virtual ~FedAmb () throw ()`  
*Destructor for the `TrickHLA FedAmb` class.*
- `Manager * get_manager ()`
- `void setup (Federate &federate, Manager &manager)`  
*Setup the required class instance associations.*
- `virtual void initialize ()`  
*Initialize the `TrickHLA Federate` Ambassador instance for this Federation Execution.*
- `bool should_print (const DebugLevelEnum &level, const DebugSourceEnum &code) const`  
*Determine if the verbose debug comments should be printed to the console.*

- virtual void `connectionLost` (std::wstring const &faultDescription) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `reportFederationExecutions` (RTI1516\_NAMESPACE::FederationExecutionInformationVector const &theFederationExecutionInformationList) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `synchronizationPointRegistrationSucceeded` (std::wstring const &label) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `synchronizationPointRegistrationFailed` (std::wstring const &label, RTI1516\_NAMESPACE::SynchronizationPointFailureReason reason) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `announceSynchronizationPoint` (std::wstring const &label, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `federationSynchronized` (std::wstring const &label, RTI1516\_NAMESPACE::FederateHandleSet const &failedToSyncSet) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `initiateFederateSave` (std::wstring const &label) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `initiateFederateSave` (std::wstring const &label, RTI1516\_NAMESPACE::LogicalTime const &theTime) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `federationSaved` () throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `federationNotSaved` (RTI1516\_NAMESPACE::SaveFailureReason theSaveFailureReason) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `federationSaveStatusResponse` (RTI1516\_NAMESPACE::FederateHandleSaveStatusPairVector const &theFederateStatusVector) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `requestFederationRestoreSucceeded` (std::wstring const &label) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `requestFederationRestoreFailed` (std::wstring const &label) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `federationRestoreBegin` () throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `initiateFederateRestore` (std::wstring const &label, std::wstring const &federateName, RTI1516\_NAMESPACE::FederateHandle handle) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `federationRestored` () throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `federationNotRestored` (RTI1516\_NAMESPACE::RestoreFailureReason theRestoreFailureReason) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `federationRestoreStatusResponse` (RTI1516\_NAMESPACE::FederateRestoreStatusVector const &theFederateRestoreStatusVector) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `startRegistrationForObjectClass` (RTI1516\_NAMESPACE::ObjectClassHandle theClass) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `stopRegistrationForObjectClass` (RTI1516\_NAMESPACE::ObjectClassHandle theClass) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `turnInteractionsOn` (RTI1516\_NAMESPACE::InteractionClassHandle theHandle) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `turnInteractionsOff` (RTI1516\_NAMESPACE::InteractionClassHandle theHandle) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `objectInstanceNameReservationSucceeded` (std::wstring const &theObjectInstanceName) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `objectInstanceNameReservationFailed` (std::wstring const &theObjectInstanceName) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `multipleObjectInstanceNameReservationSucceeded` (std::set< std::wstring > const &theObjectInstanceNames) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `multipleObjectInstanceNameReservationFailed` (std::set< std::wstring > const &theObjectInstanceNames) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `discoverObjectInstance` (RTI1516\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\_NAMESPACE::ObjectClassHandle theObjectClass, std::wstring const &theObjectInstanceName) throw ( RTI1516\_NAMESPACE::FederateInternalError )

- virtual void `discoverObjectInstance` (RTI1516\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\_NAMESPACE::ObjectClassHandle theObjectClass, std::wstring const &theObjectName, RTI1516\_NAMESPACE::FederateHandle producingFederate) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `reflectAttributeValues` (RTI1516\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\_NAMESPACE::AttributeHandleValueMap const &theAttributeValues, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag, RTI1516\_NAMESPACE::OrderType sentOrder, RTI1516\_NAMESPACE::TransportationType theType, RTI1516\_NAMESPACE::SupplementalReflectInfo theReflectInfo) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `reflectAttributeValues` (RTI1516\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\_NAMESPACE::AttributeHandleValueMap const &theAttributeValues, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag, RTI1516\_NAMESPACE::OrderType sentOrder, RTI1516\_NAMESPACE::TransportationType theType, RTI1516\_NAMESPACE::LogicalTime const &theTime, RTI1516\_NAMESPACE::OrderType receivedOrder, RTI1516\_NAMESPACE::SupplementalReflectInfo theReflectInfo) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `reflectAttributeValues` (RTI1516\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\_NAMESPACE::AttributeHandleValueMap const &theAttributeValues, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag, RTI1516\_NAMESPACE::OrderType sentOrder, RTI1516\_NAMESPACE::TransportationType theType, RTI1516\_NAMESPACE::LogicalTime const &theTime, RTI1516\_NAMESPACE::OrderType receivedOrder, RTI1516\_NAMESPACE::MessageRetractionHandle theHandle, RTI1516\_NAMESPACE::SupplementalReflectInfo theReflectInfo) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `receiveInteraction` (RTI1516\_NAMESPACE::InteractionClassHandle theInteraction, RTI1516\_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag, RTI1516\_NAMESPACE::OrderType sentOrder, RTI1516\_NAMESPACE::TransportationType theType, RTI1516\_NAMESPACE::SupplementalReceiveInfo theReceiveInfo) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `receiveInteraction` (RTI1516\_NAMESPACE::InteractionClassHandle theInteraction, RTI1516\_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag, RTI1516\_NAMESPACE::OrderType sentOrder, RTI1516\_NAMESPACE::TransportationType theType, RTI1516\_NAMESPACE::LogicalTime const &theTime, RTI1516\_NAMESPACE::OrderType receivedOrder, RTI1516\_NAMESPACE::SupplementalReceiveInfo theReceiveInfo) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `receiveInteraction` (RTI1516\_NAMESPACE::InteractionClassHandle theInteraction, RTI1516\_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag, RTI1516\_NAMESPACE::OrderType sentOrder, RTI1516\_NAMESPACE::TransportationType theType, RTI1516\_NAMESPACE::LogicalTime const &theTime, RTI1516\_NAMESPACE::OrderType receivedOrder, RTI1516\_NAMESPACE::MessageRetractionHandle theHandle, RTI1516\_NAMESPACE::SupplementalReceiveInfo theReceiveInfo) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `removeObjectInstance` (RTI1516\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag, RTI1516\_NAMESPACE::OrderType sentOrder, RTI1516\_NAMESPACE::SupplementalRemoveInfo theRemoveInfo) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `removeObjectInstance` (RTI1516\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag, RTI1516\_NAMESPACE::OrderType sentOrder, RTI1516\_NAMESPACE::LogicalTime const &theTime, RTI1516\_NAMESPACE::OrderType receivedOrder, RTI1516\_NAMESPACE::SupplementalRemoveInfo theRemoveInfo) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `removeObjectInstance` (RTI1516\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag, RTI1516\_NAMESPACE::OrderType sentOrder, RTI1516\_NAMESPACE::LogicalTime const &theTime, RTI1516\_NAMESPACE::OrderType receivedOrder, RTI1516\_NAMESPACE::MessageRetractionHandle theHandle, RTI1516\_NAMESPACE::SupplementalRemoveInfo theRemoveInfo) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `attributesInScope` (RTI1516\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes) throw ( RTI1516\_NAMESPACE::FederateInternalError )

- virtual void `attributesOutOfScope` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `provideAttributeValueUpdate` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `turnUpdatesOnForObjectInstance` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `turnUpdatesOnForObjectInstance` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes, std::wstring const &updateRateDesignator) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `turnUpdatesOffForObjectInstance` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `confirmAttributeTransportationTypeChange` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandle theAttributes, RTI1516\_NAMESPACE::TransportationType theTransportation) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `reportAttributeTransportationType` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandle theAttribute, RTI1516\_NAMESPACE::TransportationType theTransportation) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `confirmInteractionTransportationTypeChange` (RTI1516\_NAMESPACE::InteractionClassHandle theInteraction, RTI1516\_NAMESPACE::TransportationType theTransportation) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `reportInteractionTransportationType` (RTI1516\_NAMESPACE::FederateHandle federateHandle, RTI1516\_NAMESPACE::InteractionClassHandle theInteraction, RTI1516\_NAMESPACE::TransportationType theTransportation) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `requestAttributeOwnershipAssumption` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `requestDivestitureConfirmation` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `attributeOwnershipAcquisitionNotification` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &securedAttributes, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `attributeOwnershipUnavailable` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `requestAttributeOwnershipRelease` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes, RTI1516\_NAMESPACE::VariableLengthData const &theUserSuppliedTag) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `confirmAttributeOwnershipAcquisitionCancellation` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandleSet const &theAttributes) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `informAttributeOwnership` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandle theAttribute, RTI1516\_NAMESPACE::FederateHandle theOwner) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `attributeIsNotOwned` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandle theAttribute) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `attributeIsOwnedByRTI` (RTI1516\_NAMESPACE::ObjectHandle theObject, RTI1516\_NAMESPACE::AttributeHandle theAttribute) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `timeRegulationEnabled` (RTI1516\_NAMESPACE::LogicalTime const &theFederateTime) throw ( RTI1516\_NAMESPACE::FederateInternalError )

- virtual void `timeConstrainedEnabled` (RTI1516\_NAMESPACE::LogicalTime const &theFederateTime) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `timeAdvanceGrant` (RTI1516\_NAMESPACE::LogicalTime const &theTime) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- virtual void `requestRetraction` (RTI1516\_NAMESPACE::MessageRetractionHandle theHandle) throw ( RTI1516\_NAMESPACE::FederateInternalError )
- void `set_federation_restore_status_response_to_echo` ()
 

*Switch to echo (versus process) in a `federationRestoreStatusResponse()` callback...*
- void `set_federation_restore_status_response_to_process` ()
 

*Switch to process (versus echo) in a `federationRestoreStatusResponse()` callback...*
- void `set_federation_restored_rebuild_federate_handle_set` ()
 

*Enable the option to rebuild the federate handle set after a federation restore.*
- void `reset_federation_restored_rebuild_federate_handle_set` ()
 

*Disable the option to rebuild the federate handle set after a federation restore.*

## Protected Attributes

- Federate \* federate
 

**Units:** –  
Associated `TrickHLA::Federate`.
- Manager \* manager
 

**Units:** –  
Associated `TrickHLA::Manager`.

## Private Member Functions

- FedAmb (const `FedAmb` &rhs)
 

*Copy constructor for `FedAmb` class.*
- `FedAmb & operator=` (const `FedAmb` &rhs)
 

*Assignment operator for `FedAmb` class.*

## Private Attributes

- bool `federation_restore_status_response_context_switch`
- bool `federation_restored_rebuild_federate_handle_set`

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA__FedAmb` ()

### 7.20.1 Detailed Description

Definition at line 76 of file FedAmb.hh.

### 7.20.2 Constructor & Destructor Documentation

**7.20.2.1 FedAmb() [1/2]**

```
FedAmb::FedAmb ( )
```

Default constructor for the [TrickHLA FedAmb](#) class.

In most cases, we would allocate and set default names in the constructor. However, since we want this class to be Input Processor friendly, we cannot do that here since the Input Processor may not have been initialized yet. So, we have to set the name information to NULL and then allocate and set the defaults in the initialization job if not already set in the input stream. **Trick Job Class:** *initialization*

Definition at line 77 of file FedAmb.cpp.

**7.20.2.2 ~FedAmb()**

```
FedAmb::~FedAmb ( ) throw ( ) [virtual]
```

Destructor for the [TrickHLA FedAmb](#) class.

**Trick Job Class:** *shutdown*

Definition at line 92 of file FedAmb.cpp.

**7.20.2.3 FedAmb() [2/2]**

```
TrickHLA::FedAmb::FedAmb ( const FedAmb & rhs ) [private]
```

Copy constructor for [FedAmb](#) class.

This constructor is private to prevent inadvertent copies.

**7.20.3 Member Function Documentation****7.20.3.1 announceSynchronizationPoint()**

```
void FedAmb::announceSynchronizationPoint ( std::wstring const & label, RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 225 of file FedAmb.cpp.

**7.20.3.2 attributeIsNotOwned()**

```
void FedAmb::attributeIsNotOwned ( RTI1516_NAMESPACE::ObjectInstanceHandle theObject, RTI1516_NAMESPACE::AttributeHandle theAttribute ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1461 of file FedAmb.cpp.

References [THLA\\_NEWLINE](#).

**7.20.3.3 attributeIsOwnedByRTI()**

```
void FedAmb::attributeIsOwnedByRTI ( RTI1516_NAMESPACE::ObjectInstanceHandle theObject, RTI1516_NAMESPACE::AttributeHandle theAttribute ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1471 of file FedAmb.cpp.

References THLA\_NEWSLINE.

#### 7.20.3.4 attributeOwnershipAcquisitionNotification()

```
void FedAmb::attributeOwnershipAcquisitionNotification (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & securedAttributes,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1224 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, TrickHLA::Object::get\_attribute(), TrickHLA::Attribute::get\_FOM\_name(), TrickHLA::Object::get\_FOM\_name(), TrickHLA::Object::get\_name(), TrickHLA::Attribute::is\_locally\_owned(), TrickHLA::Attribute::is\_publish(), TrickHLA::Attribute::is\_remotely\_owned(), TrickHLA::Attribute::mark\_locally\_owned(), TrickHLA::Object::notify\_attribute\_ownership\_changed(), and THLA\_NEWSLINE.

#### 7.20.3.5 attributeOwnershipUnavailable()

```
void FedAmb::attributeOwnershipUnavailable (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1335 of file FedAmb.cpp.

References THLA\_NEWSLINE.

#### 7.20.3.6 attributesInScope()

```
void FedAmb::attributesInScope (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 909 of file FedAmb.cpp.

References THLA\_NEWSLINE.

#### 7.20.3.7 attributesOutOfScope()

```
void FedAmb::attributesOutOfScope (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 919 of file FedAmb.cpp.

References THLA\_NEWSLINE.

#### 7.20.3.8 confirmAttributeOwnershipAcquisitionCancellation()

```
void FedAmb::confirmAttributeOwnershipAcquisitionCancellation (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1440 of file FedAmb.cpp.

References THLA\_NEWLINE.

#### 7.20.3.9 `confirmAttributeTransportationTypeChange()`

```
void FedAmb::confirmAttributeTransportationTypeChange (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet theAttributes,
    RTI1516_NAMESPACE::TransportationType theTransportation ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 968 of file FedAmb.cpp.

References THLA\_NEWLINE.

#### 7.20.3.10 `confirmInteractionTransportationTypeChange()`

```
void FedAmb::confirmInteractionTransportationTypeChange (
    RTI1516_NAMESPACE::InteractionClassHandle theInteraction,
    RTI1516_NAMESPACE::TransportationType theTransportation ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 988 of file FedAmb.cpp.

References THLA\_NEWLINE.

#### 7.20.3.11 `connectionLost()`

```
void FedAmb::connectionLost (
    std::wstring const & faultDescription ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 174 of file FedAmb.cpp.

References THLA\_ENDL, and TrickHLA::StringUtilities::to\_string().

#### 7.20.3.12 `discoverObjectInstance()` [1/2]

```
void FedAmb::discoverObjectInstance (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::ObjectClassHandle theObjectClass,
    std::wstring const & theObjectName ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 514 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

#### 7.20.3.13 `discoverObjectInstance()` [2/2]

```
void FedAmb::discoverObjectInstance (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::ObjectClassHandle theObjectClass,
    std::wstring const & theObjectName,
    RTI1516_NAMESPACE::FederateHandle producingFederate ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 547 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

#### 7.20.3.14 **federationNotRestored()**

```
void FedAmb::federationNotRestored (
    RTI1516_NAMESPACE::RestoreFailureReason theRestoreFailureReason ) throw ( RTI1516_N←
AMESPACE::FederateInternalError)    [virtual]
```

Definition at line 373 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

#### 7.20.3.15 **federationNotSaved()**

```
void FedAmb::federationNotSaved (
    RTI1516_NAMESPACE::SaveFailureReason theSaveFailureReason ) throw ( RTI1516_NAMESPA←
CE::FederateInternalError)    [virtual]
```

Definition at line 292 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

#### 7.20.3.16 **federationRestoreBegin()**

```
void FedAmb::federationRestoreBegin ( ) throw ( RTI1516_NAMESPACE::FederateInternalError)    [virtual]
```

Definition at line 340 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, federate, TrickHLA::Federate::set\_restore\_begun(), should\_print(), and THLA\_NEWLINE.

#### 7.20.3.17 **federationRestored()**

```
void FedAmb::federationRestored ( ) throw ( RTI1516_NAMESPACE::FederateInternalError)    [virtual]
```

Definition at line 364 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, federate, TrickHLA::Federate::set\_restore\_completed(), should\_print(), and THLA\_NEWLINE.

#### 7.20.3.18 **federationRestoreStatusResponse()**

```
void FedAmb::federationRestoreStatusResponse (
    RTI1516_NAMESPACE::FederateRestoreStatusVector const & theFederateRestoreStatusVector
) throw ( RTI1516_NAMESPACE::FederateInternalError)    [virtual]
```

Definition at line 384 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

#### 7.20.3.19 **federationSaved()**

```
void FedAmb::federationSaved ( ) throw ( RTI1516_NAMESPACE::FederateInternalError)    [virtual]
```

Definition at line 281 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, federate, TrickHLA::Federate::federation\_saved(), TrickHLA::Federate::set\_save\_completed(), TrickHLA::Federate::set\_start\_to\_save(), should\_print(), and THLA\_NEWLINE.

**7.20.3.20 federationSaveStatusResponse()**

```
void FedAmb::federationSaveStatusResponse (
    RTI1516_NAMESPACE::FederateHandleSaveStatusPairVector const & theFederateStatusVector
) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 308 of file FedAmb.cpp.
```

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

**7.20.3.21 federationSynchronized()**

```
void FedAmb::federationSynchronized (
    std::wstring const & label,
    RTI1516_NAMESPACE::FederateHandleSet const & failedToSyncSet ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 232 of file FedAmb.cpp.
```

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

**7.20.3.22 get\_manager()**

```
Manager* TrickHLA::FedAmb::get_manager ( ) [inline]
```

Definition at line 99 of file FedAmb.hh.

References manager.

**7.20.3.23 informAttributeOwnership()**

```
void FedAmb::informAttributeOwnership (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandle theAttribute,
    RTI1516_NAMESPACE::FederateHandle theOwner ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 1450 of file FedAmb.cpp.
```

References THLA\_NEWLINE.

**7.20.3.24 initialize()**

```
void FedAmb::initialize ( ) [virtual]
```

Initialize the [TrickHLA Federate](#) Ambassador instance for this Federation Execution.

**Trick Job Class:** *initialization*

Definition at line 114 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, federate, TrickHLA::Federate::get\_federate\_name(), manager, should\_print(), THLA\_ENDL, THLA\_NEWLINE, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::Federate::initialize().

**7.20.3.25 initiateFederateRestore()**

```
void FedAmb::initiateFederateRestore (
    std::wstring const & label,
    std::wstring const & federateName,
```

```
RTI1516_NAMESPACE::FederateHandle handle ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 349 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

#### 7.20.3.26 **initiateFederateSave()** [1/2]

```
void FedAmb::initiateFederateSave (
    std::wstring const & label ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 258 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, and THLA_NEWLINE.
```

#### 7.20.3.27 **initiateFederateSave()** [2/2]

```
void FedAmb::initiateFederateSave (
    std::wstring const & label,
    RTI1516_NAMESPACE::LogicalTime const & theTime ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 269 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, and THLA_NEWLINE.
```

#### 7.20.3.28 **multipleObjectInstanceNameReservationFailed()**

```
void FedAmb::multipleObjectInstanceNameReservationFailed (
    std::set< std::wstring > const & theObjectInstanceNames ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 493 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

#### 7.20.3.29 **multipleObjectInstanceNameReservationSucceeded()**

```
void FedAmb::multipleObjectInstanceNameReservationSucceeded (
    std::set< std::wstring > const & theObjectInstanceNames ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 473 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

#### 7.20.3.30 **objectInstanceNameReservationFailed()**

```
void FedAmb::objectInstanceNameReservationFailed (
    std::wstring const & theObjectInstanceName ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 457 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

**7.20.3.31 objectInstanceNameReservationSucceeded()**

```
void FedAmb::objectInstanceNameReservationSucceeded (
    std::wstring const & theObjectInstanceName ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 441 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

**7.20.3.32 operator=( )**

```
FedAmb& TrickHLA::FedAmb::operator= (
    const FedAmb & rhs ) [private]
```

Assignment operator for **FedAmb** class.

This assignment operator is private to prevent inadvertent copies.

**7.20.3.33 provideAttributeValueUpdate()**

```
void FedAmb::provideAttributeValueUpdate (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 929 of file FedAmb.cpp.
```

**7.20.3.34 receiveInteraction() [1/3]**

```
void FedAmb::receiveInteraction (
    RTI1516_NAMESPACE::InteractionClassHandle theInteraction,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::OrderType sentOrder,
    RTI1516_NAMESPACE::TransportationType theType,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    RTI1516_NAMESPACE::OrderType receivedOrder,
    RTI1516_NAMESPACE::MessageRetractionHandle theHandle,
    RTI1516_NAMESPACE::SupplementalReceiveInfo theReceiveInfo ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 792 of file FedAmb.cpp.
```

References TrickHLA::DEBUG\_LEVEL\_8\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

**7.20.3.35 receiveInteraction() [2/3]**

```
void FedAmb::receiveInteraction (
    RTI1516_NAMESPACE::InteractionClassHandle theInteraction,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::OrderType sentOrder,
    RTI1516_NAMESPACE::TransportationType theType,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    RTI1516_NAMESPACE::OrderType receivedOrder,
```

```
RTI1516_NAMESPACE::SupplementalReceiveInfo theReceiveInfo ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 764 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_8_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, and THLA_NEWLINE.
```

### 7.20.3.36 receiveInteraction() [3/3]

```
void FedAmb::receiveInteraction (
    RTI1516_NAMESPACE::InteractionClassHandle theInteraction,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::OrderType sentOrder,
    RTI1516_NAMESPACE::TransportationType theType,
    RTI1516_NAMESPACE::SupplementalReceiveInfo theReceiveInfo ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 736 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_8_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, TrickHLA::Int64Time::get(), and THLA_NEWLINE.
```

### 7.20.3.37 reflectAttributeValues() [1/3]

```
void FedAmb::reflectAttributeValues (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleValueMap const & theAttributeValues,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::OrderType sentOrder,
    RTI1516_NAMESPACE::TransportationType theType,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    RTI1516_NAMESPACE::OrderType receivedOrder,
    RTI1516_NAMESPACE::MessageRetractionHandle theHandle,
    RTI1516_NAMESPACE::SupplementalReflectInfo theReflectInfo ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 686 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_8_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, TrickHLA::Object::extract_data(), TrickHLA::Object::get_name(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Object::set_last_update_time(), TrickHLA::Int64Time::setTo(), THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

### 7.20.3.38 reflectAttributeValues() [2/3]

```
void FedAmb::reflectAttributeValues (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleValueMap const & theAttributeValues,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::OrderType sentOrder,
    RTI1516_NAMESPACE::TransportationType theType,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    RTI1516_NAMESPACE::OrderType receivedOrder,
    RTI1516_NAMESPACE::SupplementalReflectInfo theReflectInfo ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
Definition at line 636 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_8_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, TrickHLA::Object::extract_data(), TrickHLA::Object::get_name(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Object::set_last_update_time(), TrickHLA::Int64Time::setTo(), THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

`_update_time()`, `TrickHLA::Int64Time::setTo()`, `THLA_NEWLINE`, and `TrickHLA::StringUtilities::to_string()`.

#### 7.20.3.39 `reflectAttributeValues()` [3/3]

```
void FedAmb::reflectAttributeValues (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleValueMap const & theAttributeValues,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::OrderType sentOrder,
    RTI1516_NAMESPACE::TransportationType theType,
    RTI1516_NAMESPACE::SupplementalReflectInfo theReflectInfo ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 565 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_LEVEL_8_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, TrickHLA::Object::extract_data(), TrickHLA::Object::get_name(), THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

#### 7.20.3.40 `removeObjectInstance()` [1/3]

```
void FedAmb::removeObjectInstance (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::OrderType sentOrder,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    RTI1516_NAMESPACE::OrderType receivedOrder,
    RTI1516_NAMESPACE::MessageRetractionHandle theHandle,
    RTI1516_NAMESPACE::SupplementalRemoveInfo theRemoveInfo ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 879 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

#### 7.20.3.41 `removeObjectInstance()` [2/3]

```
void FedAmb::removeObjectInstance (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag,
    RTI1516_NAMESPACE::OrderType sentOrder,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    RTI1516_NAMESPACE::OrderType receivedOrder,
    RTI1516_NAMESPACE::SupplementalRemoveInfo theRemoveInfo ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 850 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

#### 7.20.3.42 `removeObjectInstance()` [3/3]

```
void FedAmb::removeObjectInstance (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag,
```

```
RTI1516_NAMESPACE::OrderType sentOrder,
RTI1516_NAMESPACE::SupplementalRemoveInfo theRemoveInfo ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 821 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, THLA_NEWLINE, and TrickHLA::StringUtilities::to_string().
```

#### 7.20.3.43 reportAttributeTransportationType()

```
void FedAmb::reportAttributeTransportationType (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandle theAttribute,
    RTI1516_NAMESPACE::TransportationType theTransportation ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 978 of file FedAmb.cpp.
References THLA_NEWLINE.
```

#### 7.20.3.44 reportFederationExecutions()

```
void FedAmb::reportFederationExecutions (
    RTI1516_NAMESPACE::FederationExecutionInformationVector const & theFederationExecutionInformationList ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 188 of file FedAmb.cpp.
References THLA_NEWLINE.
```

#### 7.20.3.45 reportInteractionTransportationType()

```
void FedAmb::reportInteractionTransportationType (
    RTI1516_NAMESPACE::FederateHandle federateHandle,
    RTI1516_NAMESPACE::InteractionClassHandle theInteraction,
    RTI1516_NAMESPACE::TransportationType theTransportation ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 997 of file FedAmb.cpp.
References THLA_NEWLINE.
```

#### 7.20.3.46 requestAttributeOwnershipAssumption()

```
void FedAmb::requestAttributeOwnershipAssumption (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
Definition at line 1012 of file FedAmb.cpp.
References TrickHLA::DEBUG_LEVEL_3_TRACE, TrickHLA::DEBUG_LEVEL_8_TRACE, TrickHLA::DEBUG_SOURCE_FED_AMB, TrickHLA::Object::get_attribute(), TrickHLA::Attribute::get_FOM_name(), TrickHLA::Object::get_FOM_name(), TrickHLA::Object::get_name(), TrickHLA::Object::grant_push_request_pthread(), TrickHLA::Attribute::is_locally_owned(), TrickHLA::Attribute::is_publish(), TrickHLA::Attribute::is_remotely_owned(), TrickHLA::Object::lock(), TrickHLA::Attribute::set_push_requested(), THLA_NEWLINE, TrickHLA::StringUtilities::to_string(), and TrickHLA::Object::unlock().
```

**7.20.3.47 requestAttributeOwnershipRelease()**

```
void FedAmb::requestAttributeOwnershipRelease (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes,
    RTI1516_NAMESPACE::VariableLengthData const & theUserSuppliedTag ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1345 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_LEVEL\_8\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, TrickHLA::Object::get\_attribute(), TrickHLA::Attribute::get\_FOM\_name(), TrickHLA::Object::get\_FOM\_name(), TrickHLA::Object::get\_name(), TrickHLA::Attribute::is\_locally\_owned(), TrickHLA::Attribute::is\_remotely\_owned(), TrickHLA::Attribute::set\_pull\_requested(), TrickHLA::Object::set\_pull\_requested(), and THLA\_NEWLINE.

**7.20.3.48 requestDivestitureConfirmation()**

```
void FedAmb::requestDivestitureConfirmation (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1131 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, TrickHLA::Object::get\_attribute(), TrickHLA::Attribute::get\_FOM\_name(), TrickHLA::Object::get\_FOM\_name(), TrickHLA::Object::get\_name(), TrickHLA::Attribute::is\_locally\_owned(), TrickHLA::Attribute::is\_remotely\_owned(), TrickHLA::Attribute::set\_divest\_requested(), TrickHLA::Object::set\_divest\_requested(), THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

**7.20.3.49 requestFederationRestoreFailed()**

```
void FedAmb::requestFederationRestoreFailed (
    std::wstring const & label ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 329 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

**7.20.3.50 requestFederationRestoreSucceeded()**

```
void FedAmb::requestFederationRestoreSucceeded (
    std::wstring const & label ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 318 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

**7.20.3.51 requestRetraction()**

```
void FedAmb::requestRetraction (
    RTI1516_NAMESPACE::MessageRetractionHandle theHandle ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1551 of file FedAmb.cpp.

References THLA\_NEWLINE.

**7.20.3.52 reset\_federation\_restored\_rebuild\_federate\_handle\_set()**

```
void TrickHLA::FedAmb::reset_federation_restored_rebuild_federate_handle_set ( ) [inline]
```

Disable the option to rebuild the federate handle set after a federation restore.  
 Definition at line 494 of file FedAmb.hh.  
 References `federation_restored_rebuild_federate_handle_set`.  
 Referenced by `TrickHLA::Federate::restore_federate_handles_from_MOM()`.

#### 7.20.3.53 `set_federation_restore_status_response_to_echo()`

```
void TrickHLA::FedAmb::set_federation_restore_status_response_to_echo ( ) [inline]
Switch to echo (versus process) in a federationRestoreStatusResponse\(\) callback...
Definition at line 478 of file FedAmb.hh.
References federation_restore_status_response_context_switch.
Referenced by TrickHLA::Federate::requested_federation_restore_status().
```

#### 7.20.3.54 `set_federation_restore_status_response_to_process()`

```
void TrickHLA::FedAmb::set_federation_restore_status_response_to_process ( ) [inline]
Switch to process (versus echo) in a federationRestoreStatusResponse\(\) callback...
Definition at line 483 of file FedAmb.hh.
References federation_restore_status_response_context_switch.
Referenced by TrickHLA::Federate::initiate_restore_announce().
```

#### 7.20.3.55 `set_federation_restored_rebuild_federate_handle_set()`

```
void TrickHLA::FedAmb::set_federation_restored_rebuild_federate_handle_set ( ) [inline]
Enable the option to rebuild the federate handle set after a federation restore.
Definition at line 489 of file FedAmb.hh.
References federation_restored_rebuild_federate_handle_set.
Referenced by TrickHLA::Federate::restore_federate_handles_from_MOM().
```

#### 7.20.3.56 `setup()`

```
void FedAmb::setup (
    Federate & federate,
    Manager & manager )
```

Setup the required class instance associations.

##### Parameters

<code>federate</code>	Associated <code>TrickHLA::Federate</code> class instance.
<code>manager</code>	Associated <code>TrickHLA::Manager</code> class instance.

##### Trick Job Class: *initialization*

Definition at line 99 of file FedAmb.cpp.  
 References federate, and manager.  
 Referenced by `TrickHLA::Federate::setup()`.

#### 7.20.3.57 `should_print()`

```
bool FedAmb::should_print (
    const DebugLevelEnum & level,
```

```
const DebugSourceEnum & code ) const
```

Determine if the verbose debug comments should be printed to the console.

#### Returns

Returns true if the requested message should print level.

#### Parameters

<i>level</i>	Debug level of the incoming message.
<i>code</i>	Source code association of the incoming messages.

Definition at line 160 of file FedAmb.cpp.

References manager, and TrickHLA::Manager::should\_print().

Referenced by federationRestoreBegin(), federationRestored(), federationSaved(), initialize(), and TrickHLA::Federate::should\_print().

#### 7.20.3.58 startRegistrationForObjectClass()

```
void FedAmb::startRegistrationForObjectClass (
    RTI1516_NAMESPACE::ObjectClassHandle theClass ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 402 of file FedAmb.cpp.

References THLA\_NEWLINE.

#### 7.20.3.59 stopRegistrationForObjectClass()

```
void FedAmb::stopRegistrationForObjectClass (
    RTI1516_NAMESPACE::ObjectClassHandle theClass ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 410 of file FedAmb.cpp.

References THLA\_NEWLINE.

#### 7.20.3.60 synchronizationPointRegistrationFailed()

```
void FedAmb::synchronizationPointRegistrationFailed (
    std::wstring const & label,
    RTI1516_NAMESPACE::SynchronizationPointFailureReason reason ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 209 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

#### 7.20.3.61 synchronizationPointRegistrationSucceeded()

```
void FedAmb::synchronizationPointRegistrationSucceeded (
    std::wstring const & label ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 196 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

### 7.20.3.62 timeAdvanceGrant()

```
void FedAmb::timeAdvanceGrant (
    RTI1516_NAMESPACE::LogicalTime const & theTime ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1524 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, TrickHLA::Int64Time::getDoubleTime(), and THLA\_NEWLINE.

### 7.20.3.63 timeConstrainedEnabled()

```
void FedAmb::timeConstrainedEnabled (
    RTI1516_NAMESPACE::LogicalTime const & theFederateTime ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1504 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

### 7.20.3.64 timeRegulationEnabled()

```
void FedAmb::timeRegulationEnabled (
    RTI1516_NAMESPACE::LogicalTime const & theFederateTime ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 1484 of file FedAmb.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FED\_AMB, and THLA\_NEWLINE.

### 7.20.3.65 turnInteractionsOff()

```
void FedAmb::turnInteractionsOff (
    RTI1516_NAMESPACE::InteractionClassHandle theHandle ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 427 of file FedAmb.cpp.

References THLA\_NEWLINE.

### 7.20.3.66 turnInteractionsOn()

```
void FedAmb::turnInteractionsOn (
    RTI1516_NAMESPACE::InteractionClassHandle theHandle ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 418 of file FedAmb.cpp.

References THLA\_NEWLINE.

### 7.20.3.67 turnUpdatesOffForObjectInstance()

```
void FedAmb::turnUpdatesOffForObjectInstance (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes ) throw ( RTI1516_NAMESPACE::FederateInternalError ) [virtual]
```

Definition at line 959 of file FedAmb.cpp.

References THLA\_NEWLINE.

### 7.20.3.68 turnUpdatesOnForObjectInstance() [1/2]

```
void FedAmb::turnUpdatesOnForObjectInstance (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes ) throw ( RTI1516_NAMESPAC
E::FederateInternalError) [virtual]
```

Definition at line 940 of file FedAmb.cpp.

References THLA\_NEWLINE.

### 7.20.3.69 turnUpdatesOnForObjectInstance() [2/2]

```
void FedAmb::turnUpdatesOnForObjectInstance (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes,
    std::wstring const & updateRateDesignator ) throw ( RTI1516_NAMESPACE::FederateInternalError) [virtual]
```

Definition at line 949 of file FedAmb.cpp.

References THLA\_NEWLINE.

## 7.20.4 Friends And Related Function Documentation

### 7.20.4.1 init\_attrTrickHLA\_\_FedAmb

```
void init_attrTrickHLA__FedAmb ( ) [friend]
```

### 7.20.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 83 of file FedAmb.hh.

## 7.20.5 Field Documentation

### 7.20.5.1 federate

```
Federate* TrickHLA::FedAmb::federate [protected]
```

**Units:** –

Associated [TrickHLA::Federate](#).

Definition at line 89 of file FedAmb.hh.

Referenced by [federationRestoreBegin\(\)](#), [federationRestored\(\)](#), [federationSaved\(\)](#), [initialize\(\)](#), and [setup\(\)](#).

### 7.20.5.2 federation\_restore\_status\_response\_context\_switch

```
bool TrickHLA::FedAmb::federation_restore_status_response_context_switch [private]
```

Definition at line 500 of file FedAmb.hh.

Referenced by [set\\_federation\\_restore\\_status\\_response\\_to\\_echo\(\)](#), and [set\\_federation\\_restore\\_status\\_response\\_to\\_process\(\)](#).

### 7.20.5.3 federation\_restored\_rebuild\_federate\_handle\_set

```
bool TrickHLA::FedAmb::federation_restored_rebuild_federate_handle_set [private]
```

Definition at line 501 of file FedAmb.hh.

Referenced by `reset_federation_restored_rebuild_federate_handle_set()`, and `set_federation_restored_rebuild_federate_handle_set()`.

### 7.20.5.4 manager

```
Manager* TrickHLA::FedAmb::manager [protected]
```

**Units:** –

Associated [TrickHLA::Manager](#).

Definition at line 90 of file FedAmb.hh.

Referenced by `get_manager()`, `initialize()`, `setup()`, and `should_print()`.

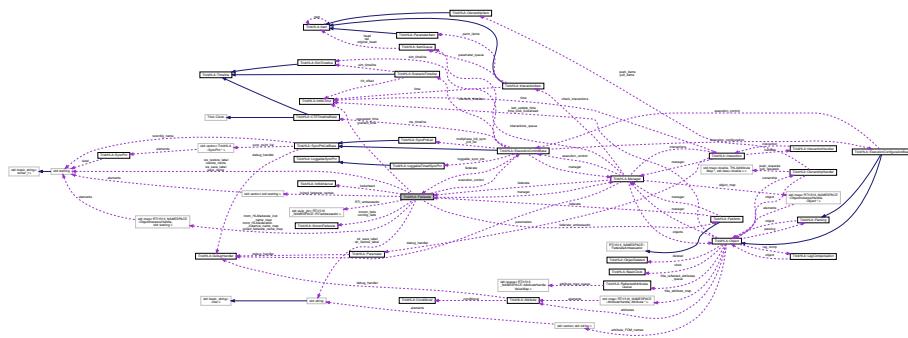
The documentation for this class was generated from the following files:

- [FedAmb.hh](#)
- [FedAmb.cpp](#)

## 7.21 TrickHLA::Federate Class Reference

```
#include <Federate.hh>
```

Collaboration diagram for TrickHLA::Federate:



### Public Member Functions

- [Federate \(\)](#)  
*Default constructor for the [TrickHLA Federate](#) class.*
- [~Federate \(\)](#)  
*Destructor for the [TrickHLA Federate](#) class.*
- [void print\\_version \(\) const](#)  
*Print the [TrickHLA](#) version string.*
- [void fix\\_FPU\\_control\\_word \(\)](#)  
*Check, and if necessary, fix the FPU Control Word.*
- [void setup \(FedAmb &federate\\_amb, Manager &federate\\_manager, ExecutionControlBase &federate\\_execution\\_control\)](#)  
*Setup the required class instance associations.*
- [void initialize \(\)](#)

- Composite initialization routine for an object instance of a [Federate](#) class.*
- `void pre_multiphase_initialization ()`

*Begin the pre-multiphase initialization process of standing up the federate in the federation execution.*
  - `void post_multiphase_initialization ()`

*Complete the post-multiphase initialization startup process prior to the federation execution going into run.*
  - `bool should_print (const DebugLevelEnum &level, const DebugSourceEnum &code) const`

*Determine if the verbose debug comments should be printed to the console.*
  - `void create_RTI_ambassador_and_connect ()`

*Create the RTI ambassador and connect to the RTI.*
  - `void create_and_join_federation ()`

*Create and then join the Federation.*
  - `void enable_async_delivery ()`

*Enable asynchronous delivery of messages for this federate.*
  - `void register_generic_sync_point (std::wstring const &label, double time=-1.0)`

*Register a generic synchronization point; i.e. not a multiphase init sync-point.*
  - `void achieve_and_wait_for_synchronization (std::wstring const &label)`

*Achieve the specified sync-point and wait for the federation to be synchronized on it.*
  - `void achieve_synchronization_point (std::wstring const &label)`

*Achieve the specified sync-point and do NOT wait for the federation to be synchronized on it.*
  - `void announce_sync_point (std::wstring const &label, RTI1516\_USERDATA const &user_supplied_tag)`

*The RTI has announced the existence of a synchronization point.*
  - `void sync_point_registration_succeeded (std::wstring const &label)`

*Marks a synchronization point as registered in the federation.*
  - `void sync_point_registration_failed (std::wstring const &label, bool not_unique)`

*Callback from [TrickHLA::FedAmb](#) through for when registration of a synchronization point fails. and is one of the sync-points created.*
  - `void federation_synchronized (std::wstring const &label)`

*Marks a synchronization point as synchronized with the federation.*
  - `std::string wait_for_required_federates_to_join ()`

*Wait for all the required federates to joined the federation.*
  - `const RTI1516\_NAMESPACE::FederateHandleSet & get_joined_federate_handles ()`

*Get a const reference to the joined federate handles.*
  - `void initialize_MOM_handles ()`

*Initialize the MOM interface handles.*
  - `void ask_MOM_for_federate_names ()`

*Request names of joined federates from the MOM.*
  - `void unsubscribe_all_HLAfederate_class_attributes_from_MOM ()`

*Unsubscribe from all MOM federate class attributes.*
  - `void unsubscribe_all_HLAfederation_class_attributes_from_MOM ()`

*Unsubscribe from all MOM federation class attributes.*
  - `void ask_MOM_for_auto_provide_setting ()`

*Ask MOM for the current "auto-provide" setting from the switches table.*
  - `void enable_MOM_auto_provide_setting (bool enable)`

*Update the MOM "auto-provide" setting from the switches table with the setting.*
  - `void backup_auto_provide_setting_from_MOM_then_disable ()`

*Backup the current "auto-provide" setting from the switches table then disable auto-provide if it was enabled.*
  - `void restore_orig_MOM_auto_provide_setting ()`

- Restore the backed up "auto-provide" state to the MOM.*
- void `add_federate_instance_id` (RTI1516\_NAMESPACE::ObjectInstanceHandle instance\_hdl)  
*Add the specified **Federate** instance ID to the list of discovered federates.*
  - void `remove_federate_instance_id` (RTI1516\_NAMESPACE::ObjectInstanceHandle instance\_hdl)  
*Remove the specified **Federate** instance ID from the list of discovered federates.*
  - void `load_and_print_running_federate_names` ()  
*Load the running federate names from the RTI.*
  - void `clear_running_feds` ()  
*Deallocate running federates based on current known information in preparation for re-size.*
  - void `update_running_feds` ()  
*Update running federates based on current known information.*
  - void `add_a_single_entry_into_running_feds` ()  
*Grow the running\_feds by one entry.*
  - int `get_running_feds_count` () const  
*Get the count of the currently running federates.*
  - void `add_MOM_HLAfederate_instance_id` (RTI1516\_NAMESPACE::ObjectInstanceHandle instance\_hdl, std::wstring const &instance\_name)  
*Add the specified MOM HLAfederate instance ID to the list of discovered federates.*
  - void `remove_MOM_HLAfederate_instance_id` (RTI1516\_NAMESPACE::ObjectInstanceHandle instance\_hdl)  
*Remove the specified **Federate** instance ID to the list of discovered federates.*
  - void `setup_checkpoint` ()  
*Perform setup for federate save.*
  - void `perform_checkpoint` ()  
*Federates that did not announce the save, perform a checkpoint.*
  - void `post_checkpoint` ()  
*Complete federate save.*
  - void `setup_restore` ()  
*Perform setup for federate restore.*
  - void `perform_restore` ()  
*Federates that did not announce the restore, perform a restore.*
  - void `post_restore` ()  
*Complete federate restore and prepare to restart execution.*
  - bool `is_HLA_save_and_restore_supported` ()  
*Returns true if HLA save and restore is supported by the user specified simulation initialization scheme.*
  - void `restore_checkpoint` (char \*file\_name)  
*Restore checkpoint.*
  - void `inform_RTI_of_restore_completion` ()  
*Inform the RTI of the success or failure of the federate restore.*
  - void `read_running_feds_file` (char \*file\_name) throw ( const char \* )  
*Read the running\_feds file, replacing the data in known federates data structure.*
  - void `copy_running_feds_into_known_feds` ()  
*Copies the contents of the checkpoint's list of federates into known federates data structure.*
  - void `restart_checkpoint` ()  
*Restart the sim from a checkpoint.*
  - void `federation_saved` ()  
*Federation save completed.*
  - void `federation_restored` ()

- Federation restore completed.*
- void `wait_for_federation_restore_begun ()`  
*Blocks until the federation restore has begun.*
  - void `wait_until_federation_is_ready_to_restore ()`  
*Blocks until the federation is ready to restore.*
  - std::string `wait_for_federation_restore_to_complete ()`  
*Blocks until the federation restore is complete.*
  - void `wait_for_restore_request_callback ()`  
*Blocks until the RTI responds with a federation request request success / failure.*
  - void `wait_for_restore_status_to_complete ()`  
*Blocks until the RTI responds with a federation status of the restore is complete.*
  - void `wait_for_save_status_to_complete ()`  
*Blocks until the RTI responds with a federation status of the save is complete.*
  - void `wait_for_federation_restore_failed_callback_to_complete ()`  
*Blocks until the RTI responds with a federation not restored callback via the federate ambassador.*
  - void `request_federation_save_status ()`  
*Requests the status of the Federation Save.*
  - void `request_federation_restore_status ()`  
*Requests the status of the Federation Restore.*
  - bool `has_restore_process_restore_request_failed () const`  
*Query if restore process restore request failed.*
  - bool `has_restore_process_restore_request_succeeded () const`  
*Query if restore process restore request succeeded.*
  - bool `has_restore_request_failed () const`  
*Query if restore request failed.*
  - bool `has_restore_request_succeeded () const`  
*Query if restore request succeeded.*
  - void `set_announce_save ()`  
*Set the announce save flag.*
  - void `set_save_completed ()`  
*Set the save completed state.*
  - void `set_restore_begun ()`  
*Set the restore begun state.*
  - void `set_restore_completed ()`  
*Set the restore completed state.*
  - void `set_restore_failed ()`  
*Set the restore failed state.*
  - void `set_restore_request_failed ()`  
*Set the restore request failed state.*
  - void `set_restore_request_succeeded ()`  
*Set the restore request succeeded state.*
  - bool `should_publish_data () const`  
*Query if federate should publish data.*
  - bool `is_start_to_restore () const`  
*Query if federate has started a restore process.*
  - void `set_restore_is_imminent ()`  
*Set the restore is imminent flag.*

- void `requested_federation_restore_status` (bool status)  
*Sets the Restore filename and flag.*
- void `print_requested_federation_restore_status` (RTI1516\_NAMESPACE::FederateRestoreStatusVector const &status\_vector)  
*Prints the federation restore status from the RTI.*
- void `process_requested_federation_restore_status` (RTI1516\_NAMESPACE::FederateRestoreStatusVector const &status\_vector)  
*Processes the federation restore status received from the RTI.*
- void `process_requested_federation_save_status` (RTI1516\_NAMESPACE::FederateHandleSaveStatusPair<Vector const &status\_vector>)  
*Processes the federation save status received from the RTI.*
- void `print_restore_failure_reason` (RTI1516\_NAMESPACE::RestoreFailureReason reason)  
*Prints the reason for the federation restore failure.*
- void `print_save_failure_reason` (RTI1516\_NAMESPACE::SaveFailureReason reason)  
*Prints the reason for the federation save failure.*
- void `set_checkpoint_file_name` (const char \*name)  
*Save the supplied checkpoint file name.*
- void `initiate_save_announce` ()  
*Sets the Save filename and flag.*
- void `initiate_restore_announce` (const char \*restore\_name)  
*Sets the Restore filename and flag.*
- bool `has_restore Been announced` () const  
*Sets the Save filename and flag.*
- void `complete_restore` ()  
*Informs of completion of federation restore.*
- bool `is_federate_executing` () const  
*Checks for the existence 'startup' initialization sync point as an indication if this federate is running.*
- void `convert_sync_pts` ()  
*Converts HLA sync points into something Trick can save in a checkpoint.*
- void `reinstate_logged_sync_pts` ()  
*Converts checkpointed sync points into HLA sync points.*
- void `set_start_to_save` ()  
*Set the start to save flag.*
- bool `check_for_shutdown` ()  
*Checks to see if shutdown has been commanded.*
- bool `check_for_shutdown_with_termination` ()  
*Checks to see if shutdown has been commanded and, if so, terminates the simulation.*
- void `check_HLA_save_directory` ()  
*Check if HLA\_save\_directory is empty. If so, ask the EXECUTIVE for info and build the absolute path of the RUN directory.*
- void `set_federate_has_begun_execution` ()  
*Set the federate has begun execution state.*
- void `restore_federate_handles_from_MOM` ()  
*Ask for all federate handles from MOM after a checkpoint reload.*
- void `rebuild_federate_handles` (RTI1516\_NAMESPACE::ObjectInstanceHandle instance\_hdl, RTI1516\_NAMESPACE::AttributeHandleValueMap const &values)  
*Reloading the federate handle set from the MOM after a checkpoint reload.*
- void `setup_time_constrained` ()

- void [setup\\_time\\_regulation \(\)](#)

*Setup this federate's constrained time management.*
- void [setup\\_time\\_management \(\)](#)

*Setup this federate's regulate time management.*
- void [time\\_advance\\_request \(\)](#)

*Setup this federate's time management.*
- void [time\\_advance\\_request\\_to\\_GALT \(\)](#)

*Increment the requested time by the lookahead time and make a HLA time advance request.*
- void [time\\_advance\\_request\\_to\\_GALT\\_LCTS\\_multiple \(\)](#)

*Moves the federates time to the Greatest Available Logical Time (GALT) that is an integer multiple of the Least-Common-Time-Step (LCTS) time if we are time constrained and Not time regulating.*
- void [wait\\_for\\_time\\_advance\\_grant \(\)](#)

*Move the requested time to an integer multiple of the Greatest Available Logical Time (GALT) and Least Common Time Step (LCTS).*
- void [set\\_startup \(bool flag\)](#)

*Wait for a HLA time-advance grant.*
- void [wait\\_for\\_time\\_advance\\_grant \(int time\\_out\\_tolerance\)](#)

*Wait for a HLA time-advance grant, but allow for an early exit if it takes longer than time\_out\_tolerance (for SSTF).*
- bool [get\\_freeze\\_announced \(\)](#)

*Set federate execution startup state.*
- void [set\\_freeze\\_announced \(bool flag\)](#)

*Set that federation execution freeze has been announced.*
- bool [get\\_freeze\\_announced \(\)](#)

*Get that federation execution freeze announced flag state.*
- bool [get\\_freeze\\_pending \(\)](#)

*Get that federation execution freeze pending flag state.*
- void [unfreeze \(\)](#)

*Perform federation execution freeze process.*
- void [shutdown \(\)](#)

*Shutdown the federate.*
- void [shutdown\\_time\\_management \(\)](#)

*Shutdown this federate's time management.*
- void [shutdown\\_time\\_constrained \(\)](#)

*Shutdown this federate's time constrained time management.*
- void [shutdown\\_time\\_regulating \(\)](#)

*Shutdown this federate's time regulating time management.*
- void [resign \(\)](#)

*Resign from the federation.*
- void [resign\\_so\\_we\\_can\\_rejoin \(\)](#)

*Resign from the federation in a way that permits rejoining later.*
- void [destroy \(\)](#)

*Destroy the federation if this is the last federate.*
- void [destroy\\_orphaned\\_federation \(\)](#)

*Destroy the federation if it was orphaned from a previous simulation run that did not shutdown cleanly.*
- bool [is\\_federate\\_instance\\_id \(RTI1516\\_NAMESPACE::ObjectInstanceHandle id\)](#)

*Determine if the specified instance ID is for one of the discovered federates.*
- bool [is\\_MOM\\_HLAfederate\\_class \(RTI1516\\_NAMESPACE::ObjectClassHandle federate\\_class\) const](#)

*Check with the MOM if the is an HLAfederate class.*

- RTI1516\_NAMESPACE::ObjectClassHandle `get_MOM_HLAfederate_class_handle () const`  
*Get the federate class handle for this federate from the MOM.*
- void `set_MOM_HLAfederate_instance_attributes (RTI1516_NAMESPACE::ObjectInstanceHandle id, RTI1516_NAMESPACE::AttributeHandleValueMap const &values)`  
*Set the Federates name given the instance ID as well as the FederateHandle ID associated with the Federate instance.*
- void `set_all_federate_MOM_instance_handles_by_name ()`  
*Set all the federate MOM instance handles by using the previously saved named for the MOM object instance associated with the federate.*
- void `determine_federate_MOM_object_instance_names ()`  
*Get the federate MOM object instance names so that we can recover the MOM instance handles associated with each federate when a checkpoint restore happens.*
- bool `is_MOM_HLAfederation_instance_id (RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl)`  
*Determine if the specified instance handle is an MOM HLAfederation instance.*
- void `add_MOM_HLAfederation_instance_id (RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl)`  
*Add the specified MOM HLAfederation instance handle to the list of running federates.*
- void `remove_MOM_HLAfederation_instance_id (RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl)`  
*Remove the specified MOM HLAfederation instance handle from the list of running federates.*
- bool `is_MOM_HLAfederation_class (RTI1516_NAMESPACE::ObjectClassHandle class_hdl) const`  
*Query if the an object class handle is a federation class.*
- void `set_MOM_HLAfederation_instance_attributes (RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl, RTI1516_NAMESPACE::AttributeHandleValueMap const &values)`  
*Set the Federation ID given the instance ID as well as the FederateHandle ID associated with the Federation instance.*
- RTI1516\_NAMESPACE::RTIambassador \* `get_RTI_ambassador ()`  
*Get the pointer to the associated HLA RTI Ambassador instance.*
- FedAmb \* `get_fed_ambassador ()`  
*Get the pointer to the associated TrickHLA Federate Ambassador instance.*
- Manager \* `get_manager ()`  
*Get the pointer to the associated TrickHLA::Manager instance.*
- ExecutionControlBase \* `get_execution_control ()`  
*Get the pointer to the associated TrickHLA::Manager instance.*
- const char \* `get_federate_name () const`  
*Get the pointer to the associated federate name.*
- const char \* `get_federate_type () const`  
*Get the pointer to the associated federate type.*
- const char \* `get_federation_name () const`  
*Get the pointer to the associated federation execution name.*
- double `get_granted_time () const`  
*Get the current granted federation execution time in seconds.*
- double `get_requested_time () const`  
*Get the requested federation execution time in seconds.*
- const Int64Time & `get_granted_fed_time () const`  
*Get the current granted federation execution time.*
- const Int64Time & `get_requested_fed_time () const`  
*Get the requested federation execution time.*
- const Int64Interval & `get_lookahead () const`  
*Get the current federate lookahead time.*
- const double `get_lookahead_time () const`  
*Get the current federate lookahead time in seconds.*

- const bool `is_zero_lookahead_time` () const  
*Query of federate has a zero lookahead time.*
- void `set_save_name` (std::wstring const &save\_label)  
*Set the name of the save.*
- void `set_restore_name` (std::wstring const &restore\_label)  
*Set the name of the restore.*
- bool `get_restart` () const  
*Get restart state.*
- bool `get_restart_cfg` () const  
*Get restart configuration state.*
- void `get_stale_data_counter` (int \*s)  
*Get stale data counter (DIS only).*
- void `set_federation_name` (const char \*const exec\_name)  
*Set the name of the federation execution.*
- bool `is_time_advance_granted` () const  
*Query if time advance has been granted.*
- void `set_time_advance_grant` (const bool &grant\_flag)  
*Set the time advance grant flag.*
- bool `in_time_regulating_state` () const  
*Query if the federate is in a time regulating state.*
- void `set_time_regulation_state` (const bool &regulation\_state)  
*Set the state of time regulation.*
- void `set_time_constrained_state` (const bool &constrained\_state)  
*Set the state of time constraint.*
- void `set_granted_time` (double time)  
*Sets the granted time from the specified double.*
- void `set_granted_time` (RTI1516\_NAMESPACE::LogicalTime const &time)  
*Sets the granted time from the specified LogicalTime.*
- void `set_requested_time` (double time)  
*Sets the requested time from the specified double.*
- void `set_requested_time` (RTI1516\_NAMESPACE::LogicalTime const &time)  
*Sets the requested time from the specified LogicalTime.*
- void `set_lookahead` (double value)  
*Sets the HLA lookahead time.*
- void `set_start_to_save` (bool save\_flag)  
*Set start to save flag.*
- void `set_start_to_restore` (bool restore\_flag)  
*Set start to restore flag.*
- void `set_restart` (bool restart\_now)  
*Set restart flag.*
- void `set_restart_cfg` (bool restart\_cfg\_now)  
*Set restart configuration flag.*
- bool `is_time_management_enabled` () const  
*Query if time management is enabled.*
- void `restart_initialization` ()  
*Perform initialization after a restart.*
- bool `federate_can_rejoin_federation` () const

- bool `is_a_required_startup_federate` (std::wstring const &fed\_name)
 

*Query if a federate is required at startup.*
- bool `is_federation_created_by_federate` () const
 

*Query if the federation was created by this federate.*
- bool `is_execution_member` ()
 

*Is the federate an execution member, which means is it connected and joined to a federation execution.*
- void `freeze_init` ()
 

*Routine to handle going from run to freeze.*
- void `check_freeze` ()
 

*Check for exit from freeze.*
- void `enter_freeze` ()
 

*Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.*
- void `exit_freeze` ()
 

*Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.*

## Data Fields

- char \* `name`

**Units:** –  
*The federate name.*
- char \* `type`

**Units:** –  
*The federate type.*
- char \* `federation_name`

**Units:** –  
*Federation execution name.*
- char \* `local_settings`

**Units:** –  
*Vendor specific HLA-Evolved local settings for the connect API.*
- char \* `FOM_modules`

**Units:** –  
*FOM filename for the IEEE 1516-2000 and SISO-STD-004.1-2004 standards, or a comma separated list of FOM-module filenames for IEEE 1516-2010.*
- char \* `MIM_module`

**Units:** –  
*Filename for the MOM and Initialization Module (MIM) for HLA IEEE 1516-2010.*
- double `lookahead_time`

**Units:** s  
*The HLA lookahead time in seconds.*
- bool `time_regulating`

**Units:** –  
*HLA Time Regulation flag (default: true).*
- bool `time_constrained`

**Units:** –  
*HLA Time Constrained flag (default: true).*
- bool `time_management`

**Units:** –  
*Enable HLA Time Management flag (default: true).*
- bool `enable_known_feds`

- **Units:** –  
Enable use of known Federates list (default: true)
  - int `known_feds_count`
    - **Units:** –  
Number of required Federates (default: 0)
  - KnownFederate \* `known_feds`
    - **Units:** –  
Array of all the known Federates in the simulation.
  - bool `can_rejoin_federation`
    - **Units:** –  
Enables this federate to resign in a way to allow re-joining of the federation at a later time.
  - double `freeze_delay_frames`
    - **Units:** –  
For `DIS`: Number of lookahead\_time frames to delay when freeze issued so all feds freeze together.
  - bool `unfreeze_after_save`
    - **Units:** –  
Flag to indicate that we should go to run immediately after a save.

## Private Member Functions

- void `write_running_feds_file` (char \*file\_name) throw ( const char \* )
 

Dumps the contents of the `running_feds` object into the supplied file name with ".running\_feds" appended to it.
- void `request_federation_save` ()
 

Request federation save from the RTI.
- void `subscribe_attributes` (RTI1516\_NAMESPACE::ObjectClassHandle class\_handle, RTI1516\_NAMESPACE::AttributeHandleSet const &attribute\_list)
 

Subscribe to the specified attributes for the given class handle.
- void `unsubscribe_attributes` (RTI1516\_NAMESPACE::ObjectClassHandle class\_handle, RTI1516\_NAMESPACE::AttributeHandleSet const &attribute\_list)
 

Unsubscribe from the specified attributes for the given class handle.
- void `request_attribute_update` (RTI1516\_NAMESPACE::ObjectClassHandle class\_handle, RTI1516\_NAMESPACE::AttributeHandleSet const &attribute\_list)
 

Request an update to the specified attributes for the given object class handle.
- void `publish_interaction_class` (RTI1516\_NAMESPACE::InteractionClassHandle class\_handle)
 

Publish `Interaction` class.
- void `unpublish_interaction_class` (RTI1516\_NAMESPACE::InteractionClassHandle class\_handle)
 

Unpublish `Interaction` class.
- void `send_interaction` (RTI1516\_NAMESPACE::InteractionClassHandle class\_handle, RTI1516\_NAMESPACE::ParameterHandleValueMap const &parameter\_list)
 

Send the `Interaction` for the specified interaction class and parameter list.
- void `create_federation` ()
 

Create the simulation federation if it does not already exist.
- void `join_federation` (const char \*const federate\_name, const char \*const federate\_type)
 

Join a federation.
- bool `is_required_federate` (std::wstring const &federate\_name)
 

Determine if the specified federate name is a required federate.
- bool `is_joined_federate` (const char \*federate\_name)
 

Determine if the specified federate name is a joined federate.
- bool `is_joined_federate` (std::wstring const &federate\_name)

- void `perform_time_advance_request ()`

*Make the HLA time-advance request using the current requested\_time value.*
- void `un_freeze ()`

*Unfreeze simulation.*
- `Federate (const Federate &rhs)`

*Copy constructor for `Federate` class.*
- `Federate & operator= (const Federate &rhs)`

*Assignment operator for `Federate` class.*

## Private Attributes

- RTI1516\_NAMESPACE::FederateHandle `federate_id`

**Data I/O:** \*\*  
*Federate ID.*
- bool `federation_created_by_federate`

**Data I/O:** \*\*  
*Federate successfully created the federation if True.*
- bool `federation_exists`

**Data I/O:** \*\*  
*Federation exists.*
- bool `federation_joined`

**Data I/O:** \*\*  
*Federate joined federation flag.*
- bool `all_federates_joined`

**Units:** –  
*Master check for all federates joined.*
- `Int64Interval lookahead`

**Units:** –  
*Lookahead time for data.*
- bool `shutdown_called`

**Units:** –  
*Flag to indicate shutdown has been called.*
- std::wstring `save_name`

**Data I/O:** \*\*  
*Name for a save file*
- std::wstring `restore_name`

**Data I/O:** \*\*  
*Name for a restore file*
- char \* `HLA_save_directory`

**Data I/O:** \*i  
**Units:** –  
*HLA Save directory*
- bool `initiate_save_flag`

**Data I/O:** \*\*  
*Save announce flag*
- `THLASaveRestoreProcEnum restore_process`

**Data I/O:** \*\*  
*Where we are in the restore process*
- `THLASaveRestoreProcEnum prev_restore_process`

- **Data I/O:** \*\*  
*previous state of the restore process*
- bool **initiate\_restore\_flag**  
**Data I/O:** \*\*  
*Restore announce flag*
- bool **restore\_in\_progress**  
**Data I/O:** \*\*  
*Restore in progress flag*
- bool **restore\_failed**  
**Data I/O:** \*\*  
*Restore of the federate failed*
- bool **restore\_is\_imminent**  
**Data I/O:** \*\*  
*Restore has been signalled by the Manager*
- char **cstr\_save\_label** [256]  
**Data I/O:** \*\*  
*Save file label in C string format*
- std::string **str\_save\_label**  
**Data I/O:** \*\*  
*Save file label in C++ string format*
- std::wstring **ws\_save\_label**  
**Data I/O:** \*\*  
*Save file label in wide string format*
- bool **announce\_save**  
**Data I/O:** \*\*  
*flag to indicate whether we have announced the federation save*
- bool **save\_label\_generated**  
**Data I/O:** \*\*  
*Save filename has been generated.*
- bool **save\_request\_complete**  
**Data I/O:** \*\*  
*save status request complete*
- bool **save\_completed**  
**Data I/O:** \*\*  
*Save completed.*
- int **stale\_data\_counter**  
**Units:** –  
*For DIS only: Number of cycles since the last time we received data via HLA.*
- char **cstr\_restore\_label** [256]  
**Data I/O:** \*\*  
*Restore file label in C string format*
- std::string **str\_restore\_label**  
**Data I/O:** \*\*  
*Restore file label in C++ string format*
- std::wstring **ws\_restore\_label**  
**Data I/O:** \*\*  
*Restore file label in wide string format*
- bool **announce\_restore**  
**Data I/O:** \*\*  
*flag to indicate whether we have announced the federation restore*
- bool **restore\_label\_generated**

- bool `restore_begun`

**Data I/O:** \*\*  
*Restore filename has been generated.*
- bool `restore_request_complete`

**Data I/O:** \*\*  
*Restore begun*
- bool `restore_completed`

**Data I/O:** \*\*  
*Restore completed.*
- bool `federation_restore_failed_callback_complete`

**Data I/O:** \*\*  
*federation not restored callback complete*
- bool `federate_has_been_restarted`

**Data I/O:** \*\*  
*Federate has restarted; so, do not restart again!*
- bool `publish_data`

**Data I/O:** \*\*  
*Default true.*
- int `running_feds_count`

**Units:** –  
*Number of running Federates (default: 0)*
- KnownFederate \* `running_feds`

**Units:** –  
*Checkpoint-able Array of running Federation Federates*
- int `running_feds_count_at_time_of_restore`

**Data I/O:** \*\*  
*Number of running Federates at the time of the restore (default: 0)*
- char `checkpoint_file_name` [256]
  - Data I/O:** \*i  
**Units:** –  
*label to attach to sync point*
- Flag `checkpoint_rt_itimer`

**Data I/O:** \*\*  
*loaded checkpoint RT ITIMER*
- bool `announce_freeze`

**Data I/O:** \*\*  
*DANNY2.7 flag to indicate that this federate is announcing go to freeze mode*
- bool `freeze_the_federation`

**Data I/O:** \*\*  
*DANNY2.7 flag to indicate the federation is going into freeze now*
- bool `execution_has_begun`

**Units:** –  
*flag to indicate if the federate has begun simulation execution.*
- bool `time_adv_grant`

**Units:** –  
*Time advance grant flag.*
- Int64Time `granted_time`

**Units:** –  
*HLA time given by RTI*

- `Int64Time requested_time`

**Units:** –  
*requested/desired HLA time*
- `double HLA_time`

**Units:** *s*  
*Current HLA time.*
- `bool start_to_save`

**Data I/O:** \*\*  
*Save flag*
- `bool start_to_restore`

**Data I/O:** \*\*  
*Restore flag*
- `bool restart_flag`

**Data I/O:** \*\*  
*Restart flag*
- `bool restart_cfg_flag`

**Data I/O:** \*\*  
*Restart flag with new configuration*
- `bool time_regulating_state`

**Units:** –  
*Internal flag, federates HLA Time Regulation state (default: false).*
- `bool time_constrained_state`

**Units:** –  
*Internal flag, federates HLA Time Constrained state (default: false).*
- `bool got_startup_sp`

**Units:** –  
*"startup" SP has been created. For DIS compatibility*
- `bool make_copy_of_run_directory`

**Units:** –  
*Make a backup of RUN directory before restarting the federation via federation manager (default: false).*
- `RTI1516_NAMESPACE::ObjectClassHandle MOM_HLAfederation_class_handle`

**Data I/O:** \*\*  
*MOM Federation class handle.*
- `RTI1516_NAMESPACE::AttributeHandle MOM_HLAfederatesInFederation_handle`

**Data I/O:** \*\*  
*MOM attribute handle to Federate-count.*
- `RTI1516_NAMESPACE::AttributeHandle MOM_HLAautoProvide_handle`

**Data I/O:** \*\*  
*MOM AutoProvide attribute handle.*
- `TrickHLAObjInstNameMap mom_HLAfederation_instance_name_map`

**Data I/O:** \*\*  
*Map of the MOM HLAfederation instances.*
- `int auto_provide_setting`

**Units:** –  
*MOM Federation wide HLAautoProvide setting.*
- `int orig_auto_provide_setting`

**Units:** –  
*Original MOM Federation wide HLAautoProvide setting when we joined the federation.*
- `RTI1516_NAMESPACE::ObjectClassHandle MOM_HLAfederate_class_handle`

**Data I/O:** \*\*  
*MOM Federate class handle.*

- RTI1516\_NAMESPACE::AttributeHandle `MOM_HLAfederateType_handle`

**Data I/O:** \*\*  
*MOM attribute handle to `Federate` type (a.k.a name in IEEE 1516-2000).*
- RTI1516\_NAMESPACE::AttributeHandle `MOM_HLAfederateName_handle`

**Data I/O:** \*\*  
*MOM attribute handle to `Federate` name.*
- RTI1516\_NAMESPACE::AttributeHandle `MOM_HLAfederate_handle`

**Data I/O:** \*\*  
*MOM attribute handle to Federate-Handle.*
- `TrickHLAObjInstanceNameMap mom_HLAfederate_inst_name_map`

**Data I/O:** \*\*  
*Map of the MOM HLAfederate instances name map.*
- `TrickHLAObjInstanceNameMap joined_federate_name_map`

**Data I/O:** \*\*  
*Map of the federate instances.*
- RTI1516\_NAMESPACE::FederateHandleSet `joined_federate_handles`

**Data I/O:** \*\*  
*FederateHandles of joined federates.*
- `VectorOfWstrings joined_federate_names`

**Data I/O:** \*\*  
*Names of the joined federates.*
- RTI1516\_NAMESPACE::InteractionClassHandle `MOM_HLAsetSwitches_class_handle`

**Data I/O:** \*\*  
*MOM HLAsetSwitches class handle.*
- RTI1516\_NAMESPACE::ParameterHandle `MOM_HLAautoProvide_param_handle`

**Data I/O:** \*\*  
*MOM HLAautoProvide parameter handle.*
- `TrickRTIAmbPtr RTI_ambassador`

**Data I/O:** \*\*  
*RTI ambassador*
- `FedAmb * federate_ambassador`

**Units:** –  
*Federate ambassador.*
- `Manager * manager`

**Units:** –  
*Associated `TrickHLA` Federate.*
- `ExecutionControlBase * execution_control`

**Units:** –  
*Execution control object.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA_Federate ()`

### 7.21.1 Detailed Description

Definition at line 82 of file Federate.hh.

### 7.21.2 Constructor & Destructor Documentation

### 7.21.2.1 Federate() [1/2]

```
Federate::Federate ( )
```

Default constructor for the [TrickHLA Federate](#) class.

NOTE: In most cases, we would allocate and set default names in the constructor. However, since we want this class to be Input Processor friendly, we cannot do that here since the Input Processor may not have been initialized yet. So, we have to set the name information to NULL and then allocate and set the defaults in the initialization job if not already set in the input stream.

**Trick Job Class:** *initialization*

Definition at line 115 of file Federate.cpp.

References `checkpoint_file_name`, `cstr_restore_label`, `cstr_save_label`, `TRICKHLA_INIT_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

### 7.21.2.2 ~Federate()

```
Federate::~Federate ( )
```

Destructor for the [TrickHLA Federate](#) class.

Free up the Trick allocated memory associated with the attributes of this class. **Trick Job Class:** *shutdown*

Definition at line 212 of file Federate.cpp.

References `clear_running_feds()`, `execution_has_begun`, `federate_ambassador`, `federation_name`, `FOM_modules`, `joined_federate_handles`, `joined_federate_name_map`, `joined_federate_names`, `known_feds`, `known_feds_count`, `local_settings`, `MIM_module`, `mom_HLAfederate_inst_name_map`, `mom_HLAfederation_instance_name_map`, `TrickHLA::KnownFederate::MOM_instance_name`, `TrickHLA::KnownFederate::name`, `name`, `shutdown()`, and `type`.

### 7.21.2.3 Federate() [2/2]

```
TrickHLA::Federate::Federate (
    const Federate & rhs ) [private]
```

Copy constructor for [Federate](#) class.

This constructor is private to prevent inadvertent copies.

## 7.21.3 Member Function Documentation

### 7.21.3.1 achieve\_and\_wait\_for\_synchronization()

```
void Federate::achieve_and_wait_for_synchronization (
    std::wstring const & label )
```

Achieve the specified sync-point and wait for the federation to be synchronized on it.

#### Parameters

<code>label</code>	Sync-point label.
--------------------	-------------------

Definition at line 2340 of file Federate.cpp.

References `TrickHLA::SyncPntListBase::achieve_and_wait_for_synchronization()`, `TrickHLA::DEBUG_LEVEL_2_T←RACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `execution_control`, `TrickHLA::SyncPntListBase::print_sync_pnts()`, `RTI1516_EXCEPTION`, `RTI_ambassador`, `should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, and `TrickHLA::StringUtilities::to_string()`.

Referenced by `DSES::ExecutionControl::post_multi_phase_init_process()`, `DIS::ExecutionControl::post_multi_phase_init_process()`, `IMSim::ExecutionControl::post_multi_phase_init_process()`, `DSES::ExecutionControl::pre_multi_phase_init_processes()`, `DIS::ExecutionControl::pre_multi_phase_init_processes()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

::pre\_multi\_phase\_init\_processes().

#### 7.21.3.2 achieve\_synchronization\_point()

```
void Federate::achieve_synchronization_point (
    std::wstring const & label )
```

Achieve the specified sync-point and do NOT wait for the federation to be synchronized on it.

##### Parameters

<i>label</i>	Sync-point label.
--------------	-------------------

Definition at line 2415 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, RTI1516\_EXCEPTION, RTI\_ambassador, should\_print(), THLA\_ENDL, THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

#### 7.21.3.3 add\_a\_single\_entry\_into\_running\_feds()

```
void Federate::add_a_single_entry_into_running_feds ( )
```

Grow the running\_feds by one entry.

Definition at line 5695 of file Federate.cpp.

References clear\_running\_feds(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, TrickHLA::StringUtilities::ip\_strdup\_wstring(), joined\_federate\_name\_map, mom\_HLAfederate\_inst\_name\_map, TrickHLA::KnownFederate::MOM\_instance\_name, TrickHLA::KnownFederate::name, name, TrickHLA::KnownFederate::required, running\_feds, running\_feds\_count, should\_print(), and THLA\_NEWLINE.

Referenced by set\_MOM\_HLAfederate\_instance\_attributes().

#### 7.21.3.4 add\_federate\_instance\_id()

```
void Federate::add_federate_instance_id (
    RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl )
```

Add the specified [Federate](#) instance ID to the list of discovered federates.

##### Parameters

<i>instance_hdl</i>	<a href="#">Federate</a> instance to add.
---------------------	---

Definition at line 824 of file Federate.cpp.

References joined\_federate\_name\_map.

Referenced by TrickHLA::Manager::discover\_object\_instance(), set\_all\_federate\_MOM\_instance\_handles\_by\_name(), and set\_MOM\_HLAfederate\_instance\_attributes().

#### 7.21.3.5 add\_MOM\_HLAfederate\_instance\_id()

```
void Federate::add_MOM_HLAfederate_instance_id (
    RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl,
    std::wstring const & instance_name )
```

Add the specified MOM HLAfederate instance ID to the list of discovered federates.

## Parameters

<i>instance_hdl</i>	<a href="#">Object</a> instance handle.
<i>instance_name</i>	<a href="#">Object</a> instance Name.

Definition at line 5748 of file Federate.cpp.

References mom\_HLAfederate\_inst\_name\_map.

Referenced by TrickHLA::Manager::discover\_object\_instance().

#### 7.21.3.6 add\_MOM\_HLAfederation\_instance\_id()

```
void Federate::add_MOM_HLAfederation_instance_id (
    RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl )
```

Add the specified MOM HLAfederation instance handle to the list of running federates.

## Parameters

<i>instance_hdl</i>	<a href="#">Object</a> instance handle.
---------------------	---

Definition at line 5839 of file Federate.cpp.

References mom\_HLAfederation\_instance\_name\_map, TrickHLA::StringUtilities::to\_string(), and TrickHLA::StringUtilities::to\_wstring().

Referenced by TrickHLA::Manager::discover\_object\_instance().

#### 7.21.3.7 announce\_sync\_point()

```
void Federate::announce_sync_point (
    std::wstring const & label,
    RTI1516_USERDATA const & user_supplied_tag )
```

The RTI has announced the existence of a synchronization point.

## Parameters

<i>label</i>	Sync-point label.
<i>user_supplied_tag</i>	Use supplied tag.

Definition at line 2486 of file Federate.cpp.

References TrickHLA::SyncPntListBase::announce\_sync\_point(), execution\_control, and RTI\_ambassador.

#### 7.21.3.8 ask\_MOM\_for\_auto\_provide\_setting()

```
void Federate::ask_MOM_for_auto_provide_setting ( )
```

Ask MOM for the current "auto-provide" setting from the switches table.

Definition at line 5293 of file Federate.cpp.

References auto\_provide\_setting, check\_for\_shutdown\_with\_termination(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, initialize\_MOM\_handles(), is\_execution\_member(), MOM\_HLAautoProvide\_handle, MOM\_HLAfederation\_class\_handle, request\_attribute\_update(), should\_print(), subscribe\_attributes(), THLA\_ENDL, THLA\_NEWLINE, and unsubscribe\_attributes().

Referenced by backup\_auto\_provide\_setting\_from\_MOM\_then\_disable().

### 7.21.3.9 ask\_MOM\_for\_federate\_names()

```
void Federate::ask_MOM_for_federate_names ( )
```

Request names of joined federates from the MOM.

Definition at line 2051 of file Federate.cpp.

References `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `initialize_MOM_handles()`, `MOM_HLAfederate_class_handle`, `MOM_HLAfederate_handle`, `MOM_HLAfederateName_handle`, `request_attribute_update()`, `should_print()`, `subscribe_attributes()`, and `THLA_NEWLINE`.

Referenced by `load_and_print_running_federate_names()`, and `wait_for_required_federates_to_join()`.

### 7.21.3.10 backup\_auto\_provide\_setting\_from\_MOM\_then\_disable()

```
void Federate::backup_auto_provide_setting_from_MOM_then_disable ( )
```

Backup the current "auto-provide" setting from the switches table then disable auto-provide if it was enabled.

Definition at line 5392 of file Federate.cpp.

References `ask_MOM_for_auto_provide_setting()`, `auto_provide_setting`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `enable_MOM_auto_provide_setting()`, `orig_auto_provide_setting`, `should_print()`, and `THLA_NEWLINE`.

Referenced by `SpaceFOM::ExecutionControl::role_determination_process()`.

### 7.21.3.11 check\_for\_shutdown()

```
bool Federate::check_for_shutdown ( )
```

Checks to see if shutdown has been commanded.

Returns

True if shutdown has been announced, else False.

**Trick Job Class:** `shutdown`

Definition at line 3826 of file Federate.cpp.

References `TrickHLA::ExecutionControlBase::check_for_shutdown()`, and `execution_control`.

### 7.21.3.12 check\_for\_shutdown\_with\_termination()

```
bool Federate::check_for_shutdown_with_termination ( )
```

Checks to see if shutdown has been commanded and, if so, terminates the simulation.

Returns

False if shutdown has NOT been announced.

NOTE: If a shutdown has been announced, this routine calls the Trick `exec_terminate()` function. So, for shutdown, it should never return. **Trick Job Class:** `shutdown`

Definition at line 3836 of file Federate.cpp.

References `TrickHLA::ExecutionControlBase::check_for_shutdown_with_termination()`, and `execution_control`.

Referenced by `ask_MOM_for_auto_provide_setting()`, `IMSim::ExecutionControl::determine_if_late_joining_or_restoring_federate()`, `TrickHLA::ExecutionControlBase::join_federation_process()`, `load_and_print_running_federate_names()`, `perform_time_advance_request()`, `TrickHLA::ExecutionControlBase::receive_execution_configuration()`, `TrickHLA::Manager::receive_init_data()`, `SpaceFOM::ExecutionControl::receive_root_ref_frame()`, `SpaceFOM::ExecutionControl::role_determination_process()`, `DSES::ExecutionControl::run_mode_transition()`, `DIS::ExecutionControl::run_mode_transition()`, `IMSim::ExecutionControl::run_mode_transition()`, `SpaceFOM::ExecutionControl::run_mode_transition()`, `setup_time_constrained()`, `setup_time_regulation()`, `DSES::ExecutionControl::wait_for_all_multiphase_init_sync_pnts()`, `IMSim::ExecutionControl::wait_for_all_multiphase_init_sync_pnts()`, `TrickHLA::SyncPnt::wait_for_announce()`, `wait_for_federation_restore_begun()`, `wait_for_federation_restore_failed_callback_to_complete()`, `wait_for_federation_restore_to_complete()`, `TrickHLA::SyncPntListBase::wait_for_list_synchronization()`.

wait\_for\_required\_federates\_to\_join(), wait\_for\_restore\_request\_callback(), wait\_for\_restore\_status\_to\_complete(), wait\_for\_save\_status\_to\_complete(), TrickHLA::ExecutionControlBase::wait\_for\_sync\_point\_announce(), TrickHLA::SyncPnt::wait\_for\_synchronization(), wait\_for\_time\_advance\_grant(), TrickHLA::Manager::wait\_on\_discovery\_of\_objects(), TrickHLA::ExecutionConfigurationBase::wait\_on\_registration(), TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects(), TrickHLA::ExecutionConfigurationBase::wait\_on\_update(), DSES::ExecutionConfiguration::wait\_on\_update(), DIS::ExecutionConfiguration::wait\_on\_update(), IMSim::ExecutionConfiguration::wait\_on\_update(), SpaceFOM::ExecutionConfiguration::wait\_on\_update(), and wait\_until\_federation\_is\_ready\_to\_restore().

#### 7.21.3.13 check\_freeze()

void Federate::check\_freeze ( )

Check for exit from freeze.

**Trick Job Class:** *freeze*

Definition at line 2589 of file Federate.cpp.

References TrickHLA::ExecutionControlBase::check\_freeze\_exit(), TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, execution\_control, should\_print(), and THLA\_NEWLINE.

#### 7.21.3.14 check\_HLA\_save\_directory()

void Federate::check\_HLA\_save\_directory ( )

Check if HLA\_save\_directory is empty. If so, ask the EXECUTIVE for info and build the absolute path of the RUN directory.

Definition at line 7092 of file Federate.cpp.

References HLA\_save\_directory.

Referenced by perform\_restore(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), set\_federate\_has\_begun\_execution(), and setup\_restore().

#### 7.21.3.15 clear\_running\_feds()

void Federate::clear\_running\_feds ( )

Deallocate running federates based on current known information in preparation for re-size.

Definition at line 5611 of file Federate.cpp.

References TrickHLA::KnownFederate::MOM\_instance\_name, TrickHLA::KnownFederate::name, name, running\_feds, and running\_feds\_count.

Referenced by add\_a\_single\_entry\_into\_running\_feds(), load\_and\_print\_running\_federate\_names(), remove\_MOM\_HLAfederate\_instance\_id(), set\_MOM\_HLAfederate\_instance\_attributes(), and ~Federate().

#### 7.21.3.16 complete\_restore()

void Federate::complete\_restore ( )

Informs of completion of federation restore.

Definition at line 6966 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, restore\_completed, TrickHLA::Restore\_In\_Progress, restore\_process, should\_print(), start\_to\_restore, and THLA\_NEWLINE.

Referenced by federation\_restored().

#### 7.21.3.17 convert\_sync\_pts()

void Federate::convert\_sync\_pts ( )

Converts HLA sync points into something Trick can save in a checkpoint.

**Trick Job Class:** *checkpoint*

Definition at line 7074 of file Federate.cpp.

References TrickHLA::ExecutionControlBase::convert\_loggable\_sync\_pts(), and execution\_control.

Referenced by setup\_checkpoint().

**7.21.3.18 copy\_running\_feds\_into\_known\_feds()**

```
void Federate::copy_running_feds_into_known_feds ( )
```

Copies the contents of the checkpoint's list of federates into known federates data structure.

Definition at line 6116 of file Federate.cpp.

References known\_feds, known\_feds\_count, TrickHLA::KnownFederate::MOM\_instance\_name, TrickHLA::KnownFederate::name, name, TrickHLA::KnownFederate::required, running\_feds, running\_feds\_count, and THLA\_NEWLINE.

Referenced by post\_restore(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

**7.21.3.19 create\_and\_join\_federation()**

```
void Federate::create_and_join_federation ( )
```

Create and then join the Federation.

**Trick Job Class:** *initialization*

Definition at line 3675 of file Federate.cpp.

References create\_federation(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, federation\_exists, federation\_joined, get\_federate\_name(), get\_federate\_type(), get\_federation\_name(), join\_federation(), should\_print(), THLA\_ENDL, and THLA\_NEWLINE.

Referenced by TrickHLA::ExecutionControlBase::join\_federation\_process(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

**7.21.3.20 create\_federation()**

```
void Federate::create_federation ( ) [private]
```

Create the simulation federation if it does not already exist.

**Trick Job Class:** *initialization*

Definition at line 3282 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, federation\_created\_by\_federate, federation\_exists, federation\_name, FOM\_modules, get\_federation\_name(), MIM\_module, RTI1516\_EXCEPTION, RTI\_ambassador, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TrickHLA::StringUtilities::tokenize(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by create\_and\_join\_federation().

**7.21.3.21 create\_RTI\_ambassador\_and\_connect()**

```
void Federate::create_RTI_ambassador_and_connect ( )
```

Create the RTI ambassador and connect to the RTI.

**Trick Job Class:** *initialization*

Definition at line 660 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, federate\_ambassador, federation\_name, local\_settings, name, RTI\_ambassador, should\_print(), THLA\_ENDL, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::ExecutionControlBase::join\_federation\_process(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.21.3.22 `destroy()`

```
void Federate::destroy ( )
```

Destroy the federation if this is the last federate.

**Trick Job Class:** *shutdown*

Definition at line 5074 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURECE\_FEDERATE, federation\_exists, federation\_joined, federation\_name, get\_federation\_name(), RTI1516\_EXCEPTION, RTI\_ambassador, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by shutdown().

#### 7.21.3.23 `destroy_orphaned_federation()`

```
void Federate::destroy_orphaned_federation ( )
```

Destroy the federation if it was orphaned from a previous simulation run that did not shutdown cleanly.

**Trick Job Class:** *initialization*

Definition at line 5214 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_9\_TRACE, TrickHLA::DEBUG\_SOURECE\_FEDERATE, federation\_name, get\_federation\_name(), RTI1516\_EXCEPTION, RTI\_ambassador, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::ExecutionControlBase::join\_federation\_process(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.21.3.24 `determine_federate_MOM_object_instance_names()`

```
void Federate::determine_federate_MOM_object_instance_names ( )
```

Get the federate MOM object instance names so that we can recover the MOM instance handles associated with each federate when a checkpoint restore happens.

**Trick Job Class:** *initialization*

Definition at line 1231 of file Federate.cpp.

References get\_RTI\_ambassador(), TrickHLA::StringUtilities::ip\_strdup\_wstring(), joined\_federate\_name\_map, known\_feds, known\_feds\_count, TrickHLA::KnownFederate::MOM\_instance\_name, name, RTI1516\_EXCEPTION, THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by wait\_for\_required\_federates\_to\_join().

#### 7.21.3.25 `enable_async_delivery()`

```
void Federate::enable_async_delivery ( )
```

Enable asynchronous delivery of messages for this federate.

**Trick Job Class:** *initialization*

Definition at line 3722 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, get\_federation\_name(), RTI1516\_EXCEPTION, RTI\_ambassador, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::ExecutionControlBase::join\_federation\_process(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.21.3.26 enable\_MOM\_auto\_provide\_setting()

```
void Federate::enable_MOM_auto_provide_setting (
    bool enable )
```

Update the MOM "auto-provide" setting from the switches table with the setting.

##### Parameters

<i>enable</i>	True to enable Auto-provide and false to disable.
---------------	---

Definition at line 5359 of file Federate.cpp.

References auto\_provide\_setting, TrickHLA::Utilities::byteswap\_int(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, TrickHLA::ENCODING\_BIG\_ENDIAN, TrickHLA::Utilities::is\_transmission\_byteswap(), MOM\_HLAAutoProvide\_param\_handle, MOM\_HLASetSwitches\_class\_handle, publish\_interaction\_class(), send\_interaction(), should\_print(), THLA\_NEWLINE, and unpublish\_interaction\_class().

Referenced by backup\_auto\_provide\_setting\_from\_MOM\_then\_disable(), and restore\_orig\_MOM\_auto\_provide\_setting().

#### 7.21.3.27 enter\_freeze()

```
void Federate::enter_freeze ( )
```

Check if a Trick freeze was commanded; if we announced freeze, tell other federates to freeze.

##### Trick Job Class: *end\_of\_frame*

Definition at line 2544 of file Federate.cpp.

References TrickHLA::ExecutionControlBase::enter\_freeze(), execution\_control, freeze\_the\_federation, and TrickHLA::ExecutionControlBase::get\_sim\_time().

#### 7.21.3.28 exit\_freeze()

```
void Federate::exit_freeze ( )
```

Routine to handle going from freeze to run; if we announced the freeze, tell other federates to run.

##### Assumptions and Limitations:

- Currently only used with **DIS** and **IMSIM** initialization schemes. **Trick Job Class: *unfreeze***

Definition at line 2570 of file Federate.cpp.

References announce\_freeze, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, execution\_control, TrickHLA::ExecutionControlBase::exit\_freeze(), freeze\_the\_federation, should\_print(), and THLA\_NEWLINE.

#### 7.21.3.29 federate\_can\_rejoin\_federation()

```
bool TrickHLA::Federate::federate_can_rejoin_federation ( ) const [inline]
```

Query if federate can rejoin federation.

**Returns**

True if federate can rejoin; False otherwise.

Definition at line 864 of file Federate.hh.

References can\_rejoin\_federation.

### 7.21.3.30 federation\_restored()

```
void Federate::federation_restored ( )
```

Federation restore completed.

**Trick Job Class:** *freeze*

Definition at line 6234 of file Federate.cpp.

References announce\_restore, complete\_restore(), cstr\_restore\_label, TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, TrickHLA::No\_Restore, restore\_begun, restore\_is\_imminent, restore\_process, save\_label\_generated, should\_print(), start\_to\_restore, str\_restore\_label, THLA\_NEWLINE, and ws\_restore\_label.

Referenced by post\_restore(), and restart\_checkpoint().

### 7.21.3.31 federation\_saved()

```
void Federate::federation_saved ( )
```

Federation save completed.

**Trick Job Class:** *freeze*

Definition at line 6208 of file Federate.cpp.

References announce\_freeze, announce\_save, checkpoint\_file\_name, cstr\_save\_label, TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, save\_label\_generated, save\_name, save\_request\_complete, should\_print(), str\_save\_label, THLA\_NEWLINE, un\_freeze(), unfreeze\_after\_save, and ws\_save\_label.

Referenced by TrickHLA::FedAmb::federationSaved().

### 7.21.3.32 federation\_synchronized()

```
void Federate::federation_synchronized (
    std::wstring const & label )
```

Marks a synchronization point as synchronized with the federation.

**Parameters**

<i>label</i>	Sync-point label.
--------------	-------------------

Definition at line 2517 of file Federate.cpp.

References execution\_control, and TrickHLA::SyncPntListBase::mark\_synchronized().

### 7.21.3.33 fix\_FPU\_control\_word()

```
void Federate::fix_FPU_control_word ( )
```

Check, and if necessary, fix the FPU Control Word.

Check that the FPU Control Word matches the value at simulation startup. If not it will reset it back to the startup value. It will use the FPU Control Word value set by the Python Input Processor.

Definition at line 351 of file Federate.cpp.

References TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

### 7.21.3.34 `freeze_init()`

`void Federate::freeze_init ( )`  
 Routine to handle going from run to freeze.

#### Assumptions and Limitations:

- Currently only used with SRFOM initialization schemes. **Trick Job Class:** `freeze_init`

Definition at line 2532 of file `Federate.cpp`.

References `execution_control`, and `TrickHLA::ExecutionControlBase::freeze_init()`.

### 7.21.3.35 `get_execution_control()`

`ExecutionControlBase* TrickHLA::Federate::get_execution_control ( ) [inline]`  
 Get the pointer to the associated `TrickHLA::Manager` instance.

#### Returns

Pointer to associated `TrickHLA::Manager`.

Definition at line 727 of file `Federate.hh`.

References `execution_control`.

Referenced by `TrickHLA::Packing::get_cte_time()`, `TrickHLA::LagCompensation::get_cte_time()`, `TrickHLA::InteractionHandler::get_cte_time()`, `TrickHLA::OwnershipHandler::get_cte_time()`, `TrickHLA::Packing::get_scenario_time()`, `TrickHLA::LagCompensation::get_scenario_time()`, `TrickHLA::InteractionHandler::get_scenario_time()`, `TrickHLA::OwnershipHandler::get_scenario_time()`, `TrickHLA::InteractionHandler::get_sim_time()`, `TrickHLA::Object::pull_ownership()`, `TrickHLA::Object::push_ownership()`, `SpaceFOM::MTRInteractionHandler::receive_interaction()`, and `SpaceFOM::MTRInteractionHandler::send_interaction()`.

### 7.21.3.36 `get_fed_ambassador()`

`FedAmb* TrickHLA::Federate::get_fed_ambassador ( ) [inline]`  
 Get the pointer to the associated `TrickHLA Federate` Ambassador instance.

#### Returns

Pointer to associated `TrickHLA::FedAmb`.

Definition at line 719 of file `Federate.hh`.

References `federate_ambassador`.

### 7.21.3.37 `get_federate_name()`

`const char* TrickHLA::Federate::get_federate_name ( ) const [inline]`  
 Get the pointer to the associated federate name.

#### Returns

Pointer to associated federate name.

Definition at line 731 of file `Federate.hh`.

References `name`.

Referenced by `SpaceFOM::ExecutionControl::check_for_shutdown_with_termination()`, `create_and_join_federation()`, `TrickHLA::FedAmb::initialize()`, `join_federation()`, and `resign_so_we_can_rejoin()`.

### 7.21.3.38 `get_federate_type()`

```
const char* TrickHLA::Federate::get_federate_type ( ) const [inline]  
Get the pointer to the associated federate type.
```

#### Returns

Pointer to associated federate type.

Definition at line 735 of file Federate.hh.

References type.

Referenced by `create_and_join_federation()`.

### 7.21.3.39 `get_federation_name()`

```
const char* TrickHLA::Federate::get_federation_name ( ) const [inline]  
Get the pointer to the associated federation execution name.
```

#### Returns

Pointer to associated federation execution name.

Definition at line 739 of file Federate.hh.

References `federation_name`.

Referenced by `SpaceFOM::ExecutionControl::check_for_shutdown_with_termination()`, `create_and_join_federation()`, `create_federation()`, `destroy()`, `destroy_orphaned_federation()`, `enable_async_delivery()`, `join_federation()`, `perform_checkpoint()`, `perform_restore()`, `perform_time_advance_request()`, `resign()`, `resign_so_we_can_rejoin()`, `restore_checkpoint()`, `setup_checkpoint()`, `setup_restore()`, `setup_time_constrained()`, `setup_time_regulation()`, `shutdown_time_constrained()`, and `shutdown_time_regulating()`.

### 7.21.3.40 `get_freeze_announced()`

```
bool TrickHLA::Federate::get_freeze_announced ( ) [inline]  
Get that federation execution freeze announced flag state.
```

#### Returns

True for federate freeze announced; False otherwise.

Definition at line 601 of file Federate.hh.

References `announce_freeze`.

Referenced by `SpaceFOM::ExecutionControl::check_freeze_exit()`, `SpaceFOM::ExecutionControl::enter_freeze()`, and `TrickHLA::ExecutionControlBase::enter_freeze()`.

### 7.21.3.41 `get_freeze_pending()`

```
bool TrickHLA::Federate::get_freeze_pending ( ) [inline]  
Get that federation execution freeze pending flag state.
```

#### Returns

True for federate freeze is pending; False otherwise.

Definition at line 605 of file Federate.hh.

References `freeze_the_federation`.

Referenced by `SpaceFOM::ExecutionControl::enter_freeze()`, and `TrickHLA::ExecutionControlBase::enter_freeze()`.

#### 7.21.3.42 get\_granted\_fed\_time()

```
const Int64Time& TrickHLA::Federate::get_granted_fed_time ( ) const [inline]
Get the current granted federation execution time.
```

##### Returns

Reference to current granted federation execution time.

Definition at line 751 of file Federate.hh.

References granted\_time.

Referenced by TrickHLA::Manager::get\_granted\_fed\_time(), TrickHLA::Object::receive\_cyclic\_data(), TrickHLA::Object::send\_cyclic\_data(), TrickHLA::Object::send\_init\_data(), and TrickHLA::Object::send\_requested\_data().

#### 7.21.3.43 get\_granted\_time()

```
double TrickHLA::Federate::get_granted_time ( ) const [inline]
Get the current granted federation execution time in seconds.
```

##### Returns

Current granted federation execution time in seconds.

Definition at line 743 of file Federate.hh.

References TrickHLA::Int64Time::getDoubleTime(), and granted\_time.

Referenced by TrickHLA::Manager::get\_granted\_time(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), DIS::ExecutionConfiguration::pack(), DSES::ExecutionConfiguration::pack(), IMSim::ExecutionConfiguration::pack(), SpaceFOM::ExecutionConfiguration::pack(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), post\_restore(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), IMSim::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::MTRInteractionHandler::receive\_interaction(), restart\_checkpoint(), SpaceFOM::MTRInteractionHandler::send\_interaction(), DSES::ExecutionConfiguration::unpack(), DIS::ExecutionConfiguration::unpack(), IMSim::ExecutionConfiguration::unpack(), SpaceFOM::ExecutionConfiguration::unpack(), and wait\_for\_time\_advance\_grant().

#### 7.21.3.44 get\_joined\_federate\_handles()

```
const RTI1516_NAMESPACE::FederateHandleSet& TrickHLA::Federate::get_joined_federate_handles ( )
[inline]
Get a const reference to the joined federate handles.
```

##### Returns

Pointer to associated federation execution name.

Definition at line 238 of file Federate.hh.

References joined\_federate\_handles.

Referenced by IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), and SpaceFOM::ExecutionControl::role\_determination\_process().

#### 7.21.3.45 get\_loookahead()

```
const Int64Interval& TrickHLA::Federate::get_loookahead ( ) const [inline]
Get the current federate lookahead time.
```

**Returns**

Reference to current federate lookahead time.

Definition at line 759 of file Federate.hh.

References lookahead.

Referenced by TrickHLA::Manager::get\_fed\_lookahead(), TrickHLA::Object::get\_update\_time\_plus\_lookahead(), D←IS::ExecutionConfiguration::pack(), DSES::ExecutionConfiguration::pack(), IMSim::ExecutionConfiguration::pack(), SpaceFOM::ExecutionConfiguration::pack(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), Trick←HLA::Object::send\_cyclic\_data(), TrickHLA::Object::send\_init\_data(), TrickHLA::Object::send\_requested\_data(), DSE←S::ExecutionConfiguration::unpack(), DIS::ExecutionConfiguration::unpack(), IMSim::ExecutionConfiguration::unpack(), and SpaceFOM::ExecutionConfiguration::unpack().

**7.21.3.46 get\_lookahead\_time()**

```
const double TrickHLA::Federate::get_lookahead_time ( ) const [inline]
```

Get the current federate lookahead time in seconds.

**Returns**

Current federate lookahead time in seconds.

Definition at line 763 of file Federate.hh.

References lookahead\_time.

Referenced by TrickHLA::Manager::determine\_job\_cycle\_time().

**7.21.3.47 get\_manager()**

```
Manager* TrickHLA::Federate::get_manager ( ) [inline]
```

Get the pointer to the associated [TrickHLA::Manager](#) instance.

**Returns**

Pointer to associated [TrickHLA::Manager](#).

Definition at line 723 of file Federate.hh.

References manager.

Referenced by SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

**7.21.3.48 get\_MOM\_HLAfederate\_class\_handle()**

```
RTI1516_NAMESPACE::ObjectClassHandle TrickHLA::Federate::get_MOM_HLAfederate_class_handle ( )
```

const [inline]

Get the federate class handle for this federate from the MOM.

**Returns**

The federate ObjectClassHandle.

Definition at line 658 of file Federate.hh.

References MOM\_HLAfederate\_class\_handle.

**7.21.3.49 get\_requested\_fed\_time()**

```
const Int64Time& TrickHLA::Federate::get_requested_fed_time ( ) const [inline]
```

Get the requested federation execution time.

**Returns**

Reference to requested federation execution time.

Definition at line 755 of file Federate.hh.

References requested\_time.

**7.21.3.50 get\_requested\_time()**

```
double TrickHLA::Federate::get_requested_time ( ) const [inline]
Get the requested federation execution time in seconds.
```

**Returns**

Requested federation execution time in seconds.

Definition at line 747 of file Federate.hh.

References TrickHLA::Int64Time::getDoubleTime(), and requested\_time.

Referenced by SpaceFOM::ExecutionControl::lateJoiner\_hla\_init\_process(), DIS::ExecutionConfiguration::pack(), DSES::ExecutionConfiguration::pack(), IMSim::ExecutionConfiguration::pack(), SpaceFOM::ExecutionConfiguration::pack(), SpaceFOM::ExecutionControl::postMultiPhase\_init\_processes(), DSES::ExecutionControl::processModeTransition\_request(), SpaceFOM::ExecutionControl::processModeTransition\_request(), DIS::ExecutionControl::processModeTransition\_request(), IMSim::ExecutionControl::processModeTransition\_request(), DSES::ExecutionConfiguration::unpack(), DIS::ExecutionConfiguration::unpack(), IMSim::ExecutionConfiguration::unpack(), and SpaceFOM::ExecutionConfiguration::unpack().

**7.21.3.51 get\_restart()**

```
bool TrickHLA::Federate::get_restart ( ) const [inline]
Get restart state.
```

**Returns**

True if in restart, False otherwise.

Definition at line 779 of file Federate.hh.

References restart\_flag.

**7.21.3.52 get\_restart\_cfg()**

```
bool TrickHLA::Federate::get_restart_cfg ( ) const [inline]
Get restart configuration state.
```

**Returns**

True if configuring restart, False otherwise.

Definition at line 783 of file Federate.hh.

References restart\_cfg\_flag.

**7.21.3.53 get\_RTI\_ambassador()**

```
RTI1516_NAMESPACE::RTIambassador* TrickHLA::Federate::get_RTI_ambassador ( ) [inline]
Get the pointer to the associated HLA RTI Ambassador instance.
```

**Returns**

Pointer to associated RTI Ambassador.

Definition at line 715 of file Federate.hh.

References RTI\_ambassador.

Referenced by IMSim::ExecutionControl::check\_freeze\_exit(), TrickHLA::ExecutionControlBase::clear\_multiphase\_init\_sync\_points(), determine\_federate\_MOM\_object\_instance\_names(), DIS::ExecutionControl::determine\_federation\_master(), SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), DIS::ExecutionControl::exit\_freeze(), IMSim::ExecutionControl::exit\_freeze(), DSES::ExecutionControl::freeze\_mode\_announce(), DIS::ExecutionControl::freeze\_mode\_announce(), IMSim::ExecutionControl::freeze\_mode\_announce(), SpaceFOM::ExecutionControl::freeze\_mode\_announce(), DSES::ExecutionControl::freeze\_mode\_transition(), DIS::ExecutionControl::freeze\_mode\_transition(), IMSim::ExecutionControl::freeze\_mode\_transition(), SpaceFOM::ExecutionControl::freeze\_mode\_transition(), TrickHLA::Manager::get\_RTI\_ambassador(), TrickHLA::Object::get\_RTI\_ambassador(), IMSim::ExecutionControl::is\_save\_initiated(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::role\_determination\_process(), DSES::ExecutionControl::run\_mode\_transition(), DIS::ExecutionControl::run\_mode\_transition(), IMSim::ExecutionControl::run\_mode\_transition(), SpaceFOM::ExecutionControl::run\_mode\_transition(), set\_all\_federate\_MOM\_instance\_handles\_by\_name(), DSES::ExecutionControl::shutdown\_mode\_transition(), DIS::ExecutionControl::shutdown\_mode\_transition(), IMSim::ExecutionControl::shutdown\_mode\_transition(), SpaceFOM::ExecutionControl::shutdown\_mode\_transition(), and SpaceFOM::ExecutionControl::wait\_on\_root\_frame\_discovered\_synchronization().

**7.21.3.54 get\_running\_feds\_count()**

```
int TrickHLA::Federate::get_running_feds_count ( ) const [inline]
```

Get the count of the currently running federates.

**Returns**

Count of the currently running federates.

Definition at line 298 of file Federate.hh.

References running\_feds\_count.

**7.21.3.55 get\_stale\_data\_counter()**

```
void TrickHLA::Federate::get_stale_data_counter (
    int * s ) [inline]
```

Get stale data counter (**DIS** only).

**Parameters**

<b>s</b>	Pointer to stale data counter.
----------	--------------------------------

Definition at line 787 of file Federate.hh.

References stale\_data\_counter.

**7.21.3.56 has\_restore Been announced()**

```
bool TrickHLA::Federate::has_restore Been announced ( ) const [inline]
```

Sets the Save filename and flag.

**Returns**

True if restore has been announced; False otherwise.

Definition at line 504 of file Federate.hh.

References restore\_begin.

Referenced by IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate().

**7.21.3.57 has\_restore\_process\_restore\_request\_failed()**

```
bool TrickHLA::Federate::has_restore_process_restore_request_failed () const [inline]  
Query if restore process restore request failed.
```

**Returns**

True if failed, False otherwise.

Definition at line 394 of file Federate.hh.

References restore\_process, and TrickHLA::Restore\_Request\_Failed.

Referenced by wait\_for\_restore\_request\_callback().

**7.21.3.58 has\_restore\_process\_restore\_request\_succeeded()**

```
bool TrickHLA::Federate::has_restore_process_restore_request_succeeded () const [inline]  
Query if restore process restore request succeeded.
```

**Returns**

True if succeeded, False otherwise.

Definition at line 398 of file Federate.hh.

References restore\_process, and TrickHLA::Restore\_Request\_Succeeded.

Referenced by wait\_for\_restore\_request\_callback().

**7.21.3.59 has\_restore\_request\_failed()**

```
bool TrickHLA::Federate::has_restore_request_failed () const [inline]  
Query if restore request failed.
```

**Returns**

True if failed, False otherwise.

Definition at line 402 of file Federate.hh.

References restore\_process, and TrickHLA::Restore\_Request\_Failed.

Referenced by IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

**7.21.3.60 has\_restore\_request\_succeeded()**

```
bool TrickHLA::Federate::has_restore_request_succeeded () const [inline]  
Query if restore request succeeded.
```

**Returns**

True if succeeded, False otherwise.

Definition at line 406 of file Federate.hh.

References restore\_process, and TrickHLA::Restore\_Request\_Succeeded.

**7.21.3.61 in\_time\_regulating\_state()**

```
bool TrickHLA::Federate::in_time_regulating_state ( ) const [inline]
Query if the federate is in a time regulating state.
```

**Returns**

True if time regulating; False otherwise.

Definition at line 804 of file Federate.hh.

References time\_regulating\_state.

Referenced by TrickHLA::Object::remove(), TrickHLA::Interaction::send(), TrickHLA::Object::send\_cyclic\_data(), and TrickHLA::Object::send\_requested\_data().

**7.21.3.62 inform\_RTI\_of\_restore\_completion()**

```
void Federate::inform_RTI_of_restore_completion ( )
Inform the RTI of the success or failure of the federate restore.
```

Definition at line 5974 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, prev\_restore\_process, TrickHLA::Restore\_Complete, TrickHLA::Restore\_Failed, restore\_process, RTI\_ambassador, should\_print(), THLA\_← NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAV← E\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by post\_restore(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

**7.21.3.63 initialize()**

```
void Federate::initialize ( )
```

Composite initialization routine for an object instance of a [Federate](#) class.

**Assumptions and Limitations:**

- The [TrickHLA::FedAmb](#) class is actually an abstract class. Therefore, the actual object instance being passed in is an instantiable polymorphic child of the RTI1516\_NAMESPACE::FederateAmbassador class. **Trick Job Class:** *initialization*

Definition at line 418 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, execution\_control, federate\_ambassador, TrickHLA::FedAmb::initialize(), TrickHLA::ExecutionControlBase::initialize(), manager, name, restart\_initialization(), should\_print(), THLA\_ENDL, THLA\_NEWLINE, TRICKHLA\_VALIDATE\_FPU\_CONTROL\_W← ORD, and type.

**7.21.3.64 initialize\_MOM\_handles()**

```
void Federate::initialize_MOM_handles ( )
```

Initialize the MOM interface handles.

**Trick Job Class:** *initialization*

Definition at line 1594 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, MOM\_HLAauto← Provide\_handle, MOM\_HLAautoProvide\_param\_handle, MOM\_HLAfederate\_class\_handle, MOM\_HLAfederate← handle, MOM\_HLAfederateName\_handle, MOM\_HLAfederatesInFederation\_handle, MOM\_HLAfederateType\_handle, MOM\_HLAfederation\_class\_handle, MOM\_HLAsetSwitches\_class\_handle, RTI\_ambassador, should\_print(), TH← LA\_NEWLINE, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by ask\_MOM\_for\_auto\_provide\_setting(), ask\_MOM\_for\_federate\_names(), load\_and\_print\_running\_federate\_names(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), TrickHLA::Manager::restart\_initialization(), restore\_federate\_handles\_from\_MO(M), and SpaceFOM::ExecutionControl::role\_determination\_process().

#### 7.21.3.65 initiate\_restore\_announce()

```
void Federate::initiate_restore_announce (
    const char * restore_name )
```

Sets the Save filename and flag.

##### Parameters

<i>restore_name</i>	Restore file name.
---------------------	--------------------

Definition at line 6888 of file Federate.cpp.

References *cstr\_restore\_label*, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, *federate\_ambassador*, TrickHLA::Initiate\_Restore, *is\_HLA\_save\_and\_restore\_supported*(), TrickHLA::No\_Restore, *request\_federation\_restore\_status*(), TrickHLA::Restore\_In\_Progress, *restore\_name*, *restore\_process*, *restore\_request\_complete*, RTI\_ambassador, *running\_feds\_count*, *running\_feds\_count\_at\_time\_of\_restore*, TrickHLA::FedAmb::set\_federation\_restore\_status\_response\_to\_process(), *should\_print*(), *str\_restore\_label*, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD, *wait\_for\_restore\_status\_to\_complete*(), and *ws\_restore\_label*.

Referenced by IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), and *setup\_restore*().

#### 7.21.3.66 initiate\_save\_announce()

```
void Federate::initiate_save_announce ( )
```

Sets the Save filename and flag.

##### Trick Job Class: *environment*

Definition at line 6854 of file Federate.cpp.

References *checkpoint\_file\_name*, *cstr\_save\_label*, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, *is\_HLA\_save\_and\_restore\_supported*(), *name*, *save\_label\_generated*, *should\_print*(), *str\_save\_label*, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_wstring(), and *ws\_save\_label*.

Referenced by TrickHLA::Manager::initiate\_federation\_save().

#### 7.21.3.67 is\_a\_required\_startup\_federate()

```
bool Federate::is_a_required_startup_federate (
    std::wstring const & fed_name )
```

Query if a federate is required at startup.

##### Returns

True if federate is required at startup; False otherwise.

##### Parameters

<i>fed_name</i>	Name of potentially required federate.
-----------------	--

Returns true if the supplied name is a required startup federate or an instance object of a required startup federate.

**Assumptions and Limitations:**

- Assumes that the instance attributes' object name is in the format 'object\_name.FOM\_name'. Otherwise, this logic fails.

Definition at line 7299 of file Federate.cpp.

References known\_feds, known\_feds\_count, name, and TrickHLA::StringUtilities::to\_wstring().

#### 7.21.3.68 `is_execution_member()`

```
bool Federate::is_execution_member ( )
```

Is the federate an execution member, which means is it connected and joined to a federation execution.

**Returns**

True if the federate is execution member; False otherwise.

**Trick Job Class: *scheduled***

Definition at line 4521 of file Federate.cpp.

References RTI\_ambassador.

Referenced by ask\_MOM\_for\_auto\_provide\_setting(), IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate(), IMSim::ExecutionControl::exit\_freeze(), IMSim::ExecutionControl::is\_save\_initiated(), load\_and\_print\_running\_federate\_names(), TrickHLA::Object::pull\_ownership\_upon\_rejoin(), TrickHLA::ExecutionControlBase::receive\_execution\_configuration(), TrickHLA::Manager::receive\_init\_data(), SpaceFOM::ExecutionControl::receive\_root\_ref\_frame(), restore\_federate\_handles\_from\_MOM(), SpaceFOM::ExecutionControl::role\_determination\_process(), setup\_checkpoint(), setup\_restore(), setup\_time\_constrained(), setup\_time\_regulation(), DSES::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), TrickHLA::SyncPnt::wait\_for\_announce(), wait\_for\_federation\_restore\_began(), wait\_for\_federation\_restore\_failed\_callback\_to\_complete(), wait\_for\_federation\_restore\_to\_complete(), TrickHLA::SyncPntListBase::wait\_for\_list\_synchronization(), wait\_for\_required\_federates\_to\_join(), wait\_for\_restore\_request\_callback(), wait\_for\_restore\_status\_to\_complete(), wait\_for\_save\_status\_to\_complete(), TrickHLA::ExecutionControlBase::wait\_for\_sync\_point\_announce(), TrickHLA::SyncPnt::wait\_for\_synchronization(), wait\_for\_time\_advance\_grant(), TrickHLA::Manager::wait\_on\_discovery\_of\_objects(), TrickHLA::Object::wait\_on\_object\_name\_reservation(), TrickHLA::Object::wait\_on\_object\_registration(), TrickHLA::ExecutionConfigurationBase::wait\_on\_registration(), TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects(), TrickHLA::ExecutionConfigurationBase::wait\_on\_update(), DIS::ExecutionConfiguration::wait\_on\_update(), DSES::ExecutionConfiguration::wait\_on\_update(), IMSim::ExecutionConfiguration::wait\_on\_update(), SpaceFOM::ExecutionConfiguration::wait\_on\_update(), wait\_until\_federation\_is\_ready\_to\_restore(), and TrickHLA::Manager::~Manager().

#### 7.21.3.69 `is_federate_executing()`

```
bool Federate::is_federate_executing ( ) const
```

Checks for the existence 'startup' initialization sync point as an indication if this federate is running.

**Returns**

True if federate is running; False otherwise.

Definition at line 6990 of file Federate.cpp.

References execution\_has\_begun.

Referenced by set\_MOM\_HLAfederate\_instance\_attributes(), and setup\_restore().

### 7.21.3.70 is\_federate\_instance\_id()

```
bool Federate::is_federate_instance_id (
    RTI1516_NAMESPACE::ObjectInstanceHandle id )
```

Determine if the specified instance ID is for one of the discovered federates.

#### Returns

True if ID is for a federate.

#### Parameters

<i>id</i>	MOM HLAfederate instance ID.
-----------	------------------------------

Definition at line 841 of file Federate.cpp.

References joined\_federate\_name\_map.

Referenced by set\_MOM\_HLAfederate\_instance\_attributes().

### 7.21.3.71 is\_federation\_created\_by\_federate()

```
bool TrickHLA::Federate::is_federation_created_by_federate ( ) const [inline]
```

Query if the federation was created by this federate.

#### Returns

True if created by this federate; False otherwise.

Definition at line 873 of file Federate.hh.

References federation\_created\_by\_federate.

Referenced by TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre←multi\_phase\_init\_processes().

### 7.21.3.72 is\_HLA\_save\_and\_restore\_supported()

```
bool Federate::is_HLA_save_and_restore_supported ( )
```

Returns true if HLA save and restore is supported by the user specified simulation initialization scheme.

#### Returns

True if HLA save and restore are supported, false otherwise.

#### Assumptions and Limitations:

- Currently only used with [DIS](#) and [IMSIM](#) initialization schemes.

Definition at line 2631 of file Federate.cpp.

References execution\_control, and TrickHLA::ExecutionControlBase::is\_save\_and\_restore\_supported().

Referenced by initiate\_restore\_announce(), initiate\_save\_announce(), perform\_checkpoint(), perform\_restore(), post←\_checkpoint(), post\_restore(), request\_federation\_save(), and setup\_restore().

### 7.21.3.73 is\_joined\_federate() [1/2]

```
bool Federate::is_joined_federate (
    const char * federate_name ) [private]
```

Determine if the specified federate name is a joined federate.

**Returns**

True if a name of joined federate, otherwise false.

**Parameters**

<i>federate_name</i>	Federate name to test.
----------------------	------------------------

Definition at line 1328 of file Federate.cpp.

References TrickHLA::StringUtilities::to\_wstring().

Referenced by `wait_for_required_federates_to_join()`.

**7.21.3.74 is\_joined\_federate() [2/2]**

```
bool Federate::is_joined_federate (
    std::wstring const & federate_name ) [private]
```

Determine if the specified federate name is a joined federate.

**Returns**

True if a name of joined federate, otherwise false.

**Parameters**

<i>federate_name</i>	Federate name to test.
----------------------	------------------------

Definition at line 1336 of file Federate.cpp.

References `joined_federate_names`.

**7.21.3.75 is\_MOM\_HLAfederate\_class()**

```
bool TrickHLA::Federate::is_MOM_HLAfederate_class (
    RTI1516_NAMESPACE::ObjectClassHandle federate_class ) const [inline]
```

Check with the MOM if the is an HLAfederate class.

**Returns**

True if a MOM HLAfederate class.

**Parameters**

<i>federate_class</i>	Object class handle to check.
-----------------------	-------------------------------

Definition at line 651 of file Federate.hh.

References `MOM_HLAfederate_class_handle`.

Referenced by `TrickHLA::Manager::discover_object_instance()`.

**7.21.3.76 is\_MOM\_HLAfederation\_class()**

```
bool TrickHLA::Federate::is_MOM_HLAfederation_class (
    RTI1516_NAMESPACE::ObjectClassHandle class_hdl ) const [inline]
```

Query if the an object class handle is a federation class.

#### Returns

True if class handle is a federation class; False otherwise.

#### Parameters

<code>class_hdl</code>	HLA object class handle to test.
------------------------	----------------------------------

Definition at line 698 of file Federate.hh.

References MOM\_HLAfederation\_class\_handle.

Referenced by TrickHLA::Manager::discover\_object\_instance().

### 7.21.3.77 `is_MOM_HLAfederation_instance_id()`

```
bool Federate::is_MOM_HLAfederation_instance_id (
    RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl )
```

Determine if the specified instance handle is an MOM HLAfederation instance.

#### Returns

True if ID is for a federate; False otherwise.

#### Parameters

<code>instance_hdl</code>	Federate instance handle.
---------------------------	---------------------------

Definition at line 6997 of file Federate.cpp.

References mom\_HLAfederation\_instance\_name\_map.

Referenced by set\_MOM\_HLAfederation\_instance\_attributes().

### 7.21.3.78 `is_required_federate()`

```
bool Federate::is_required_federate (
    std::wstring const & federate_name ) [private]
```

Determine if the specified federate name is a required federate.

#### Returns

True if a name of required federate, otherwise false.

#### Parameters

<code>federate_name</code>	Federate name to test.
----------------------------	------------------------

Definition at line 1313 of file Federate.cpp.

References known\_feds, known\_feds\_count, name, and TrickHLA::StringUtilities::to\_wstring().

Referenced by wait\_for\_required\_federates\_to\_join().

### 7.21.3.79 `is_start_to_restore()`

```
bool TrickHLA::Federate::is_start_to_restore ( ) const [inline]  
Query if federate has started a restore process.
```

#### Returns

True if restore has started; False otherwise.

Definition at line 459 of file Federate.hh.

References `start_to_restore`.

Referenced by `IMSim::ExecutionControl::determine_if_late_joining_or_restoring_federate()`.

### 7.21.3.80 `is_time_advance_granted()`

```
bool TrickHLA::Federate::is_time_advance_granted ( ) const [inline]  
Query if time advance has been granted.
```

#### Returns

True if time advance has been granted; False otherwise.

Definition at line 796 of file Federate.hh.

References `time_adv_grant`.

Referenced by `IMSim::FreezeInteractionHandler::send_scenario_freeze_interaction()`.

### 7.21.3.81 `is_time_management_enabled()`

```
bool TrickHLA::Federate::is_time_management_enabled ( ) const [inline]  
Query if time management is enabled.
```

#### Returns

True if time management is enabled; False otherwise.

Definition at line 852 of file Federate.hh.

References `time_management`.

Referenced by `IMSim::ExecutionControl::pre_multi_phase_init_processes()`, `TrickHLA::Object::pull_ownership()`, and `TrickHLA::Object::push_ownership()`.

### 7.21.3.82 `is_zero_lookingahead_time()`

```
const bool TrickHLA::Federate::is_zero_lookingahead_time ( ) const [inline]  
Query of federate has a zero lookahead time.
```

#### Returns

True if lookahead time is zero; False otherwise.

Definition at line 767 of file Federate.hh.

References `lookahead_time`.

Referenced by `TrickHLA::Object::send_cyclic_data()`, and `TrickHLA::Object::send_requested_data()`.

### 7.21.3.83 `join_federation()`

```
void Federate::join_federation (
    const char *const federate_name,
    const char *const federate_type ) [private]
```

Join a federation.

**Parameters**

<i>federate_name</i>	Name of this federate.
<i>federate_type</i>	Type for this federate.

**Trick Job Class: initialization**

Definition at line 3454 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, federate\_ambassador, federate\_id, federation\_created\_by\_federate, federation\_exists, federation\_joined, federation\_name, FOM\_modules, get\_federate\_name(), get\_federation\_name(), RTI\_ambassador, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TrickHLA::StringUtilities::tokenize(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by create\_and\_join\_federation().

**7.21.3.84 load\_and\_print\_running\_federate\_names()**

```
void Federate::load_and_print_running_federate_names ( )
```

Load the running federate names from the RTI.

Definition at line 5434 of file Federate.cpp.

References all\_federates\_joined, ask\_MOM\_for\_federate\_names(), check\_for\_shutdown\_with\_termination(), clear\_running\_feds(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, initialize\_MOM\_handles(), is\_execution\_member(), joined\_federate\_handles, joined\_federate\_name\_map, joined\_federate\_names, MOM\_HLAfederatesInFederation\_handle, MOM\_HLAfederation\_class\_handle, TrickHLA::KnownFederate::name, request\_attribute\_update(), running\_feds, running\_feds\_count, should\_print(), subscribe\_attributes(), THLA\_ENDL, THLA\_NEWLINE, unsubscribe\_attributes(), and update\_running\_feds().

Referenced by DIS::ExecutionControl::post\_multi\_phase\_init\_process(), and IMSim::ExecutionControl::post\_multi\_phase\_init\_process().

**7.21.3.85 operator=()**

```
Federate& TrickHLA::Federate::operator= (
    const Federate & rhs ) [private]
```

Assignment operator for **Federate** class.

This assignment operator is private to prevent inadvertent copies.

**7.21.3.86 perform\_checkpoint()**

```
void Federate::perform_checkpoint ( )
```

Federates that did not announce the save, perform a checkpoint.

**Assumptions and Limitations:**

- Currently only used with **DIS** and **IMSIM** initialization schemes. **Trick Job Class: freeze**

Definition at line 2642 of file Federate.cpp.

References announce\_save, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, execution\_control, get\_federation\_name(), is\_HLA\_save\_and\_restore\_supported(), TrickHLA::ExecutionControlBase::perform\_save(), post\_checkpoint(), save\_name, should\_print(), start\_to\_save, str\_save\_label, THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

### 7.21.3.87 `perform_restore()`

```
void Federate::perform_restore ( )
```

Federates that did not announce the restore, perform a restore.

**Assumptions and Limitations:**

- Currently only used with **DIS** and IMSIM initialization schemes. **Trick Job Class:** *freeze*

Definition at line 2888 of file Federate.cpp.

References `announce_restore`, `check_HLA_save_directory()`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `get_federation_name()`, `HLA_save_directory`, `is_HLA_save_and_restore_supported()`, `post_restore()`, `restore_name`, `should_print()`, `start_to_restore`, `str_restore_label`, `THLA_NEWLINE`, and `TrickHLA::StringUtilities::to_string()`.

### 7.21.3.88 `perform_time_advance_request()`

```
void Federate::perform_time_advance_request ( ) [private]
```

Make the HLA time-advance request using the current `requested_time` value.

**Trick Job Class:** *scheduled*

Definition at line 4252 of file Federate.cpp.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `TrickHLA::Int64Time::get()`, `get_federation_name()`, `TrickHLA::Int64Time::getDoubleTime()`, `requested_time`, `RTI_ambassador`, `save_completed`, `should_print()`, `THLA_NEWLINE`, `time_adv_grant`, `time_management`, `TrickHLA::StringUtilities::to_string()`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `time_advance_request()`, `time_advance_request_to_GALT()`, and `time_advance_request_to_GALT_LCTS_multiple()`.

### 7.21.3.89 `post_checkpoint()`

```
void Federate::post_checkpoint ( )
```

Complete federate save.

**Assumptions and Limitations:**

- Currently only used with **DIS** and IMSIM initialization schemes. **Trick Job Class:** *post\_checkpoint*

Definition at line 2835 of file Federate.cpp.

References `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `is_HLA_save_and_restore_supported()`, `RTI_ambassador`, `should_print()`, `start_to_save`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `perform_checkpoint()`.

### 7.21.3.90 `post_multiphase_initialization()`

```
void Federate::post_multiphase_initialization ( )
```

Complete the post-multiphase initialization startup process prior to the federation execution going into run.

This performs all the startup steps after any multi-phase initialization process defined by the user.

**Trick Job Class:** *initialization*

Definition at line 639 of file Federate.cpp.

References `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `execution_control`, `TrickHLA::ExecutionControlBase::post_multi_phase_init_processes()`, `set_federate_has_begun_execution()`, `should_print()`, and `THLA_NEWLINE`.

### 7.21.3.91 post\_restore()

```
void Federate::post_restore ( )
Complete federate restore and prepare to restart execution.
```

#### Assumptions and Limitations:

- Currently only used with [DIS](#) and [IMSIM](#) initialization schemes.

Definition at line 3025 of file Federate.cpp.

References [announce\\_restore](#), [copy\\_running\\_feds\\_into\\_known\\_feds\(\)](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [federation\\_restored\(\)](#), [get\\_granted\\_time\(\)](#), [TrickHLA::Manager::get\\_object\\_count\(\)](#), [TrickHLA::Manager::get\\_objects\(\)](#), [granted\\_time](#), [HLA\\_time](#), [inform\\_RTI\\_of\\_restore\\_completion\(\)](#), [is\\_HLA\\_save\\_and\\_restore\\_supported\(\)](#), [manager](#), [prev\\_restore\\_process](#), [reinstate\\_logged\\_sync\\_pts\(\)](#), [requested\\_time](#), [TrickHLA::Manager::reset\\_mgr\\_initialized\(\)](#), [TrickHLA::Restore\\_Complete](#), [restore\\_federate\\_handles\\_from\\_MOM\(\)](#), [TrickHLA::Manager::restore\\_interactions\(\)](#), [TrickHLA::Object::restore\\_ownership\\_transfer\\_checkpointed\\_data\(\)](#), [restore\\_process](#), [RTI\\_ambassador](#), [set\\_all\\_federate\\_MOM\\_instance\\_handles\\_by\\_name\(\)](#), [TrickHLA::Manager::set\\_all\\_object\\_instance\\_handles\\_by\\_name\(\)](#), [set\\_granted\\_time\(\)](#), [TrickHLA::Manager::setup\\_all\\_ref\\_attributes\(\)](#), [TrickHLA::Manager::setup\\_all\\_RTI\\_handles\(\)](#), [should\\_print\(\)](#), [start\\_to\\_restore](#), [THLA\\_NEWLINE](#), [TRICKHLA\\_RESTORE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_SAVE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_VALIDATE\\_FPU\\_CONTROL\\_WORD](#), [wait\\_for\\_federation\\_restore\\_began\(\)](#), [wait\\_for\\_federation\\_restore\\_failed\\_callback\\_to\\_complete\(\)](#), and [wait\\_for\\_federation\\_restore\\_to\\_complete\(\)](#).

Referenced by [perform\\_restore\(\)](#).

### 7.21.3.92 pre\_multiphase\_initialization()

```
void Federate::pre_multiphase_initialization ( )
```

Begin the pre-multiphase initialization process of standing up the federate in the federation execution.

This performs all the startup steps prior to any multi-phase initialization process defined by the user. The multi-phase initialization will be performed as initialization jobs between [P\\_INIT](#) and [P\\_LAST](#) phased initialization jobs.

#### Trick Job Class: *initialization*

Definition at line 605 of file Federate.cpp.

References [TrickHLA::DEBUG\\_LEVEL\\_1\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [execution\\_control](#), [TrickHLA::Manager::initialize\(\)](#), [manager](#), [TrickHLA::ExecutionControlBase::pre\\_multi\\_phase\\_init\\_processes\(\)](#), [should\\_print\(\)](#), [THLA\\_ENDL](#), and [THLA\\_NEWLINE](#).

### 7.21.3.93 print\_requested\_federation\_restore\_status()

```
void Federate::print_requested_federation_restore_status (
    RTI1516_NAMESPACE::FederateRestoreStatusVector const & status_vector )
```

Prints the federation restore status from the RTI.

#### Parameters

<i>status_vector</i>	Save status.
----------------------	--------------

Definition at line 6693 of file Federate.cpp.

References [name](#), and [TrickHLA::StringUtilities::to\\_string\(\)](#).

### 7.21.3.94 print\_restore\_failure\_reason()

```
void Federate::print_restore_failure_reason (
    RTI1516_NAMESPACE::RestoreFailureReason reason )
```

Prints the reason for the federation restore failure.

#### Parameters

<i>reason</i>	Restore failure reason.
---------------	-------------------------

Definition at line 6785 of file Federate.cpp.

References federation\_restore\_failed\_callback\_complete.

### 7.21.3.95 print\_save\_failure\_reason()

```
void Federate::print_save_failure_reason (
    RTI1516_NAMESPACE::SaveFailureReason reason )
```

Prints the reason for the federation save failure.

#### Parameters

<i>reason</i>	Save failure reason.
---------------	----------------------

Definition at line 6812 of file Federate.cpp.

### 7.21.3.96 print\_version()

```
void Federate::print_version ( ) const
```

Print the [TrickHLA](#) version string.

**Trick Job Class:** *initialization*

Definition at line 312 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, TrickHLA::Utilities::get\_release\_date(), TrickHLA::Utilities::get\_version(), should\_print(), THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

### 7.21.3.97 process\_requested\_federation\_restore\_status()

```
void Federate::process_requested_federation_restore_status (
    RTI1516_NAMESPACE::FederateRestoreStatusVector const & status_vector )
```

Processes the federation restore status received from the RTI.

#### Parameters

<i>status_vector</i>	Save status.
----------------------	--------------

Definition at line 6733 of file Federate.cpp.

References TrickHLA::Initiate\_Restore, initiate\_restore\_flag, restore\_process, and restore\_request\_complete.

### 7.21.3.98 process\_requested\_federation\_save\_status()

```
void Federate::process_requested_federation_save_status (
    RTI1516_NAMESPACE::FederateHandleSaveStatusPairVector const & status_vector )
```

Processes the federation save status received from the RTI.

**Parameters**

<i>status_vector</i>	Save status.
----------------------	--------------

Definition at line 6762 of file Federate.cpp.

References `initiate_save_flag`, and `save_request_complete`.

**7.21.3.99 publish\_interaction\_class()**

```
void Federate::publish_interaction_class (
    RTI1516_NAMESPACE::InteractionClassHandle class_handle ) [private]
```

Publish [Interaction](#) class.

**Parameters**

<i>class_handle</i>	<a href="#">Interaction</a> class handle.
---------------------	---

Definition at line 2151 of file Federate.cpp.

References `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `RTI_ambassador`, `should_print()`, `THLA_NEWLINE`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `enable_MOM_auto_provide_setting()`.

**7.21.3.100 read\_running\_feds\_file()**

```
void Federate::read_running_feds_file (
    char * file_name ) throw ( const char *)
```

Read the `running_feds` file, replacing the data in known federates data structure.

**Parameters**

<i>file_name</i>	Checkpoint file name.
------------------	-----------------------

Definition at line 6047 of file Federate.cpp.

References `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `IMSim::ExecutionControl::pre_multi_phase_init_processes()`, and `setup_restore()`.

**7.21.3.101 rebuild\_federate\_handles()**

```
void Federate::rebuild_federate_handles (
    RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl,
    RTI1516_NAMESPACE::AttributeHandleValueMap const & values )
```

Reloads the federate handle set from the MOM after a checkpoint reload.

**Parameters**

<i>instance_hdl</i>	<a href="#">Object</a> instance handle.
<i>values</i>	<a href="#">Attribute</a> values.

Definition at line 7168 of file Federate.cpp.

References `TrickHLA::Utilities::byteswap_int()`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE`,

FEDERATE, TrickHLA::ENCODING\_BIG\_ENDIAN, TrickHLA::Utilities::is\_transmission\_byteswap(), joined\_federate\_handles, RTI\_ambassador, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

#### 7.21.3.102 register\_generic\_sync\_point()

```
void Federate::register_generic_sync_point (
    std::wstring const & label,
    double time = -1.0 )
```

Register a generic synchronization point; i.e. not a multiphase init sync-point.

##### Parameters

<i>label</i>	Sync-point label.
<i>time</i>	Optional Sync-point time in seconds.

Definition at line 2280 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, RTI1516\_EXCEPTION, RTI1516\_USERDATA, RTI\_ambassador, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::Int64Interval::toMicroseconds(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by IMSim::ExecutionControl::check\_freeze\_time(), DIS::ExecutionControl::enter\_freeze(), IMSim::ExecutionControl::exit\_freeze(), IMSim::ExecutionControl::is\_save\_initiated(), IMSim::ExecutionControl::post\_multi\_phase\_init\_process(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.21.3.103 reinstate\_logged\_sync\_pts()

```
void Federate::reinstate_logged_sync_pts ( )
```

Converts checkpointed sync points into HLA sync points.

Definition at line 7083 of file Federate.cpp.

References execution\_control, and TrickHLA::ExecutionControlBase::reinstate\_logged\_sync\_pts().

Referenced by post\_restore(), and restart\_checkpoint().

#### 7.21.3.104 remove\_federate\_instance\_id()

```
void Federate::remove_federate_instance_id (
    RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl )
```

Remove the specified [Federate](#) instance ID from the list of discovered federates.

##### Parameters

<i>instance_hdl</i>	<a href="#">Federate</a> instance to remove.
---------------------	--

Definition at line 830 of file Federate.cpp.

References joined\_federate\_name\_map.

Referenced by remove\_MOM\_HLAfederate\_instance\_id().

**7.21.3.105 remove\_MOM\_HLAfederate\_instance\_id()**

```
void Federate::remove_MOM_HLAfederate_instance_id (
    RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl )
```

Remove the specified [Federate](#) instance ID to the list of discovered federates.

**Parameters**

<i>instance_hdl</i>	<a href="#">Object</a> instance handle.
---------------------	---

Definition at line 5755 of file Federate.cpp.

References [clear\\_running\\_feds\(\)](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [TrickHLA::StringUtilities::ip\\_strdup\\_wstring\(\)](#), [mom\\_HLAfederate\\_inst\\_name\\_map](#), [TrickHLA::KnownFederate::MO\\_M\\_instance\\_name](#), [TrickHLA::KnownFederate::name](#), [name](#), [remove\\_federate\\_instance\\_id\(\)](#), [remove\\_MOM\\_HLAfederation\\_instance\\_id\(\)](#), [TrickHLA::KnownFederate::required](#), [running\\_feds](#), [running\\_feds\\_count](#), [should\\_print\(\)](#), [THLA\\_NEWLINE](#), and [TrickHLA::StringUtilities::to\\_string\(\)](#).

**7.21.3.106 remove\_MOM\_HLAfederation\_instance\_id()**

```
void Federate::remove_MOM_HLAfederation_instance_id (
    RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl )
```

Remove the specified MOM HLAfederation instance handle from the list of running federates.

**Parameters**

<i>instance_hdl</i>	<a href="#">Object</a> instance handle.
---------------------	---

Definition at line 5849 of file Federate.cpp.

References [mom\\_HLAfederation\\_instance\\_name\\_map](#).

Referenced by [remove\\_MOM\\_HLAfederate\\_instance\\_id\(\)](#).

**7.21.3.107 request\_attribute\_update()**

```
void Federate::request_attribute_update (
    RTI1516_NAMESPACE::ObjectClassHandle class_handle,
    RTI1516_NAMESPACE::AttributeHandleSet const & attribute_list ) [private]
```

Request an update to the specified attributes for the given object class handle.

**Parameters**

<i>class_handle</i>	Class handle.
<i>attribute_list</i>	Attributes handles.

Definition at line 1993 of file Federate.cpp.

References [TrickHLA::DEBUG\\_LEVEL\\_4\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [RTI1516\\_USERDATA](#), [RTI\\_ambassador](#), [should\\_print\(\)](#), [THLA\\_NEWLINE](#), [TRICKHLA\\_RESTORE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_SAVE\\_FPU\\_CONTROL\\_WORD](#), and [TRICKHLA\\_VALIDATE\\_FPU\\_CONTROL\\_WORD](#).

Referenced by [ask\\_MOM\\_for\\_auto\\_provide\\_setting\(\)](#), [ask\\_MOM\\_for\\_federate\\_names\(\)](#), [load\\_and\\_print\\_running\\_federate\\_names\(\)](#), and [restore\\_federate\\_handles\\_from\\_MOM\(\)](#).

### 7.21.3.108 request\_federation\_restore\_status()

```
void Federate::request_federation_restore_status ( )
```

Requests the status of the Federation Restore.

Definition at line 6618 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, RTI\_ambassador, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by initiate\_restore\_announce().

### 7.21.3.109 request\_federation\_save()

```
void Federate::request_federation_save ( ) [private]
```

Request federation save from the RTI.

**Trick Job Class:** *freeze*

Definition at line 5892 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, is\_HLA\_save\_and\_restore\_supported(), RTI\_ambassador, save\_name, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by setup\_checkpoint().

### 7.21.3.110 request\_federation\_save\_status()

```
void Federate::request_federation_save_status ( )
```

Requests the status of the Federation Save.

Definition at line 6584 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, RTI\_ambassador, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by setup\_checkpoint().

### 7.21.3.111 requested\_federation\_restore\_status()

```
void Federate::requested_federation_restore_status (
    bool status )
```

Sets the Restore filename and flag.

#### Parameters

<i>status</i>	Restore success status from RTI.
---------------	----------------------------------

**Trick Job Class:** *freeze*

Definition at line 6658 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, federate\_ambassador, RTI\_ambassador, TrickHLA::FedAmb::set\_federation\_restore\_status\_response\_to\_echo(), should\_print(), THLA\_NE\_WLINE, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

### 7.21.3.112 resign()

```
void Federate::resign ( )
```

Resign from the federation.

**Trick Job Class:** *shutdown*

Definition at line 4758 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, federation\_joined, get\_federation\_name(), RTI1516\_EXCEPTION, RTI\_ambassador, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by shutdown().

### 7.21.3.113 resign\_so\_we\_can\_rejoin()

```
void Federate::resign_so_we_can_rejoin ( )
```

Resign from the federation in a way that permits rejoining later.

Resign from the federation but divest ownership of my attributes and do not delete the federate from the federation when resigning. **Trick Job Class:** *logging*

Definition at line 4920 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, federation\_joined, get\_federation\_name(), get\_federation\_name(), RTI1516\_EXCEPTION, RTI\_ambassador, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by shutdown().

### 7.21.3.114 restart\_checkpoint()

```
void Federate::restart_checkpoint ( )
```

Restart the sim from a checkpoint.

**Assumptions and Limitations:**

- Currently only used with IMSIM initialization scheme; only for restore at simulation startup. **Trick Job Class:** *environment*

Definition at line 6161 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, federation\_restored(), get\_granted\_time(), granted\_time, HLA\_time, TrickHLA::No\_Restore, reinstate\_logged\_sync\_pts(), requested\_time, restore\_process, RTI\_ambassador, set\_granted\_time(), should\_print(), THLA\_NEWLINE, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

### 7.21.3.115 restart\_initialization()

```
void Federate::restart_initialization ( )
```

Perform initialization after a restart.

**Trick Job Class:** *initialization*

Definition at line 487 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, enable\_known\_feds, federate\_ambassador, federation\_name, FOM\_modules, known\_feds, known\_feds\_count, lookahead\_time, name, set\_lookahead(), should\_print(), THLA\_ENDL, THLA\_NEWLINE, time\_constrained, time\_management, time\_regulating, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by `initialize()`, and `TrickHLA::Manager::restart_initialization()`.

#### 7.21.3.116 `restore_checkpoint()`

```
void Federate::restore_checkpoint (
    char * file_name )
```

Restore checkpoint.

##### Parameters

<code>file_name</code>	Checkpoint file name.
------------------------	-----------------------

Definition at line 5934 of file `Federate.cpp`.

References `execution_control`, `get_federation_name()`, `HLA_save_directory`, `TrickHLA::ExecutionControlBase::is_master()`, `prev_restore_process`, `TrickHLA::Restore_Complete`, `restore_process`, and `THLA_NEWLINE`.

Referenced by `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

#### 7.21.3.117 `restore_federate_handles_from_MOM()`

```
void Federate::restore_federate_handles_from_MOM ( )
```

Ask for all federate handles from MOM after a checkpoint reload.

Definition at line 7109 of file `Federate.cpp`.

References `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `federate_ambassador`, `initialize_MOM_handles()`, `is_execution_member()`, `joined_federate_handles`, `MOM_HLAfederate_class_handle`, `MOM_HLAfederate_handle`, `request_attribute_update()`, `TrickHLA::FedAmb::reset_federation_restored_rebuild_federate_handle_set()`, `running_feds_count`, `TrickHLA::FedAmb::set_federation_restored_rebuild_federate_handle_set()`, `should_print()`, `subscribe_attributes()`, `THLA_ENDL`, `THLA_NEWLINE`, and `unsubscribe_attributes()`.

Referenced by `post_restore()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

#### 7.21.3.118 `restore_orig_MOM_auto_provide_setting()`

```
void Federate::restore_orig_MOM_auto_provide_setting ( )
```

Restore the backed up "auto-provide" state to the MOM.

Definition at line 5412 of file `Federate.cpp`.

References `auto_provide_setting`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `enable_MOM_auto_provide_setting()`, `orig_auto_provide_setting`, `should_print()`, and `THLA_NEWLINE`.

Referenced by `SpaceFOM::ExecutionControl::post_multi_phase_init_processes()`.

#### 7.21.3.119 `send_interaction()`

```
void Federate::send_interaction (
    RTI1516_NAMESPACE::InteractionClassHandle class_handle,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & parameter_list ) [private]
```

Send the `Interaction` for the specified interaction class and parameter list.

##### Parameters

<code>class_handle</code>	<code>Interaction</code> class handle.
<code>parameter_list</code>	<code>Parameter</code> values in a map.

Definition at line 2227 of file Federate.cpp.

References RTI1516\_USERDATA, RTI\_ambassador, THLA\_NEWLINE, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by enable\_MOM\_auto\_provide\_setting().

#### 7.21.3.120 set\_all\_federate\_MOM\_instance\_handles\_by\_name()

```
void Federate::set_all_federate_MOM_instance_handles_by_name ( )
```

Set all the federate MOM instance handles by using the previously saved named for the MOM object instance associated with the federate.

Definition at line 1088 of file Federate.cpp.

References add\_federate\_instance\_id(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, get\_RTI\_ambassador(), joined\_federate\_name\_map, known\_feds, known\_feds\_count, TrickHLA::KnownFederate::name, RTI1516\_EXCEPTION, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by post\_restore(), and TrickHLA::Manager::restart\_initialization().

#### 7.21.3.121 set\_announce\_save()

```
void TrickHLA::Federate::set_announce_save ( ) [inline]
```

Set the announce save flag.

Definition at line 409 of file Federate.hh.

References announce\_save.

Referenced by IMSim::ExecutionControl::start\_federation\_save\_at\_scenario\_time().

#### 7.21.3.122 set\_checkpoint\_file\_name()

```
void Federate::set_checkpoint_file_name (
    const char * name )
```

Save the supplied checkpoint file name.

##### Parameters

<i>name</i>	Checkpoint file name.
-------------	-----------------------

##### Trick Job Class: *environment*

Definition at line 6844 of file Federate.cpp.

References checkpoint\_file\_name, save\_name, and TrickHLA::StringUtilities::to\_wstring().

Referenced by TrickHLA::Manager::initiate\_federation\_save().

#### 7.21.3.123 set\_federate\_has\_begun\_execution()

```
void TrickHLA::Federate::set_federate_has_begun_execution ( ) [inline]
```

Set the federate has begun execution state.

Definition at line 536 of file Federate.hh.

References check\_HLA\_save\_directory(), execution\_has\_begun, and joined\_federate\_name\_map.

Referenced by post\_multiphase\_initialization(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

### 7.21.3.124 set\_federation\_name()

```
void Federate::set_federation_name (
    const char *const exec_name )
```

Set the name of the federation execution.

#### Parameters

<i>exec_name</i>	Federation execution name.
------------------	----------------------------

#### Trick Job Class: *initialization*

Definition at line 5263 of file Federate.cpp.

References `federation_name`.

### 7.21.3.125 set\_freeze\_announced()

```
void TrickHLA::Federate::set_freeze_announced (
    bool flag ) [inline]
```

Set that federation execution freeze has been announced.

#### Parameters

<i>flag</i>	True for federate freeze announce; False otherwise.
-------------	---

Definition at line 597 of file Federate.hh.

References `announce_freeze`.

Referenced by `TrickHLA::ExecutionControlBase::check_pause_at_init()`, and `SpaceFOM::ExecutionControl::freeze_init()`.

### 7.21.3.126 set\_granted\_time() [1/2]

```
void Federate::set_granted_time (
    double time )
```

Sets the granted time from the specified double.

#### Parameters

<i>time</i>	Granted time in seconds.
-------------	--------------------------

Definition at line 3134 of file Federate.cpp.

References `granted_time`, and `TrickHLA::Int64Time::setTo()`.

Referenced by `post_restore()`, and `restart_checkpoint()`.

### 7.21.3.127 set\_granted\_time() [2/2]

```
void Federate::set_granted_time (
    RTI1516_NAMESPACE::LogicalTime const & time )
```

Sets the granted time from the specified LogicalTime.

#### Parameters

<i>time</i>	Granted time in HLA logical time.
-------------	-----------------------------------

Definition at line 3140 of file Federate.cpp.  
 References granted\_time, and TrickHLA::Int64Time::setTo().

#### 7.21.3.128 set\_lookinghead()

```
void Federate::set_lookinghead (
    double value )
```

Sets the HLA lookahead time.

##### Parameters

<i>value</i>	HLA lookahead time in seconds.
--------------	--------------------------------

Definition at line 3158 of file Federate.cpp.  
 References lookahead, lookahead\_time, and TrickHLA::Int64Interval::setTo().  
 Referenced by restart\_initialization().

#### 7.21.3.129 set\_MOM\_HLAfederate\_instance\_attributes()

```
void Federate::set_MOM_HLAfederate_instance_attributes (
    RTI1516_NAMESPACE::ObjectInstanceHandle id,
    RTI1516_NAMESPACE::AttributeHandleValueMap const & values )
```

Set the Federates name given the instance ID as well as the FederateHandle ID associated with the [Federate](#) instance.

##### Parameters

<i>id</i>	<a href="#">Object</a> instance handle.
<i>values</i>	<a href="#">Attribute</a> values.

Definition at line 847 of file Federate.cpp.  
 References add\_a\_single\_entry\_into\_running\_feds(), add\_federate\_instance\_id(), TrickHLA::Utilities::byteswap\_int(), clear\_running\_feds(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, TrickHLA::ENCODING\_BIG\_ENDIAN, TrickHLA::StringUtilities::ip\_strdup\_wstring(), is\_federate\_executing(), is\_federate\_instance\_id(), TrickHLA::Utilities::is\_transmission\_byteswap(), joined\_federate\_handles, joined\_federate\_name\_map, joined\_federate\_names, MOM\_HLAfederate\_handle, MOM\_HLAfederateName\_handle, name, RTI\_ambassador, running\_feds, running\_feds\_count, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, TRICKHLA\_VALID\_FPU\_CONTROL\_WORD, and update\_running\_feds().

#### 7.21.3.130 set\_MOM\_HLAfederation\_instance\_attributes()

```
void Federate::set_MOM_HLAfederation_instance_attributes (
    RTI1516_NAMESPACE::ObjectInstanceHandle instance_hdl,
    RTI1516_NAMESPACE::AttributeHandleValueMap const & values )
```

Set the Federation ID given the instance ID as well as the FederateHandle ID associated with the Federation instance.

##### Parameters

<i>instance_hdl</i>	<a href="#">Object</a> instance handle.
<i>values</i>	<a href="#">Attribute</a> values.

Definition at line 7003 of file Federate.cpp.

References `auto_provide_setting`, `TrickHLA::Utilities::byteswap_int()`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `TrickHLA::ENCODING_BIG_ENDIAN`, `is_MOM_HLAfederation_instance_id()`, `TrickHLA::Utilities::is_transmission_byteswap()`, `MOM_HLAautoProvide_handle`, `MOM_HLAfederatesInFederation_handle`, `running_feds_count`, `should_print()`, and `THLA_NEWLINE`.

#### 7.21.3.131 `set_requested_time()` [1/2]

```
void Federate::set_requested_time (
    double time )
```

Sets the requested time from the specified double.

Parameters

<i>time</i>	Requested time in seconds.
-------------	----------------------------

Definition at line 3146 of file Federate.cpp.

References `requested_time`, and `TrickHLA::Int64Time::setTo()`.

Referenced by `time_advance_request_to_GALT()`, and `time_advance_request_to_GALT_LCTS_multiple()`.

#### 7.21.3.132 `set_requested_time()` [2/2]

```
void Federate::set_requested_time (
    RTI1516_NAMESPACE::LogicalTime const & time )
```

Sets the requested time from the specified LogicalTime.

Parameters

<i>time</i>	Requested time in HLA logical time.
-------------	-------------------------------------

Definition at line 3152 of file Federate.cpp.

References `requested_time`, and `TrickHLA::Int64Time::setTo()`.

#### 7.21.3.133 `set_restart()`

```
void TrickHLA::Federate::set_restart (
    bool restart_now ) [inline]
```

Set restart flag.

Parameters

<i>restart_now</i>	True for federate restart; False otherwise.
--------------------	---

Definition at line 844 of file Federate.hh.

References `restart_flag`.

Referenced by `IMSim::ExecutionControl::check_freeze_exit()`.

#### 7.21.3.134 `set_restart_cfg()`

```
void TrickHLA::Federate::set_restart_cfg (
```

```
bool restart_cfg_now ) [inline]
```

Set restart configuration flag.

#### Parameters

<i>restart_cfg_now</i>	True for configuring restart; False otherwise.
------------------------	--

Definition at line 848 of file Federate.hh.

References restart\_cfg\_flag.

Referenced by IMSim::ExecutionControl::check\_freeze\_exit().

#### 7.21.3.135 set\_restore\_begun()

```
void TrickHLA::Federate::set_restore_begun ( ) [inline]
```

Set the restore begun state.

Definition at line 420 of file Federate.hh.

References publish\_data, restore\_begun, and restore\_completed.

Referenced by TrickHLA::FedAmb::federationRestoreBegin().

#### 7.21.3.136 set\_restore\_completed()

```
void TrickHLA::Federate::set_restore_completed ( ) [inline]
```

Set the restore completed state.

Definition at line 428 of file Federate.hh.

References publish\_data, restore\_begun, TrickHLA::Restore\_Complete, restore\_completed, restore\_process, and start\_to\_restore.

Referenced by TrickHLA::FedAmb::federationRestored().

#### 7.21.3.137 set\_restore\_failed()

```
void TrickHLA::Federate::set_restore_failed ( ) [inline]
```

Set the restore failed state.

Definition at line 438 of file Federate.hh.

References publish\_data, restore\_begun, restore\_completed, TrickHLA::Restore\_Failed, restore\_process, and start\_to\_restore.

#### 7.21.3.138 set\_restore\_is\_imminent()

```
void TrickHLA::Federate::set_restore_is_imminent ( ) [inline]
```

Set the restore is imminent flag.

Definition at line 462 of file Federate.hh.

References restore\_is\_imminent.

Referenced by IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.21.3.139 set\_restore\_name()

```
void TrickHLA::Federate::set_restore_name (
    std::wstring const & restore_label ) [inline]
```

Set the name of the restore.

**Parameters**

<i>restore_label</i>	Restore name.
----------------------	---------------

Definition at line 775 of file Federate.hh.

References `restore_name`.

**7.21.3.140 `set_restore_request_failed()`**

```
void TrickHLA::Federate::set_restore_request_failed ( ) [inline]
```

Set the restore request failed state.

Definition at line 448 of file Federate.hh.

References `restore_process`, and `TrickHLA::Restore_Request_Failed`.

**7.21.3.141 `set_restore_request_succeeded()`**

```
void TrickHLA::Federate::set_restore_request_succeeded ( ) [inline]
```

Set the restore request succeeded state.

Definition at line 451 of file Federate.hh.

References `restore_process`, and `TrickHLA::Restore_Request_Succeeded`.

**7.21.3.142 `set_save_completed()`**

```
void TrickHLA::Federate::set_save_completed ( ) [inline]
```

Set the save completed state.

Definition at line 412 of file Federate.hh.

References `publish_data`, `save_completed`, and `start_to_save`.

Referenced by `TrickHLA::FedAmb::federationSaved()`.

**7.21.3.143 `set_save_name()`**

```
void TrickHLA::Federate::set_save_name ( std::wstring const & save_label ) [inline]
```

Set the name of the save.

**Parameters**

<i>save_label</i>	Save name.
-------------------	------------

Definition at line 771 of file Federate.hh.

References `save_name`.

Referenced by `setup_checkpoint()`.

**7.21.3.144 `set_start_to_restore()`**

```
void TrickHLA::Federate::set_start_to_restore ( bool restore_flag ) [inline]
```

Set start to restore flag.

## Parameters

<i>restore_flag</i>	True if restore started; False otherwise.
---------------------	---

Definition at line 840 of file Federate.hh.

References start\_to\_restore.

#### 7.21.3.145 set\_start\_to\_save() [1/2]

```
void TrickHLA::Federate::set_start_to_save ( ) [inline]
```

Set the start to save flag.

Definition at line 521 of file Federate.hh.

References start\_to\_save.

Referenced by TrickHLA::FedAmb::federationSaved().

#### 7.21.3.146 set\_start\_to\_save() [2/2]

```
void TrickHLA::Federate::set_start_to_save ( bool save_flag ) [inline]
```

Set start to save flag.

## Parameters

<i>save_flag</i>	True if save started; False otherwise.
------------------	--

Definition at line 836 of file Federate.hh.

References start\_to\_save.

#### 7.21.3.147 set\_startup()

```
void TrickHLA::Federate::set_startup ( bool flag ) [inline]
```

Set federate execution startup state.

## Parameters

<i>flag</i>	True for federate started; False otherwise.
-------------	---

Definition at line 591 of file Federate.hh.

References got\_startup\_sp.

Referenced by DIS::ExecutionControl::sync\_point\_registration\_failed(), and DIS::ExecutionControl::sync\_point\_registration\_succeeded().

#### 7.21.3.148 set\_time\_advance\_grant()

```
void TrickHLA::Federate::set_time_advance_grant ( const bool & grant_flag ) [inline]
```

Set the time advance grant flag.

**Parameters**

<i>grant_flag</i>	Status of time advance grant.
-------------------	-------------------------------

Definition at line 800 of file Federate.hh.

References time\_adv\_grant.

**7.21.3.149 set\_time\_constrained\_state()**

```
void TrickHLA::Federate::set_time_constrained_state (
    const bool & constrained_state ) [inline]
```

Set the state of time constraint.

**Parameters**

<i>constrained_state</i>	Desired state of time constraint for this federate.
--------------------------	---

Definition at line 812 of file Federate.hh.

References time\_constrained\_state.

**7.21.3.150 set\_time\_regulation\_state()**

```
void TrickHLA::Federate::set_time_regulation_state (
    const bool & regulation_state ) [inline]
```

Set the state of time regulation.

**Parameters**

<i>regulation_state</i>	Desired state of time regulation for this federate.
-------------------------	---

Definition at line 808 of file Federate.hh.

References time\_regulating\_state.

**7.21.3.151 setup()**

```
void Federate::setup (
    FedAmb & federate_amb,
    Manager & federate_manager,
    ExecutionControlBase & federate_execution_control )
```

Setup the required class instance associations.

**Parameters**

<i>federate_amb</i>	Associated federate ambassador class instance.
<i>federate_manager</i>	Associated federate manager class instance.
<i>federate_execution_control</i>	Associated federate execution control class instance.

**Assumptions and Limitations:**

- The [TrickHLA::FedAmb](#) class is actually an abstract class. Therefore, the actual object instance being passed in is an instantiable polymorphic child of the RTI1516\_NAMESPACE::FederateAmbassador class.
- The [TrickHLA::ExecutionControlBase](#) class is actually an abstract class. Therefore, the actual object instance being passed in is an instantiable polymorphic child of the [TrickHLA::ExecutionControlBase](#) class.

**Trick Job Class:** *default\_data*

Definition at line 384 of file Federate.cpp.

References [execution\\_control](#), [federate\\_ambassador](#), [manager](#), [TrickHLA::FedAmb::setup\(\)](#), [TrickHLA::ExecutionControlBase::setup\(\)](#), and [TrickHLA::Manager::setup\(\)](#).**7.21.3.152 setup\_checkpoint()**`void Federate::setup_checkpoint ( )`

Perform setup for federate save.

**Assumptions and Limitations:**

- Currently only used with [DIS](#) and [IMSIM](#) initialization schemes. **Trick Job Class:** *checkpoint*

Definition at line 2688 of file Federate.cpp.

References [announce\\_save](#), [convert\\_sync\\_pts\(\)](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [execution\\_control](#), [get\\_federation\\_name\(\)](#), [initiate\\_save\\_flag](#), [is\\_execution\\_member\(\)](#), [TrickHLA::ExecutionControlBase::is\\_save\\_initiated\(\)](#), [manager](#), [request\\_federation\\_save\(\)](#), [request\\_federation\\_save\\_status\(\)](#), [RTI\\_ambassador](#), [save\\_name](#), [set\\_save\\_name\(\)](#), [TrickHLA::Manager::setup\\_checkpoint\(\)](#), [should\\_print\(\)](#), [start\\_to\\_save](#), [str\\_save\\_label](#), [the\\_cpr](#), [THLA\\_ENDL](#), [THLA\\_NEWLINE](#), [TrickHLA::StringUtilities::to\\_string\(\)](#), [TrickHLA::StringUtilities::to\\_wstring\(\)](#), [TRICKHLA\\_RESTORE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_SAVE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_VALIDATE\\_FPU\\_CONTROL\\_WORD](#), [wait\\_for\\_save\\_status\\_to\\_complete\(\)](#), and [write\\_running\\_feds\\_file\(\)](#).**7.21.3.153 setup\_restore()**`void Federate::setup_restore ( )`

Perform setup for federate restore.

**Assumptions and Limitations:**

- Currently only used with [DIS](#) and [IMSIM](#) initialization schemes. **Trick Job Class:** *preload\_checkpoint*

Definition at line 2939 of file Federate.cpp.

References [announce\\_freeze](#), [announce\\_restore](#), [check\\_HLA\\_save\\_directory\(\)](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [get\\_federation\\_name\(\)](#), [initiate\\_restore\\_announce\(\)](#), [is\\_execution\\_member\(\)](#), [is\\_federate\\_executing\(\)](#), [is\\_HLA\\_save\\_and\\_restore\\_supported\(\)](#), [read\\_running\\_feds\\_file\(\)](#), [TrickHLA::Restore\\_In\\_Progress](#), [restore\\_process](#), [should\\_print\(\)](#), [start\\_to\\_restore](#), [str\\_restore\\_label](#), [THLA\\_ENDL](#), [THLA\\_NEWLINE](#), and [wait\\_for\\_required\\_federates\\_to\\_join\(\)](#).**7.21.3.154 setup\_time\_constrained()**`void Federate::setup_time_constrained ( )`

Setup this federate's constrained time management.

**Trick Job Class:** *initialization*.

Definition at line 3898 of file Federate.cpp.

References [check\\_for\\_shutdown\\_with\\_termination\(\)](#), [TrickHLA::DEBUG\\_LEVEL\\_2\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [get\\_federation\\_name\(\)](#), [is\\_execution\\_member\(\)](#), [RTI1516\\_EXCEPTION](#), [RTI\\_ambassador](#),

should\_print(), THLA\_ENDL, THLA\_NEWLINE, time\_adv\_grant, time\_constrained, time\_constrained\_state, time\_management, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.  
Referenced by setup\_time\_management().

#### 7.21.3.155 setup\_time\_management()

```
void Federate::setup_time_management ( )
```

Setup this federate's time management.

**Trick Job Class:** *initialization*.

Definition at line 3844 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, setup\_time\_constrained(), setup\_time\_regulation(), should\_print(), shutdown\_time\_constrained(), shutdown\_time\_regulating(), THLA\_NEWLINE, time\_constrained, time\_constrained\_state, time\_management, time\_regulating, and time\_regulating\_state.

Referenced by DSES::ExecutionControl::post\_multi\_phase\_init\_process(), DIS::ExecutionControl::post\_multi\_phase\_init\_process(), IMSim::ExecutionControl::post\_multi\_phase\_init\_process(), TrickHLA::ExecutionControl::post\_multi\_phase\_init\_processes(), and SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes().

#### 7.21.3.156 setup\_time\_regulation()

```
void Federate::setup_time_regulation ( )
```

Setup this federate's regulate time management.

**Trick Job Class:** *initialization*.

Definition at line 4052 of file Federate.cpp.

References check\_for\_shutdown\_with\_termination(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, TrickHLA::Int64Interval::get(), get\_federation\_name(), TrickHLA::Int64Interval::getDoubleTime(), is\_execution\_member(), lookahead, RTI1516\_EXCEPTION, RTI\_ambassador, should\_print(), THLA\_ENDL, THLA\_NEWLINE, time\_adv\_grant, time\_management, time\_regulating, time\_regulating\_state, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by setup\_time\_management().

#### 7.21.3.157 should\_print()

```
bool Federate::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const
```

Determine if the verbose debug comments should be printed to the console.

##### Returns

Returns true if the requested message should print level.

##### Parameters

<i>level</i>	Debug level of the incoming message.
<i>code</i>	Source code association of the incoming messages.

Definition at line 336 of file Federate.cpp.

References federate\_ambassador, and TrickHLA::FedAmb::should\_print().

Referenced by achieve\_and\_wait\_for\_synchronization(), achieve\_synchronization\_point(), add\_a\_single\_entry\_into\_running\_feds(), ask\_MOM\_for\_auto\_provide\_setting(), ask\_MOM\_for\_federate\_names(), backup\_auto\_provide\_setting\_from\_MOM\_then\_disable(), check\_freeze(), complete\_restore(), create\_and\_join\_federation(), create\_federation(), create\_RTI\_ambassador\_and\_connect(), destroy(), destroy\_orphaned\_federation(), enable\_async\_delivery(), enable\_MOM\_auto\_provide\_setting(), exit\_freeze(), federation\_restored(), federation\_saved(), inform\_RTI\_of\_restore\_completion(), initialize(), initialize\_MOM\_handles(), initiate\_restore\_announce(), initiate\_save\_announce(), join\_federation(), TrickHLA::ExecutionControlBase::join\_federation\_process(), SpaceFOM::ExecutionControl::late\_joine\_hla\_init\_process(), load\_and\_print\_running\_federate\_names(), perform\_checkpoint(), perform\_restore(), perform\_time\_advance\_request(), post\_checkpoint(), SpaceFOM::ExecutionControl::post\_multiphase\_init\_processes(), post\_multiphase\_initialization(), post\_restore(), pre\_multiphase\_initialization(), print\_version(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), IMSim::ExecutionControl::process\_mode\_transition\_request(), publish\_interaction\_class(), rebuild\_federate\_handles(), register\_generic\_sync\_point(), remove\_MOM\_HLAfederate\_instance\_id(), request\_attribute\_update(), request\_federation\_restore\_status(), request\_federation\_save(), request\_federation\_save\_status(), requested\_federation\_restore\_status(), resign(), resign\_so\_we\_can\_rejoin(), restart\_checkpoint(), restart\_initialization(), restore\_federate\_handles\_from\_MOM(), restore\_orig\_MOM\_auto\_provide\_setting(), set\_all\_federate\_MOM\_instance\_handles\_by\_name(), set\_MOM\_HLAfederate\_instance\_attributes(), set\_MOM\_HLAfederation\_instance\_attributes(), setup\_checkpoint(), SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes(), setup\_restore(), setup\_time\_constrained(), setup\_time\_management(), setup\_time\_regulation(), TrickHLA::ExecutionControlBase::should\_print(), shutdown(), shutdown\_time\_constrained(), shutdown\_time\_regulating(), subscribe\_attributes(), time\_advance\_request(), time\_advance\_request\_to\_GALT(), time\_advance\_request\_to\_GALT\_LCTS\_multiple(), unpublish\_interaction\_class(), unsubscribe\_all\_HLAfederate\_class\_attributes\_from\_MOM(), unsubscribe\_all\_HLAfederation\_class\_attributes\_from\_MOM(), unsubscribe\_attributes(), wait\_for\_federation\_restore\_began(), wait\_for\_federation\_restore\_failed\_callback\_to\_complete(), wait\_for\_federation\_restore\_to\_complete(), wait\_for\_required\_federates\_to\_join(), wait\_for\_restore\_request\_callback(), wait\_for\_restore\_status\_to\_complete(), wait\_for\_save\_status\_to\_complete(), wait\_for\_time\_advance\_grant(), TrickHLA::ExecutionConfigurationBase::wait\_on\_update(), DIS::ExecutionConfiguration::wait\_on\_update(), DSES::ExecutionConfiguration::wait\_on\_update(), IMSim::ExecutionConfiguration::wait\_on\_update(), SpaceFOM::ExecutionConfiguration::wait\_on\_update(), and wait\_until\_federation\_is\_ready\_to\_restore().

### 7.21.3.158 should\_publish\_data()

```
bool TrickHLA::Federate::should_publish_data ( ) const [inline]
Query if federate should publish data.
```

#### Returns

True if data should be published; False otherwise.

Definition at line 455 of file Federate.hh.

References publish\_data.

Referenced by TrickHLA::Interaction::send(), TrickHLA::Object::send\_cyclic\_data(), TrickHLA::Object::send\_init\_data(), and TrickHLA::Object::send\_requested\_data().

### 7.21.3.159 shutdown()

```
void Federate::shutdown ( )
```

Shutdown the federate.

Shutdown the federate by shutting down the time management, resigning from the federation, and then attempt to destroy the federation. **Trick Job Class:** *shutdown*

Definition at line 4546 of file Federate.cpp.

References `_FPU_PC_MASK`, `can_rejoin_federation`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `destroy()`, `execution_control_manager`, `TrickHLA::ExecutionControlBase::remove_execution_configuration()`, `resign()`, `resign_so_we_can_rejoin()`, `should_print()`, `TrickHLA::Manager::shutdown()`, `TrickHLA::ExecutionControlBase::shutdown()`, `shutdown_called`, `shutdown_time_management()`, `THLA_NEWLINE`, `TRICKHLA_A_RESTORE_FPU_CONTROL_WORD`, and `TRICKHLA_SAVE_FPU_CONTROL_WORD`.  
 Referenced by `DSES::ExecutionControl::check_freeze_exit()`, `SpaceFOM::ExecutionControl::check_freeze_exit()`, and `~Federate()`.

#### 7.21.3.160 `shutdown_time_constrained()`

```
void Federate::shutdown_time_constrained ( )
```

Shutdown this federate's time constrained time management.

**Trick Job Class:** `shutdown`

Definition at line 4636 of file `Federate.cpp`.

References `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `get_federation_name()`, `RTI1516_EXCEPTION`, `RTI_ambassador`, `should_print()`, `THLA_NEWLINE`, `time_constrained_state`, `TrickHLA::StringUtilities::to_string()`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `setup_time_management()`, and `shutdown_time_management()`.

#### 7.21.3.161 `shutdown_time_management()`

```
void Federate::shutdown_time_management ( )
```

Shutdown this federate's time management.

Shutdown this federate's time management by shutting down time constraint management and time regulating management. **Trick Job Class:** `shutdown`

Definition at line 4627 of file `Federate.cpp`.

References `shutdown_time_constrained()`, and `shutdown_time_regulating()`.

Referenced by `shutdown()`.

#### 7.21.3.162 `shutdown_time_regulating()`

```
void Federate::shutdown_time_regulating ( )
```

Shutdown this federate's time regulating time management.

**Trick Job Class:** `shutdown`

Definition at line 4697 of file `Federate.cpp`.

References `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `get_federation_name()`, `RTI1516_EXCEPTION`, `RTI_ambassador`, `should_print()`, `THLA_NEWLINE`, `time_constrained_state`, `time_regulating_state`, `TrickHLA::StringUtilities::to_string()`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `setup_time_management()`, and `shutdown_time_management()`.

#### 7.21.3.163 `subscribe_attributes()`

```
void Federate::subscribe_attributes (
    RTI1516_NAMESPACE::ObjectClassHandle class_handle,
    RTI1516_NAMESPACE::AttributeHandleSet const & attribute_list ) [private]
```

Subscribe to the specified attributes for the given class handle.

**Parameters****Parameters**

<i>class_handle</i>	Class handle.
<i>attribute_list</i>	Attributes handles.

Definition at line 1879 of file Federate.cpp.

References `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `RTI_ambassador`, `should_print()`, `THLA_NEWLINE`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `ask_MOM_for_auto_provide_setting()`, `ask_MOM_for_federate_names()`, `load_and_print_running_federate_names()`, and `restore_federate_handles_from_MOM()`.

**7.21.3.164 sync\_point\_registration\_failed()**

```
void Federate::sync_point_registration_failed (
    std::wstring const & label,
    bool not_unique )
```

Callback from `TrickHLA::FedAmb` through for when registration of a synchronization point fails. and is one of the sync-points created.

**Parameters**

<i>label</i>	Sync-point label.
<i>not_unique</i>	True if not unique label.

Definition at line 2507 of file Federate.cpp.

References `execution_control`, and `TrickHLA::SyncPntListBase::sync_point_registration_failed()`.

**7.21.3.165 sync\_point\_registration\_succeeded()**

```
void Federate::sync_point_registration_succeeded (
    std::wstring const & label )
```

Marks a synchronization point as registered in the federation.

**Parameters**

<i>label</i>	Sync-point label.
--------------	-------------------

Definition at line 2498 of file Federate.cpp.

References `execution_control`, and `TrickHLA::SyncPntListBase::sync_point_registration_succeeded()`.

**7.21.3.166 time\_advance\_request()**

```
void Federate::time_advance_request ( )
```

Increment the requested time by the lookahead time and make a HLA time advance request.

**Trick Job Class:** *scheduled*

Definition at line 4221 of file Federate.cpp.

References `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `lookahead_time`,

perform\_time\_advance\_request(), requested\_time, save\_completed, should\_print(), THLA\_NEWLINE, and time\_management.

#### 7.21.3.167 time\_advance\_request\_to\_GALT()

void Federate::time\_advance\_request\_to\_GALT ( )

Moves the federates time to the Greatest Available Logical Time (GALT) that is an integer multiple of the Least-Common-Time-Step (LCTS) time if we are time constrained and Not time regulating.

Definition at line 3165 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, execution\_control, TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Int64Interval::getTimeInMicros(), TrickHLA::ExecutionControlBase::is\_late\_joiner(), TrickHLA::ExecutionControlBase::is\_master(), lookahead, perform\_time\_advance\_request(), requested\_time, RTI\_ambassador, set\_requested\_time(), should\_print(), THLA\_NEWLINE, time\_management, TRI-CKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by IMSim::ExecutionControl::post\_multi\_phase\_init\_process(), and TrickHLA::ExecutionControl::post\_multi\_phase\_init\_processes().

#### 7.21.3.168 time\_advance\_request\_to\_GALT\_LCTS\_multiple()

void Federate::time\_advance\_request\_to\_GALT\_LCTS\_multiple ( )

Move the requested time to an integer multiple of the Greatest Available Logical Time (GALT) and Least Common Time Step (LCTS).

Definition at line 3219 of file Federate.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, execution\_control, TrickHLA::ExecutionControlBase::get\_least\_common\_time\_step(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Int64Interval::getTimeInMicros(), TrickHLA::ExecutionControlBase::is\_late\_joiner(), TrickHLA::ExecutionControlBase::is\_master(), lookahead, perform\_time\_advance\_request(), requested\_time, RTI\_ambassador, set\_requested\_time(), should\_print(), THLA\_NEWLINE, time\_management, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes().

#### 7.21.3.169 un\_freeze()

void Federate::un\_freeze ( ) [private]

Unfreeze simulation.

Definition at line 2617 of file Federate.cpp.

References execution\_control, and TrickHLA::ExecutionControlBase::un\_freeze().

Referenced by IMSim::ExecutionControl::check\_freeze\_exit(), DIS::ExecutionControl::enter\_freeze(), IMSim::ExecutionControl::enter\_freeze(), federation\_saved(), IMSim::ExecutionControl::mark\_synchronized(), and unfreeze().

#### 7.21.3.170 unfreeze()

void TrickHLA::Federate::unfreeze ( ) [inline]

Perform federation execution freeze process.

Definition at line 608 of file Federate.hh.

References un\_freeze().

Referenced by SpaceFOM::ExecutionControl::enter\_freeze().

**7.21.3.171 `unpublish_interaction_class()`**

```
void Federate::unpublish_interaction_class (
    RTI1516_NAMESPACE::InteractionClassHandle class_handle ) [private]
```

Unpublish [Interaction](#) class.

**Parameters**

<i>class_handle</i>	<a href="#">Interaction</a> class handle.
---------------------	---

Definition at line 2189 of file Federate.cpp.

References [TrickHLA::DEBUG\\_LEVEL\\_4\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [RTI\\_ambassador](#), [should\\_print\(\)](#), [THLA\\_NEWLINE](#), [TRICKHLA\\_RESTORE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_SAVE\\_FPU\\_CONTROL\\_WORD](#), and [TRICKHLA\\_VALIDATE\\_FPU\\_CONTROL\\_WORD](#).

Referenced by [enable\\_MOM\\_auto\\_provide\\_setting\(\)](#).

**7.21.3.172 `unsubscribe_all_HLAfederate_class_attributes_from_MOM()`**

```
void Federate::unsubscribe_all_HLAfederate_class_attributes_from_MOM ( )
```

Unsubscribe from all MOM federate class attributes.

Definition at line 2074 of file Federate.cpp.

References [TrickHLA::DEBUG\\_LEVEL\\_4\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [MOM\\_HLAfederate](#), [class\\_handle](#), [RTI\\_ambassador](#), [should\\_print\(\)](#), [THLA\\_NEWLINE](#), [TRICKHLA\\_RESTORE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_SAVE\\_FPU\\_CONTROL\\_WORD](#), and [TRICKHLA\\_VALIDATE\\_FPU\\_CONTROL\\_WORD](#).

Referenced by [wait\\_for\\_required\\_federates\\_to\\_join\(\)](#).

**7.21.3.173 `unsubscribe_all_HLAfederation_class_attributes_from_MOM()`**

```
void Federate::unsubscribe_all_HLAfederation_class_attributes_from_MOM ( )
```

Unsubscribe from all MOM federation class attributes.

Definition at line 2113 of file Federate.cpp.

References [TrickHLA::DEBUG\\_LEVEL\\_4\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [MOM\\_HLAfederation](#), [class\\_handle](#), [RTI\\_ambassador](#), [should\\_print\(\)](#), [THLA\\_NEWLINE](#), [TRICKHLA\\_RESTORE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_SAVE\\_FPU\\_CONTROL\\_WORD](#), and [TRICKHLA\\_VALIDATE\\_FPU\\_CONTROL\\_WORD](#).

**7.21.3.174 `unsubscribe_attributes()`**

```
void Federate::unsubscribe_attributes (
    RTI1516_NAMESPACE::ObjectClassHandle class_handle,
    RTI1516_NAMESPACE::AttributeHandleSet const & attribute_list ) [private]
```

Unsubscribe from the specified attributes for the given class handle.

**Parameters**

<i>class_handle</i>	Class handle.
<i>attribute_list</i>	Attributes handles.

Definition at line 1938 of file Federate.cpp.

References [TrickHLA::DEBUG\\_LEVEL\\_4\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_FEDERATE](#), [RTI\\_ambassador](#), [should\\_print\(\)](#), [THLA\\_NEWLINE](#), [TRICKHLA\\_RESTORE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_SAVE\\_FPU\\_CONTROL\\_WORD](#), and [TRICKHLA\\_VALIDATE\\_FPU\\_CONTROL\\_WORD](#).

Referenced by `ask_MOM_for_auto_provide_setting()`, `load_and_print_running_federate_names()`, and `restore_federate_handles_from_MOM()`.

#### 7.21.3.175 `update_running_feds()`

```
void Federate::update_running_feds ( )
```

Update running federates based on current known information.

Definition at line 5633 of file `Federate.cpp`.

References `TrickHLA::StringUtilities::ip_strdup_wstring()`, `joined_federate_name_map`, `mom_HLAfederate_inst_name_map`, `TrickHLA::KnownFederate::MOM_instance_name`, `TrickHLA::KnownFederate::name`, `name`, `TrickHLA::KnownFederate::required`, `running_feds`, `running_feds_count`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `load_and_print_running_federate_names()`, and `set_MOM_HLAfederate_instance_attributes()`.

#### 7.21.3.176 `wait_for_federation_restore_begun()`

```
void Federate::wait_for_federation_restore_begun ( )
```

Blocks until the federation restore has begun.

Definition at line 6252 of file `Federate.cpp`.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `is_execution_member()`, `restore_begun`, `should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `post_restore()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

#### 7.21.3.177 `wait_for_federation_restore_failed_callback_to_complete()`

```
void Federate::wait_for_federation_restore_failed_callback_to_complete ( )
```

Blocks until the RTI responds with a federation not restored callback via the federate ambassador.

Definition at line 6536 of file `Federate.cpp`.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `federation_restore_failed_callback_complete`, `is_execution_member()`, `restore_completed`, `should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `post_restore()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

#### 7.21.3.178 `wait_for_federation_restore_to_complete()`

```
string Federate::wait_for_federation_restore_to_complete ( )
```

Blocks until the federation restore is complete.

Returns

Empty string if successful, descriptive string on failure.

Definition at line 6330 of file `Federate.cpp`.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `federation_restore_failed_callback_complete`, `is_execution_member()`, `restore_completed`, `TrickHLA::Restore_Failed`, `restore_failed`, `restore_process`, `running_feds_count`, `running_feds_count_at_time_of_restore`, `should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `post_restore()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

#### 7.21.3.179 `wait_for_required_federates_to_join()`

```
string Federate::wait_for_required_federates_to_join ( )
```

Wait for all the required federates to joined the federation.

**Returns**

A non-empty string whrn there is a problem.

**Trick Job Class: *initialization***

Definition at line 1350 of file Federate.cpp.

References `all_federates_joined`, `ask_MOM_for_federate_names()`, `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `determine_federate_MOM_object_instance_names()`, `enable_known_feds`, `is_execution_member()`, `is_joined_federate()`, `is_required_federate()`, `joined_federate_handles`, `joined_federate_names`, `known_feds`, `known_feds_count`, `TrickHLA::KnownFederate::name`, `name`, `restore_is_imminent`, `should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, and `unsubscribe_all_HLAfederate_class_attributes_from_MOM()`.

Referenced by `DSES::ExecutionControl::pre_multi_phase_init_processes()`, `DIS::ExecutionControl::pre_multi_phase_init_processes()`, `IMSim::ExecutionControl::pre_multi_phase_init_processes()`, `TrickHLA::Manager::restart_initialization()`, `SpaceFOM::ExecutionControl::role_determination_process()`, and `setup_restore()`.

**7.21.3.180 `wait_for_restore_request_callback()`**

```
void Federate::wait_for_restore_request_callback ( )
```

Blocks until the RTI responds with a federation request request success / failure.

Definition at line 6419 of file Federate.cpp.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `has_restore_process_restore_request_failed()`, `has_restore_process_restore_request_succeeded()`, `is_execution_member()`, `should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

**7.21.3.181 `wait_for_restore_status_to_complete()`**

```
void Federate::wait_for_restore_status_to_complete ( )
```

Blocks until the RTI responds with a federation status of the restore is complete.

Definition at line 6458 of file Federate.cpp.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `is_execution_member()`, `restore_request_complete`, `should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `initiate_restore_announce()`.

**7.21.3.182 `wait_for_save_status_to_complete()`**

```
void Federate::wait_for_save_status_to_complete ( )
```

Blocks until the RTI responds with a federation status of the save is complete.

Definition at line 6497 of file Federate.cpp.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `is_execution_member()`, `save_request_complete`, `should_print()`, `THLA_ENDL`, and `THLA_NEWLINE`.

Referenced by `setup_checkpoint()`.

**7.21.3.183 `wait_for_time_advance_grant()` [1/2]**

```
void Federate::wait_for_time_advance_grant ( )
```

Wait for a HLA time-advance grant.

**Trick Job Class: *scheduled***

Definition at line 4375 of file Federate.cpp.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_LEVEL_5_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `get_granted_time()`, `TrickHLA::Int64Time::getDoubleTime()`, `HLA_time`, `is_execution_member()`, `requested_time`, `should_print()`, `THLA_ENDIAN`, `THLA_NEWLINE`, `time_adv_grant`, and `time_management`.

Referenced by `IMSim::FreezeInteractionHandler::send_scenario_freeze_interaction()`.

#### 7.21.3.184 `wait_for_time_advance_grant()` [2/2]

```
void Federate::wait_for_time_advance_grant (
    int time_out_tolerance )
```

Wait for a HLA time-advance grant, but allow for an early exit if it takes longer than `time_out_tolerance` (for SSTF).

##### Parameters

<code>time_out_tolerance</code>	Timeout tolerance in nanoseconds.
---------------------------------	-----------------------------------

##### Assumptions and Limitations:

- Currently only used with **DIS** initialization scheme. **Trick Job Class:** *scheduled*

Definition at line 4467 of file `Federate.cpp`.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_LEVEL_4_TRACE`, `TrickHLA::DEBUG_LEVEL_5_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `get_granted_time()`, `TrickHLA::Int64Time::getDoubleTime()`, `HLA_time`, `requested_time`, `should_print()`, `stale_data_counter`, `THLA_ENDIAN`, `THLA_NEWLINE`, `time_adv_grant`, and `time_management`.

#### 7.21.3.185 `wait_until_federation_is_ready_to_restore()`

```
void Federate::wait_until_federation_is_ready_to_restore ( )
```

Blocks until the federation is ready to restore.

Definition at line 6291 of file `Federate.cpp`.

References `check_for_shutdown_with_termination()`, `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_FEDERATE`, `is_execution_member()`, `should_print()`, `start_to_restore`, `THLA_ENDIAN`, and `THLA_NEWLINE`.

Referenced by `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

#### 7.21.3.186 `write_running_feds_file()`

```
void Federate::write_running_feds_file (
    char * file_name ) throw ( const char * ) [private]
```

Dumps the contents of the `running_feds` object into the supplied file name with `".running_feds"` appended to it.

##### Parameters

<code>file_name</code>	Checkpoint file name.
------------------------	-----------------------

Definition at line 5860 of file `Federate.cpp`.

References `THLA_ENDIAN`.

Referenced by `setup_checkpoint()`.

## 7.21.4 Friends And Related Function Documentation

### 7.21.4.1 init\_attrTrickHLA\_Federate

```
void init_attrTrickHLA_Federate ( ) [friend]
```

### 7.21.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 89 of file Federate.hh.

## 7.21.5 Field Documentation

### 7.21.5.1 all\_federates\_joined

```
bool TrickHLA::Federate::all_federates_joined [private]
```

#### Units: -

Master check for all federates joined.

Definition at line 905 of file Federate.hh.

Referenced by load\_and\_print\_running\_federate\_names(), and wait\_for\_required\_federates\_to\_join().

### 7.21.5.2 announce\_freeze

```
bool TrickHLA::Federate::announce_freeze [private]
```

#### Data I/O: \*\*

DANNY2.7 flag to indicate that this federate is announcing go to freeze mode

Definition at line 960 of file Federate.hh.

Referenced by DIS::ExecutionControl::check\_freeze\_exit(), IMSim::ExecutionControl::check\_freeze\_exit(), IMSim::ExecutionControl::check\_freeze\_time(), DIS::ExecutionControl::enter\_freeze(), IMSim::ExecutionControl::enter\_freeze(), DIS::ExecutionControl::exit\_freeze(), IMSim::ExecutionControl::exit\_freeze(), exit\_freeze(), federation\_saved(), get\_freeze\_announced(), set\_freeze\_announced(), and setup\_restore().

### 7.21.5.3 announce\_restore

```
bool TrickHLA::Federate::announce_restore [private]
```

#### Data I/O: \*\*

flag to indicate whether we have announced the federation restore

Definition at line 938 of file Federate.hh.

Referenced by federation\_restored(), perform\_restore(), post\_restore(), and setup\_restore().

### 7.21.5.4 announce\_save

```
bool TrickHLA::Federate::announce_save [private]
```

#### Data I/O: \*\*

flag to indicate whether we have announced the federation save

Definition at line 928 of file Federate.hh.

Referenced by `IMSim::ExecutionControl::add_freeze_scenario_time()`, `IMSim::ExecutionControl::check_freeze_time()`, `federation_saved()`, `IMSim::ExecutionControl::is_save_initiated()`, `IMSim::ExecutionControl::mark_synchronized()`, `perform_checkpoint()`, `set_announce_save()`, and `setup_checkpoint()`.

#### 7.21.5.5 `auto_provide_setting`

```
int TrickHLA::Federate::auto_provide_setting [private]
```

**Units:** –

MOM Federation wide HLAautoProvide setting.

Definition at line 988 of file `Federate.hh`.

Referenced by `ask_MOM_for_auto_provide_setting()`, `backup_auto_provide_setting_from_MOM_then_disable()`, `enable_MOM_auto_provide_setting()`, `restore_orig_MOM_auto_provide_setting()`, and `set_MOM_HLAfederation←instance_attributes()`.

#### 7.21.5.6 `can_rejoin_federation`

```
bool TrickHLA::Federate::can_rejoin_federation
```

**Units:** –

Enables this federate to resign in a way to allow re-joining of the federation at a later time.

Definition at line 127 of file `Federate.hh`.

Referenced by `federate_can_rejoin_federation()`, and `shutdown()`.

#### 7.21.5.7 `checkpoint_file_name`

```
char TrickHLA::Federate::checkpoint_file_name[256] [private]
```

**Data I/O:** *\*i*

**Units:** –

label to attach to sync point

Definition at line 958 of file `Federate.hh`.

Referenced by `Federate()`, `federation_saved()`, `initiate_save_announce()`, and `set_checkpoint_file_name()`.

#### 7.21.5.8 `checkpoint_rt_itimer`

```
Flag TrickHLA::Federate::checkpoint_rt_itimer [private]
```

**Data I/O:** *\*\**

loaded checkpoint RT ITIMER

Definition at line 959 of file `Federate.hh`.

#### 7.21.5.9 `cstr_restore_label`

```
char TrickHLA::Federate::cstr_restore_label[256] [private]
```

**Data I/O:** *\*\**

Restore file label in C string format

Definition at line 935 of file `Federate.hh`.

Referenced by `Federate()`, `federation_restored()`, and `initiate_restore_announce()`.

#### 7.21.5.10 `cstr_save_label`

```
char TrickHLA::Federate::cstr_save_label[256] [private]
```

**Data I/O: \*\***

Save file label in C string format

Definition at line 925 of file Federate.hh.

Referenced by Federate(), federation\_saved(), and initiate\_save\_announce().

**7.21.5.11 enable\_known\_feds**

```
bool TrickHLA::Federate::enable_known_feds
```

**Units: –**

Enable use of known Federates list (default: true)

Definition at line 123 of file Federate.hh.

Referenced by TrickHLA::ExecutionControl::initialize(), restart\_initialization(), and wait\_for\_required\_federates\_to\_join().

**7.21.5.12 execution\_control**

```
ExecutionControlBase* TrickHLA::Federate::execution_control [private]
```

**Units: –**

Execution control object.

This has to point to an allocated execution control class that inherits from the [ExecutionControlBase](#) interface class. For instance SRFOM::ExecutionControl.

Definition at line 1008 of file Federate.hh.

Referenced by achieve\_and\_wait\_for\_synchronization(), announce\_sync\_point(), check\_for\_shutdown(), check\_for\_shutdown\_with\_termination(), check\_freeze(), convert\_sync\_pts(), enter\_freeze(), exit\_freeze(), federation\_synchronized(), freeze\_init(), get\_execution\_control(), initialize(), is\_HLA\_save\_and\_restore\_supported(), perform\_checkpoint(), post\_multiphase\_initialization(), pre\_multiphase\_initialization(), reinstate\_logged\_sync\_pts(), restore\_checkpoint(), setup(), setup\_checkpoint(), shutdown(), sync\_point\_registration\_failed(), sync\_point\_registration\_succeeded(), time\_advance\_request\_to\_GALT(), time\_advance\_request\_to\_GALT\_LCTS\_multiple(), and un\_freeze().

**7.21.5.13 execution\_has\_begun**

```
bool TrickHLA::Federate::execution_has_begun [private]
```

**Units: –**

flag to indicate if the federate has begun simulation execution.

Definition at line 962 of file Federate.hh.

Referenced by is\_federate\_executing(), set\_federate\_has\_begun\_execution(), and ~Federate().

**7.21.5.14 federate\_ambassador**

```
FedAmb* TrickHLA::Federate::federate_ambassador [private]
```

**Units: –**

[Federate](#) ambassador.

Definition at line 1006 of file Federate.hh.

Referenced by create\_RTI\_ambassador\_and\_connect(), get\_fed\_ambassador(), initialize(), initiate\_restore\_announce(), join\_federation(), requested\_federation\_restore\_status(), restart\_initialization(), restore\_federate\_handles\_from\_MOM(), setup(), should\_print(), and ~Federate().

**7.21.5.15 federate\_has\_been\_restarted**

```
bool TrickHLA::Federate::federate_has_been_restarted [private]
```

**Data I/O: \*\*****Federate** has restarted; so, do not restart again!

Definition at line 945 of file Federate.hh.

**7.21.5.16 federate\_id**

RTI1516\_NAMESPACE::FederateHandle TrickHLA::Federate::federate\_id [private]

**Data I/O: \*\*****Federate** ID.

Definition at line 901 of file Federate.hh.

Referenced by join\_federation().

**7.21.5.17 federation\_created\_by\_federate**

bool TrickHLA::Federate::federation\_created\_by\_federate [private]

**Data I/O: \*\*****Federate** successfully created the federation if True.

Definition at line 902 of file Federate.hh.

Referenced by create\_federation(), is\_federation\_created\_by\_federate(), and join\_federation().

**7.21.5.18 federation\_exists**

bool TrickHLA::Federate::federation\_exists [private]

**Data I/O: \*\***

Federation exists.

Definition at line 903 of file Federate.hh.

Referenced by create\_and\_join\_federation(), create\_federation(), destroy(), and join\_federation().

**7.21.5.19 federation\_joined**

bool TrickHLA::Federate::federation\_joined [private]

**Data I/O: \*\*****Federate** joined federation flag.

Definition at line 904 of file Federate.hh.

Referenced by create\_and\_join\_federation(), destroy(), join\_federation(), resign(), and resign\_so\_we\_can\_rejoin().

**7.21.5.20 federation\_name**

char\* TrickHLA::Federate::federation\_name

**Units: -**

Federation execution name.

Definition at line 100 of file Federate.hh.

Referenced by create\_federation(), create\_RTI\_ambassador\_and\_connect(), destroy(), destroy\_orphaned\_federation(), get\_federation\_name(), join\_federation(), restart\_initialization(), set\_federation\_name(), and ~Federate().

**7.21.5.21 federation\_restore\_failed\_callback\_complete**

bool TrickHLA::Federate::federation\_restore\_failed\_callback\_complete [private]

**Data I/O: \*\***

federation not restored callback complete

Definition at line 943 of file Federate.hh.

Referenced by `print_restore_failure_reason()`, `wait_for_federation_restore_failed_callback_to_complete()`, and `wait_for_federation_restore_to_complete()`.

**7.21.5.22 FOM\_modules**

```
char* TrickHLA::Federate::FOM_modules
```

**Units: -**

FOM filename for the IEEE 1516-2000 and SISO-STD-004.1-2004 standards, or a comma separated list of FOM-module filenames for IEEE 1516-2010.

Definition at line 107 of file Federate.hh.

Referenced by `create_federation()`, `join_federation()`, `restart_initialization()`, and `~Federate()`.

**7.21.5.23 freeze\_delay\_frames**

```
double TrickHLA::Federate::freeze_delay_frames
```

**Units: -**

For [DIS](#): Number of lookahead\_time frames to delay when freeze issued so all feds freeze together.

Definition at line 131 of file Federate.hh.

Referenced by `DIS::ExecutionControl::enter_freeze()`.

**7.21.5.24 freeze\_the\_federation**

```
bool TrickHLA::Federate::freeze_the_federation [private]
```

**Data I/O: \*\***

DANNY2.7 flag to indicate the federation is going into freeze now

Definition at line 961 of file Federate.hh.

Referenced by `IMSim::ExecutionControl::check_pause()`, `IMSim::ExecutionControl::check_scenario_freeze_time()`, `DIS::ExecutionControl::enter_freeze()`, `IMSim::ExecutionControl::enter_freeze()`, `enter_freeze()`, `DIS::ExecutionControl::exit_freeze()`, `IMSim::ExecutionControl::exit_freeze()`, `exit_freeze()`, and `get_freeze_pending()`.

**7.21.5.25 got\_startup\_sp**

```
bool TrickHLA::Federate::got_startup_sp [private]
```

**Units: -**

"startup" SP has been created. For [DIS](#) compatibility

Definition at line 981 of file Federate.hh.

Referenced by `set_startup()`.

**7.21.5.26 granted\_time**

```
Int64Time TrickHLA::Federate::granted_time [private]
```

**Units: -**

HLA time given by RTI

Definition at line 968 of file Federate.hh.

Referenced by `IMSim::ExecutionControl::check_scenario_freeze_time()`, `get_granted_fed_time()`, `get_granted_time()`, `post_restore()`, `restart_checkpoint()`, and `set_granted_time()`.

### 7.21.5.27 HLA\_save\_directory

char\* TrickHLA::Federate::HLA\_save\_directory [private]

**Data I/O:** *\*i*

**Units:** –

HLA Save directory

Definition at line 915 of file Federate.hh.

Referenced by check\_HLA\_save\_directory(), perform\_restore(), and restore\_checkpoint().

### 7.21.5.28 HLA\_time

double TrickHLA::Federate::HLA\_time [private]

**Units:** *s*

Current HLA time.

Definition at line 970 of file Federate.hh.

Referenced by post\_restore(), restart\_checkpoint(), and wait\_for\_time\_advance\_grant().

### 7.21.5.29 initiate\_restore\_flag

bool TrickHLA::Federate::initiate\_restore\_flag [private]

**Data I/O:** *\*\**

Restore announce flag

Definition at line 920 of file Federate.hh.

Referenced by process\_requested\_federation\_restore\_status().

### 7.21.5.30 initiate\_save\_flag

bool TrickHLA::Federate::initiate\_save\_flag [private]

**Data I/O:** *\*\**

Save announce flag

Definition at line 916 of file Federate.hh.

Referenced by DIS::ExecutionControl::is\_save\_initiated(), IMSim::ExecutionControl::is\_save\_initiated(), IMSim::ExecutionControl::mark\_synchronized(), process\_requested\_federation\_save\_status(), and setup\_checkpoint().

### 7.21.5.31 joined\_federate\_handles

RTI1516\_NAMESPACE::FederateHandleSet TrickHLA::Federate::joined\_federate\_handles [private]

**Data I/O:** *\*\**

FederateHandles of joined federates.

Definition at line 997 of file Federate.hh.

Referenced by get\_joined\_federate\_handles(), load\_and\_print\_running\_federate\_names(), rebuild\_federate\_handles(), restore\_federate\_handles\_from\_MOM(), set\_MOM\_HLAfederate\_instance\_attributes(), wait\_for\_required\_federates\_to\_join(), and ~Federate().

### 7.21.5.32 joined\_federate\_name\_map

TrickHLAObjInstanceNameMap TrickHLA::Federate::joined\_federate\_name\_map [private]

**Data I/O:** *\*\**

Map of the federate instances.

Definition at line 996 of file Federate.hh.

Referenced by `add_a_single_entry_into_running_feds()`, `add_federate_instance_id()`, `determine_federate_MOM_object_instance_names()`, `is_federate_instance_id()`, `load_and_print_running_federate_names()`, `remove_federate_instance_id()`, `set_all_federate_MOM_instance_handles_by_name()`, `set_federate_has_begun_execution()`, `set_MOM_HLAfederate_instance_attributes()`, `update_running_feds()`, and `~Federate()`.

#### 7.21.5.33 joined\_federate\_names

`VectorOfWstrings TrickHLA::Federate::joined_federate_names [private]`

**Data I/O:** \*\*

Names of the joined federates.

Definition at line 998 of file `Federate.hh`.

Referenced by `is_joined_federate()`, `load_and_print_running_federate_names()`, `set_MOM_HLAfederate_instance_attributes()`, `wait_for_required_federates_to_join()`, and `~Federate()`.

#### 7.21.5.34 known\_feds

`KnownFederate* TrickHLA::Federate::known_feds`

**Units:** –

Array of all the known Federates in the simulation.

Definition at line 125 of file `Federate.hh`.

Referenced by `TrickHLA::ExecutionConfiguration::configure()`, `copy_running_feds_into_known_feds()`, `determine_federate_MOM_object_instance_names()`, `is_a_required_startup_federate()`, `is_required_federate()`, `restart_initialization()`, `set_all_federate_MOM_instance_handles_by_name()`, `wait_for_required_federates_to_join()`, and `~Federate()`.

#### 7.21.5.35 known\_feds\_count

`int TrickHLA::Federate::known_feds_count`

**Units:** –

Number of required Federates (default: 0)

Definition at line 124 of file `Federate.hh`.

Referenced by `TrickHLA::ExecutionConfiguration::configure()`, `copy_running_feds_into_known_feds()`, `determine_federate_MOM_object_instance_names()`, `TrickHLA::ExecutionControl::initialize()`, `is_a_required_startup_federate()`, `is_required_federate()`, `restart_initialization()`, `set_all_federate_MOM_instance_handles_by_name()`, `wait_for_required_federates_to_join()`, and `~Federate()`.

#### 7.21.5.36 local\_settings

`char* TrickHLA::Federate::local_settings`

**Units:** –

Vendor specific HLA-Evolved local settings for the connect API.

Pitch RTI: "crcHost = 192.168.1.1\ncrcPort = 8989"

MAK RTI: "(setqb RTI\_tcpForwarderAddr \"192.168.1.1\") (setqb RTI\_distributedForwarderPort 5000)"

Definition at line 102 of file `Federate.hh`.

Referenced by `create_RTI_ambassador_and_connect()`, and `~Federate()`.

#### 7.21.5.37 lookahead

`Int64Interval TrickHLA::Federate::lookahead [private]`

**Units:** –

Lookahead time for data.

Definition at line 907 of file Federate.hh.

Referenced by `get_lookinghead()`, `set_lookinghead()`, `setup_time_regulation()`, `time_advance_request_to_GALT()`, and `time_advance_request_to_GALT_LCTS_multiple()`.

**7.21.5.38 lookahead\_time**

```
double TrickHLA::Federate::lookahead_time
```

**Units:** s

The HLA lookahead time in seconds.

Definition at line 115 of file Federate.hh.

Referenced by `DIS::ExecutionControl::enter_freeze()`, `get_lookinghead_time()`, `is_zero_lookinghead_time()`, `restart_initialization()`, `set_lookinghead()`, and `time_advance_request()`.

**7.21.5.39 make\_copy\_of\_run\_directory**

```
bool TrickHLA::Federate::make_copy_of_run_directory [private]
```

**Units:** –

Make a backup of RUN directory before restarting the federation via federation manager (default: false).

Definition at line 982 of file Federate.hh.

**7.21.5.40 manager**

```
Manager* TrickHLA::Federate::manager [private]
```

**Units:** –

Associated [TrickHLA Federate](#).

Definition at line 1007 of file Federate.hh.

Referenced by `get_manager()`, `initialize()`, `post_restore()`, `pre_multiphase_initialization()`, `setup()`, `setup_checkpoint()`, and `shutdown()`.

**7.21.5.41 MIM\_module**

```
char* TrickHLA::Federate::MIM_module
```

**Units:** –

Filename for the MOM and Initialization Module (MIM) for HLA IEEE 1516-2010.

Definition at line 110 of file Federate.hh.

Referenced by `create_federation()`, and `~Federate()`.

**7.21.5.42 MOM\_HLAautoProvide\_handle**

```
RTI1516_NAMESPACE::AttributeHandle TrickHLA::Federate::MOM_HLAautoProvide_handle [private]
```

**Data I/O:** \*\*

MOM AutoProvide attribute handle.

Definition at line 986 of file Federate.hh.

Referenced by `ask_MOM_for_auto_provide_setting()`, `initialize_MOM_handles()`, and `set_MOM_HLAfederation_instance_attributes()`.

#### 7.21.5.43 MOM\_HLAautoProvide\_param\_handle

```
RTI1516_NAMESPACE::ParameterHandle TrickHLA::Federate::MOM_HLAautoProvide_param_handle [private]  
Data I/O: **
```

MOM HLAautoProvide parameter handle.

Definition at line 1001 of file Federate.hh.

Referenced by enable\_MOM\_auto\_provide\_setting(), and initialize\_MOM\_handles().

#### 7.21.5.44 MOM\_HLAfederate\_class\_handle

```
RTI1516_NAMESPACE::ObjectClassHandle TrickHLA::Federate::MOM_HLAfederate_class_handle [private]  
Data I/O: **
```

MOM Federate class handle.

Definition at line 991 of file Federate.hh.

Referenced by ask\_MOM\_for\_federate\_names(), get\_MOM\_HLAfederate\_class\_handle(), initialize\_MOM\_handles(), is\_MOM\_HLAfederate\_class(), restore\_federate\_handles\_from\_MOM(), and unsubscribe\_all\_HLAfederate\_class\_attributes\_from\_MOM().

#### 7.21.5.45 MOM\_HLAfederate\_handle

```
RTI1516_NAMESPACE::AttributeHandle TrickHLA::Federate::MOM_HLAfederate_handle [private]
```

**Data I/O:** \*\*

MOM attribute handle to Federate-Handle.

Definition at line 994 of file Federate.hh.

Referenced by ask\_MOM\_for\_federate\_names(), initialize\_MOM\_handles(), restore\_federate\_handles\_from\_MOM(), and set\_MOM\_HLAfederate\_instance\_attributes().

#### 7.21.5.46 mom\_HLAfederate\_inst\_name\_map

```
TrickHLAObjInstanceNameMap TrickHLA::Federate::mom_HLAfederate_inst_name_map [private]
```

**Data I/O:** \*\*

Map of the MOM HLAfederate instances name map.

Definition at line 995 of file Federate.hh.

Referenced by add\_a\_single\_entry\_into\_running\_feds(), add\_MOM\_HLAfederate\_instance\_id(), remove\_MOM\_HLAfederate\_instance\_id(), update\_running\_feds(), and ~Federate().

#### 7.21.5.47 MOM\_HLAfederateName\_handle

```
RTI1516_NAMESPACE::AttributeHandle TrickHLA::Federate::MOM_HLAfederateName_handle [private]
```

**Data I/O:** \*\*

MOM attribute handle to Federate name.

Definition at line 993 of file Federate.hh.

Referenced by ask\_MOM\_for\_federate\_names(), initialize\_MOM\_handles(), and set\_MOM\_HLAfederate\_instance\_attributes().

#### 7.21.5.48 MOM\_HLAfederatesInFederation\_handle

```
RTI1516_NAMESPACE::AttributeHandle TrickHLA::Federate::MOM_HLAfederatesInFederation_handle [private]
```

**Data I/O:** \*\*

MOM attribute handle to Federate-count.

Definition at line 985 of file Federate.hh.

Referenced by initialize\_MOM\_handles(), load\_and\_print\_running\_federate\_names(), and set\_MOM\_HLAfederation<instance\_attributes().

#### 7.21.5.49 MOM\_HLAfederateType\_handle

RTI1516\_NAMESPACE::AttributeHandle TrickHLA::Federate::MOM\_HLAfederateType\_handle [private]

**Data I/O:** \*\*

MOM attribute handle to [Federate](#) type (a.k.a name in IEEE 1516-2000).

Definition at line 992 of file Federate.hh.

Referenced by initialize\_MOM\_handles().

#### 7.21.5.50 MOM\_HLAfederation\_class\_handle

RTI1516\_NAMESPACE::ObjectClassHandle TrickHLA::Federate::MOM\_HLAfederation\_class\_handle [private]

**Data I/O:** \*\*

MOM Federation class handle.

Definition at line 984 of file Federate.hh.

Referenced by ask\_MOM\_for\_auto\_provide\_setting(), initialize\_MOM\_handles(), is\_MOM\_HLAfederation\_class(), load\_and\_print\_running\_federate\_names(), and unsubscribe\_all\_HLAfederation\_class\_attributes\_from\_MOM().

#### 7.21.5.51 mom\_HLAfederation\_instance\_name\_map

TrickHLAObjInstanceNameMap TrickHLA::Federate::mom\_HLAfederation\_instance\_name\_map [private]

**Data I/O:** \*\*

Map of the MOM HLAfederation instances.

Definition at line 987 of file Federate.hh.

Referenced by add\_MOM\_HLAfederation\_instance\_id(), is\_MOM\_HLAfederation\_instance\_id(), remove\_MOM\_HLAfederation\_instance\_id(), and ~Federate().

#### 7.21.5.52 MOM\_HLAssetSwitches\_class\_handle

RTI1516\_NAMESPACE::InteractionClassHandle TrickHLA::Federate::MOM\_HLAssetSwitches\_class\_handle [private]

**Data I/O:** \*\*

MOM HLAssetSwitches class handle.

Definition at line 1000 of file Federate.hh.

Referenced by enable\_MOM\_auto\_provide\_setting(), and initialize\_MOM\_handles().

#### 7.21.5.53 name

char\* TrickHLA::Federate::name

**Units:** –

The federate name.

Definition at line 98 of file Federate.hh.

Referenced by add\_a\_single\_entry\_into\_running\_feds(), clear\_running\_feds(), copy\_running\_feds\_into\_known\_feds(), create\_RTI\_ambassador\_and\_connect(), determine\_federate\_MOM\_object\_instance\_names(), get\_federate\_name(), initialize(), initiate\_save\_announce(), is\_a\_required\_startup\_federate(), is\_required\_federate(), print\_requested\_federation\_restore\_status(), remove\_MOM\_HLAfederate\_instance\_id(), restart\_initialization(), set\_MOM\_HLAfederate\_instance\_attributes(), update\_running\_feds(), wait\_for\_required\_federates\_to\_join(), and ~Federate().

#### 7.21.5.54 orig\_auto\_provide\_setting

```
int TrickHLA::Federate::orig_auto_provide_setting [private]
```

**Units:** -

Original MOM Federation wide HLAautoProvide setting when we joined the federation.

Definition at line 989 of file Federate.hh.

Referenced by backup\_auto\_provide\_setting\_from\_MOM\_then\_disable(), and restore\_orig\_MOM\_auto\_provide\_setting().

#### 7.21.5.55 prev\_restore\_process

```
THLASaveRestoreProcEnum TrickHLA::Federate::prev_restore_process [private]
```

**Data I/O:** \*\*

previous state of the restore process

Definition at line 919 of file Federate.hh.

Referenced by inform\_RTI\_of\_restore\_completion(), post\_restore(), and restore\_checkpoint().

#### 7.21.5.56 publish\_data

```
bool TrickHLA::Federate::publish_data [private]
```

**Data I/O:** \*\*

Default true.

indicates if this federate's data & interactions should be processed.

Definition at line 948 of file Federate.hh.

Referenced by set\_restore\_begun(), set\_restore\_completed(), set\_restore\_failed(), set\_save\_completed(), and should\_publish\_data().

#### 7.21.5.57 requested\_time

```
Int64Time TrickHLA::Federate::requested_time [private]
```

**Units:** -

requested/desired HLA time

Definition at line 969 of file Federate.hh.

Referenced by IMSim::ExecutionControl::check\_pause(), get\_requested\_fed\_time(), get\_requested\_time(), perform\_time\_advance\_request(), post\_restore(), restart\_checkpoint(), set\_requested\_time(), time\_advance\_request(), time\_advance\_request\_to\_GALT(), time\_advance\_request\_to\_GALT\_LCTS\_multiple(), and wait\_for\_time\_advance\_grant().

#### 7.21.5.58 restart\_cfg\_flag

```
bool TrickHLA::Federate::restart_cfg_flag [private]
```

**Data I/O:** \*\*

Restart flag with new configuration

Definition at line 974 of file Federate.hh.

Referenced by get\_restart\_cfg(), and set\_restart\_cfg().

#### 7.21.5.59 restart\_flag

```
bool TrickHLA::Federate::restart_flag [private]
```

**Data I/O: \*\***

Restart flag

Definition at line 973 of file Federate.hh.

Referenced by `get_restart()`, and `set_restart()`.**7.21.5.60 `restore_begun`**`bool TrickHLA::Federate::restore_begun [private]`**Data I/O: \*\***

Restore begun

Definition at line 940 of file Federate.hh.

Referenced by `federation_restored()`, `has_restore_been_announced()`, `set_restore_begun()`, `set_restore_completed()`, `set_restore_failed()`, and `wait_for_federation_restore_begun()`.**7.21.5.61 `restore_completed`**`bool TrickHLA::Federate::restore_completed [private]`**Data I/O: \*\***

Restore completed.

Definition at line 942 of file Federate.hh.

Referenced by `complete_restore()`, `set_restore_begun()`, `set_restore_completed()`, `set_restore_failed()`, `wait_for_federation_restore_failed_callback_to_complete()`, and `wait_for_federation_restore_to_complete()`.**7.21.5.62 `restore_failed`**`bool TrickHLA::Federate::restore_failed [private]`**Data I/O: \*\***

Restore of the federate failed

Definition at line 922 of file Federate.hh.

Referenced by `wait_for_federation_restore_to_complete()`.**7.21.5.63 `restore_in_progress`**`bool TrickHLA::Federate::restore_in_progress [private]`**Data I/O: \*\***

Restore in progress flag

Definition at line 921 of file Federate.hh.

**7.21.5.64 `restore_is_imminent`**`bool TrickHLA::Federate::restore_is_imminent [private]`**Data I/O: \*\***Restore has been signalled by the [Manager](#)

Definition at line 923 of file Federate.hh.

Referenced by `federation_restored()`, `set_restore_is_imminent()`, and `wait_for_required_federates_to_join()`.**7.21.5.65 `restore_label_generated`**`bool TrickHLA::Federate::restore_label_generated [private]`

**Data I/O: \*\***

Restore filename has been generated.

Definition at line 939 of file Federate.hh.

**7.21.5.66 restore\_name**

```
std::wstring TrickHLA::Federate::restore_name [private]
```

**Data I/O: \*\***

Name for a restore file

Definition at line 912 of file Federate.hh.

Referenced by `initiate_restore_announce()`, `perform_restore()`, and `set_restore_name()`.

**7.21.5.67 restore\_process**

```
THLA_SaveRestoreProcEnum TrickHLA::Federate::restore_process [private]
```

**Data I/O: \*\***

Where we are in the restore process

Definition at line 918 of file Federate.hh.

Referenced by `complete_restore()`, `federation_restored()`, `has_restore_process_restore_request_failed()`, `has_restore_process_restore_request_succeeded()`, `has_restore_request_failed()`, `has_restore_request_succeeded()`, `inform_RTI_of_restore_completion()`, `initiate_restore_announce()`, `post_restore()`, `process_requested_federation_restore_status()`, `restart_checkpoint()`, `restore_checkpoint()`, `set_restore_completed()`, `set_restore_failed()`, `set_restore_request_failed()`, `set_restore_request_succeeded()`, `setup_restore()`, and `wait_for_federation_restore_to_complete()`.

**7.21.5.68 restore\_request\_complete**

```
bool TrickHLA::Federate::restore_request_complete [private]
```

**Data I/O: \*\***

restore status request complete

Definition at line 941 of file Federate.hh.

Referenced by `initiate_restore_announce()`, `process_requested_federation_restore_status()`, and `wait_for_restore_status_to_complete()`.

**7.21.5.69 RTI\_ambassador**

```
TrickRTIAmbPtr TrickHLA::Federate::RTI_ambassador [private]
```

**Data I/O: \*\***

RTI ambassador

Definition at line 1005 of file Federate.hh.

Referenced by `achieve_and_wait_for_synchronization()`, `achieve_synchronization_point()`, `announce_sync_point()`, `create_federation()`, `create_RTI_ambassador_and_connect()`, `destroy()`, `destroy_orphaned_federation()`, `enable_async_delivery()`, `get_RTI_ambassador()`, `inform_RTI_of_restore_completion()`, `initialize_MOM_handles()`, `initiate_restore_announce()`, `is_execution_member()`, `join_federation()`, `perform_time_advance_request()`, `post_checkpoint()`, `post_restore()`, `publish_interaction_class()`, `rebuild_federate_handles()`, `register_generic_sync_point()`, `request_attribute_update()`, `request_federation_restore_status()`, `request_federation_save()`, `request_federation_save_status()`, `requested_federation_restore_status()`, `resign()`, `resign_so_we_can_rejoin()`, `restart_checkpoint()`, `send_interaction()`, `set_MOM_HLAfederate_instance_attributes()`, `setup_checkpoint()`, `setup_time_constrained()`, `setup_time_regulation()`, `shutdown_time_constrained()`, `shutdown_time_regulating()`, `subscribe_attributes()`, `time_advance_request_to_GALT()`, `time_advance_request_to_GALT_LCTS_multiple()`, `unpublish_interaction_class()`, `unsubscribe_all_HLAfederate`.

class\_attributes\_from\_MOM(), unsubscribe\_all\_HLAfederation\_class\_attributes\_from\_MOM(), and unsubscribe\_attributes().

#### 7.21.5.70 running\_feds

`KnownFederate* TrickHLA::Federate::running_feds [private]`

**Units:** –

Checkpoint-able Array of running Federation Federates

Definition at line 955 of file Federate.hh.

Referenced by add\_a\_single\_entry\_into\_running\_feds(), clear\_running\_feds(), copy\_running\_feds\_into\_known\_feds(), load\_and\_print\_running\_federate\_names(), remove\_MOM\_HLAfederate\_instance\_id(), set\_MOM\_HLAfederate\_instance\_attributes(), and update\_running\_feds().

#### 7.21.5.71 running\_feds\_count

`int TrickHLA::Federate::running_feds_count [private]`

**Units:** –

Number of running Federates (default: 0)

Definition at line 954 of file Federate.hh.

Referenced by add\_a\_single\_entry\_into\_running\_feds(), clear\_running\_feds(), copy\_running\_feds\_into\_known\_feds(), get\_running\_feds\_count(), initiate\_restore\_announce(), load\_and\_print\_running\_federate\_names(), remove\_MOM\_HLAfederate\_instance\_id(), restore\_federate\_handles\_from\_MOM(), set\_MOM\_HLAfederate\_instance\_attributes(), set\_MOM\_HLAfederation\_instance\_attributes(), update\_running\_feds(), and wait\_for\_federation\_restore\_to\_complete().

#### 7.21.5.72 running\_feds\_count\_at\_time\_of\_restore

`int TrickHLA::Federate::running_feds_count_at_time_of_restore [private]`

**Data I/O:** \*\*

Number of running Federates at the time of the restore (default: 0)

Definition at line 956 of file Federate.hh.

Referenced by initiate\_restore\_announce(), and wait\_for\_federation\_restore\_to\_complete().

#### 7.21.5.73 save\_completed

`bool TrickHLA::Federate::save_completed [private]`

**Data I/O:** \*\*

Save completed.

Definition at line 931 of file Federate.hh.

Referenced by IMSim::ExecutionControl::is\_save\_initiated(), perform\_time\_advance\_request(), set\_save\_completed(), and time\_advance\_request().

#### 7.21.5.74 save\_label\_generated

`bool TrickHLA::Federate::save_label_generated [private]`

**Data I/O:** \*\*

Save filename has been generated.

Definition at line 929 of file Federate.hh.

Referenced by federation\_restored(), federation\_saved(), and initiate\_save\_announce().

### 7.21.5.75 `save_name`

```
std::wstring TrickHLA::Federate:::save_name [private]
```

**Data I/O:** \*\*

Name for a save file

Definition at line 911 of file Federate.hh.

Referenced by `federation_saved()`, `perform_checkpoint()`, `request_federation_save()`, `set_checkpoint_file_name()`, `set_save_name()`, and `setup_checkpoint()`.

### 7.21.5.76 `save_request_complete`

```
bool TrickHLA::Federate:::save_request_complete [private]
```

**Data I/O:** \*\*

save status request complete

Definition at line 930 of file Federate.hh.

Referenced by `federation_saved()`, `process_requested_federation_save_status()`, and `wait_for_save_status_to_complete()`.

### 7.21.5.77 `shutdown_called`

```
bool TrickHLA::Federate:::shutdown_called [private]
```

**Units:** -

Flag to indicate shutdown has been called.

Definition at line 909 of file Federate.hh.

Referenced by `shutdown()`.

### 7.21.5.78 `stale_data_counter`

```
int TrickHLA::Federate:::stale_data_counter [private]
```

**Units:** -

For **DIS** only: Number of cycles since the last time we received data via HLA.

Definition at line 933 of file Federate.hh.

Referenced by `get_stale_data_counter()`, and `wait_for_time_advance_grant()`.

### 7.21.5.79 `start_to_restore`

```
bool TrickHLA::Federate:::start_to_restore [private]
```

**Data I/O:** \*\*

Restore flag

Definition at line 972 of file Federate.hh.

Referenced by `complete_restore()`, `federation_restored()`, `is_start_to_restore()`, `perform_restore()`, `post_restore()`, `set_restore_completed()`, `set_restore_failed()`, `set_start_to_restore()`, `setup_restore()`, and `wait_until_federation_is_ready_to_restore()`.

### 7.21.5.80 `start_to_save`

```
bool TrickHLA::Federate:::start_to_save [private]
```

**Data I/O:** \*\*

Save flag

Definition at line 971 of file Federate.hh.

Referenced by `perform_checkpoint()`, `post_checkpoint()`, `set_save_completed()`, `set_start_to_save()`, and `setup_checkpoint()`.

#### 7.21.5.81 str\_restore\_label

`std::string TrickHLA::Federate::str_restore_label [private]`

**Data I/O:** \*\*

Restore file label in C++ string format

Definition at line 936 of file `Federate.hh`.

Referenced by `federation_restored()`, `initiate_restore_announce()`, `perform_restore()`, and `setup_restore()`.

#### 7.21.5.82 str\_save\_label

`std::string TrickHLA::Federate::str_save_label [private]`

**Data I/O:** \*\*

Save file label in C++ string format

Definition at line 926 of file `Federate.hh`.

Referenced by `federation_saved()`, `initiate_save_announce()`, `perform_checkpoint()`, and `setup_checkpoint()`.

#### 7.21.5.83 time\_adv\_grant

`bool TrickHLA::Federate::time_adv_grant [private]`

**Units:** –

Time advance grant flag.

Definition at line 967 of file `Federate.hh`.

Referenced by `is_time_advance_granted()`, `perform_time_advance_request()`, `set_time_advance_grant()`, `setup_time_constrained()`, `setup_time_regulation()`, and `wait_for_time_advance_grant()`.

#### 7.21.5.84 time\_constrained

`bool TrickHLA::Federate::time_constrained`

**Units:** –

HLA Time Constrained flag (default: true).

Definition at line 118 of file `Federate.hh`.

Referenced by `DSES::ExecutionControl::initialize()`, `DIS::ExecutionControl::initialize()`, `IMSim::ExecutionControl::initialize()`, `SpaceFOM::ExecutionControl::initialize()`, `restart_initialization()`, `setup_time_constrained()`, and `setup_time_management()`.

#### 7.21.5.85 time\_constrained\_state

`bool TrickHLA::Federate::time_constrained_state [private]`

**Units:** –

Internal flag, federates HLA Time Constrained state (default: false).

Definition at line 979 of file `Federate.hh`.

Referenced by `set_time_constrained_state()`, `setup_time_constrained()`, `setup_time_management()`, `shutdown_time_constrained()`, and `shutdown_time_regulating()`.

**7.21.5.86 time\_management**

```
bool TrickHLA::Federate::time_management
```

**Units:** –

Enable HLA Time Management flag (default: true).

Definition at line 119 of file Federate.hh.

Referenced by IMSim::ExecutionControl::check\_scenario\_freeze\_time(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), is\_time\_management\_enabled(), perform\_time\_advance\_request(), restart\_initialization(), setup\_time\_constrained(), setup\_time\_management(), setup\_time\_regulation(), time\_advance\_request(), time\_advance\_request\_to\_GALT(), time\_advance\_request\_to\_GALT\_LCTS\_multiple(), and wait\_for\_time\_advance\_grant().

**7.21.5.87 time\_regulating**

```
bool TrickHLA::Federate::time_regulating
```

**Units:** –

HLA Time Regulation flag (default: true).

Definition at line 117 of file Federate.hh.

Referenced by DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), restart\_initialization(), setup\_time\_management(), and setup\_time\_regulation().

**7.21.5.88 time\_regulating\_state**

```
bool TrickHLA::Federate::time_regulating_state [private]
```

**Units:** –

Internal flag, federates HLA Time Regulation state (default: false).

Definition at line 978 of file Federate.hh.

Referenced by in\_time\_regulating\_state(), set\_time\_regulation\_state(), setup\_time\_management(), setup\_time\_regulation(), and shutdown\_time\_regulating().

**7.21.5.89 type**

```
char* TrickHLA::Federate::type
```

**Units:** –

The federate type.

Definition at line 99 of file Federate.hh.

Referenced by get\_federate\_type(), initialize(), and ~Federate().

**7.21.5.90 unfreeze\_after\_save**

```
bool TrickHLA::Federate::unfreeze_after_save
```

**Units:** –

Flag to indicate that we should go to run immediately after a save.

Definition at line 135 of file Federate.hh.

Referenced by federation\_saved().

**7.21.5.91 ws\_restore\_label**

```
std::wstring TrickHLA::Federate::ws_restore_label [private]
```

**Data I/O: \*\***

Restore file label in wide string format

Definition at line 937 of file Federate.hh.

Referenced by federation\_restored(), and initiate\_restore\_announce().

**7.21.5.92 ws\_save\_label**

```
std::wstring TrickHLA::Federate::ws_save_label [private]
```

**Data I/O: \*\***

Save file label in wide string format

Definition at line 927 of file Federate.hh.

Referenced by federation\_saved(), and initiate\_save\_announce().

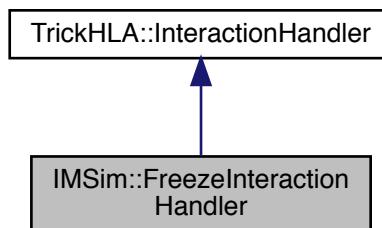
The documentation for this class was generated from the following files:

- [Federate.hh](#)
- [Federate.cpp](#)

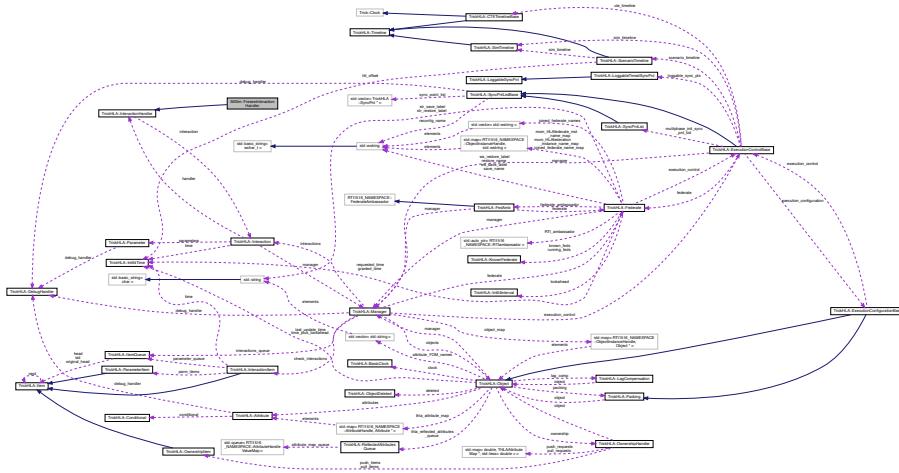
## 7.22 IMSim::FreezeInteractionHandler Class Reference

```
#include <FreezeInteractionHandler.hh>
```

Inheritance diagram for IMSim::FreezeInteractionHandler:



Collaboration diagram for IMSim::FreezeInteractionHandler:



## Public Member Functions

- `FreezeInteractionHandler ()`  
*Default constructor for the `TrickHLA FreezeInteractionHandler` class.*
- `virtual ~FreezeInteractionHandler ()`  
*Destructor for the `TrickHLA FreezeInteractionHandler` class.*
- `void send_scenario_freeze_interaction (double &freeze_time, bool late_joining_federate=false)`  
*Send the freeze interaction scenario time immediately using Timestamp Order.*
- `void receive_interaction (RTI1516_USERDATA const &theUserSuppliedTag)`  
*Called when the interaction is received from the RTI.*
- `double * get_address_of_interaction_time ()`  
*Get the address of the interaction time.*

## Private Member Functions

- `FreezeInteractionHandler (const FreezeInteractionHandler &rhs)`
- `FreezeInteractionHandler & operator= (const FreezeInteractionHandler &rhs)`

## Static Private Member Functions

- `static bool check_values (const double &v1, const double &v2)`  
*Value comparison check.*

## Private Attributes

- `double time`

**Units:** s

*Scenario Time on which to freeze simulation execution on a major frame boundary.*

## Friends

- class `InputProcessor`
- `void init_attrTrickHLA_FreezeInteractionHandler ()`

## Additional Inherited Members

### 7.22.1 Detailed Description

Definition at line 52 of file FreezeInteractionHandler.hh.

### 7.22.2 Constructor & Destructor Documentation

#### 7.22.2.1 `FreezeInteractionHandler()` [1/2]

`FreezeInteractionHandler::FreezeInteractionHandler ( )`  
Default constructor for the [TrickHLA FreezeInteractionHandler](#) class.

**Trick Job Class:** *initialization*

Definition at line 56 of file FreezeInteractionHandler.cpp.

#### 7.22.2.2 `~FreezeInteractionHandler()`

`FreezeInteractionHandler::~FreezeInteractionHandler ( ) [virtual]`  
Destructor for the [TrickHLA FreezeInteractionHandler](#) class.

**Trick Job Class:** *shutdown*

Definition at line 64 of file FreezeInteractionHandler.cpp.

#### 7.22.2.3 `FreezeInteractionHandler()` [2/2]

```
IMSim::FreezeInteractionHandler::FreezeInteractionHandler (
    const FreezeInteractionHandler & rhs ) [private]
```

### 7.22.3 Member Function Documentation

#### 7.22.3.1 `check_values()`

```
static bool IMSim::FreezeInteractionHandler::check_values (
    const double & v1,
    const double & v2 ) [inline], [static], [private]
```

Value comparison check.

#### Returns

True if values match to within a tolerance; False otherwise.

#### Parameters

<code>v1</code>	First value in comparison.
<code>v2</code>	Second value in comparison.

Definition at line 102 of file FreezeInteractionHandler.hh.

### 7.22.3.2 `get_address_of_interaction_time()`

```
double* IMSim::FreezeInteractionHandler::get_address_of_interaction_time ( ) [inline]
Get the address of the interaction time.
```

#### Returns

Pointer to the interaction time.

Definition at line 86 of file FreezeInteractionHandler.hh.

References time.

Referenced by IMSim::ExecutionControl::setup\_interaction\_ref\_attributes().

### 7.22.3.3 `operator=()`

```
FreezeInteractionHandler& IMSim::FreezeInteractionHandler::operator= (
    const FreezeInteractionHandler & rhs ) [private]
```

### 7.22.3.4 `receive_interaction()`

```
void FreezeInteractionHandler::receive_interaction (
    RTI1516_USERDATA const & theUserSuppliedTag ) [virtual]
```

Called when the interaction is received from the RTI.

#### Parameters

<code>theUserSuppliedTag</code>	User tag.
---------------------------------	-----------

Reimplemented from [TrickHLA::InteractionHandler](#).

Definition at line 260 of file FreezeInteractionHandler.cpp.

References TrickHLA::Interaction::get\_fed\_lookingahead(), TrickHLA::Interaction::get\_federate(), TrickHLA::Interaction::get\_granted\_fed\_time(), TrickHLA::Int64Interval::getDoubleTime(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::InteractionHandler::interaction, THLA\_ENDL, and time.

### 7.22.3.5 `send_scenario_freeze_interaction()`

```
void FreezeInteractionHandler::send_scenario_freeze_interaction (
    double & freeze_time,
    bool late_joining_federate = false )
```

Send the freeze interaction scenario time immediately using Timestamp Order.

#### Parameters

<code>freeze_time</code>	Reference to the freeze scenario time, typically Terrestrial Time (TT).
<code>late_joining_federate</code>	True if late joining federate; False otherwise.

The interaction time is computed to be the current granted time plus lookahead. If the supplied time does not fall on the HLA and lookahead time boundaries, the time is rounded up to the next highest integer multiple of the lookahead time. If the time is less than the granted time plus one lookahead, this means that the freeze time is invalid so it is updated to be 'granted time plus one lookahead'. If the federate is a late-joining federate, the freeze time is slipped one frame because it would occur in a previous frame when the other non-late-joiner federates receive the freeze interaction time. We will calculate two important times related to the freeze scenario time:

1. The HLA logical timestamp the freeze interaction is sent for (TSO), which is the interaction\_hla\_time variable.
2. The Freeze scenario time that is an integer multiple of the lookahead and is  $\geq$  the equivalent freeze interaction scenario time from step 1, which is the freeze\_scenario\_time variable.

We need to find the equivalent simulation-time and HLA-time for a given freeze scenario-time so that we can do the correct time comparisons. Also, if we are a late joining federate the sim-time and HLA-time will not be aligned as shown in this example.

```

*      HLA-time |-----|-----|
*          101.0          102.0          103.0
*
*      Scenario-time |-----|-----|
*          March 2, 2032 @ 19:20:07  March 2, 2032 @ 19:20:08  March 2, 2032 @ 19:20:09
*
*      Sim-time |-----|-----|
*          0.0          1.0          2.0
*

```

Scenario-time and Sim-time change at the same rate but they have different starting epochs.

freeze-sim-time = current-sim-time + (freeze-scenario-time - current-scenario-time)

freeze-hla-time = granted-hla-time + (freeze-scenario-time - current-scenario-time)

We must wait for a valid Time Advance Grant (TAG) so the HLA granted time will be valid and with proper alignment with the scenario and simulation timelines. NOTE: We can only do the blocking wait for the time advance grant here because this is an end\_of\_frame job and if we don't have a granted time then we are at the end of the frame that made the TAR call. The wait for Time Advance Grant will be at the top of the next frame.

Definition at line 79 of file FreezeInteractionHandler.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_5\_TRACE, TrickHLA::DEBUG\_SO←URCE\_INTERACTION, TrickHLA::Interaction::get\_fed\_looking(), TrickHLA::Interaction::get\_federate(), TrickHLA::Interaction::get\_granted\_fed\_time(), TrickHLA::Int64Interval::getDoubleTime(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Int64Time::getTimeInMicros(), TrickHLA::InteractionHandler::interaction, TrickHLA::Federate::is←\_time\_advance\_granted(), TrickHLA::InteractionHandler::should\_print(), THLA\_ENDL, THLA\_NEWLINE, time, and TrickHLA::Federate::wait\_for\_time\_advance\_grant().

Referenced by IMSim::ExecutionControl::trigger\_freeze\_interaction().

## 7.22.4 Friends And Related Function Documentation

### 7.22.4.1 init\_attrTrickHLA\_FreezeInteractionHandler

```
void init_attrTrickHLA_FreezeInteractionHandler ( ) [friend]
```

### 7.22.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 59 of file FreezeInteractionHandler.hh.

## 7.22.5 Field Documentation

### 7.22.5.1 time

```
double IMSim::FreezeInteractionHandler::time [private]
```

**Units:** s

Scenario Time on which to freeze simulation execution on a major frame boundary.

Definition at line 92 of file FreezeInteractionHandler.hh.

Referenced by `get_address_of_interaction_time()`, `receive_interaction()`, and `send_scenario_freeze_interaction()`. The documentation for this class was generated from the following files:

- [FreezeInteractionHandler.hh](#)
- [FreezeInteractionHandler.cpp](#)

## 7.23 TrickHLA::Int64Interval Class Reference

```
#include <Int64Interval.hh>
```

### Public Member Functions

- **Int64Interval ()**  
*Default constructor for the `TrickHLA Int64Interval` class.*
- **Int64Interval (int64\_t value)**  
*Initialization constructor for the `TrickHLA Int64Interval` class.*
- **Int64Interval (double value)**  
*Initialization constructor for the `TrickHLA Int64Interval` class.*
- **Int64Interval (RTI1516\_NAMESPACE::LogicalTimeInterval const &value)**  
*Initialization constructor for the `TrickHLA Int64Interval` class.*
- **Int64Interval (RTI1516\_NAMESPACE::HLAinteger64Interval const &value)**  
*Initialization constructor for the `TrickHLA Int64Interval` class.*
- **Int64Interval (Int64Interval const &value)**  
*Copy constructor for the `TrickHLA Int64Interval` class.*
- **virtual ~Int64Interval ()**  
*Destructor for the `TrickHLA Int64Interval` class.*
- **virtual Int64Interval & operator= (double lhs)**  
*Assignment operator from double time value.*
- **virtual Int64Interval & operator= (int64\_t lhs)**  
*Assignment operator from 64bit integer time value.*
- **virtual Int64Interval & operator= (Int64Interval const &lhs)**  
*Assignment operator from `TrickHLA::Int64Interval` time interval value.*
- **bool operator> (int64\_t lhs)**  
*Interval time greater than comparison operator.*
- **bool operator> (double lhs)**  
*Interval time greater than comparison operator.*
- **RTI1516\_NAMESPACE::HLAinteger64Interval get () const**  
*Get the HLA integer time.*
- **long getSeconds () const**  
*Return the seconds contained in the current timestamp.*
- **int getMicros () const**  
*Return the microseconds seconds contained in the current timestamp.*
- **int64\_t getTimeInMicros () const**  
*Return the time, in microseconds, contained in the current timestamp as a 64-bit integer value.*
- **double getDoubleTime () const**  
*Return the current timestamp as a double precision floating point value.*
- **std::wstring toString () const**  
*Returns a summary of the time.*

- void [setTo](#) (const int64\_t value)  
*Set the time interval to the given value.*
- void [setTo](#) (const double value)  
*Set the time interval to the given value.*
- void [setTo](#) (RTI1516\_NAMESPACE::LogicalTimeInterval const &value)  
*Set the time interval to the given value.*

## Static Public Member Functions

- static int64\_t [toMicroseconds](#) (const double value)  
*Converts the given floating point time to an integer representing microseconds.*

## Private Attributes

- RTI1516\_NAMESPACE::HLAinteger64Interval [\\_HLAinterval](#)  
**Data I/O:** \*\*  
*The HLA standard's class representation of integer64 interval.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_\\_Int64Interval](#) ()

### 7.23.1 Detailed Description

Definition at line 52 of file Int64Interval.hh.

### 7.23.2 Constructor & Destructor Documentation

#### 7.23.2.1 [Int64Interval\(\)](#) [1/6]

TrickHLA::Int64Interval::Int64Interval ( ) [inline]

Default constructor for the [TrickHLA Int64Interval](#) class.

Definition at line 69 of file Int64Interval.hh.

#### 7.23.2.2 [Int64Interval\(\)](#) [2/6]

```
Int64Interval::Int64Interval (
    int64_t value) [explicit]
```

Initialization constructor for the [TrickHLA Int64Interval](#) class.

#### Parameters

<code>value</code>	64bit integer value initialization.
--------------------	-------------------------------------

#### Trick Job Class: *initialization*

Definition at line 53 of file Int64Interval.cpp.

### 7.23.2.3 Int64Interval() [3/6]

```
Int64Interval::Int64Interval (
    double value ) [explicit]
```

Initialization constructor for the [TrickHLA Int64Interval](#) class.

#### Parameters

<code>value</code>	Floating point double value initialization.
--------------------	---

**Trick Job Class:** *initialization*

Definition at line 62 of file Int64Interval.cpp.

### 7.23.2.4 Int64Interval() [4/6]

```
Int64Interval::Int64Interval (
    RTI1516_NAMESPACE::LogicalTimeInterval const & value ) [explicit]
```

Initialization constructor for the [TrickHLA Int64Interval](#) class.

#### Parameters

<code>value</code>	HLA Logical Time Interval value initialization.
--------------------	---

**Trick Job Class:** *initialization*

Definition at line 71 of file Int64Interval.cpp.

### 7.23.2.5 Int64Interval() [5/6]

```
Int64Interval::Int64Interval (
    RTI1516_NAMESPACE::HLAinteger64Interval const & value ) [explicit]
```

Initialization constructor for the [TrickHLA Int64Interval](#) class.

#### Parameters

<code>value</code>	HLA 64bit Integer Time Interval value initialization.
--------------------	---

**Trick Job Class:** *initialization*

Definition at line 80 of file Int64Interval.cpp.

### 7.23.2.6 Int64Interval() [6/6]

```
Int64Interval::Int64Interval (
    Int64Interval const & value )
```

Copy constructor for the [TrickHLA Int64Interval](#) class.

#### Parameters

<code>value</code>	TrickHLA Long Integer Time Interval value initialization.
--------------------	---

**Trick Job Class:** *initialization*

Definition at line 89 of file Int64Interval.cpp.

### 7.23.2.7 ~Int64Interval()

Int64Interval::~Int64Interval ( ) [virtual]

Destructor for the [TrickHLA Int64Interval](#) class.

**Trick Job Class:** *shutdown*

Definition at line 98 of file Int64Interval.cpp.

## 7.23.3 Member Function Documentation

### 7.23.3.1 get()

RTI1516\_NAMESPACE::HLAinteger64Interval [TrickHLA::Int64Interval::get](#) ( ) const [inline]

Get the HLA integer time.

**Returns**

A copy of the encapsulated HLAinteger64Interval class.

Definition at line 142 of file Int64Interval.hh.

References [\\_HLAinterval](#).

Referenced by [TrickHLA::Federate::setup\\_time\\_regulation\(\)](#).

### 7.23.3.2 getDoubleTime()

double Int64Interval::getDoubleTime ( ) const

Return the current timestamp as a double precision floating point value.

**Returns**

Time in seconds as a floating point double.

Definition at line 117 of file Int64Interval.cpp.

References [getMicros\(\)](#), [getSeconds\(\)](#), and [TrickHLA::MICROS\\_MULTIPLIER](#).

Referenced by [IMSim::FreezeInteractionHandler::receive\\_interaction\(\)](#), [TrickHLA::Object::send\\_cyclic\\_data\(\)](#), [TrickHLA::Object::send\\_init\\_data\(\)](#), [TrickHLAModel::SineLagCompensation::send\\_lag\\_compensation\(\)](#), [TrickHLA::Object::send\\_requested\\_data\(\)](#), [IMSim::FreezeInteractionHandler::send\\_scenario\\_freeze\\_interaction\(\)](#), [TrickHLAModel::SineInteractionHandler::send\\_sine\\_interaction\(\)](#), [TrickHLA::Federate::setup\\_time\\_regulation\(\)](#), and [toString\(\)](#).

### 7.23.3.3 getMicros()

int Int64Interval::getMicros ( ) const

Return the microseconds seconds contained in the current timestamp.

**Returns**

The current timestamp in integer microseconds.

Definition at line 107 of file Int64Interval.cpp.

References [\\_HLAinterval](#), and [TrickHLA::MICROS\\_MULTIPLIER](#).

Referenced by [getDoubleTime\(\)](#).

**7.23.3.4 getSeconds()**

```
long Int64Interval::getSeconds ( ) const
Return the seconds contained in the current timestamp.
```

**Returns**

The current timestamp in seconds.

Definition at line 102 of file Int64Interval.cpp.

References \_HLAinterval, and TrickHLA::MICROS\_MULTIPLIER.

Referenced by getDoubleTime().

**7.23.3.5 getTimeInMicros()**

```
int64_t Int64Interval::getTimeInMicros ( ) const
Return the time, in microseconds, contained in the current timestamp as a 64-bit integer value.
```

**Returns**

Time in integer microseconds.

Definition at line 112 of file Int64Interval.cpp.

References \_HLAinterval.

Referenced by TrickHLA::Int64Time::operator%(), TrickHLA::Int64Time::operator\*(), TrickHLA::Int64Time::operator+(), TrickHLA::Int64Time::operator+=(), TrickHLA::Int64Time::operator-(), TrickHLA::Int64Time::operator/(), operator=(), operator>(), DIS::ExecutionConfiguration::pack(), DSES::ExecutionConfiguration::pack(), IMSim::ExecutionConfiguration::pack(), SpaceFOM::ExecutionConfiguration::pack(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), TrickHLA::Object::send\_cyclic\_data(), TrickHLA::Object::send\_init\_data(), TrickHLA::Object::send\_requested\_data(), TrickHLA::Federate::time\_advance\_request\_to\_GALT(), TrickHLA::Federate::time\_advance\_request\_to\_GALT\_LCTS\_multiple(), DSES::ExecutionConfiguration::unpack(), DIS::ExecutionConfiguration::unpack(), IMSim::ExecutionConfiguration::unpack(), and SpaceFOM::ExecutionConfiguration::unpack().

**7.23.3.6 operator=() [1/3]**

```
virtual Int64Interval& TrickHLA::Int64Interval::operator=
    double lhs ) [inline], [virtual]
```

Assignment operator from double time value.

**Returns**

A corresponding [TrickHLA::Int64Interval](#) time value.

**Parameters**

<i>lhs</i>	Left hand side operand as floating point time interval in seconds.
------------	--

Definition at line 100 of file Int64Interval.hh.

References \_HLAinterval, and toMicroseconds().

**7.23.3.7 operator=() [2/3]**

```
virtual Int64Interval& TrickHLA::Int64Interval::operator=
    int64_t lhs ) [inline], [virtual]
```

Assignment operator from 64bit integer time value.

**Returns**

A corresponding [TrickHLA::Int64Interval](#) time value.

**Parameters**

<i>lhs</i>	Left hand side operand as 64bit integer time interval in microseconds.
------------	--

Definition at line 109 of file `Int64Interval.hh`.

References `_HLAinterval`.

**7.23.3.8 operator=() [3/3]**

```
virtual Int64Interval& TrickHLA::Int64Interval::operator= (
    Int64Interval const & lhs ) [inline], [virtual]
```

Assignment operator from `TrickHLA::Int64Interval` time interval value.

**Returns**

A corresponding [TrickHLA::Int64Interval](#) time value.

**Parameters**

<i>lhs</i>	Left hand side operand as <a href="#">TrickHLA::Int64Interval</a> time interval.
------------	--

Definition at line 118 of file `Int64Interval.hh`.

References `_HLAinterval`, and `getTimeInMicros()`.

**7.23.3.9 operator>() [1/2]**

```
bool TrickHLA::Int64Interval::operator> (
    double lhs ) [inline]
```

Interval time greater than comparison operator.

**Returns**

True if right operand is greater than the left operand; False otherwise.

**Parameters**

<i>lhs</i>	Left hand side operand as floating point time interval in seconds.
------------	--

Definition at line 132 of file `Int64Interval.hh`.

References `getTimeInMicros()`, and `toMicroseconds()`.

**7.23.3.10 operator>() [2/2]**

```
bool TrickHLA::Int64Interval::operator> (
    int64_t lhs ) [inline]
```

Interval time greater than comparison operator.

#### Returns

True if right operand is greater than the left operand; False otherwise.

#### Parameters

<i>lhs</i>	Left hand side operand as 64bit integer time interval in microseconds.
------------	--

Definition at line 127 of file Int64Interval.hh.

References `getTimeInMicros()`.

### 7.23.3.11 `setTo()` [1/3]

```
void Int64Interval::setTo (
    const double value )
```

Set the time interval to the given value.

#### Parameters

<i>value</i>	The desired time interval in seconds.
--------------	---------------------------------------

Definition at line 140 of file Int64Interval.cpp.

References `_HLAinterval`, and `toMicroseconds()`.

### 7.23.3.12 `setTo()` [2/3]

```
void Int64Interval::setTo (
    const int64_t value )
```

Set the time interval to the given value.

#### Parameters

<i>value</i>	The desired time interval in integer microseconds.
--------------	--

Definition at line 134 of file Int64Interval.cpp.

References `_HLAinterval`.

Referenced by `TrickHLA::Federate::set_lookinghead()`.

### 7.23.3.13 `setTo()` [3/3]

```
void Int64Interval::setTo (
    RTI1516_NAMESPACE::LogicalTimeInterval const & value )
```

Set the time interval to the given value.

#### Parameters

<i>value</i>	The desired time interval as an HLA LogicalTimeInterval.
--------------	--

Definition at line 146 of file Int64Interval.cpp.

References `_HLAinterval`.

#### 7.23.3.14 `toMicroseconds()`

```
int64_t Int64Interval::toMicroseconds (
    const double value ) [static]
```

Converts the given floating point time to an integer representing microseconds.

##### Returns

Time value in microseconds.

##### Parameters

<code>value</code>	Time value as a floating point double in seconds.
--------------------	---

Definition at line 153 of file Int64Interval.cpp.

References `TrickHLA::MAX_LOGICAL_TIME_SECONDS`, `TrickHLA::MAX_VALUE_IN_MICROS`, and `TrickHLA::MICROS_MULTIPLIER`.

Referenced by `TrickHLA::Int64Time::operator+=()`, `operator=()`, `TrickHLA::Int64Time::operator=()`, `operator>()`, `TrickHLA::Int64Time::operator>()`, `SpaceFOM::MTRInteractionHandler::receive_interaction()`, `TrickHLA::Federate::register_generic_sync_point()`, `TrickHLA::Object::send_cyclic_data()`, `SpaceFOM::MTRInteractionHandler::send_interaction()`, `TrickHLA::Object::send_requested_data()`, `setTo()`, and `TrickHLA::Int64Time::setTo()`.

#### 7.23.3.15 `toString()`

```
wstring Int64Interval::toString ( ) const
```

Returns a summary of the time.

##### Returns

Summary of time as a string.

Definition at line 124 of file Int64Interval.cpp.

References `getDoubleTime()`.

### 7.23.4 Friends And Related Function Documentation

#### 7.23.4.1 `init_attrTrickHLA__Int64Interval`

```
void init_attrTrickHLA__Int64Interval ( ) [friend]
```

#### 7.23.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 59 of file Int64Interval.hh.

### 7.23.5 Field Documentation

### 7.23.5.1 \_HLAinterval

```
RTI1516_NAMESPACE::HLAinteger64Interval TrickHLA::Int64Interval::_HLAinterval [private]
```

#### Data I/O: \*\*

The HLA standard's class representation of integer64 interval.

Definition at line 197 of file Int64Interval.hh.

Referenced by `get()`, `getMicros()`, `getSeconds()`, `getTimeInMicros()`, `operator=()`, and `setTo()`.

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

- [Int64Interval.hh](#)
- [Int64Interval.cpp](#)

## 7.24 TrickHLA::Int64Time Class Reference

```
#include <Int64Time.hh>
```

### Public Member Functions

- [Int64Time \(\)](#)  
*Default constructor for the `TrickHLA Int64Time` class.*
- [Int64Time \(int64\\_t value\)](#)  
*Initialization constructor for the `TrickHLA Int64Time` class.*
- [Int64Time \(double value\)](#)  
*Initialization constructor for the `TrickHLA Int64Time` class.*
- [Int64Time \(RTI1516\\_NAMESPACE::LogicalTime const &value\)](#)  
*Initialization constructor for the `TrickHLA Int64Time` class.*
- [Int64Time \(RTI1516\\_NAMESPACE::HLAinteger64Time const &value\)](#)  
*Initialization constructor for the `TrickHLA Int64Time` class.*
- [Int64Time \(Int64Time const &value\)](#)  
*Copy constructor for the `TrickHLA Int64Time` class.*
- virtual [~Int64Time \(\)](#)  
*Destructor for the `TrickHLA Int64Time` class.*
- virtual [Int64Time & operator= \(double lhs\)](#)  
*Assignment operator.*
- virtual [Int64Time & operator= \(int64\\_t lhs\)](#)  
*Assignment operator.*
- virtual [Int64Time & operator= \(Int64Time const &lhs\)](#)  
*Assignment operator.*
- [Int64Time operator+= \(double lhs\)](#)  
*Addition then assignment operator.*
- [Int64Time operator+= \(const Int64Interval &lhs\)](#)  
*Addition then assignment operator.*
- [Int64Time operator+ \(const Int64Time &lhs\) const](#)  
*Addition operator.*
- [Int64Time operator+ \(const Int64Interval &lhs\) const](#)  
*Addition operator.*
- [Int64Time operator- \(const Int64Time &lhs\) const](#)  
*Subtraction operator.*
- [Int64Time operator- \(const Int64Interval &lhs\) const](#)  
*Subtraction operator.*

- **Int64Time operator\*** (const Int64Interval &lhs) const
  - Subtraction operator.*
- **Int64Time operator/** (const Int64Interval &lhs) const
  - Multiplication operator.*
  - Division operator.*
- **Int64Time operator%** (const Int64Interval &lhs) const
  - Modulo operator.*
- bool **operator<** (const Int64Time &lhs) const
  - Less than comparison operator.*
- bool **operator>** (int64\_t lhs)
  - Greater than comparison operator.*
- bool **operator>** (double lhs)
  - Greater than comparison operator.*
- bool **operator>** (const Int64Time &lhs) const
  - Greater than comparison operator.*
- bool **operator<=** (const Int64Time &lhs) const
  - Less than or equal to comparison operator.*
- bool **operator>=** (const Int64Time &lhs) const
  - Greater than or equal to comparison operator.*
- bool **operator==** (int64\_t lhs) const
  - Equals comparison operator.*
- bool **operator==** (const Int64Time &lhs) const
  - Equals comparison operator.*
- bool **operator!=** (int64\_t lhs) const
  - Not equal to comparison operator.*
- bool **operator!=** (const Int64Time &lhs) const
  - Not equal to comparison operator.*
- RTI1516\_NAMESPACE::HLAinteger64Time **get** () const
  - Get the HLA integer time.*
- void **decode** (RTI1516\_USERDATA const &user\_supplied\_tag)
  - Saves the incoming HLA encoded LogicalTime into the encapsulated class.*
- long **getSeconds** () const
  - Return the seconds contained in the current timestamp.*
- int **getMicros** () const
  - Return the microseconds seconds contained in the current timestamp.*
- int64\_t **getTimeInMicros** () const
  - Return the time, in microseconds, contained in the current timestamp as a 64-bit integer value.*
- double **getDoubleTime** () const
  - Return the current timestamp as a double precision floating point value.*
- std::wstring **toString** () const
  - Returns a wide string representing the time.*
- void **setTo** (const int64\_t value)
  - Set the time to the given value.*
- void **setTo** (const double value)
  - Set the time to the given value.*
- void **setTo** (RTI1516\_NAMESPACE::LogicalTime const &value)
  - Set the time to the given value.*
- void **setTo** (Int64Time const &value)
  - Set the time to the given value.*

## Private Attributes

- RTI1516\_NAMESPACE::HLAinteger64Time [\\_HLAtime](#)

**Data I/O:** \*\*

HLA standard's class representation of integer64 time.

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_\\_Int64Time \(\)](#)

### 7.24.1 Detailed Description

Definition at line 53 of file Int64Time.hh.

### 7.24.2 Constructor & Destructor Documentation

#### 7.24.2.1 Int64Time() [1/6]

`TrickHLA::Int64Time::Int64Time ( ) [inline]`

Default constructor for the [TrickHLA Int64Time](#) class.

Definition at line 70 of file Int64Time.hh.

#### 7.24.2.2 Int64Time() [2/6]

```
Int64Time::Int64Time (
    int64_t value ) [explicit]
```

Initialization constructor for the [TrickHLA Int64Time](#) class.

##### Parameters

<code>value</code>	64bit integer value initialization.
--------------------	-------------------------------------

**Trick Job Class:** *initialization*

Definition at line 49 of file Int64Time.cpp.

#### 7.24.2.3 Int64Time() [3/6]

```
Int64Time::Int64Time (
    double value ) [explicit]
```

Initialization constructor for the [TrickHLA Int64Time](#) class.

##### Parameters

<code>value</code>	Floating point double value initialization.
--------------------	---

**Trick Job Class:** *initialization*

Definition at line 58 of file Int64Time.cpp.

#### 7.24.2.4 `Int64Time()` [4/6]

```
Int64Time::Int64Time (
    RTI1516_NAMESPACE::LogicalTime const & value ) [explicit]
```

Initialization constructor for the [TrickHLA Int64Time](#) class.

##### Parameters

<code>value</code>	HLA Logical Time value initialization.
--------------------	--

**Trick Job Class:** *initialization*

Definition at line 67 of file Int64Time.cpp.

#### 7.24.2.5 `Int64Time()` [5/6]

```
Int64Time::Int64Time (
    RTI1516_NAMESPACE::HLAinteger64Time const & value ) [explicit]
```

Initialization constructor for the [TrickHLA Int64Time](#) class.

##### Parameters

<code>value</code>	HLA 64bit Integer Time value initialization.
--------------------	--

**Trick Job Class:** *initialization*

Definition at line 76 of file Int64Time.cpp.

#### 7.24.2.6 `Int64Time()` [6/6]

```
Int64Time::Int64Time (
    Int64Time const & value )
```

Copy constructor for the [TrickHLA Int64Time](#) class.

##### Parameters

<code>value</code>	<a href="#">TrickHLA</a> Long Integer Time value initialization.
--------------------	--

**Trick Job Class:** *initialization*

Definition at line 85 of file Int64Time.cpp.

#### 7.24.2.7 `~Int64Time()`

```
Int64Time::~Int64Time ( ) [virtual]
```

Destructor for the [TrickHLA Int64Time](#) class.

**Trick Job Class:** *shutdown*

Definition at line 94 of file Int64Time.cpp.

### 7.24.3 Member Function Documentation

### 7.24.3.1 decode()

```
void Int64Time::decode (
    RTI1516_USERDATA const & user_supplied_tag )
```

Saves the incoming HLA encoded LogicalTime into the encapsulated class.

#### Parameters

<i>user_supplied_tag</i>	Time encoded in user supplied tag.
--------------------------	------------------------------------

Definition at line 98 of file Int64Time.cpp.

References \_HLAtime.

Referenced by DIS::ExecutionControl::announce\_sync\_point(), and IMSim::ExecutionControl::announce\_sync\_point().

### 7.24.3.2 get()

```
RTI1516_NAMESPACE::HLAinteger64Time TrickHLA::Int64Time::get ( ) const [inline]
```

Get the HLA integer time.

#### Returns

A copy of the encapsulated HLAinteger64Time class.

Definition at line 292 of file Int64Time.hh.

References \_HLAtime.

Referenced by TrickHLA::Federate::perform\_time\_advance\_request(), TrickHLA::FedAmb::receiveInteraction(), TrickHLA::Object::remove(), TrickHLA::Interaction::send(), TrickHLA::Object::send\_cyclic\_data(), TrickHLA::Object::send\_init\_data(), and TrickHLA::Object::send\_requested\_data().

### 7.24.3.3 getDoubleTime()

```
double Int64Time::getDoubleTime ( ) const
```

Return the current timestamp as a double precision floating point value.

#### Returns

Time in seconds as a floating point double.

Definition at line 119 of file Int64Time.cpp.

References getMicros(), getSeconds(), and TrickHLA::MICROS\_MULTIPLIER.

Referenced by DIS::ExecutionControl::announce\_sync\_point(), IMSim::ExecutionControl::announce\_sync\_point(), IMSim::ExecutionControl::check\_scenario\_freeze\_time(), TrickHLA::Federate::get\_granted\_time(), TrickHLA::Federate::get\_requested\_time(), TrickHLA::Federate::perform\_time\_advance\_request(), DIS::PausePointList::print\_sync\_pnts(), IMSim::PausePointList::print\_sync\_pnts(), TrickHLA::TimedSyncPntList::print\_sync\_pnts(), TrickHLA::Interaction::process\_interaction(), TrickHLA::Object::pull\_ownership(), TrickHLA::Object::push\_ownership(), TrickHLA::Object::receive\_cyclic\_data(), IMSim::FreezeInteractionHandler::receive\_interaction(), SpaceFOM::ExecutionControl::receive\_interaction(), IMSim::ExecutionControl::receive\_interaction(), TrickHLA::Manager::receive\_interaction(), TrickHLA::FedAmb::reflectAttributeValues(), TrickHLA::Interaction::send(), TrickHLA::Object::send\_cyclic\_data(), TrickHLA::Object::send\_init\_data(), TrickHLA::Object::send\_requested\_data(), IMSim::FreezeInteractionHandler::send\_scenario\_freeze\_interaction(), TrickHLAModel::SineInteractionHandler::send\_sine\_interaction(), TrickHLA::Federate::time\_advance\_request\_to\_GALT(), TrickHLA::Federate::time\_advance\_request\_to\_GALT\_LCTS\_multiple(), TrickHLA::ScenarioTimeline::time\_from\_HLT(), TrickHLA::FedAmb::timeAdvanceGrant(), toString(), and TrickHLA::Federate::wait\_for\_time\_advance\_grant().

#### 7.24.3.4 `getMicros()`

```
int Int64Time::getMicros ( ) const
Return the microseconds seconds contained in the current timestamp.
```

##### Returns

The current timestamp in integer microseconds.

Definition at line 109 of file Int64Time.cpp.

References \_HLAtime, and TrickHLA::MICROS\_MULTIPLIER.

Referenced by `getDoubleTime()`.

#### 7.24.3.5 `getSeconds()`

```
long Int64Time::getSeconds ( ) const
Return the seconds contained in the current timestamp.
```

##### Returns

The current timestamp in seconds.

Definition at line 104 of file Int64Time.cpp.

References \_HLAtime, and TrickHLA::MICROS\_MULTIPLIER.

Referenced by `getDoubleTime()`.

#### 7.24.3.6 `getTimeInMicros()`

```
int64_t Int64Time::getTimeInMicros ( ) const
Return the time, in microseconds, contained in the current timestamp as a 64-bit integer value.
```

##### Returns

Time in integer microseconds.

Definition at line 114 of file Int64Time.cpp.

References \_HLAtime.

Referenced by TrickHLA::TimedSyncPnt::convert(), TrickHLA::Manager::dump\_interactions(), operator!=(), operator%(), operator\*(), operator+(), operator+=(), operator-(), operator/(), operator<(), operator<=(), operator=(), operator==(), operator>(), operator>=(), TrickHLA::Interaction::send(), TrickHLA::Object::send\_cyclic\_data(), TrickHLA::Object::send\_init\_data(), TrickHLA::Object::send\_requested\_data(), IMSim::FreezeInteractionHandler::send\_scenario\_freeze\_interaction(), and `setTo()`.

#### 7.24.3.7 `operator"!="() [1/2]`

```
bool TrickHLA::Int64Time::operator!= (
    const Int64Time & lhs ) const [inline]
```

Not equal to comparison operator.

##### Returns

True if right operand is not equal to the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand as a <a href="#">TrickHLA::Int64Time</a> .
------------------	---

Definition at line 282 of file Int64Time.hh.

References `getTimeInMicros()`.

#### 7.24.3.8 `operator"!="()` [2/2]

```
bool TrickHLA::Int64Time::operator!= (
    int64_t lhs ) const [inline]
```

Not equal to comparison operator.

##### Returns

True if right operand is not equal to the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand as a 64bit integer time in microseconds.
------------------	---

Definition at line 274 of file Int64Time.hh.

References `getTimeInMicros()`.

#### 7.24.3.9 `operator%()`

```
Int64Time TrickHLA::Int64Time::operator% (
    const Int64Interval & lhs ) const [inline]
```

Modulo operator.

##### Returns

A corresponding `TrickHLA::Int64Time` time value.

##### Parameters

<code>lhs</code>	Left hand side operand as a <code>TrickHLA::Int64Interval</code> .
------------------	--

Definition at line 204 of file Int64Time.hh.

References `TrickHLA::Int64Interval::getTimeInMicros()`, and `getTimeInMicros()`.

#### 7.24.3.10 `operator*()`

```
Int64Time TrickHLA::Int64Time::operator* (
    const Int64Interval & lhs ) const [inline]
```

Multiplication operator.

##### Returns

A corresponding `TrickHLA::Int64Time` time value.

##### Parameters

<code>lhs</code>	Left hand side operand as a <code>TrickHLA::Int64Interval</code> .
------------------	--

Definition at line 186 of file Int64Time.hh.

References TrickHLA::Int64Interval::getTimeInMicros(), and getTimeInMicros().

#### 7.24.3.11 operator+() [1/2]

```
Int64Time TrickHLA::Int64Time::operator+ (
    const Int64Interval & lhs ) const [inline]
```

Addition operator.

##### Returns

A corresponding [TrickHLA::Int64Time](#) time value.

##### Parameters

<i>lhs</i>	Left hand side operand as a <a href="#">TrickHLA::Int64Interval</a> .
------------	---

Definition at line 159 of file Int64Time.hh.

References TrickHLA::Int64Interval::getTimeInMicros(), and getTimeInMicros().

#### 7.24.3.12 operator+() [2/2]

```
Int64Time TrickHLA::Int64Time::operator+ (
    const Int64Time & lhs ) const [inline]
```

Addition operator.

##### Returns

A corresponding [TrickHLA::Int64Time](#) time value.

##### Parameters

<i>lhs</i>	Left hand side operand as a <a href="#">TrickHLA::Int64Time</a> .
------------	---

Definition at line 150 of file Int64Time.hh.

References [getTimeInMicros\(\)](#).

#### 7.24.3.13 operator+=() [1/2]

```
Int64Time TrickHLA::Int64Time::operator+= (
    const Int64Interval & lhs ) [inline]
```

Addition then assignment operator.

##### Returns

A corresponding [TrickHLA::Int64Time](#) time value.

##### Parameters

<i>lhs</i>	Left hand side operand as a <a href="#">TrickHLA::Int64Interval</a> .
------------	---

Definition at line 139 of file Int64Time.hh.

References TrickHLA::Int64Interval::getTimeInMicros(), getTimeInMicros(), and setTo().

#### 7.24.3.14 operator+=() [2/2]

```
Int64Time TrickHLA::Int64Time::operator+= (
    double lhs ) [inline]
```

Addition then assignment operator.

Returns

A corresponding [TrickHLA::Int64Time](#) time value.

Parameters

<i>lhs</i>	Left hand side operand as a floating point double time in seconds.
------------	--

Definition at line 128 of file Int64Time.hh.

References getTimeInMicros(), setTo(), and TrickHLA::Int64Interval::toMicroseconds().

#### 7.24.3.15 operator-() [1/2]

```
Int64Time TrickHLA::Int64Time::operator- (
    const Int64Interval & lhs ) const [inline]
```

Subtraction operator.

Returns

A corresponding [TrickHLA::Int64Time](#) time value.

Parameters

<i>lhs</i>	Left hand side operand as a <a href="#">TrickHLA::Int64Interval</a> .
------------	---

Definition at line 177 of file Int64Time.hh.

References TrickHLA::Int64Interval::getTimeInMicros(), and getTimeInMicros().

#### 7.24.3.16 operator-() [2/2]

```
Int64Time TrickHLA::Int64Time::operator- (
    const Int64Time & lhs ) const [inline]
```

Subtraction operator.

Returns

A corresponding [TrickHLA::Int64Time](#) time value.

Parameters

<i>lhs</i>	Left hand side operand as a <a href="#">TrickHLA::Int64Time</a> .
------------	---

Definition at line 168 of file Int64Time.hh.

References `getTimeInMicros()`.

#### 7.24.3.17 `operator/()`

```
Int64Time TrickHLA::Int64Time::operator/ (
    const Int64Interval & lhs ) const [inline]
```

Division operator.

##### Returns

A corresponding `TrickHLA::Int64Time` time value.

##### Parameters

<code>lhs</code>	Left hand side operand as a <code>TrickHLA::Int64Interval</code> .
------------------	--

Definition at line 195 of file Int64Time.hh.

References `TrickHLA::Int64Interval::getTimeInMicros()`, and `getTimeInMicros()`.

#### 7.24.3.18 `operator<()`

```
bool TrickHLA::Int64Time::operator< (
    const Int64Time & lhs ) const [inline]
```

Less than comparison operator.

##### Returns

True if right operand is greater than the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand as a <code>TrickHLA::Int64Time</code> .
------------------	--

Definition at line 213 of file Int64Time.hh.

References `getTimeInMicros()`.

#### 7.24.3.19 `operator<=()`

```
bool TrickHLA::Int64Time::operator<= (
    const Int64Time & lhs ) const [inline]
```

Less than or equal to comparison operator.

##### Returns

True if right operand is less than or equal to the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand a <code>TrickHLA::Int64Time</code> .
------------------	---

Definition at line 242 of file Int64Time.hh.

References `getTimelnMicros()`.

#### 7.24.3.20 `operator=()` [1/3]

```
virtual Int64Time& TrickHLA::Int64Time::operator= (
    double lhs ) [inline], [virtual]
```

Assignment operator.

##### Returns

A corresponding [TrickHLA::Int64Time](#) time value.

##### Parameters

<i>lhs</i>	Left hand side operand as floating point double time in seconds.
------------	--

Definition at line 101 of file Int64Time.hh.

References `_HLAtime`, and `TrickHLA::Int64Interval::toMicroseconds()`.

#### 7.24.3.21 `operator=()` [2/3]

```
virtual Int64Time& TrickHLA::Int64Time::operator= (
    int64_t lhs ) [inline], [virtual]
```

Assignment operator.

##### Returns

A corresponding [TrickHLA::Int64Time](#) time value.

##### Parameters

<i>lhs</i>	Left hand side operand as 64bit integer time in microseconds.
------------	---

Definition at line 110 of file Int64Time.hh.

References `_HLAtime`.

#### 7.24.3.22 `operator=()` [3/3]

```
virtual Int64Time& TrickHLA::Int64Time::operator= (
    Int64Time const & lhs ) [inline], [virtual]
```

Assignment operator.

##### Returns

A corresponding [TrickHLA::Int64Time](#) time value.

##### Parameters

<i>lhs</i>	Left hand side operand as <a href="#">TrickHLA::Int64Time</a> .
------------	---

Definition at line 119 of file Int64Time.hh.  
 References `_HLAtime`, and `getTimeInMicros()`.

#### 7.24.3.23 `operator==()` [1/2]

```
bool TrickHLA::Int64Time::operator== (
    const Int64Time & lhs ) const [inline]
```

Equals comparison operator.

##### Returns

True if right operand is equal to the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand as a <code>TrickHLA::Int64Time</code> .
------------------	--

Definition at line 266 of file Int64Time.hh.  
 References `getTimeInMicros()`.

#### 7.24.3.24 `operator==()` [2/2]

```
bool TrickHLA::Int64Time::operator== (
    int64_t lhs ) const [inline]
```

Equals comparison operator.

##### Returns

True if right operand is equal to the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand as a 64bit integer time in microseconds.
------------------	---

Definition at line 258 of file Int64Time.hh.  
 References `getTimeInMicros()`.

#### 7.24.3.25 `operator>()` [1/3]

```
bool TrickHLA::Int64Time::operator> (
    const Int64Time & lhs ) const [inline]
```

Greater than comparison operator.

##### Returns

True if right operand is greater than the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand a <code>TrickHLA::Int64Time</code> .
------------------	---

Definition at line 234 of file Int64Time.hh.

References `getTimeInMicros()`.

#### 7.24.3.26 `operator>()` [2/3]

```
bool TrickHLA::Int64Time::operator> (
    double lhs ) [inline]
```

Greater than comparison operator.

##### Returns

True if right operand is greater than the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand as floating point time in seconds.
------------------	---

Definition at line 226 of file Int64Time.hh.

References `getTimeInMicros()`, and `TrickHLA::Int64Interval::toMicroseconds()`.

#### 7.24.3.27 `operator>()` [3/3]

```
bool TrickHLA::Int64Time::operator> (
    int64_t lhs ) [inline]
```

Greater than comparison operator.

##### Returns

True if right operand is greater than the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand as a 64bit integer time in microseconds.
------------------	---

Definition at line 221 of file Int64Time.hh.

References `getTimeInMicros()`.

#### 7.24.3.28 `operator>=()`

```
bool TrickHLA::Int64Time::operator>= (
    const Int64Time & lhs ) const [inline]
```

Greater than or equal to comparison operator.

##### Returns

True if right operand is greater than or equal to the left operand; False otherwise.

##### Parameters

<code>lhs</code>	Left hand side operand as a <code>TrickHLA::Int64Time</code> .
------------------	--

Definition at line 250 of file Int64Time.hh.

References `getTimeInMicros()`.

#### 7.24.3.29 `setTo()` [1/4]

```
void Int64Time::setTo (
    const double value )
```

Set the time to the given value.

##### Parameters

<code>value</code>	The desired time interval in seconds.
--------------------	---------------------------------------

Definition at line 142 of file Int64Time.cpp.

References `_HLAtime`, and `TrickHLA::Int64Interval::toMicroseconds()`.

#### 7.24.3.30 `setTo()` [2/4]

```
void Int64Time::setTo (
    const int64_t value )
```

Set the time to the given value.

##### Parameters

<code>value</code>	The desired time in integer microseconds.
--------------------	---

Definition at line 136 of file Int64Time.cpp.

References `_HLAtime`.

Referenced by `DIS::ExecutionControl::announce_sync_point()`, `IMSim::ExecutionControl::announce_sync_point()`, `IMSim::ExecutionControl::check_pause()`, `TrickHLA::Interaction::extract_data()`, `TrickHLA::InteractionItem::InteractionItem()`, `operator+=()`, `SpaceFOM::ExecutionControl::receive_interaction()`, `IMSim::ExecutionControl::receive_interaction()`, `TrickHLA::Manager::receive_interaction()`, `TrickHLA::FedAmb::reflectAttributeValues()`, `IMSim::ExecutionControl::reinstate_logged_sync_pts()`, `TrickHLA::Interaction::send()`, `TrickHLA::Object::send_cyclic_data()`, `TrickHLA::Object::send_requested_data()`, `TrickHLA::Federate::set_granted_time()`, `TrickHLA::Object::set_last_update_time()`, and `TrickHLA::Federate::set_requested_time()`.

#### 7.24.3.31 `setTo()` [3/4]

```
void Int64Time::setTo (
    Int64Time const & value )
```

Set the time to the given value.

##### Parameters

<code>value</code>	The desired time as a <code>TrickHLA::Int64Time</code> .
--------------------	--

Definition at line 155 of file Int64Time.cpp.

References `_HLAtime`, and `getTimeInMicros()`.

**7.24.3.32 setTo() [4/4]**

```
void Int64Time::setTo (
    RTI1516_NAMESPACE::LogicalTime const & value )
```

Set the time to the given value.

**Parameters**

<b>value</b>	The desired time as an HLA LogicalTime.
--------------	---

Definition at line 148 of file Int64Time.cpp.

References `_HLAtime`.

**7.24.3.33 toString()**

```
wstring Int64Time::toString () const
```

Returns a wide string representing the time.

**Returns**

Wide string representing the time.

Definition at line 126 of file Int64Time.cpp.

References `getDoubleTime()`.

Referenced by `TrickHLA::TimedSyncPnt::to_string()`.

**7.24.4 Friends And Related Function Documentation****7.24.4.1 init\_attrTrickHLA\_\_Int64Time**

```
void init_attrTrickHLA__Int64Time () [friend]
```

**7.24.4.2 InputProcessor**

```
friend class InputProcessor [friend]
```

Definition at line 60 of file Int64Time.hh.

**7.24.5 Field Documentation****7.24.5.1 \_HLAtime**

```
RTI1516_NAMESPACE::HLAinteger64Time TrickHLA::Int64Time::_HLAtime [private]
```

**Data I/O: \*\***

HLA standard's class representation of integer64 time.

Definition at line 345 of file Int64Time.hh.

Referenced by `decode()`, `get()`, `getMicros()`, `getSeconds()`, `getTimeInMicros()`, `operator=()`, and `setTo()`.

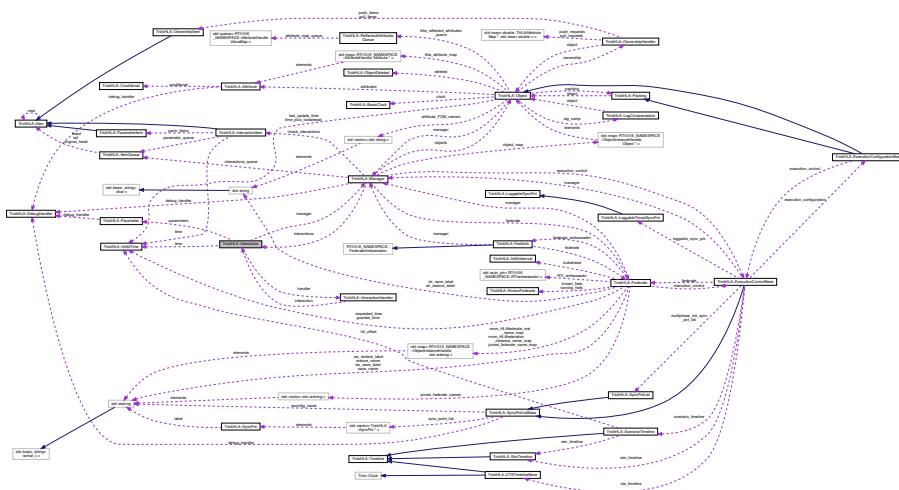
The documentation for this class was generated from the following files:

- [Int64Time.hh](#)
- [Int64Time.cpp](#)

## 7.25 TrickHLA::Interaction Class Reference

```
#include <Interaction.hh>
```

## Collaboration diagram for TrickHLA::Interaction:



## Public Member Functions

- **Interaction ()**  
*Default constructor for the [TrickHLA Interaction](#) class.*
  - **virtual ~Interaction ()**  
*Destructor for the [TrickHLA Interaction](#) class.*
  - **void initialize (Manager \*trickhla\_mngr)**  
*Initializes the [TrickHLA Interaction](#) class.*
  - **void publish\_interaction ()**  
*Publishes the interaction to the RTI.*
  - **void unpublish\_interaction ()**  
*Unpublish the [Interaction](#).*
  - **void subscribe\_to\_interaction ()**  
*Subscribes to the [Interaction](#).*
  - **void unsubscribe\_from\_interaction ()**  
*Unsubscribes from the [Interaction](#).*
  - **void setup\_preferred\_order\_with\_RTI ()**  
*Setup the interaction preferred order with the RTI.*
  - **void remove ()**  
*Will unpublish or unsubscribe the interaction.*
  - **bool send (RTI1516\_USERDATA const &the\_user\_supplied\_tag)**  
*Sends the interaction to the RTI using Receive Order.*
  - **bool send (double send\_HLA\_time, RTI1516\_USERDATA const &the\_user\_supplied\_tag)**  
*ends the interaction to the RTI using Timestamp Order.*
  - **void process\_interaction ()**  
*Process the interaction by decoding the parameter data into the users simulation variables and calling the users interaction-handler.*

- void `extract_data` (`InteractionItem` \*`interaction_item`)  
*Extracts the parameters for the received `Interaction`.*
- const `char` \* `get_FOM_name` () const  
*Get the FOM name for this interaction.*
- bool `is_publish` () const  
*Query if this interaction is published.*
- bool `is_subscribe` () const  
*Query if this interaction is subscribed.*
- `RTI1516_NAMESPACE::InteractionClassHandle` `get_class_handle` () const  
*Get this interactions `InteractionClassHandle`.*
- void `set_class_handle` (`RTI1516_NAMESPACE::InteractionClassHandle` `id`)  
*Set the interaction `InteractionClassHandle`.*
- int `get_parameter_count` () const  
*Get the parameter count for this interaction.*
- `Parameter` \* `get_parameters` ()  
*Get the `TrickHLA::Parameter` array associated with this interaction.*
- void `lock` ()  
*Lock the thread mutex.*
- void `unlock` ()  
*Unlock the thread mutex.*
- bool `is_changed` () const  
*Query if the interaction data has changed.*
- void `mark_changed` ()  
*Mark the data as cahnged.*
- void `mark_unchanged` ()  
*Mark the data as unchanged, and clear the change flag for all the parameters as well.*
- `InteractionHandler` \* `get_handler` ()  
*Get the `TrickHLA::InteractionHandler` associated with this interaction.*
- void `set_handler` (`InteractionHandler` \*`ptr`)  
*Set the `TrickHLA::InteractionHandler` for this interaction.*
- `Manager` \* `get_manager` ()  
*Get the associated `TrickHLA::Manager` instance.*
- `Federate` \* `get_federate` ()  
*Returns a pointer to our federate, or `NULL` if one does not exist yet.*
- `RTI1516_NAMESPACE::RTIambassador` \* `get_RTI_ambassador` ()  
*Returns a pointer to the RTI ambassador, or `NULL` if one does not exist yet.*
- `Int64Interval` `get_fed_looking_ahead` () const  
*Return a copy of the federate's lookahead time.*
- `Int64Time` `get_granted_fed_time` () const  
*Return a copy of the granted HLA logical time.*
- bool `should_print` (const `DebugLevelEnum` &`level`, const `DebugSourceEnum` &`code`) const  
*Determine if the verbose debug comments should be printed to the console.*
- void `set_FOM_name` (`char` \*`in_name`)  
*Set the FOM name for this interaction.*
- void `set_user_supplied_tag` (`unsigned char` \*`tag`, `size_t` `tag_size`)  
*Set the received user supplied tag.*
- void `set_publish` ()

- void `set_subscribe ()`

*Mark this interaction as published.*
- void `set_parameter_count (int in_num)`

*Set the interaction parameter count.*
- void `set_parameters (Parameter *ptr)`

*Set the parameter array.*
- `TransportationEnum get_preferred_order () const`

*Get the preferred transport order for this interaction.*

## Data Fields

- `char * FOM_name`

**Units:** –  
*FOM name for the interaction.*
- `bool publish`

**Units:** –  
*True to publish interaction.*
- `bool subscribe`

**Units:** –  
*True to subscribe to interaction.*
- `TransportationEnum preferred_order`

**Units:** –  
*Either Timestamp (default) or Receive Order.*
- `int param_count`

**Units:** –  
*Number of interaction parameters.*
- `Parameter * parameters`

**Units:** –  
*Array of interaction parameters.*
- `InteractionHandler * handler`

**Units:** –  
*Interaction handler.*

## Private Member Functions

- `Interaction (const Interaction &rhs)`

*Copy constructor for `Interaction` class.*
- `Interaction & operator= (const Interaction &rhs)`

*Assignment operator for `Interaction` class.*

## Private Attributes

- `pthread_mutex_t mutex`

**Data I/O:** \*\*  
*Mutex to lock thread over critical code sections.*
- `bool changed`

**Units:** –  
*Flag indicating the data has changed.*
- `bool received_as_TSO`

**Units:** –*True if received interaction as Timestamp order.*

- `Int64Time time`

**Units:** –*Time used for Timestamp Order interaction.*

- `Manager * manager`

**Units:** –*TrickHLA Manager.*

- `RTI1516_NAMESPACE::InteractionClassHandle class_handle`

**Data I/O:** \*\**RTI Interaction Class handle.*

- `size_t user_supplied_tag_size`

**Units:** –*Number of bytes in the user supplied tag.*

- `size_t user_supplied_tag_capacity`

**Units:** –*Capacity of the user supplied tag.*

- `unsigned char * user_supplied_tag`

**Units:** –*User supplied tag data.***Friends**

- class `InputProcessor`
- void `init_attrTrickHLA__Interaction ()`

**7.25.1 Detailed Description**

Definition at line 82 of file `Interaction.hh`.

**7.25.2 Constructor & Destructor Documentation****7.25.2.1 `Interaction()` [1/2]**

`Interaction::Interaction ( )`

Default constructor for the `TrickHLA Interaction` class.

**Trick Job Class:** *initialization*

Definition at line 66 of file `Interaction.cpp`.

References `mutex`.

**7.25.2.2 `~Interaction()`**

`Interaction::~Interaction ( ) [virtual]`

Destructor for the `TrickHLA Interaction` class.

**Trick Job Class:** *shutdown*

Definition at line 89 of file `Interaction.cpp`.

References `mutex`, `remove()`, `user_supplied_tag`, and `user_supplied_tag_size`.

### 7.25.2.3 `Interaction()` [2/2]

```
TrickHLA::Interaction::Interaction (
    const Interaction & rhs ) [private]
```

Copy constructor for `Interaction` class.

This constructor is private to prevent inadvertent copies.

## 7.25.3 Member Function Documentation

### 7.25.3.1 `extract_data()`

```
void Interaction::extract_data (
    InteractionItem * interaction_item )
```

Extracts the parameters for the received `Interaction`.

#### Parameters

<code>interaction_item</code>	<code>Interaction</code> item.
-------------------------------	--------------------------------

Definition at line 1111 of file `Interaction.cpp`.

References `class_handle`, `TrickHLA::ParameterItem::data`, `TrickHLA::DEBUG_LEVEL_7_TRACE`, `TrickHLA::DEBUG_SOURCE_INTERACTION`, `TrickHLA::ItemQueue::empty()`, `TrickHLA::Parameter::extract_data()`, `TrickHLA::ItemQueue::front()`, `TrickHLA::Parameter::get_FOM_name()`, `get_FOM_name()`, `TrickHLA::ParameterItem::index`, `is_subscribe()`, `TrickHLA::InteractionItem::is_timestamp_order()`, `lock()`, `mark_changed()`, `param_count`, `TrickHLA::InteractionItem::parameter_queue`, `parameters`, `TrickHLA::ItemQueue::pop()`, `received_as_TSO`, `set_user_supplied_tag()`, `TrickHLA::Int64Time::setTo()`, `should_print()`, `TrickHLA::ParameterItem::size`, `THLA_NEWLINE`, `TrickHLA::InteractionItem::time`, `time`, `TrickHLA::StringUtilities::to_string()`, `unlock()`, `TrickHLA::InteractionItem::user_supplied_tag`, and `TrickHLA::InteractionItem::user_supplied_tag_size`.

Referenced by `TrickHLA::Manager::process_interactions()`, `SpaceFOM::ExecutionControl::receive_interaction()`, and `IMSim::ExecutionControl::receive_interaction()`.

### 7.25.3.2 `get_class_handle()`

```
RTI1516_NAMESPACE::InteractionClassHandle TrickHLA::Interaction::get_class_handle () const [inline]
```

Get this interactions `InteractionClassHandle`.

#### Returns

Copy of this interactions `InteractionClassHandle`.

Definition at line 188 of file `Interaction.hh`.

References `class_handle`.

Referenced by `SpaceFOM::ExecutionControl::receive_interaction()`, `remove()`, `TrickHLA::Manager::setup_interaction_RTI_handles()`, `IMSim::ExecutionControl::unpublish()`, `TrickHLA::Manager::unpublish()`, `IMSim::ExecutionControl::unsubscribe()`, and `TrickHLA::Manager::unsubscribe()`.

### 7.25.3.3 `get_fed_lookahead()`

```
Int64Interval Interaction::get_fed_lookahead () const
```

Return a copy of the federate's lookahead time.

**Returns**

A copy of the federate's lookahead time.

If the manager does not exist, -1.0 seconds is assigned to the returned object.

Definition at line 1191 of file Interaction.cpp.

References TrickHLA::Manager::get\_fed\_lookahead(), and manager.

Referenced by TrickHLA::InteractionHandler::get\_fed\_lookahead(), IMSim::FreezeInteractionHandler::receive\_← interaction(), and IMSim::FreezeInteractionHandler::send\_scenario\_freeze\_interaction().

**7.25.3.4 get\_federate()**

```
Federate * Interaction::get_federate ( )
```

Returns a pointer to our federate, or NULL if one does not exist yet.

**Returns**

A pointer to this federate's [TrickHLA::Federate](#) instance.

Definition at line 1217 of file Interaction.cpp.

References TrickHLA::Manager::get\_federate(), and manager.

Referenced by TrickHLA::InteractionHandler::get\_cte\_time(), TrickHLA::InteractionHandler::get\_scenario\_time(), Trick← HLA::InteractionHandler::get\_sim\_time(), IMSim::FreezeInteractionHandler::receive\_interaction(), SpaceFOM::MTR← InteractionHandler::receive\_interaction(), send(), SpaceFOM::MTRInteractionHandler::send\_interaction(), and IMSim← ::FreezeInteractionHandler::send\_scenario\_freeze\_interaction().

**7.25.3.5 get\_FOM\_name()**

```
const char* TrickHLA::Interaction::get_FOM_name ( ) const [inline]
```

Get the FOM name for this interaction.

**Returns**

Constant string with the FOM name for this interaction.

Definition at line 176 of file Interaction.hh.

References FOM\_name.

Referenced by extract\_data(), process\_interaction(), publish\_interaction(), remove(), send(), IMSim::Execution← Control::setup\_interaction\_ref\_attributes(), SpaceFOM::ExecutionControl::setup\_interaction\_ref\_attributes(), Trick← HLA::Manager::setup\_interaction\_ref\_attributes(), TrickHLA::Manager::setup\_interaction\_RTI\_handles(), setup← preferred\_order\_with\_RTI(), subscribe\_to\_interaction(), unpublish\_interaction(), and unsubscribe\_from\_interaction().

**7.25.3.6 get\_granted\_fed\_time()**

```
Int64Time Interaction::get_granted_fed_time ( ) const
```

Return a copy of the granted HLA logical time.

**Returns**

A copy of the federation granted time.

If the manager does not exist, MAX\_LOGICAL\_TIME\_SECONDS is assigned to the returned object.

Definition at line 1206 of file Interaction.cpp.

References TrickHLA::Manager::get\_granted\_fed\_time(), manager, and TrickHLA::MAX\_LOGICAL\_TIME\_SECONDS.

Referenced by TrickHLA::InteractionHandler::get\_granted\_fed\_time(), IMSim::FreezeInteractionHandler::receive\_← interaction(), and IMSim::FreezeInteractionHandler::send\_scenario\_freeze\_interaction().

### 7.25.3.7 `get_handler()`

`InteractionHandler* TrickHLA::Interaction::get_handler ( ) [inline]`  
 Get the `TrickHLA::InteractionHandler` associated with this interaction.

#### Returns

The interaction handler.

Definition at line 224 of file `Interaction.hh`.

References handler.

Referenced by `IMSim::ExecutionControl::start_federation_save_at_scenario_time()`, `IMSim::ExecutionControl::trigger_freeze_interaction()`, and `IMSim::ExecutionControl::~ExecutionControl()`.

### 7.25.3.8 `get_manager()`

`Manager* TrickHLA::Interaction::get_manager ( ) [inline]`  
 Get the associated `TrickHLA::Manager` instance.

#### Returns

Pointer to the associated `TrickHLA::Manager` instance.

Definition at line 232 of file `Interaction.hh`.

References manager.

### 7.25.3.9 `get_parameter_count()`

`int TrickHLA::Interaction::get_parameter_count ( ) const [inline]`  
 Get the parameter count for this interaction.

#### Returns

The parameter count for this interaction.

Definition at line 196 of file `Interaction.hh`.

References param\_count.

Referenced by `SpaceFOM::ExecutionControl::receive_interaction()`, `IMSim::ExecutionControl::setup_interaction_ref_attributes()`, `SpaceFOM::ExecutionControl::setup_interaction_ref_attributes()`, `TrickHLA::Manager::setup_interaction_ref_attributes()`, and `TrickHLA::Manager::setup_interaction_RTI_handles()`.

### 7.25.3.10 `get_parameters()`

`Parameter* TrickHLA::Interaction::get_parameters ( ) [inline]`  
 Get the `TrickHLA::Parameter` array associated with this interaction.

#### Returns

The `Parameter` array.

Definition at line 200 of file `Interaction.hh`.

References parameters.

Referenced by `SpaceFOM::ExecutionControl::receive_interaction()`, `TrickHLA::Manager::setup_interaction_ref_attributes()`, and `TrickHLA::Manager::setup_interaction_RTI_handles()`.

**7.25.3.11 get\_preferred\_order()**

```
TransportationEnum TrickHLA::Interaction::get_preferred_order ( ) const [inline]
Get the preferred transport order for this interaction.
```

**Returns**

The preferred transport order for this interaction.

Definition at line 290 of file Interaction.hh.

References preferred\_order.

**7.25.3.12 get\_RTI\_ambassador()**

```
RTIambassador * Interaction::get_RTI_ambassador ( )
Returns a pointer to the RTI ambassador, or NULL if one does not exist yet.
```

**Returns**

Pointer to this federate's associated RTIambassador instance.

Definition at line 1222 of file Interaction.cpp.

References TrickHLA::Manager::get\_RTI\_ambassador(), and manager.

Referenced by publish\_interaction(), remove(), send(), setup\_preferred\_order\_with\_RTI(), subscribe\_to\_interaction(), unpublish\_interaction(), and unsubscribe\_from\_interaction().

**7.25.3.13 initialize()**

```
void Interaction::initialize (
    Manager * trickhla_mgr )
Initializes the TrickHLA Interaction class.
```

**Parameters**

<i>trickhla_mgr</i>	Pointer to the associated <a href="#">TrickHLA::Manager</a> class.
---------------------	--

**Trick Job Class: initialization**

Definition at line 109 of file Interaction.cpp.

References FOM\_name, handler, TrickHLA::InteractionHandler::initialize\_callback(), manager, param\_count, parameters, preferred\_order, THLA\_ENDL, THLA\_NEWLINE, TrickHLA::TRANSPORT\_FIRST\_VALUE, TrickHLA::TRANSPORT\_LAST\_VALUE, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by IMSim::ExecutionControl::setup\_interaction\_ref\_attributes(), SpaceFOM::ExecutionControl::setup\_interaction\_ref\_attributes(), and TrickHLA::Manager::setup\_interaction\_ref\_attributes().

**7.25.3.14 is\_changed()**

```
bool TrickHLA::Interaction::is_changed ( ) const [inline]
Query if the interaction data has changed.
```

**Returns**

True if data has changed; False otherwise.

Definition at line 211 of file Interaction.hh.

References changed.

Referenced by `process_interaction()`.

#### 7.25.3.15 `is_publish()`

```
bool TrickHLA::Interaction::is_publish ( ) const [inline]
```

Query if this interaction is published.

##### Returns

True if this interaction is published; False otherwise.

Definition at line 180 of file `Interaction.hh`.

References `publish`.

Referenced by `publish_interaction()`, `remove()`, `send()`, `setup_preferred_order_with_RTI()`, `SpaceFOM::ExecutionControl::unpublish()`, and `unpublish_interaction()`.

#### 7.25.3.16 `is_subscribe()`

```
bool TrickHLA::Interaction::is_subscribe ( ) const [inline]
```

Query if this interaction is subscribed.

##### Returns

True if this interaction is subscribed; False otherwise.

Definition at line 184 of file `Interaction.hh`.

References `subscribe`.

Referenced by `extract_data()`, `TrickHLA::Manager::process_interactions()`, `SpaceFOM::ExecutionControl::receive_interaction()`, `subscribe_to_interaction()`, `SpaceFOM::ExecutionControl::unsubscribe()`, and `unsubscribe_from_interaction()`.

#### 7.25.3.17 `lock()`

```
void TrickHLA::Interaction::lock ( ) [inline]
```

Lock the thread mutex.

Definition at line 203 of file `Interaction.hh`.

References `mutex`.

Referenced by `extract_data()`, `process_interaction()`, and `send()`.

#### 7.25.3.18 `mark_changed()`

```
void TrickHLA::Interaction::mark_changed ( ) [inline]
```

Mark the data as cahnged.

Definition at line 214 of file `Interaction.hh`.

References `changed`.

Referenced by `extract_data()`.

#### 7.25.3.19 `mark_unchanged()`

```
void Interaction::mark_unchanged ( )
```

Mark the data as unchanged, and clear the change flag for all the parameters as well.

Definition at line 1178 of file `Interaction.cpp`.

References changed, TrickHLA::Parameter::mark\_unchanged(), param\_count, and parameters.  
Referenced by process\_interaction().

### 7.25.3.20 operator=( )

```
Interaction& TrickHLA::Interaction::operator= (
    const Interaction & rhs ) [private]
```

Assignment operator for [Interaction](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.25.3.21 process\_interaction()

```
void Interaction::process_interaction ( )
```

Process the interaction by decoding the parameter data into the users simulation variables and calling the users interaction-handler.

Definition at line 1061 of file [Interaction.cpp](#).

References class\_handle, TrickHLA::DEBUG\_LEVEL\_5\_TRACE, TrickHLA::DEBUG\_SOURCE\_INTERACTION, get\_FOM\_name(), TrickHLA::Int64Time::getDoubleTime(), handler, is\_changed(), lock(), mark\_unchanged(), param\_count, parameters, TrickHLA::InteractionHandler::receive\_interaction(), received\_as\_TSO, RTI1516\_USERDATA, should\_print(), THLA\_NEWLINE, time, TrickHLA::StringUtilities::to\_string(), unlock(), TrickHLA::Parameter::unpack\_parameter\_buffer(), user\_supplied\_tag, and user\_supplied\_tag\_size.

Referenced by TrickHLA::Manager::process\_interactions(), SpaceFOM::ExecutionControl::receive\_interaction(), and IMSim::ExecutionControl::receive\_interaction().

### 7.25.3.22 publish\_interaction()

```
void Interaction::publish_interaction ( )
```

Publishes the interaction to the RTI.

Definition at line 412 of file [Interaction.cpp](#).

References class\_handle, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_INTERACTION, get\_FOM\_name(), get\_RTI\_ambassador(), is\_publish(), RTI1516\_EXCEPTION, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by SpaceFOM::ExecutionControl::publish(), IMSim::ExecutionControl::publish(), and TrickHLA::Manager::publish().

### 7.25.3.23 remove()

```
void Interaction::remove ( )
```

Will unpublish or unsubscribe the interaction.

Called from the virtual destructor. **Trick Job Class:** *shutdown*

Definition at line 234 of file [Interaction.cpp](#).

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_INTERACTION, get\_class\_handle(), get\_FOM\_name(), get\_RTI\_ambassador(), is\_publish(), TrickHLA::Manager::is\_shutdown(), manager, RTI1516\_EXCEPTION, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD, and unsubscribe\_from\_interaction().

Referenced by ~Interaction().

### 7.25.3.24 send() [1/2]

```
bool Interaction::send (
    double send_HLA_time,
    RTI1516_USERDATA const & the_user_supplied_tag )
ends the interaction to the RTI using Timestamp Order.
```

#### Returns

True if interaction was sent; False otherwise.

#### Parameters

<i>send_HLA_time</i>	The HLA logical time the user wants to send the interaction.
<i>the_user_supplied_tag</i>	Users tag.

Definition at line 935 of file Interaction.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_7\_TRACE, TrickHLA::DEBUG\_SO←URCE\_INTERACTION, TrickHLA::Int64Time::get(), TrickHLA::Parameter::get\_encoded\_parameter\_value(), get←federate(), get\_FOM\_name(), TrickHLA::Parameter::get\_parameter\_handle(), get\_RTI\_ambassador(), TrickHLA::←Int64Time::getDoubleTime(), TrickHLA::Int64Time::getTimeInMicros(), TrickHLA::Federate::in\_time\_regulating\_state(), is\_publish(), lock(), param\_count, parameters, preferred\_order, RTI1516\_EXCEPTION, TrickHLA::Int64Time::setTo(), should\_print(), TrickHLA::Federate::should\_publish\_data(), THLA\_ENDL, THLA\_NEWLINE, time, TrickHLA::String←Utilities::to\_string(), TrickHLA::TRANSPORT\_RECEIVE\_ORDER, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD, and unlock().

### 7.25.3.25 send() [2/2]

```
bool Interaction::send (
    RTI1516_USERDATA const & the_user_supplied_tag )
```

Sends the interaction to the RTI using Receive Order.

#### Returns

True if interaction was sent; False otherwise.

#### Parameters

<i>the_user_supplied_tag</i>	Users tag.
------------------------------	------------

Definition at line 864 of file Interaction.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_INTERACTION, TrickHLA::←Parameter::get\_encoded\_parameter\_value(), get\_federate(), get\_FOM\_name(), TrickHLA::Parameter::get←parameter\_handle(), get\_RTI\_ambassador(), is\_publish(), lock(), param\_count, parameters, RTI1516\_EXCEPTION, should\_print(), TrickHLA::Federate::should\_publish\_data(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD, and unlock().

Referenced by TrickHLA::InteractionHandler::send\_interaction().

### 7.25.3.26 set\_class\_handle()

```
void TrickHLA::Interaction::set_class_handle (
```

```
RTI1516_NAMESPACE::InteractionClassHandle id ) [inline]
```

Set the interaction InteractionClassHandle.

#### Parameters

<i>id</i>	The interaction InteractionClassHandle.
-----------	---

Definition at line 192 of file Interaction.hh.

References class\_handle.

Referenced by TrickHLA::Manager::setup\_interaction\_RTI\_handles().

### 7.25.3.27 set\_FOM\_name()

```
void TrickHLA::Interaction::set_FOM_name (
    char * in_name ) [inline]
```

Set the FOM name for this interaction.

#### Parameters

<i>in_name</i>	The FOM name for this interaction.
----------------	------------------------------------

Definition at line 258 of file Interaction.hh.

References FOM\_name.

Referenced by IMSim::ExecutionControl::setup\_interaction\_ref\_attributes(), and SpaceFOM::ExecutionControl::setup\_interaction\_ref\_attributes().

### 7.25.3.28 set\_handler()

```
void TrickHLA::Interaction::set_handler (
    InteractionHandler * ptr ) [inline]
```

Set the [TrickHLA::InteractionHandler](#) for this interaction.

#### Parameters

<i>ptr</i>	The <a href="#">TrickHLA::InteractionHandler</a> instance to use.
------------	---

Definition at line 227 of file Interaction.hh.

Referenced by IMSim::ExecutionControl::setup\_interaction\_ref\_attributes(), SpaceFOM::ExecutionControl::setup\_interaction\_ref\_attributes(), and IMSim::ExecutionControl::~ExecutionControl().

### 7.25.3.29 set\_parameter\_count()

```
void TrickHLA::Interaction::set_parameter_count (
    int in_num ) [inline]
```

Set the interaction parameter count.

#### Parameters

<i>in_num</i>	The number of parameters in this interaction.
---------------	---

Definition at line 282 of file Interaction.hh.

References `param_count`.

Referenced by `IMSim::ExecutionControl::setup_interaction_ref_attributes()`, and `SpaceFOM::ExecutionControl::setup_interaction_ref_attributes()`.

### 7.25.3.30 `set_parameters()`

```
void TrickHLA::Interaction::set_parameters (
    Parameter * ptr ) [inline]
```

Set the parameter array.

#### Parameters

<code>ptr</code>	Pointer to the <code>TrickHLA::Parameter</code> array associated with this interaction.
------------------	---

Definition at line 286 of file `Interaction.hh`.

References `parameters`.

Referenced by `IMSim::ExecutionControl::setup_interaction_ref_attributes()`, and `SpaceFOM::ExecutionControl::setup_interaction_ref_attributes()`.

### 7.25.3.31 `set_publish()`

```
void TrickHLA::Interaction::set_publish ( ) [inline]
```

Mark this interaction as published.

Definition at line 275 of file `Interaction.hh`.

References `publish`.

Referenced by `IMSim::ExecutionControl::setup_interaction_ref_attributes()`, and `SpaceFOM::ExecutionControl::setup_interaction_ref_attributes()`.

### 7.25.3.32 `set_subscribe()`

```
void TrickHLA::Interaction::set_subscribe ( ) [inline]
```

Mark this interaction as subscribed.

Definition at line 278 of file `Interaction.hh`.

References `subscribe`.

Referenced by `IMSim::ExecutionControl::setup_interaction_ref_attributes()`, and `SpaceFOM::ExecutionControl::setup_interaction_ref_attributes()`.

### 7.25.3.33 `set_user_supplied_tag()`

```
void Interaction::set_user_supplied_tag (
    unsigned char * tag,
    size_t tag_size )
```

Set the received user supplied tag.

#### Parameters

<code>tag</code>	The user supplied tag.
<code>tag_size</code>	Size of the user supplied tag.

Definition at line 210 of file `Interaction.cpp`.

References `user_supplied_tag`, `user_supplied_tag_capacity`, and `user_supplied_tag_size`. Referenced by `extract_data()`.

#### 7.25.3.34 `setup_preferred_order_with_RTI()`

```
void Interaction::setup_preferred_order_with_RTI ( )
```

Setup the interaction preferred order with the RTI.

Definition at line 273 of file `Interaction.cpp`.

References `class_handle`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_INTERACTION`, `get_FOM_name()`, `get_RTI_ambassador()`, `is_publish()`, `preferred_order`, `RTI1516_EXCEPTION`, `should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, `TrickHLA::TRANSPORT_RECEIVE_ORDER`, `TrickHLA::TRANSPORT_SPECIFIED_IN_FOM`, `TrickHLA::TRANSPORT_TIMESTAMP_ORDER`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `TrickHLA::Manager::setup_preferred_order_with_RTI()`.

#### 7.25.3.35 `should_print()`

```
bool Interaction::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const
```

Determine if the verbose debug comments should be printed to the console.

Returns

Returns true if the requested message should print level.

##### Parameters

<code>level</code>	Debug level of the incoming message.
<code>code</code>	Source code association of the incoming messages.

Definition at line 200 of file `Interaction.cpp`.

References `manager`, and `TrickHLA::Manager::should_print()`.

Referenced by `extract_data()`, `process_interaction()`, `publish_interaction()`, `remove()`, `send()`, `setup_preferred_order_with_RTI()`, `TrickHLA::InteractionHandler::should_print()`, `subscribe_to_interaction()`, `unpublish_interaction()`, and `unsubscribe_from_interaction()`.

#### 7.25.3.36 `subscribe_to_interaction()`

```
void Interaction::subscribe_to_interaction ( )
```

Subscribes to the [Interaction](#).

Definition at line 632 of file `Interaction.cpp`.

References `class_handle`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_INTERACTION`, `get_FOM_name()`, `get_RTI_ambassador()`, `is_subscribe()`, `RTI1516_EXCEPTION`, `should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `SpaceFOM::ExecutionControl::subscribe()`, `IMSim::ExecutionControl::subscribe()`, and `TrickHLA::Manager::subscribe()`.

### 7.25.3.37 unlock()

```
void TrickHLA::Interaction::unlock ( ) [inline]
Unlock the thread mutex.
Definition at line 206 of file Interaction.hh.
References mutex.
Referenced by extract_data(), process_interaction(), and send().
```

### 7.25.3.38 unpublish\_interaction()

```
void Interaction::unpublish_interaction ( )
Unpublish the Interaction.
Definition at line 523 of file Interaction.cpp.
References class_handle, TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_INTERACTION, get<-
_FOM_name(), get_RTI_ambassador(), is_publish(), RTI1516_EXCEPTION, should_print(), THLA_ENDL, THLA_N<-
EWLINE, TrickHLA::StringUtilities::to_string(), TRICKHLA_RESTORE_FPU_CONTROL_WORD, TRICKHLA_SAVE<-
_FPU_CONTROL_WORD, and TRICKHLA_VALIDATE_FPU_CONTROL_WORD.
Referenced by SpaceFOM::ExecutionControl::unpublish(), IMSim::ExecutionControl::unpublish(), and TrickHLA::-
Manager::unpublish().
```

### 7.25.3.39 unsubscribe\_from\_interaction()

```
void Interaction::unsubscribe_from_interaction ( )
Unsubscribes from the Interaction.
Definition at line 753 of file Interaction.cpp.
References class_handle, TrickHLA::DEBUG_LEVEL_2_TRACE, TrickHLA::DEBUG_SOURCE_INTERACTION, get<-
_FOM_name(), get_RTI_ambassador(), is_subscribe(), RTI1516_EXCEPTION, should_print(), THLA_ENDL, THLA_N<-
EWLINE, TrickHLA::StringUtilities::to_string(), TRICKHLA_RESTORE_FPU_CONTROL_WORD, TRICKHLA_SAVE<-
_FPU_CONTROL_WORD, and TRICKHLA_VALIDATE_FPU_CONTROL_WORD.
Referenced by remove(), SpaceFOM::ExecutionControl::unsubscribe(), IMSim::ExecutionControl::unsubscribe(), and
TrickHLA::Manager::unsubscribe().
```

## 7.25.4 Friends And Related Function Documentation

### 7.25.4.1 init\_attrTrickHLA\_\_Interaction

```
void init_attrTrickHLA__Interaction ( ) [friend]
```

### 7.25.4.2 InputProcessor

```
friend class InputProcessor [friend]
Definition at line 89 of file Interaction.hh.
```

## 7.25.5 Field Documentation

### 7.25.5.1 changed

```
bool TrickHLA::Interaction::changed [private]
```

**Units:** –

Flag indicating the data has changed.

Definition at line 295 of file Interaction.hh.

Referenced by is\_changed(), mark\_changed(), and mark\_unchanged().

**7.25.5.2 class\_handle**

```
RTI1516_NAMESPACE::InteractionClassHandle TrickHLA::Interaction::class_handle [private]
```

**Data I/O:** \*\*

RTI [Interaction](#) Class handle.

Definition at line 302 of file Interaction.hh.

Referenced by extract\_data(), get\_class\_handle(), process\_interaction(), publish\_interaction(), set\_class\_handle(), setup\_preferred\_order\_with\_RTI(), subscribe\_to\_interaction(), unsubscribe\_from\_interaction().

**7.25.5.3 FOM\_name**

```
char* TrickHLA::Interaction::FOM_name
```

**Units:** –

FOM name for the interaction.

Definition at line 98 of file Interaction.hh.

Referenced by get\_FOM\_name(), initialize(), and set\_FOM\_name().

**7.25.5.4 handler**

```
InteractionHandler* TrickHLA::Interaction::handler
```

**Units:** –

[Interaction](#) handler.

Definition at line 108 of file Interaction.hh.

Referenced by get\_handler(), initialize(), and process\_interaction().

**7.25.5.5 manager**

```
Manager* TrickHLA::Interaction::manager [private]
```

**Units:** –

[TrickHLA Manager](#).

Definition at line 301 of file Interaction.hh.

Referenced by get\_fed\_lookahead(), get\_federate(), get\_granted\_fed\_time(), get\_manager(), get\_RTI\_ambassador(), initialize(), remove(), and should\_print().

**7.25.5.6 mutex**

```
pthread_mutex_t TrickHLA::Interaction::mutex [private]
```

**Data I/O:** \*\*

Mutex to lock thread over critical code sections.

Definition at line 293 of file Interaction.hh.

Referenced by Interaction(), lock(), unlock(), and ~Interaction().

### 7.25.5.7 `param_count`

```
int TrickHLA::Interaction::param_count
```

**Units:** –

Number of interaction parameters.

Definition at line 105 of file Interaction.hh.

Referenced by `extract_data()`, `get_parameter_count()`, `initialize()`, `mark_unchanged()`, `process_interaction()`, `send()`, and `set_parameter_count()`.

### 7.25.5.8 `parameters`

```
Parameter* TrickHLA::Interaction::parameters
```

**Units:** –

Array of interaction parameters.

Definition at line 106 of file Interaction.hh.

Referenced by `extract_data()`, `get_parameters()`, `initialize()`, `mark_unchanged()`, `process_interaction()`, `send()`, and `set_parameters()`.

### 7.25.5.9 `preferred_order`

```
TransportationEnum TrickHLA::Interaction::preferred_order
```

**Units:** –

Either Timestamp (default) or Receive Order.

Definition at line 103 of file Interaction.hh.

Referenced by `get_preferred_order()`, `initialize()`, `send()`, and `setup_preferred_order_with_RTI()`.

### 7.25.5.10 `publish`

```
bool TrickHLA::Interaction::publish
```

**Units:** –

True to publish interaction.

Definition at line 100 of file Interaction.hh.

Referenced by `is_publish()`, and `set_publish()`.

### 7.25.5.11 `received_as_TSO`

```
bool TrickHLA::Interaction::received_as_TSO [private]
```

**Units:** –

True if received interaction as Timestamp order.

Definition at line 297 of file Interaction.hh.

Referenced by `extract_data()`, and `process_interaction()`.

### 7.25.5.12 `subscribe`

```
bool TrickHLA::Interaction::subscribe
```

**Units:** –

True to subscribe to interaction.

Definition at line 101 of file Interaction.hh.

Referenced by `is_subscribe()`, and `set_subscribe()`.

### 7.25.5.13 time

```
Int64Time TrickHLA::Interaction::time [private]
```

**Units:** –

Time used for Timestamp Order interaction.

Definition at line 299 of file Interaction.hh.

Referenced by extract\_data(), process\_interaction(), and send().

### 7.25.5.14 user\_supplied\_tag

```
unsigned char* TrickHLA::Interaction::user_supplied_tag [private]
```

**Units:** –

User supplied tag data.

Definition at line 306 of file Interaction.hh.

Referenced by process\_interaction(), set\_user\_supplied\_tag(), and ~Interaction().

### 7.25.5.15 user\_supplied\_tag\_capacity

```
size_t TrickHLA::Interaction::user_supplied_tag_capacity [private]
```

**Units:** –

Capacity of the user supplied tag.

Definition at line 305 of file Interaction.hh.

Referenced by set\_user\_supplied\_tag().

### 7.25.5.16 user\_supplied\_tag\_size

```
size_t TrickHLA::Interaction::user_supplied_tag_size [private]
```

**Units:** –

Number of bytes in the user supplied tag.

Definition at line 304 of file Interaction.hh.

Referenced by process\_interaction(), set\_user\_supplied\_tag(), and ~Interaction().

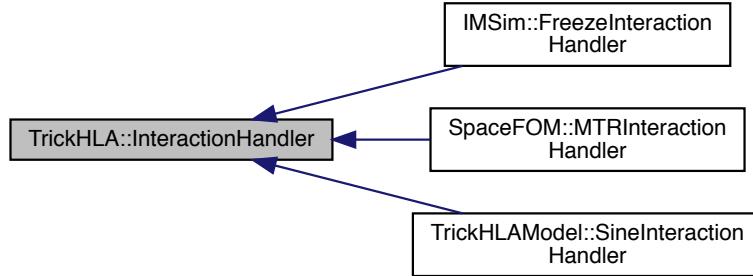
The documentation for this class was generated from the following files:

- [Interaction.hh](#)
- [Interaction.cpp](#)

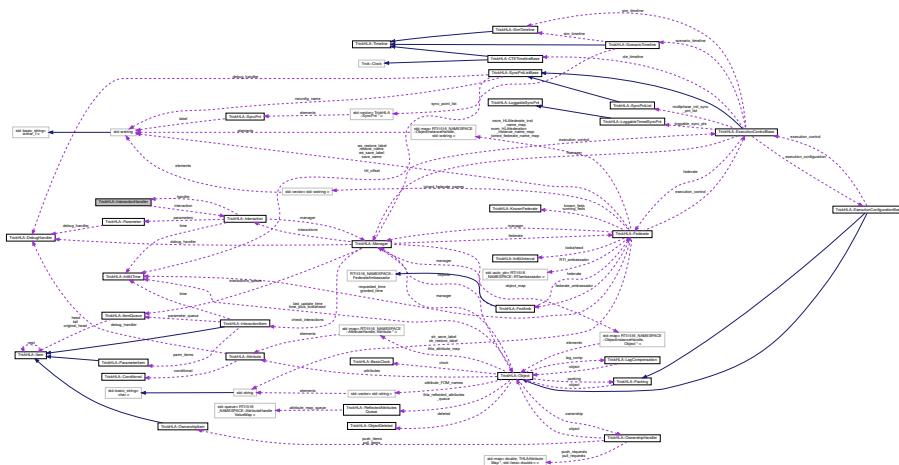
## 7.26 TrickHLA::InteractionHandler Class Reference

```
#include <InteractionHandler.hh>
```

## Inheritance diagram for TrickHLA::InteractionHandler:



## Collaboration diagram for TrickHLA::InteractionHandler:



## Public Member Functions

- **InteractionHandler ()**  
*Default constructor for the `TrickHLA InteractionHandler` class.*
  - **virtual ~InteractionHandler ()**  
*Destructor for the `TrickHLA InteractionHandler` class.*
  - **virtual void initialize\_callback (Interaction \*inter)**  
*Initializes the callback to the interaction.*
  - **bool should\_print (const DebugLevelEnum &level, const DebugSourceEnum &code) const**  
*Determine if the verbose debug comments should be printed to the console.*
  - **bool send\_interaction ()**  
*Sends the interaction to to RTI using Receive Order.*
  - **bool send\_interaction (RTI1516\_USERDATA const &the\_user\_supplied\_tag)**  
*Sends the interaction to to RTI using Receive Order.*
  - **bool send\_interaction (double send\_HLA\_time)**

- `bool send_interaction (double send_HLA_time, RTI1516_USERDATA const &the_user_supplied_tag)`

*Sends the interaction to to RTI using Timestamp Order.*
- `Int64Interval get_fed_looking_ahead () const`

*Returns a copy of the interactions looking\_ahead time.*
- `Int64Time get_granted_fed_time () const`

*Returns a copy of the granted HLA logical time.*
- `double get_sim_time ()`

*Returns the current simulation time.*
- `double get_scenario_time ()`

*Returns the current scenario time.*
- `double get_cte_time ()`

*Returns the current Central Timing Equipment (CTE) time.*
- `virtual void receive_interaction (RTI1516_USERDATA const &the_user_supplied_tag)`

*Called when the interaction is received from the RTI.*

## Protected Attributes

- `Interaction * interaction`
- Data I/O:** \*\*  
*Pointer to the TrickHLA interaction.*

## Private Member Functions

- `InteractionHandler (const InteractionHandler &rhs)`  
*Copy constructor for InteractionHandler class.*
- `InteractionHandler & operator= (const InteractionHandler &rhs)`  
*Assignment operator for InteractionHandler class.*

## Friends

- class `InputProcessor`
- `void init_attrTrickHLA__InteractionHandler ()`

### 7.26.1 Detailed Description

Definition at line 56 of file InteractionHandler.hh.

### 7.26.2 Constructor & Destructor Documentation

#### 7.26.2.1 InteractionHandler() [1/2]

`InteractionHandler::InteractionHandler ( )`  
 Default constructor for the TrickHLA InteractionHandler class.

**Trick Job Class:** *initialization*

Definition at line 55 of file InteractionHandler.cpp.

### 7.26.2.2 ~InteractionHandler()

`InteractionHandler::~InteractionHandler ( ) [virtual]`  
 Destructor for the [TrickHLA InteractionHandler](#) class.

**Trick Job Class:** `shutdown`

Definition at line 63 of file `InteractionHandler.cpp`.

### 7.26.2.3 InteractionHandler() [2/2]

`TrickHLA::InteractionHandler::InteractionHandler (`  
`const InteractionHandler & rhs ) [private]`  
 Copy constructor for [InteractionHandler](#) class.

This constructor is private to prevent inadvertent copies.

## 7.26.3 Member Function Documentation

### 7.26.3.1 get\_cte\_time()

`double InteractionHandler::get_cte_time ( )`  
 Returns the current Central Timing Equipment (CTE) time.

**Returns**

Current CTE time.

Definition at line 157 of file `InteractionHandler.cpp`.

References `TrickHLA::ExecutionControlBase::does_cte_timeline_exist()`, `TrickHLA::ExecutionControlBase::get_cte_time()`, `TrickHLA::Federate::get_execution_control()`, `TrickHLA::Interaction::get_federate()`, and `interaction`.

Referenced by `SpaceFOM::MTRInteractionHandler::receive_interaction()`, and `SpaceFOM::MTRInteractionHandler::send_interaction()`.

### 7.26.3.2 get\_fed\_lookahead()

`Int64Interval InteractionHandler::get_fed_lookahead ( ) const`  
 Return a copy of the interactions lookahead time.

**Returns**

A copy of the federation lookahead time.

Definition at line 117 of file `InteractionHandler.cpp`.

References `TrickHLA::Interaction::get_fed_lookahead()`, and `interaction`.

Referenced by `TrickHLAModel::SineInteractionHandler::send_sine_interaction()`.

### 7.26.3.3 get\_granted\_fed\_time()

`Int64Time InteractionHandler::get_granted_fed_time ( ) const`  
 Return a copy of the granted HLA logical time.

**Returns**

A copy of the federation granted time.

Definition at line 128 of file InteractionHandler.cpp.

References TrickHLA::Interaction::get\_granted\_fed\_time(), interaction, and TrickHLA::MAX\_LOGICAL\_TIME\_SECO\_NDS.

Referenced by TrickHLAModel::SineInteractionHandler::send\_sine\_interaction().

**7.26.3.4 get\_scenario\_time()**

```
double InteractionHandler::get_scenario_time ( )
```

Returns the current scenario time.

**Returns**

Current scenario time.

Definition at line 148 of file InteractionHandler.cpp.

References TrickHLA::Federate::get\_execution\_control(), TrickHLA::Interaction::get\_federate(), TrickHLA::ExecutionControlBase::get\_scenario\_time(), and interaction.

Referenced by TrickHLAModel::SineInteractionHandler::receive\_interaction(), SpaceFOM::MTRInteractionHandler::receive\_interaction(), and SpaceFOM::MTRInteractionHandler::send\_interaction().

**7.26.3.5 get\_sim\_time()**

```
double InteractionHandler::get_sim_time ( )
```

Returns the current simulation time.

**Returns**

Current simulation time.

Definition at line 139 of file InteractionHandler.cpp.

References TrickHLA::Federate::get\_execution\_control(), TrickHLA::Interaction::get\_federate(), TrickHLA::ExecutionControlBase::get\_sim\_time(), and interaction.

Referenced by SpaceFOM::MTRInteractionHandler::receive\_interaction(), and SpaceFOM::MTRInteractionHandler::send\_interaction().

**7.26.3.6 initialize\_callback()**

```
void InteractionHandler::initialize_callback (
    Interaction * inter ) [virtual]
```

Initializes the callback to the interaction.

**Parameters**

<i>inter</i>	Associated interaction for this handler.
--------------	--

**Trick Job Class: initialization**

Definition at line 70 of file InteractionHandler.cpp.

References interaction.

Referenced by TrickHLA::Interaction::initialize().

### 7.26.3.7 operator=( )

```
InteractionHandler& TrickHLA::InteractionHandler::operator= (
    const InteractionHandler & rhs ) [private]
```

Assignment operator for `InteractionHandler` class.

This assignment operator is private to prevent inadvertent copies.

### 7.26.3.8 receive\_interaction()

```
void InteractionHandler::receive_interaction (
    RTI1516_USERDATA const & the_user_supplied_tag ) [virtual]
```

Called when the interaction is received from the RTI.

#### Parameters

<code>the_user_supplied_tag</code>	Users tag.
------------------------------------	------------

Reimplemented in `IMSim::FreezeInteractionHandler`, `SpaceFOM::MTRInteractionHandler`, and `TrickHLAModel::SineInteractionHandler`.

Definition at line 110 of file `InteractionHandler.cpp`.

References `THLA_NEWSLINE`.

Referenced by `TrickHLA::Interaction::process_interaction()`.

### 7.26.3.9 send\_interaction() [1/4]

```
bool InteractionHandler::send_interaction ( )
```

Sends the interaction to to RTI using Receive Order.

#### Returns

True if the interaction was sent; False otherwise.

Definition at line 86 of file `InteractionHandler.cpp`.

References `interaction`, `RTI1516_USERDATA`, and `TrickHLA::Interaction::send()`.

Referenced by `SpaceFOM::MTRInteractionHandler::send_interaction()`.

### 7.26.3.10 send\_interaction() [2/4]

```
bool InteractionHandler::send_interaction (
    double send_HLA_time )
```

Sends the interaction to to RTI using Timestamp Order.

#### Returns

True if the interaction was sent; False otherwise.

#### Parameters

<code>send_HLA_time</code>	User specified HLA logical time to send the interaction.
----------------------------	--

Definition at line 97 of file `InteractionHandler.cpp`.

References `interaction`, `RTI1516_USERDATA`, and `TrickHLA::Interaction::send()`.

**7.26.3.11 send\_interaction() [3/4]**

```
bool InteractionHandler::send_interaction (
    double send_HLA_time,
    RTI1516_USERDATA const & the_user_supplied_tag )
```

Sends the interaction to to RTI using Timestamp Order.

**Returns**

True if the interaction was sent; False otherwise.

**Parameters**

<i>send_HLA_time</i>	User specified HLA logical time to send the interaction.
<i>the_user_supplied_tag</i>	Users tag.

Definition at line 103 of file InteractionHandler.cpp.

References [interaction](#), and [TrickHLA::Interaction::send\(\)](#).

**7.26.3.12 send\_interaction() [4/4]**

```
bool InteractionHandler::send_interaction (
    RTI1516_USERDATA const & the_user_supplied_tag )
```

Sends the interaction to to RTI using Receive Order.

**Returns**

True if the interaction was sent; False otherwise.

**Parameters**

<i>the_user_supplied_tag</i>	Users tag.
------------------------------	------------

Definition at line 91 of file InteractionHandler.cpp.

References [interaction](#), and [TrickHLA::Interaction::send\(\)](#).

**7.26.3.13 should\_print()**

```
bool InteractionHandler::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const
```

Determine if the verbose debug comments should be printed to the console.

**Returns**

Returns true if the requested message should print level.

**Parameters**

<i>level</i>	Debug level of the incoming message.
<i>code</i>	Source code association of the incoming messages.

Definition at line 76 of file InteractionHandler.cpp.

References interaction, and TrickHLA::Interaction::should\_print().

Referenced by TrickHLAModel::SineInteractionHandler::receive\_interaction(), SpaceFOM::MTRInteractionHandler::receive\_interaction(), SpaceFOM::MTRInteractionHandler::send\_interaction(), IMSim::FreezeInteractionHandler::send\_scenario\_freeze\_interaction(), and TrickHLAModel::SineInteractionHandler::send\_sine\_interaction().

## 7.26.4 Friends And Related Function Documentation

### 7.26.4.1 init\_attrTrickHLA\_\_InteractionHandler

```
void init_attrTrickHLA__InteractionHandler ( ) [friend]
```

### 7.26.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 63 of file InteractionHandler.hh.

## 7.26.5 Field Documentation

### 7.26.5.1 interaction

```
Interaction* TrickHLA::InteractionHandler::interaction [protected]
```

**Data I/O: \*\***

Pointer to the [TrickHLA](#) interaction.

Definition at line 136 of file InteractionHandler.hh.

Referenced by get\_cte\_time(), get\_fed\_lookahead(), get\_granted\_fed\_time(), get\_scenario\_time(), get\_sim\_time(), initialize\_callback(), IMSim::FreezeInteractionHandler::receive\_interaction(), SpaceFOM::MTRInteractionHandler::receive\_interaction(), SpaceFOM::MTRInteractionHandler::send\_interaction(), send\_interaction(), IMSim::FreezeInteractionHandler::send\_scenario\_freeze\_interaction(), and should\_print().

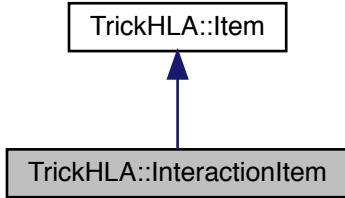
The documentation for this class was generated from the following files:

- [InteractionHandler.hh](#)
- [InteractionHandler.cpp](#)

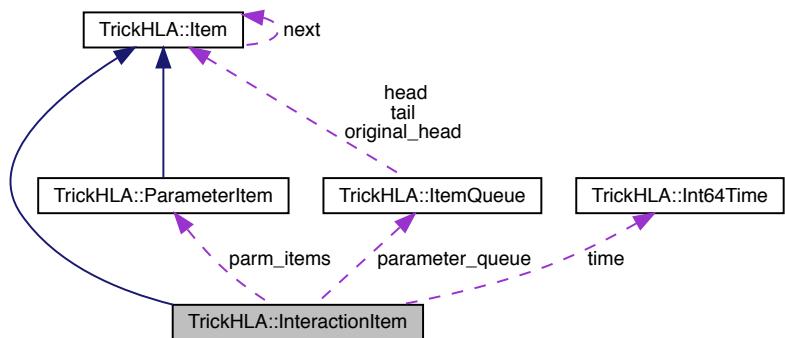
## 7.27 TrickHLA::InteractionItem Class Reference

```
#include <InteractionItem.hh>
```

Inheritance diagram for TrickHLA::InteractionItem:



Collaboration diagram for TrickHLA::InteractionItem:



## Public Member Functions

- [InteractionItem \(\)](#)  
*Default constructor for the `TrickHLA::InteractionItem` class.*
- [InteractionItem \(int interaction\\_index, int interaction\\_type, int param\\_count, Parameter \\*parameters, RTI1516\\_\\_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516\\_USERDATA const &theUserSuppliedTag\)](#)  
*Initialization constructor for the `TrickHLA::InteractionItem` class.*
- [InteractionItem \(int interaction\\_index, int interaction\\_type, int param\\_count, Parameter \\*parameters, RTI1516\\_\\_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516\\_USERDATA const &theUserSuppliedTag, RTI1516\\_NAMESPACE::LogicalTime const &theTime\)](#)  
*Initialization constructor for the `TrickHLA::InteractionItem` class.*
- [virtual ~InteractionItem \(\)](#)  
*Destructor for the `TrickHLA::InteractionItem` class.*
- [void checkpoint\\_queue \(\)](#)  
*Decode all the parameter\_queue values into parm\_items linear array.*

- void `clear_parm_items ()`  
*removes all the parm\_items values.*
- void `restore_queue ()`  
*Encode all the parm\_items values into this `InteractionItem`.*
- bool `is_timestamp_order () const`  
*Query if this `InteractionItem` is sent TimeStamp Order (TSO).*
- bool `is_receive_order () const`  
*Query if this `InteractionItem` is sent Receive Order (RO).*

## Data Fields

- int `index`  
**Units:** –  
*Index to the applicable `Interaction`.*
- `ItemQueue parameter_queue`  
**Data I/O:** \*\*  
*Linked list queue of parameter items.*
- int `interaction_type`  
**Units:** –  
*type of the containing interaction*
- int `parm_items_count`  
**Units:** –  
*Number of array elements*
- `ParameterItem * parm_items`  
**Units:** –  
*checkpoint-able parameter items array*
- `size_t user_supplied_tag_size`  
**Units:** –  
*Number of bytes in the user supplied tag.*
- `unsigned char * user_supplied_tag`  
**Units:** –  
*User supplied tag data.*
- bool `order_is_TSO`  
**Units:** –  
*True if Timestamp Order, false for Receive Order.*
- `Int64Time time`  
**Units:** –  
*Time associated with TSO interaction.*

## Private Member Functions

- void `initialize (int inter_type, int param_count, Parameter *parameters, RTI1516_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516_NAMESPACE::Userdata const &theUserSuppliedTag)`  
*Decode the `Interaction` values into this `Item`.*
- `InteractionItem (const InteractionItem &rhs)`  
*Copy constructor for `InteractionItem` class.*
- `InteractionItem & operator= (const InteractionItem &rhs)`  
*Assignment operator for `InteractionItem` class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_\\_InteractionItem \(\)](#)

### 7.27.1 Detailed Description

Definition at line 62 of file [InteractionItem.hh](#).

### 7.27.2 Constructor & Destructor Documentation

#### 7.27.2.1 [InteractionItem\(\)](#) [1/4]

`InteractionItem::InteractionItem ( )`  
Default constructor for the [TrickHLA InteractionItem](#) class.

**Trick Job Class:** *initialization*

Definition at line 68 of file [InteractionItem.cpp](#).

#### 7.27.2.2 [InteractionItem\(\)](#) [2/4]

```
InteractionItem::InteractionItem (
    int interaction_index,
    int interaction_type,
    int param_count,
    Parameter * parameters,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_NAMESPACE::Userdata const & theUserSuppliedTag )
```

Initialization constructor for the [TrickHLA InteractionItem](#) class.

#### Parameters

<code>interaction_index</code>	<a href="#">Interaction</a> index.
<code>interaction_type</code>	Type of the containing interaction.
<code>param_count</code>	Number of parameters.
<code>parameters</code>	<a href="#">Interaction</a> Parameters.
<code>theParameterValues</code>	<a href="#">Parameter</a> values.
<code>theUserSuppliedTag</code>	User supplied tag.

**Trick Job Class:** *initialization*

Definition at line 84 of file [InteractionItem.cpp](#).

References [initialize\(\)](#), and `interaction_type`.

#### 7.27.2.3 [InteractionItem\(\)](#) [3/4]

```
InteractionItem::InteractionItem (
    int interaction_index,
    int interaction_type,
    int param_count,
    Parameter * parameters,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
```

```
RTI1516_USERDATA const & theUserSuppliedTag,
RTI1516_NAMESPACE::LogicalTime const & theTime )
```

Initialization constructor for the [TrickHLA InteractionItem](#) class.

#### Parameters

<i>interaction_index</i>	<a href="#">Interaction</a> index.
<i>interaction_type</i>	Type of the containing interaction.
<i>param_count</i>	Number of parameters.
<i>parameters</i>	<a href="#">Interaction Parameters</a> .
<i>theParameterValues</i>	Parameter values.
<i>theUserSuppliedTag</i>	User supplied tag.
<i>theTime</i>	Time for TSO interaction.

#### Trick Job Class: *initialization*

Definition at line 106 of file [InteractionItem.cpp](#).

References [initialize\(\)](#), [interaction\\_type](#), [TrickHLA::Int64Time::setTo\(\)](#), and [time](#).

#### 7.27.2.4 [~InteractionItem\(\)](#)

```
InteractionItem::~InteractionItem ( ) [virtual]
```

Destructor for the [TrickHLA InteractionItem](#) class.

#### Trick Job Class: *shutdown*

Definition at line 131 of file [InteractionItem.cpp](#).

References [clear\\_parm\\_items\(\)](#), [user\\_supplied\\_tag](#), and [user\\_supplied\\_tag\\_size](#).

#### 7.27.2.5 [InteractionItem\(\)](#) [4/4]

```
TrickHLA::InteractionItem::InteractionItem (
    const InteractionItem & rhs ) [private]
```

Copy constructor for [InteractionItem](#) class.

This constructor is private to prevent inadvertent copies.

### 7.27.3 Member Function Documentation

#### 7.27.3.1 [checkpoint\\_queue\(\)](#)

```
void InteractionItem::checkpoint_queue ( )
```

Decode all the [parameter\\_queue](#) values into [parm\\_items](#) linear array.

Definition at line 187 of file [InteractionItem.cpp](#).

References [TrickHLA::ParameterItem::data](#), [TrickHLA::ItemQueue::front\(\)](#), [TrickHLA::ParameterItem::index](#), [TrickHLA::ItemQueue::lock\(\)](#), [TrickHLA::ItemQueue::next\(\)](#), [parameter\\_queue](#), [parm\\_items](#), [parm\\_items\\_count](#), [TrickHLA::ItemQueue::rewind\(\)](#), [TrickHLA::ParameterItem::size](#), [TrickHLA::ItemQueue::size\(\)](#), [THLA\\_ENDL](#), and [TrickHLA::ItemQueue::unlock\(\)](#).

Referenced by [TrickHLA::Manager::setup\\_checkpoint\\_interactions\(\)](#).

#### 7.27.3.2 [clear\\_parm\\_items\(\)](#)

```
void InteractionItem::clear_parm_items ( )
```

removes all the parm\_items values.

Definition at line 229 of file InteractionItem.cpp.

References TrickHLA::ParameterItem::clear(), parm\_items, and parm\_items\_count.

Referenced by TrickHLA::Manager::clear\_interactions(), and ~InteractionItem().

### 7.27.3.3 initialize()

```
void InteractionItem::initialize (
    int inter_type,
    int param_count,
    Parameter * parameters,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_USERDATA const & theUserSuppliedTag ) [private]
```

Decode the [Interaction](#) values into this [Item](#).

#### Parameters

<i>inter_type</i>	Type of the containing interaction.
<i>param_count</i>	Number of parameters.
<i>parameters</i>	<a href="#">Interaction</a> Parameters.
<i>theParameterValues</i>	<a href="#">Parameter</a> values.
<i>theUserSuppliedTag</i>	User supplied tag.

#### Trick Job Class: initialization

Definition at line 146 of file InteractionItem.cpp.

References interaction\_type, parameter\_queue, TrickHLA::ItemQueue::push(), user\_supplied\_tag, and user\_supplied←\_tag\_size.

Referenced by InteractionItem().

### 7.27.3.4 is\_receive\_order()

```
bool TrickHLA::InteractionItem::is_receive_order ( ) const [inline]
```

Query if this [InteractionItem](#) is sent Receive Order (RO).

#### Returns

True if sent Receive Order; False otherwise.

Definition at line 152 of file InteractionItem.hh.

References order\_is\_TSO.

### 7.27.3.5 is\_timestamp\_order()

```
bool TrickHLA::InteractionItem::is_timestamp_order ( ) const [inline]
```

Query if this [InteractionItem](#) is sent TimeStamp Order (TSO).

#### Returns

True if sent TimeStamp Order; False otherwise.

Definition at line 148 of file InteractionItem.hh.

References order\_is\_TSO.

Referenced by TrickHLA::Interaction::extract\_data().

### 7.27.3.6 operator=( )

```
InteractionItem& TrickHLA::InteractionItem::operator= (
    const InteractionItem & rhs ) [private]
```

Assignment operator for [InteractionItem](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.27.3.7 restore\_queue( )

```
void InteractionItem::restore_queue ( )
```

Encode all the `parm_items` values into this [InteractionItem](#).

Definition at line 241 of file `InteractionItem.cpp`.

References `TrickHLA::ParameterItem::data`, `TrickHLA::ParameterItem::index`, `parameter_queue`, `parm_items`, `parm_items_count`, `TrickHLA::ItemQueue::push()`, and `TrickHLA::ParameterItem::size()`.

Referenced by `TrickHLA::Manager::restore_interactions()`.

## 7.27.4 Friends And Related Function Documentation

### 7.27.4.1 init\_attrTrickHLA\_\_InteractionItem

```
void init_attrTrickHLA__InteractionItem ( ) [friend]
```

### 7.27.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 69 of file `InteractionItem.hh`.

## 7.27.5 Field Documentation

### 7.27.5.1 index

```
int TrickHLA::InteractionItem::index
```

**Units:** –

Index to the applicable [Interaction](#).

Definition at line 78 of file `InteractionItem.hh`.

Referenced by `TrickHLA::Manager::dump_interactions()`, `TrickHLA::Manager::process_interactions()`, `TrickHLA::Manager::restore_interactions()`, and `TrickHLA::Manager::setup_checkpoint_interactions()`.

### 7.27.5.2 interaction\_type

```
int TrickHLA::InteractionItem::interaction_type
```

**Units:** –

type of the containing interaction

Definition at line 82 of file `InteractionItem.hh`.

Referenced by `TrickHLA::Manager::dump_interactions()`, `initialize()`, `InteractionItem()`, `TrickHLA::Manager::process_interactions()`, `TrickHLA::Manager::restore_interactions()`, and `TrickHLA::Manager::setup_checkpoint_interactions()`.

### 7.27.5.3 order\_is\_TSO

`bool TrickHLA::InteractionItem::order_is_TSO`

**Units:** –

True if Timestamp Order, false for Receive Order.

Definition at line 90 of file InteractionItem.hh.

Referenced by `TrickHLA::Manager::dump_interactions()`, `is_receive_order()`, `is_timestamp_order()`, `TrickHLA::Manager::restore_interactions()`, and `TrickHLA::Manager::setup_checkpoint_interactions()`.

### 7.27.5.4 parameter\_queue

`ItemQueue TrickHLA::InteractionItem::parameter_queue`

**Data I/O:** \*\*

Linked list queue of parameter items.

Definition at line 80 of file InteractionItem.hh.

Referenced by `checkpoint_queue()`, `TrickHLA::Interaction::extract_data()`, `initialize()`, and `restore_queue()`.

### 7.27.5.5 parm\_items

`ParameterItem* TrickHLA::InteractionItem::parm_items`

**Units:** –

checkpoint-able parameter items array

Definition at line 85 of file InteractionItem.hh.

Referenced by `checkpoint_queue()`, `clear_parm_items()`, `TrickHLA::Manager::dump_interactions()`, `TrickHLA::Manager::restore_interactions()`, `restore_queue()`, and `TrickHLA::Manager::setup_checkpoint_interactions()`.

### 7.27.5.6 parm\_items\_count

`int TrickHLA::InteractionItem::parm_items_count`

**Units:** –

Number of array elements

Definition at line 84 of file InteractionItem.hh.

Referenced by `checkpoint_queue()`, `clear_parm_items()`, `TrickHLA::Manager::dump_interactions()`, `TrickHLA::Manager::restore_interactions()`, `restore_queue()`, and `TrickHLA::Manager::setup_checkpoint_interactions()`.

### 7.27.5.7 time

`Int64Time TrickHLA::InteractionItem::time`

**Units:** –

Time associated with TSO interaction.

Definition at line 91 of file InteractionItem.hh.

Referenced by `TrickHLA::Manager::dump_interactions()`, `TrickHLA::Interaction::extract_data()`, `InteractionItem()`, `TrickHLA::Manager::restore_interactions()`, and `TrickHLA::Manager::setup_checkpoint_interactions()`.

### 7.27.5.8 user\_supplied\_tag

`unsigned char* TrickHLA::InteractionItem::user_supplied_tag`

**Units:** –

User supplied tag data.

Definition at line 88 of file InteractionItem.hh.

Referenced by TrickHLA::Interaction::extract\_data(), initialize(), TrickHLA::Manager::restore\_interactions(), TrickHLA::Manager::setup\_checkpoint\_interactions(), and ~InteractionItem().

#### 7.27.5.9 user\_supplied\_tag\_size

size\_t TrickHLA::InteractionItem::user\_supplied\_tag\_size

**Units:** –

Number of bytes in the user supplied tag.

Definition at line 87 of file InteractionItem.hh.

Referenced by TrickHLA::Manager::dump\_interactions(), TrickHLA::Interaction::extract\_data(), initialize(), TrickHLA::Manager::restore\_interactions(), TrickHLA::Manager::setup\_checkpoint\_interactions(), and ~InteractionItem().

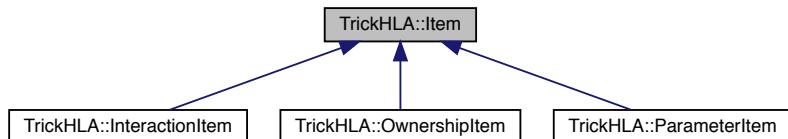
The documentation for this class was generated from the following files:

- [InteractionItem.hh](#)
- [InteractionItem.cpp](#)

## 7.28 TrickHLA::Item Class Reference

#include <Item.hh>

Inheritance diagram for TrickHLA::Item:



Collaboration diagram for TrickHLA::Item:



### Public Member Functions

- [Item \(\)](#)

*Default constructor for the [TrickHLA Item](#) class.*

- [virtual ~Item \(\)](#)

*Destructor for the [TrickHLA Item](#) class.*

## Data Fields

- `Item * next`

**Units:** –

*Next item in linked-list.*

## Private Member Functions

- `Item (const Item &rhs)`  
*Copy constructor for `Item` class.*
- `Item & operator= (const Item &rhs)`  
*Assignment operator for `Item` class.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA__Item ()`

### 7.28.1 Detailed Description

Definition at line 39 of file Item.hh.

### 7.28.2 Constructor & Destructor Documentation

#### 7.28.2.1 Item() [1/2]

`Item::Item ( )`

Default constructor for the `TrickHLA Item` class.

**Trick Job Class:** *initialization*

Definition at line 40 of file Item.cpp.

#### 7.28.2.2 ~Item()

`Item::~Item ( ) [virtual]`

Destructor for the `TrickHLA Item` class.

**Trick Job Class:** *shutdown*

Definition at line 48 of file Item.cpp.

#### 7.28.2.3 Item() [2/2]

`TrickHLA::Item::Item (`  
    `const Item & rhs ) [private]`

Copy constructor for `Item` class.

This constructor is private to prevent inadvertent copies.

### 7.28.3 Member Function Documentation

### 7.28.3.1 `operator=()`

```
Item& TrickHLA::Item::operator= (
    const Item & rhs ) [private]
```

Assignment operator for `Item` class.

This assignment operator is private to prevent inadvertent copies.

## 7.28.4 Friends And Related Function Documentation

### 7.28.4.1 `init_attrTrickHLA__Item`

```
void init_attrTrickHLA__Item ( ) [friend]
```

### 7.28.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 46 of file `Item.hh`.

## 7.28.5 Field Documentation

### 7.28.5.1 `next`

```
Item* TrickHLA::Item::next
```

**Units:** –

Next item in linked-list.

Definition at line 52 of file `Item.hh`.

Referenced by `TrickHLA::ItemQueue::dump_head_pointers()`, `TrickHLA::ItemQueue::next()`, `TrickHLA::ItemQueue::pop()`, and `TrickHLA::ItemQueue::push()`.

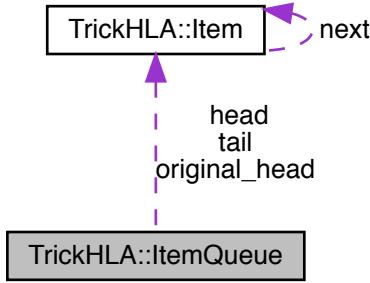
The documentation for this class was generated from the following files:

- [Item.hh](#)
- [Item.cpp](#)

## 7.29 `TrickHLA::ItemQueue` Class Reference

```
#include <ItemQueue.hh>
```

Collaboration diagram for TrickHLA::ItemQueue:



## Public Member Functions

- `ItemQueue ()`  
*Default constructor for the `TrickHLA ItemQueue` class.*
- `virtual ~ItemQueue ()`  
*Destructor for the `TrickHLA ItemQueue` class.*
- `bool empty () const`  
*Query if the item queue is empty.*
- `Item * front ()`  
*Gets the front item on the item queue.*
- `void dump_head_pointers (const char *name)`  
*Prints the 'head' pointers for all elements in the queue.*
- `void next (Item *item)`  
*Sets head to the passed-in element's next value.*
- `void pop ()`  
*Pop an item off the queue.*
- `void push (Item *item)`  
*Push the item onto the queue.*
- `void rewind ()`  
*Re-established original 'head' queue pointer after the queue has been walked.*
- `int size () const`  
*Get the size of the item queue.*
- `void lock ()`  
*Lock the thread mutex.*
- `void unlock ()`  
*Unlock the thread mutex.*

## Private Member Functions

- `ItemQueue (const ItemQueue &rhs)`  
*Copy constructor for `ItemQueue` class.*
- `ItemQueue & operator= (const ItemQueue &rhs)`  
*Assignment operator for `ItemQueue` class.*

## Private Attributes

- `pthread_mutex_t mutex`  
**Data I/O:** \*\*  
*Mutex to lock thread over critical code sections.*
- `int count`  
**Units:** count  
*Number of elements in the queue.*
- `Item * head`  
**Units:** –  
*First item in linked-list queue.*
- `Item * tail`  
**Units:** –  
*Last item in linked-list queue.*
- `Item * original_head`  
**Units:** –  
*copy of the original head of queue*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA__ItemQueue ()`

### 7.29.1 Detailed Description

Definition at line 46 of file `ItemQueue.hh`.

### 7.29.2 Constructor & Destructor Documentation

#### 7.29.2.1 `ItemQueue()` [1/2]

`ItemQueue::ItemQueue ( )`  
 Default constructor for the `TrickHLA ItemQueue` class.

**Trick Job Class:** *initialization*  
 Definition at line 47 of file `ItemQueue.cpp`.  
 References `mutex`.

#### 7.29.2.2 `~ItemQueue()`

`ItemQueue::~ItemQueue ( ) [virtual]`  
 Destructor for the `TrickHLA ItemQueue` class.  
**Trick Job Class:** *shutdown*  
 Definition at line 60 of file `ItemQueue.cpp`.

References empty(), mutex, and pop().

### 7.29.2.3 ItemQueue() [2/2]

```
TrickHLA::ItemQueue::ItemQueue (
    const ItemQueue & rhs ) [private]
```

Copy constructor for [ItemQueue](#) class.

This constructor is private to prevent inadvertent copies.

## 7.29.3 Member Function Documentation

### 7.29.3.1 dump\_head\_pointers()

```
void ItemQueue::dump_head_pointers (
    const char * name )
```

Prints the 'head' pointers for all elements in the queue.

Parameters

<i>name</i>	Name of the caller.
-------------	---------------------

**Trick Job Class:** *initialization*

Definition at line 74 of file ItemQueue.cpp.

References head, TrickHLA::Item::next, and THLA\_NEWLINE.

### 7.29.3.2 empty()

```
bool TrickHLA::ItemQueue::empty ( ) const [inline]
```

Query if the item queue is empty.

Returns

True if empty; False otherwise.

Definition at line 70 of file ItemQueue.hh.

References head.

Referenced by TrickHLA::Interaction::extract\_data(), pop(), TrickHLA::Manager::process\_interactions(), TrickHLA::Manager::setup\_checkpoint\_interactions(), and ~ItemQueue().

### 7.29.3.3 front()

```
Item* TrickHLA::ItemQueue::front ( ) [inline]
```

Gets the front item on the item queue.

Returns

The front or head item on the item queue.

Definition at line 74 of file ItemQueue.hh.

References head.

Referenced by TrickHLA::InteractionItem::checkpoint\_queue(), TrickHLA::Interaction::extract\_data(), TrickHLA::Manager::process\_interactions(), and TrickHLA::Manager::setup\_checkpoint\_interactions().

#### 7.29.3.4 lock()

```
void TrickHLA::ItemQueue::lock ( ) [inline]
```

Lock the thread mutex.

Definition at line 100 of file ItemQueue.hh.

References mutex.

Referenced by TrickHLA::InteractionItem::checkpoint\_queue(), pop(), push(), and TrickHLA::Manager::setup\_checkpoint\_interactions().

#### 7.29.3.5 next()

```
void ItemQueue::next (
    Item * item )
```

Sets head to the passed-in element's next value.

##### Parameters

<i>item</i>	<a href="#">Item</a> to extract the 'next' data pointer.
-------------	--

##### Trick Job Class: *initialization*

Definition at line 100 of file ItemQueue.cpp.

References head, TrickHLA::Item::next, and original\_head.

Referenced by TrickHLA::InteractionItem::checkpoint\_queue(), and TrickHLA::Manager::setup\_checkpoint\_interactions().

#### 7.29.3.6 operator=()

```
ItemQueue& TrickHLA::ItemQueue::operator= (
    const ItemQueue & rhs ) [private]
```

Assignment operator for [ItemQueue](#) class.

This assignment operator is private to prevent inadvertent copies.

#### 7.29.3.7 pop()

```
void ItemQueue::pop ( )
```

Pop an item off the queue.

##### Trick Job Class: *initialization*

Definition at line 123 of file ItemQueue.cpp.

References count, empty(), head, lock(), TrickHLA::Item::next, tail, and unlock().

Referenced by TrickHLA::Interaction::extract\_data(), TrickHLA::Manager::process\_interactions(), and ~ItemQueue().

#### 7.29.3.8 push()

```
void ItemQueue::push (
    Item * item )
```

Push the item onto the queue.

##### Parameters

<i>item</i>	<a href="#">Item</a> to put into the queue.
-------------	---

##### Trick Job Class: *initialization*

Definition at line 149 of file ItemQueue.cpp.

References count, head, lock(), TrickHLA::Item::next, tail, and unlock().

Referenced by TrickHLA::InteractionItem::initialize(), TrickHLA::Manager::receive\_interaction(), TrickHLA::Manager::restore\_interactions(), and TrickHLA::InteractionItem::restore\_queue().

#### 7.29.3.9 `rewind()`

```
void ItemQueue::rewind ( )
```

Re-established original 'head' queue pointer after the queue has been walked.

**Trick Job Class:** *initialization*

Definition at line 172 of file ItemQueue.cpp.

References head, and original\_head.

Referenced by TrickHLA::InteractionItem::checkpoint\_queue(), and TrickHLA::Manager::setup\_checkpoint\_interactions().

#### 7.29.3.10 `size()`

```
int TrickHLA::ItemQueue::size ( ) const [inline]
```

Get the size of the item queue.

**Returns**

Number of elements in the queue.

Definition at line 97 of file ItemQueue.hh.

References count.

Referenced by TrickHLA::InteractionItem::checkpoint\_queue(), and TrickHLA::Manager::setup\_checkpoint\_interactions().

#### 7.29.3.11 `unlock()`

```
void TrickHLA::ItemQueue::unlock ( ) [inline]
```

Unlock the thread mutex.

Definition at line 103 of file ItemQueue.hh.

References mutex.

Referenced by TrickHLA::InteractionItem::checkpoint\_queue(), pop(), push(), and TrickHLA::Manager::setup\_checkpoint\_interactions().

### 7.29.4 Friends And Related Function Documentation

#### 7.29.4.1 `init_attrTrickHLA__ItemQueue`

```
void init_attrTrickHLA__ItemQueue ( ) [friend]
```

#### 7.29.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 53 of file ItemQueue.hh.

## 7.29.5 Field Documentation

### 7.29.5.1 count

```
int TrickHLA::ItemQueue::count [private]
```

**Units:** count

Number of elements in the queue.

Definition at line 108 of file ItemQueue.hh.

Referenced by pop(), push(), and size().

### 7.29.5.2 head

```
Item* TrickHLA::ItemQueue::head [private]
```

**Units:** –

First item in linked-list queue.

Definition at line 110 of file ItemQueue.hh.

Referenced by dump\_head\_pointers(), empty(), front(), next(), pop(), push(), and rewind().

### 7.29.5.3 mutex

```
pthread_mutex_t TrickHLA::ItemQueue::mutex [private]
```

**Data I/O:** \*\*

Mutex to lock thread over critical code sections.

Definition at line 106 of file ItemQueue.hh.

Referenced by ItemQueue(), lock(), unlock(), and ~ItemQueue().

### 7.29.5.4 original\_head

```
Item* TrickHLA::ItemQueue::original_head [private]
```

**Units:** –

copy of the original head of queue

Definition at line 113 of file ItemQueue.hh.

Referenced by next(), and rewind().

### 7.29.5.5 tail

```
Item* TrickHLA::ItemQueue::tail [private]
```

**Units:** –

Last item in linked-list queue.

Definition at line 111 of file ItemQueue.hh.

Referenced by pop(), and push().

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

- [ItemQueue.hh](#)
- [ItemQueue.cpp](#)

## 7.30 TrickHLA::KnownFederate Class Reference

```
#include <KnownFederate.hh>
```

## Public Member Functions

- [KnownFederate \(\)](#)  
*Default constructor for the TrickHLA KnownFederate class.*
- [~KnownFederate \(\)](#)  
*Destructor for the TrickHLA KnownFederate class.*

## Data Fields

- [char \\* MOM\\_instance\\_name](#)  
**Units:** –  
*MOM instance name for the federate object.*
- [char \\* name](#)  
**Units:** –  
*Name of a Federate in the Federation.*
- [bool required](#)  
**Units:** –  
*True requires federate to be in federation before continuing.*

## Private Member Functions

- [KnownFederate \(const KnownFederate &rhs\)](#)  
*Copy constructor for KnownFederate class.*
- [KnownFederate & operator= \(const KnownFederate &rhs\)](#)  
*Assignment operator for KnownFederate class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_KnownFederate \(\)](#)

### 7.30.1 Detailed Description

Definition at line 36 of file KnownFederate.hh.

### 7.30.2 Constructor & Destructor Documentation

#### 7.30.2.1 KnownFederate() [1/2]

```
TrickHLA::KnownFederate::KnownFederate ( ) [inline]
Default constructor for the TrickHLA KnownFederate class.
Definition at line 60 of file KnownFederate.hh.
```

#### 7.30.2.2 ~KnownFederate()

```
TrickHLA::KnownFederate::~KnownFederate ( ) [inline]
Destructor for the TrickHLA KnownFederate class.
Definition at line 62 of file KnownFederate.hh.
```

### 7.30.2.3 KnownFederate() [2/2]

```
TrickHLA::KnownFederate::KnownFederate (
    const KnownFederate & rhs ) [private]
```

Copy constructor for [KnownFederate](#) class.

This constructor is private to prevent inadvertent copies.

## 7.30.3 Member Function Documentation

### 7.30.3.1 operator=()

```
KnownFederate& TrickHLA::KnownFederate::operator= (
    const KnownFederate & rhs ) [private]
```

Assignment operator for [KnownFederate](#) class.

This assignment operator is private to prevent inadvertent copies.

## 7.30.4 Friends And Related Function Documentation

### 7.30.4.1 init\_attrTrickHLA\_KnownFederate

```
void init_attrTrickHLA_KnownFederate ( ) [friend]
```

### 7.30.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 43 of file [KnownFederate.hh](#).

## 7.30.5 Field Documentation

### 7.30.5.1 MOM\_instance\_name

```
char* TrickHLA::KnownFederate::MOM_instance_name
```

#### Units: -

MOM instance name for the federate object.

Definition at line 49 of file [KnownFederate.hh](#).

Referenced by [TrickHLA::Federate::add\\_a\\_single\\_entry\\_into\\_running\\_feds\(\)](#), [TrickHLA::Federate::clear\\_running\\_feds\(\)](#), [TrickHLA::Federate::copy\\_running\\_feds\\_into\\_known\\_feds\(\)](#), [TrickHLA::Federate::determine\\_federate\\_MO\\_M\\_object\\_instance\\_names\(\)](#), [TrickHLA::Federate::remove\\_MOM\\_HLAfederate\\_instance\\_id\(\)](#), [TrickHLA::Federate::update\\_running\\_feds\(\)](#), and [TrickHLA::Federate::~Federate\(\)](#).

### 7.30.5.2 name

```
char* TrickHLA::KnownFederate::name
```

#### Units: -

Name of a [Federate](#) in the Federation.

Definition at line 55 of file [KnownFederate.hh](#).

Referenced by TrickHLA::Federate::add\_a\_single\_entry\_into\_running\_feds(), TrickHLA::Federate::clear\_running\_feds(), TrickHLA::ExecutionConfiguration::configure(), TrickHLA::Federate::copy\_running\_feds\_into\_known\_feds(), TrickHLAModel::SimpleSimConfig::initialize(), TrickHLA::Federate::load\_and\_print\_running\_federate\_names(), TrickHLA::Federate::remove\_MOM\_HLAfederate\_instance\_id(), TrickHLA::Federate::set\_all\_federate\_MOM\_instance\_handles\_by\_name(), TrickHLA::Federate::update\_running\_feds(), TrickHLA::Federate::wait\_for\_required\_federates\_to\_join(), and TrickHLA::Federate::~Federate().

### 7.30.5.3 required

```
bool TrickHLA::KnownFederate::required
```

**Units:** –

True requires federate to be in federation before continuing.

Definition at line 56 of file KnownFederate.hh.

Referenced by TrickHLA::Federate::add\_a\_single\_entry\_into\_running\_feds(), TrickHLA::ExecutionConfiguration::configure(), TrickHLA::Federate::copy\_running\_feds\_into\_known\_feds(), TrickHLA::Federate::remove\_MOM\_HLAfederate\_instance\_id(), and TrickHLA::Federate::update\_running\_feds().

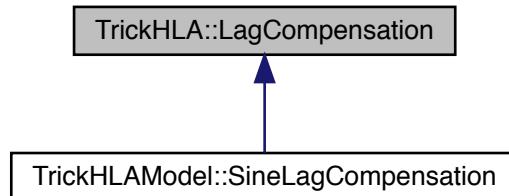
The documentation for this class was generated from the following file:

- [KnownFederate.hh](#)

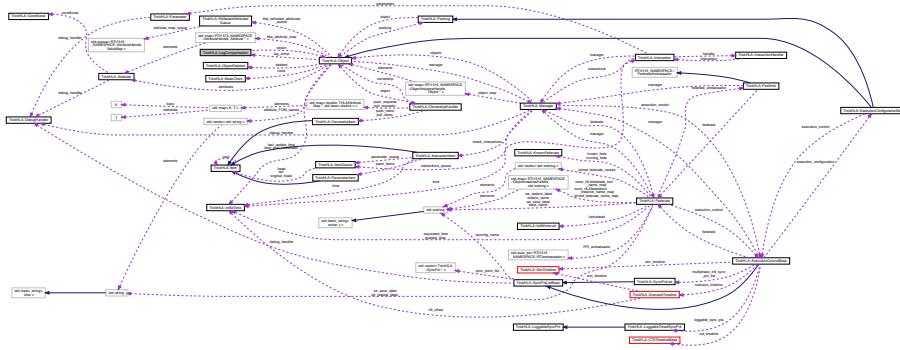
## 7.31 TrickHLA::LagCompensation Class Reference

```
#include <LagCompensation.hh>
```

Inheritance diagram for TrickHLA::LagCompensation:



Collaboration diagram for TrickHLA::LagCompensation:



## Public Member Functions

- `LagCompensation ()`  
*Default constructor for the [TrickHLA LagCompensation](#) class.*
- `virtual ~LagCompensation ()`  
*Destructor for the [TrickHLA LagCompensation](#) class.*
- `bool should_print (const DebugLevelEnum &level, const DebugSourceEnum &code) const`  
*Determine if the verbose debug comments should be printed to the console.*
- `Attribute * get_attribute (const char *attr_FOM_name)`  
*Get the [Attribute](#) by FOM name.*
- `Attribute * get_attribute_and_validate (const char *attr_FOM_name)`  
*Get the [Attribute](#) for the given attribute FOM name and validate that it exists.*
- `Int64Interval get_fed_lookahead () const`  
*Returns a copy of the object's lookahead time.*
- `Int64Time get_granted_fed_time () const`  
*Returns a copy of the object's granted federation time.*
- `double get_scenario_time ()`  
*Returns the current scenario time.*
- `double get_cte_time ()`  
*Returns the current Central Timing Equipment (CTE) time.*
- `virtual void initialize_callback (Object *obj)`  
*Initialize the callback object to the supplied [Object](#) pointer.*
- `virtual void send_lag_compensation ()`  
*Send side lag compensation callback.*
- `virtual void receive_lag_compensation ()`  
*Receive side lag compensation callback.*

## Protected Attributes

- `Object * object`  
**Data I/O: \*\***  
*Object associated with this lag-comp class.*

## Private Member Functions

- [LagCompensation \(const LagCompensation &rhs\)](#)  
*Copy constructor for LagCompensation class.*
- [LagCompensation & operator= \(const LagCompensation &rhs\)](#)  
*Assignment operator for LagCompensation class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_LagCompensation \(\)](#)

### 7.31.1 Detailed Description

Definition at line 53 of file LagCompensation.hh.

### 7.31.2 Constructor & Destructor Documentation

#### 7.31.2.1 LagCompensation() [1/2]

`TrickHLA::LagCompensation::LagCompensation ( ) [inline]`

Default constructor for the [TrickHLA LagCompensation](#) class.

Definition at line 70 of file LagCompensation.hh.

#### 7.31.2.2 ~LagCompensation()

`virtual TrickHLA::LagCompensation::~LagCompensation ( ) [inline], [virtual]`

Destructor for the [TrickHLA LagCompensation](#) class.

Definition at line 72 of file LagCompensation.hh.

#### 7.31.2.3 LagCompensation() [2/2]

`TrickHLA::LagCompensation::LagCompensation (`  
    `const LagCompensation & rhs ) [private]`

Copy constructor for [LagCompensation](#) class.

This constructor is private to prevent inadvertent copies.

### 7.31.3 Member Function Documentation

#### 7.31.3.1 get\_attribute()

`Attribute * LagCompensation::get_attribute (`  
    `const char * attr_FOM_name )`

Get the [Attribute](#) by FOM name.

Returns

[Attribute](#) for the given name.

**Parameters**

<code>attr_FOM_name</code>	Attribute FOM name.
----------------------------	---------------------

Definition at line 95 of file LagCompensation.cpp.

**7.31.3.2 `get_attribute_and_validate()`**

```
Attribute * LagCompensation::get_attribute_and_validate (
    const char * attr_FOM_name )
```

Get the [Attribute](#) for the given attribute FOM name and validate that it exists.

**Returns**

[Attribute](#) for the given name.

**Parameters**

<code>attr_FOM_name</code>	Attribute FOM name.
----------------------------	---------------------

If the attribute is not found then an error message is displayed and exec-terminate is called.

Definition at line 105 of file LagCompensation.cpp.

References [THLA\\_ENDL](#).

Referenced by [TrickHLAModel::SineLagCompensation::initialize\\_callback\(\)](#).

**7.31.3.3 `get_cte_time()`**

```
double LagCompensation::get_cte_time ( )
```

Returns the current Central Timing Equipment (CTE) time.

**Returns**

Current CTE time.

Definition at line 166 of file LagCompensation.cpp.

References [TrickHLA::ExecutionControlBase::does\\_cte\\_timeline\\_exist\(\)](#), [TrickHLA::ExecutionControlBase::get\\_cte\\_time\(\)](#), [TrickHLA::Federate::get\\_execution\\_control\(\)](#), and [TrickHLA::ExecutionControlBase::get\\_federate\(\)](#).

**7.31.3.4 `get_fed_lookahead()`**

```
Int64Interval LagCompensation::get_fed_lookahead ( ) const
```

Returns a copy of the object's lookahead time.

**Returns**

A copy of the federate's lookahead time.

Definition at line 137 of file LagCompensation.cpp.

Referenced by [TrickHLAModel::SineLagCompensation::send\\_lag\\_compensation\(\)](#).

**7.31.3.5 `get_granted_fed_time()`**

```
Int64Time LagCompensation::get_granted_fed_time ( ) const
```

Returns a copy of the object's granted federation time.

**Returns**

A copy of the federate's current granted time.

Definition at line 147 of file LagCompensation.cpp.

References TrickHLA::MAX\_LOGICAL\_TIME\_SECONDS.

**7.31.3.6 get\_scenario\_time()**

```
double LagCompensation::get_scenario_time ( )
```

Returns the current scenario time.

**Returns**

Current scenario time..

Definition at line 157 of file LagCompensation.cpp.

References TrickHLA::Federate::get\_execution\_control(), TrickHLA::ExecutionControlBase::get\_federate(), and TrickHLA::ExecutionControlBase::get\_scenario\_time().

Referenced by TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), and TrickHLAModel::SineLagCompensation::send\_lag\_compensation().

**7.31.3.7 initialize\_callback()**

```
void LagCompensation::initialize_callback (
    Object * obj )  [virtual]
```

Initialize the callback object to the supplied **Object** pointer.

**Parameters**

<i>obj</i>	Associated object for this class.
------------	-----------------------------------

**Trick Job Class: initialization**

Reimplemented in [TrickHLAModel::SineLagCompensation](#).

Definition at line 57 of file LagCompensation.cpp.

Referenced by TrickHLA::Object::initialize().

**7.31.3.8 operator=()**

```
LagCompensation& TrickHLA::LagCompensation::operator= (
    const LagCompensation & rhs )  [private]
```

Assignment operator for **LagCompensation** class.

This assignment operator is private to prevent inadvertent copies.

**7.31.3.9 receive\_lag\_compensation()**

```
void LagCompensation::receive_lag_compensation ( )  [virtual]
```

Receive side lag compensation callback.

Reimplemented in [TrickHLAModel::SineLagCompensation](#).

Definition at line 84 of file LagCompensation.cpp.

References THLA\_ENDL.

Referenced by TrickHLA::Object::receive\_cyclic\_data().

### 7.31.3.10 send\_lag\_compensation()

```
void LagCompensation::send_lag_compensation ( ) [virtual]
Send side lag compensation callback.
```

Reimplemented in [TrickHLAModel::SineLagCompensation](#).

Definition at line 73 of file LagCompensation.cpp.

References THLA\_ENDL.

Referenced by [TrickHLA::Object::send\\_cyclic\\_data\(\)](#), and [TrickHLA::Object::send\\_requested\\_data\(\)](#).

### 7.31.3.11 should\_print()

```
bool LagCompensation::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const
```

Determine if the verbose debug comments should be printed to the console.

#### Returns

Returns true if the requested message should print level.

#### Parameters

<i>level</i>	Debug level of the incoming message.
<i>code</i>	Source code association of the incoming messages.

Definition at line 63 of file LagCompensation.cpp.

References [TrickHLA::Object::should\\_print\(\)](#).

Referenced by [TrickHLAModel::SineLagCompensation::receive\\_lag\\_compensation\(\)](#), and [TrickHLAModel::SineLagCompensation::send\\_lag\\_compensation\(\)](#).

## 7.31.4 Friends And Related Function Documentation

### 7.31.4.1 init\_attrTrickHLA\_\_LagCompensation

```
void init_attrTrickHLA__LagCompensation ( ) [friend]
```

### 7.31.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 60 of file LagCompensation.hh.

## 7.31.5 Field Documentation

### 7.31.5.1 object

```
Object* TrickHLA::LagCompensation::object [protected]
```

#### Data I/O: \*\*

[Object](#) associated with this lag-comp class.

Definition at line 123 of file LagCompensation.hh.

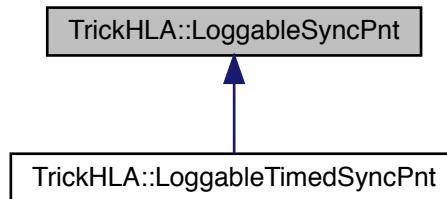
The documentation for this class was generated from the following files:

- [LagCompensation.hh](#)
- [LagCompensation.cpp](#)

## 7.32 TrickHLA::LoggableSyncPnt Class Reference

```
#include <LoggableSyncPnt.hh>
```

Inheritance diagram for TrickHLA::LoggableSyncPnt:



### Public Member Functions

- [LoggableSyncPnt \(\)](#)  
*Default constructor for the `TrickHLA LoggableSyncPnt` class.*
- [virtual ~LoggableSyncPnt \(\)](#)  
*Destructor for the `TrickHLA LoggableSyncPnt` class.*
- [virtual void clear \(\)](#)  
*Clear the Trick allocated memory.*

### Data Fields

- [char \\* label](#)  
**Units:** –  
*Sync-point name.*
- [int state](#)  
**Units:** –  
*Sync-point state.*

### Private Member Functions

- [LoggableSyncPnt \(const LoggableSyncPnt &rhs\)](#)  
*Copy constructor for `LoggableSyncPnt` class.*
- [LoggableSyncPnt & operator= \(const LoggableSyncPnt &rhs\)](#)  
*Assignment operator for `LoggableSyncPnt` class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_LoggableSyncPnt\(\)](#)

### 7.32.1 Detailed Description

Definition at line 45 of file LoggableSyncPnt.hh.

### 7.32.2 Constructor & Destructor Documentation

#### 7.32.2.1 LoggableSyncPnt() [1/2]

`TrickHLA::LoggableSyncPnt::LoggableSyncPnt ( ) [inline]`

Default constructor for the [TrickHLA LoggableSyncPnt](#) class.

Definition at line 59 of file LoggableSyncPnt.hh.

#### 7.32.2.2 ~LoggableSyncPnt()

`virtual TrickHLA::LoggableSyncPnt::~LoggableSyncPnt ( ) [inline], [virtual]`

Destructor for the [TrickHLA LoggableSyncPnt](#) class.

Definition at line 61 of file LoggableSyncPnt.hh.

References [clear\(\)](#).

#### 7.32.2.3 LoggableSyncPnt() [2/2]

`TrickHLA::LoggableSyncPnt::LoggableSyncPnt (`  
`const LoggableSyncPnt & rhs ) [private]`

Copy constructor for [LoggableSyncPnt](#) class.

This constructor is private to prevent inadvertent copies.

### 7.32.3 Member Function Documentation

#### 7.32.3.1 clear()

`virtual void TrickHLA::LoggableSyncPnt::clear ( ) [inline], [virtual]`

Clear the Trick allocated memory.

Definition at line 65 of file LoggableSyncPnt.hh.

References [label](#).

Referenced by [TrickHLA::ExecutionControlBase::~ExecutionControlBase\(\)](#), and [~LoggableSyncPnt\(\)](#).

#### 7.32.3.2 operator=()

`LoggableSyncPnt& TrickHLA::LoggableSyncPnt::operator= (`  
`const LoggableSyncPnt & rhs ) [private]`

Assignment operator for [LoggableSyncPnt](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.32.4 Friends And Related Function Documentation

#### 7.32.4.1 init\_attrTrickHLA\_\_LoggableSyncPnt

```
void init_attrTrickHLA__LoggableSyncPnt ( ) [friend]
```

#### 7.32.4.2 InputProcessor

```
friend class InputProcessor [friend]  
Definition at line 52 of file LoggableSyncPnt.hh.
```

### 7.32.5 Field Documentation

#### 7.32.5.1 label

```
char* TrickHLA::LoggableSyncPnt::label
```

**Units:** –

Sync-point name.

Definition at line 75 of file LoggableSyncPnt.hh.

Referenced by clear(), TrickHLA::TimedSyncPnt::convert(), and TrickHLA::SyncPnt::convert().

#### 7.32.5.2 state

```
int TrickHLA::LoggableSyncPnt::state
```

**Units:** –

Sync-point state.

Definition at line 76 of file LoggableSyncPnt.hh.

Referenced by TrickHLA::TimedSyncPnt::convert(), and TrickHLA::SyncPnt::convert().

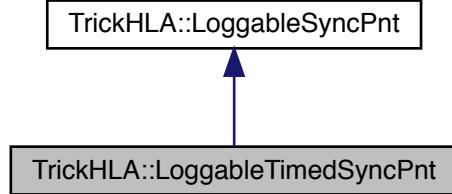
The documentation for this class was generated from the following file:

- [LoggableSyncPnt.hh](#)

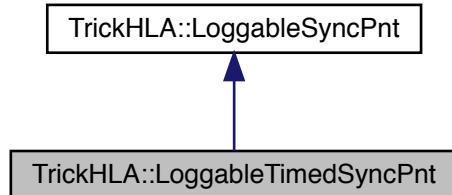
## 7.33 TrickHLA::LoggableTimedSyncPnt Class Reference

```
#include <LoggableTimedSyncPnt.hh>
```

Inheritance diagram for TrickHLA::LoggableTimedSyncPnt:



Collaboration diagram for TrickHLA::LoggableTimedSyncPnt:



## Public Member Functions

- [LoggableTimedSyncPnt \(\)](#)  
*Default constructor for the `TrickHLA LoggableTimedSyncPnt` class.*
- [~LoggableTimedSyncPnt \(\)](#)  
*Destructor for the `TrickHLA LoggableTimedSyncPnt` class.*

## Data Fields

- `int64_t time`  
**Units:** –  
*Freeze time.*

## Private Member Functions

- [LoggableTimedSyncPnt \(const LoggableTimedSyncPnt &rhs\)](#)  
*Copy constructor for `LoggableTimedSyncPnt` class.*
- [LoggableTimedSyncPnt & operator= \(const LoggableTimedSyncPnt &rhs\)](#)  
*Assignment operator for `LoggableTimedSyncPnt` class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_LoggableTimedSyncPnt\(\)](#)

### 7.33.1 Detailed Description

Definition at line 45 of file LoggableTimedSyncPnt.hh.

### 7.33.2 Constructor & Destructor Documentation

#### 7.33.2.1 LoggableTimedSyncPnt() [1/2]

`TrickHLA::LoggableTimedSyncPnt::LoggableTimedSyncPnt ( ) [inline]`

Default constructor for the [TrickHLA LoggableTimedSyncPnt](#) class.

Definition at line 59 of file LoggableTimedSyncPnt.hh.

#### 7.33.2.2 ~LoggableTimedSyncPnt()

`TrickHLA::LoggableTimedSyncPnt::~LoggableTimedSyncPnt ( ) [inline]`

Destructor for the [TrickHLA LoggableTimedSyncPnt](#) class.

Definition at line 61 of file LoggableTimedSyncPnt.hh.

#### 7.33.2.3 LoggableTimedSyncPnt() [2/2]

`TrickHLA::LoggableTimedSyncPnt::LoggableTimedSyncPnt (`  
    `const LoggableTimedSyncPnt & rhs ) [private]`

Copy constructor for [LoggableTimedSyncPnt](#) class.

This constructor is private to prevent inadvertent copies.

### 7.33.3 Member Function Documentation

#### 7.33.3.1 operator=()

`LoggableTimedSyncPnt & TrickHLA::LoggableTimedSyncPnt::operator= (`  
    `const LoggableTimedSyncPnt & rhs ) [private]`

Assignment operator for [LoggableTimedSyncPnt](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.33.4 Friends And Related Function Documentation

#### 7.33.4.1 init\_attrTrickHLA\_LoggableTimedSyncPnt

`void init_attrTrickHLA_LoggableTimedSyncPnt ( ) [friend]`

#### 7.33.4.2 InputProcessor

```
friend class InputProcessor [friend]
Definition at line 52 of file LoggableTimedSyncPnt.hh.
```

### 7.33.5 Field Documentation

### 7.33.5.1 time

```
int64_t TrickHLA::LoggableTimedSyncPnt::time
```

**Units: –**

Freeze time.

Definition at line 64 of file LoggableTimedSyncPnt.hh.

Referenced by TrickHLA::TimedSyncPnt::convert().

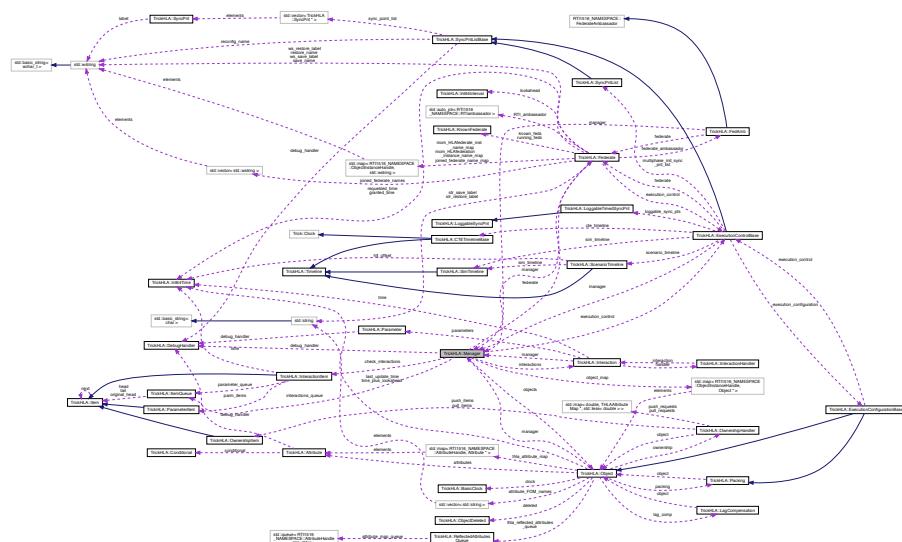
The documentation for this class was generated from the following file:

- `LoggableTimedSyncPnt.hh`

## 7.34 TrickHLA::Manager Class Reference

```
#include <Manager.hh>
```

## Collaboration diagram for TrickHLA::Manager::



## Public Member Functions

- Manager ()

*Default constructor for the [TrickHLA Manager](#) class.*

- virtual ~Manager ()

*Destructor for the [TrickHLA Manager](#) class.*

- void `setup (Federate &federate, ExecutionControlBase &execution control)`

*Setup the required class instance associations.*

- void [initialize \(\)](#)  
*Initializes the federate using the a multiphase initialization scheme, which must occur after the [Federate](#) and [FedAmb](#) have been initialized.*
- void [shutdown \(\)](#)  
*Shutdown the federate by shutting down time management, resigning from the federation, and then attempt to destroy the federation.*
- bool [is\\_shutdown \(\) const](#)  
*Check if federate is shutting down.*
- bool [is\\_RTI\\_ready \(const char \\*method\\_name\)](#)  
*Checks to make sure the RTI is ready by making sure the [TrickHLA::Federate](#) and [TrickHLA::FedAmb](#) exist and the RTI handles are initialized.*
- bool [is\\_late\\_joining\\_federate \(\) const](#)  
*Check if this is a late joining federate.*
- void [send\\_init\\_data \(\)](#)  
*Sends all the initialization data.*
- void [send\\_init\\_data \(const char \\*instance\\_name\)](#)  
*Sends the initialization data for the specified object instance name.*
- void [receive\\_init\\_data \(\)](#)  
*Wait to receive all the initialization data that is marked as required.*
- void [receive\\_init\\_data \(const char \\*instance\\_name\)](#)  
*Wait to receive the initialization data for the specified object instance name.*
- void [clear\\_init\\_sync\\_points \(\)](#)  
*Clear any remaining initialization sync-points.*
- void [wait\\_for\\_init\\_sync\\_point \(const char \\*sync\\_point\\_label\)](#)  
*Achieve then wait for the federation to become synchronized for the specified sync-point label.*
- void [send\\_requested\\_data \(double current\\_time\)](#)  
*Send the attribute value requested data to the remote federates.*
- void [request\\_data\\_update \(std::wstring const &instance\\_name\)](#)  
*Request an update to the object attributes for the given object instance name.*
- void [request\\_data\\_update \(const char \\*instance\\_name\)](#)  
*Request an update to the object attributes for the given object instance name.*
- void [send\\_cyclic\\_data \(double current\\_time\)](#)  
*Send cyclic data to remote federates.*
- void [receive\\_cyclic\\_data \(double current\\_time\)](#)  
*Handle the received cyclic data.*
- bool [discover\\_object\\_instance \(RTI1516\\_NAMESPACE::ObjectInstanceHandle theObject, RTI1516\\_NAMESPACE::ObjectClassHandle theObjectClass, std::wstring const &theObjectName\)](#)  
*Process the object discovery.*
- [Object \\* get\\_trickhla\\_object \(RTI1516\\_NAMESPACE::ObjectInstanceHandle const &instance\\_id\)](#)  
*Gets the [TrickHLA Object](#) for the specified RTI [Object](#) Instance Handle.*
- [Object \\* get\\_trickhla\\_object \(std::wstring const &obj\\_instance\\_name\)](#)  
*Gets the [TrickHLA Object](#) for the specified RTI [Object](#) Instance Name.*
- void [object\\_instance\\_name\\_reservation\\_succeeded \(std::wstring const &obj\\_instance\\_name\)](#)  
*The object instance name reservation succeeded for the given name.*
- void [object\\_instance\\_name\\_reservation\\_failed \(std::wstring const &obj\\_instance\\_name\)](#)  
*The object instance name reservation failed for the given name.*
- void [add\\_object\\_to\\_map \(Object \\*object\)](#)  
*Add a [TrickHLA::Object](#) to the manager object map.*

- `Federate * get_federate ()`  
`Get the pointer to the associated TrickHLA::Federate instance.`
- `ExecutionControlBase * get_execution_control ()`  
`Get the pointer to the associated TrickHLA::ExecutionControlBase instance.`
- `RTI1516_NAMESPACE::RTIambassador * get_RTI_ambassador ()`  
`Returns a pointer to the RTI ambassador, or NULL if one does not exist yet.`
- `double get_granted_time () const`  
`Get the granted HLA time.`
- `Int64Interval get_fed_looking_ahead () const`  
`Return a copy of the federate's lookahead time.`
- `Int64Time get_granted_fed_time () const`  
`Return a copy of the granted HLA logical time.`
- `void process_interactions ()`  
`Process the received interactions.`
- `void receive_interaction (RTI1516_NAMESPACE::InteractionClassHandle const &theInteraction, RTI1516_NAMESPACE::ParameterHandleValueMap const &theParameterValues, RTI1516_USERDATA const &theUserSuppliedTag, RTI1516_NAMESPACE::LogicalTime const &theTime, bool received_as_TSO)`  
`Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.`
- `void process_ownership ()`  
`Process the ownership requests.`
- `void mark_object_as_deleted_from_federation (RTI1516_NAMESPACE::ObjectInstanceHandle const &instance_id)`  
`Identifies the object as deleted from the RTI.`
- `void process_deleted_objects ()`  
`Scheduled method used as a callback to identify if any objects were deleted from the RTI.`
- `void restart_initialization ()`  
`Perform initialization after a checkpoint restart.`
- `void start_federation_save (const char *file_name)`  
`Start the federation save as soon as possible.`
- `void start_federation_save_at_sim_time (double freeze_sim_time, const char *file_name)`  
`Start the Federation save at the specified simulation time.`
- `void start_federation_save_at_scenario_time (double freeze_scenario_time, const char *file_name)`  
`Start the Federation save at the specified scenario time.`
- `void setup_checkpoint ()`  
`Setup the checkpoint data structures.`
- `void publish ()`  
`Publishes Object & Interaction classes and their member data.`
- `void unpublish ()`  
`Unpublish the Object & Interaction classes.`
- `void subscribe ()`  
`Subscribe to Object and Interaction classes and their member data.`
- `void unsubscribe ()`  
`Unsubscribe from the Object and Interaction classes.`
- `void publish_and_subscribe ()`  
`Publish and Subscribe to Object and Interaction classes and their member data.`
- `void reserve_object_names_with_RTI ()`  
`Reserve the RTI object instance names with the RTI, but only for the objects that are locally owned.`

- void `wait_on_reservation_of_object_names ()`  
*Waits on the reservation of the RTI object instance names for the locally owned objects.*
- void `wait_on_discovery_of_objects ()`  
*Waits on the discovery of object instances.*
- bool `is_this_a_rejoining_federate ()`  
*Checks if any object discoveries have occurred.*
- void `register_objects_with_RTI ()`  
*Creates an RTI object instance and registers it with the RTI, but only for the objects that are locally owned.*
- void `wait_on_registration_of_required_objects ()`
- void `setup_all_RTI_handles ()`  
*Sets the RTI run-time type IDs/handles for the object, attributes, interactions, and parameters.*
- void `setup_object_RTI_handles (const int data_obj_count, Object *data_objects)`  
*Sets the RTI run-time type IDs/handles for the object and attributes.*
- void `setup_interaction_RTI_handles (const int interactions_counter, Interaction *in_interactions)`  
*Sets the RTI run-time type IDs/handles for the specified interactions and parameters.*
- void `setup_all_ref_attributes ()`  
*Set up the Trick ref-attributes for the user specified objects, attributes, interactions, and parameters.*
- void `setup_preferred_order_with_RTI ()`  
*Setup the preferred order (TSO or RO) for all the object attributes and interactions.*
- void `provide_attribute_update (RTI1516_NAMESPACE::ObjectInstanceHandle const &theObject, RTI1516_NAMESPACE::AttributeHandleSet const &theAttributes)`  
*Requesting an attribute value update for the given object instance and attributes.*
- const `DebugHandler & get_debug_handler () const`  
*Get the debug handler for this `TrickHLA::Manager`.*
- int `get_object_count () const`  
*Get the `TrickHLA::Object` count.*
- `Object * get_objects ()`  
*Get the array of `TrickHLA::Object` instances.*
- int `get_interaction_count () const`  
*Get the number of `TrickHLA::Interactions`.*
- `Interaction * get_interactions ()`  
*Get the array containing the `TrickHLA::Interaction` instances.*
- void `reset_mgr_initialized ()`  
*Reset the manager as initialized.*
- bool `has_federate_been_restored () const`  
*Check if the federate has been restored.*
- void `set_execution_configuration (ExecutionConfigurationBase *exec_config)`  
*Set the execution configuration object.*
- `ExecutionConfigurationBase * get_execution_configuration ()`  
*Get the execution configuration object.*
- bool `is_execution_configuration_used ()`  
*Test is an execution configuration object is used.*
- bool `should_print (const DebugLevelEnum &level, const DebugSourceEnum &code) const`  
*Determine if the verbose debug comments should be printed to the console.*

## Data Fields

- int `obj_count`

**Units:** –  
*Number of TrickHLA Objects.*
- `Object * objects`

**Units:** –  
*Array of TrickHLA object.*
- int `inter_count`

**Units:** –  
*Number of TrickHLA Interactions.*
- `Interaction * interactions`

**Units:** –  
*Array of TrickHLA Interactions.*
- `DebugHandler debug_handler`

**Units:** –  
*Decides whether to print any debug messages.*
- bool `restore_federation`

**Data I/O:** `*i`  
**Units:** –  
*flag indicating whether to trigger the restore*
- char \* `restore_file_name`

**Data I/O:** `*i`  
**Units:** –  
*file name, which will be the label name*
- bool `initiated_a_federation_save`

**Data I/O:** `**`  
*did this manager initiate the federation save?*

## Private Member Functions

- void `initialization_execution_control ()`

*Initializes the federation execution control scheme, which must occur after the `TrickHLA::Federate` and `TrickHLA::FedAmb` has been initialized.*
- bool `is_restore_federate () const`

*Check to see if this is a restored federate.*
- void `setup_object_ref_attributes (const int data_obj_count, Object *data_objects)`

*Set up the Trick ref-attributes for the user specified objects and attributes.*
- void `setup_interaction_ref_attributes ()`

*Set up the Trick ref-attributes for the user specified interactions and parameters.*
- void `set_all_object_instance_handles_by_name ()`

*Set all the object instance handles by using the object instance names.*
- void `set_object_instance_handles_by_name (const int data_obj_count, Object *data_objects)`

*Set object instance handles by using the object instance names.*
- `Object * get_unregistered_object (RTI1516_NAMESPACE::ObjectClassHandle const &theObjectClass, std::wstring const &theObjectName)`

*Returns the first object that matches the specified Object-Class, object instance name, and is not registered, i.e. the instance ID == 0.*
- `Object * get_unregistered_remote_object (RTI1516_NAMESPACE::ObjectClassHandle const &theObjectClass)`

*Returns the first object that is remotely owned, has the same Object-Class, is not registered, and does not have an `Object` Instance Name associated with it.*

- void `determine_job_cycle_time ()`  
*Determines the job cycle time.*
- void `pull_ownership ()`  
*Pull ownership from the other federates if the pull ownership flag has been enabled.*
- void `pull_ownership_upon_rejoin ()`  
*Pull ownership from the other federates when this federate has rejoined the Federation.*
- void `push_ownership ()`  
*Push ownership to the other federates if the push ownership flag has been enabled.*
- void `grant_pull_request ()`  
*Grant any request to pull the ownership.*
- void `release_ownership ()`  
*Release ownership if we have a request to divest.*
- void `initiate_federation_save (const char *file_name)`  
*Tell the federate to initiate a save announce with the user-supplied checkpoint name set for the current frame.*
- void `setup_checkpoint_interactions ()`  
*Decodes interactions queue into an `InteractionItem` linear array.*
- void `clear_interactions ()`  
*Clears `InteractionItem` linear array.*
- void `dump_interactions ()`  
*Echoes the contents of checkpoint `InteractionItem` linear array.*
- void `restore_interactions ()`  
*Encodes checkpoint `InteractionItem` linear arrays back into the main interaction queue.*
- Manager (const Manager &rhs)  
*Copy constructor for `Manager` class.*
- Manager & `operator=` (const Manager &rhs)  
*Assignment operator for `Manager` class.*

## Private Attributes

- bool `shutdown_called`

**Units:** –  
*Flag to indicate that shutdown has been called.*
- ItemQueue `interactions_queue`

**Data I/O:** \*\*  
*Interactions queue.*
- int `check_interactions_count`

**Units:** –  
*Number of checkpointed interactions*
- InteractionItem \* `check_interactions`

**Units:** –  
*checkpointable version of interactions\_queue*
- double `job_cycle_time`
- bool `rejoining_federate`

**Units:** –  
*Internal flag to indicate if the federate is rejoining the federation.*
- bool `restore_determined`

**Data I/O:** \*\*  
*Internal flag to indicate that the restore status has been determined.*
- bool `restore_federate`

- **Data I/O:** `**`  
*Internal flag to indicate if the federate is to be restored*
- `bool mgr_initialized`
  - Units:** –  
*Internal flag to indicate Manager is initialized.*
- `ObjectInstanceMap object_map`
  - Data I/O:** `**`  
*Map of all the Objects this federate uses, the Key is the object instance-handle.*
- `bool federate_has_been_restored`
  - Data I/O:** `**`  
*Federate has been restored. do not reserve the object names again!*
- `Federate * federate`
  - Units:** –  
*Associated TrickHLA Federate.*
- `ExecutionControlBase * execution_control`
  - Units:** –  
*Execution control object.*

## Friends

- `class InputProcessor`
- `class Federate`
- `void init_attrTrickHLA_Manager()`

### 7.34.1 Detailed Description

Definition at line 86 of file Manager.hh.

### 7.34.2 Constructor & Destructor Documentation

#### 7.34.2.1 Manager() [1/2]

`Manager::Manager ( )`  
 Default constructor for the TrickHLA Manager class.

**Trick Job Class:** *initialization*

Definition at line 82 of file Manager.cpp.

#### 7.34.2.2 ~Manager()

`Manager::~Manager ( ) [virtual]`  
 Destructor for the TrickHLA Manager class.

**Trick Job Class:** *shutdown*

Definition at line 111 of file Manager.cpp.

References `clear_interactions()`, `federate`, `TrickHLA::Federate::is_execution_member()`, and `object_map`.

#### 7.34.2.3 Manager() [2/2]

`TrickHLA::Manager::Manager (`  
`const Manager & rhs ) [private]`

Copy constructor for Manager class.

This constructor is private to prevent inadvertent copies.

### 7.34.3 Member Function Documentation

#### 7.34.3.1 add\_object\_to\_map()

```
void Manager::add_object_to_map (
    Object * object )
```

Add a [TrickHLA::Object](#) to the manager object map.

##### Parameters

<code>object</code>	<a href="#">TrickHLA::Object</a> to add to the manager object map.
---------------------	--

##### Trick Job Class: *initialization*

Definition at line 846 of file Manager.cpp.

References [TrickHLA::Object::get\\_instance\\_handle\(\)](#), [TrickHLA::Object::is\\_instance\\_handle\\_valid\(\)](#), and [object\\_map](#).

Referenced by [TrickHLA::ExecutionControlBase::add\\_object\\_to\\_map\(\)](#).

#### 7.34.3.2 clear\_init\_sync\_points()

```
void Manager::clear_init_sync_points ( )
```

Clear any remaining initialization sync-points.

##### Trick Job Class: *initialization*

Definition at line 608 of file Manager.cpp.

References [TrickHLA::ExecutionControlBase::clear\\_multiphase\\_init\\_sync\\_points\(\)](#), and [execution\\_control](#).

#### 7.34.3.3 clear\_interactions()

```
void Manager::clear_interactions ( ) [private]
```

Clears [InteractionItem](#) linear array.

Definition at line 2810 of file Manager.cpp.

References [check\\_interactions](#), [check\\_interactions\\_count](#), [TrickHLA::InteractionItem::clear\\_parm\\_items\(\)](#), and [trick\\_MM](#).

Referenced by [process\\_interactions\(\)](#), [setup\\_checkpoint\\_interactions\(\)](#), and [~Manager\(\)](#).

#### 7.34.3.4 determine\_job\_cycle\_time()

```
void Manager::determine_job_cycle_time ( ) [private]
```

Determines the job cycle time.

##### Trick Job Class: *scheduled*

Definition at line 2095 of file Manager.cpp.

References [debug\\_handler](#), [TrickHLA::DEBUG\\_LEVEL\\_4\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_MANAGER](#), [federate](#), [TrickHLA::Federate::get\\_lookahead\\_time\(\)](#), [job\\_cycle\\_time](#), [obj\\_count](#), [objects](#), [TrickHLA::Object::set\\_core\\_job\\_cycle\\_time\(\)](#), [TrickHLA::DebugHandler::should\\_print\(\)](#), [THLA\\_ENDL](#), and [THLA\\_NEWLINE](#).

Referenced by [receive\\_cyclic\\_data\(\)](#), [send\\_cyclic\\_data\(\)](#), and [send\\_requested\\_data\(\)](#).

#### 7.34.3.5 discover\_object\_instance()

```
bool Manager::discover_object_instance (
    RTI1516_NAMESPACE::ObjectInstanceHandle theObject,
```

```
RTI1516_NAMESPACE::ObjectClassHandle theObjectClass,
std::wstring const & theObjectName )
```

Process the object discovery.

#### Returns

True if the instance was recognized, false otherwise.

#### Parameters

<i>theObject</i>	Instance handle to a <a href="#">Federate</a> or <a href="#">Object</a> instance.
<i>theObjectClass</i>	Class of the object.
<i>theObjectName</i>	Name of the instance.

#### Trick Job Class: *scheduled*

Definition at line 2389 of file Manager.cpp.

References `TrickHLA::Federate::add_federate_instance_id()`, `TrickHLA::Federate::add_MOM_HLAfederate_instance_id()`, `TrickHLA::Federate::add_MOM_HLAfederation_instance_id()`, `debug_handler`, `TrickHLA::DEBUG_LEVEL_2`  $\leftarrow$  `TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `federate`, `TrickHLA::Object::get_instance_handle()`, `TrickHLA::Object::get_name()`, `get_unregistered_object()`, `get_unregistered_remote_object()`, `TrickHLA::Federate::is_MOM_HLAfederate_class()`, `TrickHLA::Federate::is_MOM_HLAfederation_class()`, `object_map`, `TrickHLA::Object::set_instance_handle_and_name()`, `TrickHLA::DebugHandler::should_print()`, `THLA_NEWLINE`, and `TrickHLA::StringUtilities::to_string()`.

#### 7.34.3.6 `dump_interactions()`

```
void Manager::dump_interactions ( ) [private]
```

Echoes the contents of checkpoint [InteractionItem](#) linear array.

Definition at line 2822 of file Manager.cpp.

References `check_interactions`, `check_interactions_count`, `TrickHLA::Int64Time::getTimeInMicros()`, `TrickHLA::InteractionItem::index`, `TrickHLA::ParameterItem::index`, `TrickHLA::InteractionItem::interaction_type`, `TrickHLA::InteractionItem::order_is_TSO`, `TrickHLA::InteractionItem::parm_items`, `TrickHLA::InteractionItem::parm_items_count`, `TrickHLA::ParameterItem::size`, `TrickHLA::InteractionItem::time`, and `TrickHLA::InteractionItem::user_supplied_tag_size`.

#### 7.34.3.7 `get_debug_handler()`

```
const DebugHandler& TrickHLA::Manager::get_debug_handler ( ) const [inline]
```

Get the debug handler for this [TrickHLA::Manager](#).

#### Returns

The associated debug handler.

Definition at line 385 of file Manager.hh.

References `debug_handler`.

Referenced by `TrickHLA::ExecutionControlBase::initialize()`.

#### 7.34.3.8 `get_execution_configuration()`

```
ExecutionConfigurationBase* TrickHLA::Manager::get_execution_configuration ( ) [inline]
```

Get the execution configuration object.

**Returns**

Pointer to the associated execution configuration object.

Definition at line 422 of file Manager.hh.

References `execution_control`, and `TrickHLA::ExecutionControlBase::get_execution_configuration()`.

Referenced by `request_data_update()`, `set_all_object_instance_handles_by_name()`, `setup_all_ref_attributes()`, `setup_preferred_order_with_RTI()`, and `wait_on_registration_of_required_objects()`.

### 7.34.3.9 `get_execution_control()`

```
ExecutionControlBase* TrickHLA::Manager::get_execution_control ( ) [inline]
```

Get the pointer to the associated [TrickHLA::ExecutionControlBase](#) instance.

**Returns**

Pointer to associated [TrickHLA::ExecutionControlBase](#) instance.

Definition at line 243 of file Manager.hh.

References `execution_control`.

Referenced by `IMSim::ExecutionControl::announce_sync_point()`, `IMSim::ExecutionControl::check_pause_at_init()`, and `wait_for_init_sync_point()`.

### 7.34.3.10 `get_fed_lookahead()`

```
Int64Interval Manager::get_fed_lookahead ( ) const
```

Return a copy of the federate's lookahead time.

**Returns**

This federate's lookahead time interval.

If the federate does not exist, -1.0 seconds is assigned to the returned object.

Definition at line 2628 of file Manager.cpp.

References `federate`, and [TrickHLA::Federate::get\\_fed\\_lookahead\(\)](#).

Referenced by [TrickHLA::Interaction::get\\_fed\\_lookahead\(\)](#), and [TrickHLA::Object::get\\_fed\\_lookahead\(\)](#).

### 7.34.3.11 `get_federate()`

```
Federate* TrickHLA::Manager::get_federate ( ) [inline]
```

Get the pointer to the associated [TrickHLA::Federate](#) instance.

**Returns**

Pointer to the associated [TrickHLA::Federate](#) instance.

Definition at line 239 of file Manager.hh.

References `federate`.

Referenced by `TrickHLA::ExecutionConfiguration::configure()`, `TrickHLA::Object::get_federate()`, and `TrickHLA::Interaction::get_federate()`.

### 7.34.3.12 `get_granted_fed_time()`

```
Int64Time Manager::get_granted_fed_time ( ) const
```

Return a copy of the granted HLA logical time.

**Returns**

The granted federation time.

If the federate does not exist, MAX\_LOGICAL\_TIME\_SECONDS is assigned to the returned object.

Definition at line 2643 of file Manager.cpp.

References federate, TrickHLA::Federate::get\_granted\_fed\_time(), and TrickHLA::MAX\_LOGICAL\_TIME\_SECONDS.

Referenced by TrickHLA::Interaction::get\_granted\_fed\_time(), and TrickHLA::Object::get\_granted\_fed\_time().

**7.34.3.13 get\_granted\_time()**

```
double Manager::get_granted_time ( ) const
```

Get the granted HLA time.

**Returns**

Granted HLA time.

Definition at line 2619 of file Manager.cpp.

References federate, and TrickHLA::Federate::get\_granted\_time().

Referenced by TrickHLA::Object::get\_granted\_time().

**7.34.3.14 get\_interaction\_count()**

```
int TrickHLA::Manager::get_interaction_count ( ) const [inline]
```

Get the number of TrickHLA::Interactions.

**Returns**

The number of [TrickHLA::Interaction](#) instances.

Definition at line 397 of file Manager.hh.

References inter\_count.

**7.34.3.15 get\_interactions()**

```
Interaction* TrickHLA::Manager::get_interactions ( ) [inline]
```

Get the array containing the [TrickHLA::Interaction](#) instances.

**Returns**

Array of [TrickHLA::Interaction](#) instances.

Definition at line 401 of file Manager.hh.

References interactions.

**7.34.3.16 get\_object\_count()**

```
int TrickHLA::Manager::get_object_count ( ) const [inline]
```

Get the [TrickHLA::Object](#) count.

**Returns**

The number of registered [TrickHLA::Object](#) instances.

Definition at line 389 of file Manager.hh.

References obj\_count.

Referenced by TrickHLA::Federate::post\_restore().

**7.34.3.17 get\_objects()**

```
Object* TrickHLA::Manager::get_objects ( ) [inline]
Get the array of TrickHLA::Object instances.
```

**Returns**

Array of [TrickHLA::Object](#) instances.

Definition at line 393 of file Manager.hh.

References objects.

Referenced by [TrickHLA::Federate::post\\_restore\(\)](#).

**7.34.3.18 get\_RTI\_ambassador()**

```
RTIambassador * Manager::get_RTI_ambassador ( )
```

Returns a pointer to the RTI ambassador, or NULL if one does not exist yet.

**Returns**

Pointer to the RTI ambassador.

Definition at line 3033 of file Manager.cpp.

References federate, and [TrickHLA::Federate::get\\_RTI\\_ambassador\(\)](#).

Referenced by [TrickHLA::Interaction::get\\_RTI\\_ambassador\(\)](#), [is\\_RTI\\_ready\(\)](#), [set\\_object\\_instance\\_handles\\_by\\_name\(\)](#), [setup\\_interaction\\_RTI\\_handles\(\)](#), [setup\\_object\\_RTI\\_handles\(\)](#), and [wait\\_for\\_init\\_sync\\_point\(\)](#).

**7.34.3.19 get\_trickhla\_object() [1/2]**

```
Object * Manager::get_trickhla_object (
    RTI1516_NAMESPACE::ObjectInstanceHandle const & instance_id )
```

Gets the [TrickHLA Object](#) for the specified RTI [Object](#) Instance Handle.

**Returns**

[TrickHLA Object](#).

**Parameters**

<i>instance_id</i>	<a href="#">Object</a> instance handle.
--------------------	---

**Trick Job Class: *scheduled***

Definition at line 2357 of file Manager.cpp.

References [object\\_map](#).

Referenced by [mark\\_object\\_as\\_deleted\\_from\\_federation\(\)](#), [object\\_instance\\_name\\_reservation\\_succeeded\(\)](#), [provide\\_attribute\\_update\(\)](#), [receive\\_init\\_data\(\)](#), [request\\_data\\_update\(\)](#), and [send\\_init\\_data\(\)](#).

**7.34.3.20 get\_trickhla\_object() [2/2]**

```
Object * Manager::get_trickhla_object (
    std::wstring const & obj_instance_name )
```

Gets the [TrickHLA Object](#) for the specified RTI [Object](#) Instance Name.

**Returns**

TrickHLA Object.

**Parameters**

<i>obj_instance_name</i>	Object instance name.
--------------------------	-----------------------

**Trick Job Class:** *scheduled*

Definition at line 2368 of file Manager.cpp.

References `execution_control`, `TrickHLA::ExecutionControlBase::get_trickhla_object()`, `obj_count`, `objects`, and `TrickHLA::StringUtilities::to_wstring()`.

**7.34.3.21 get\_unregistered\_object()**

```
Object * Manager::get_unregistered_object (
    RTI1516_NAMESPACE::ObjectClassHandle const & theObjectClass,
    std::wstring const & theObjectInstanceName ) [private]
```

Returns the first object that matches the specified Object-Class, object instance name, and is not registered, i.e. the instance ID == 0.

**Returns**

TrickHLA Object

**Parameters**

<i>theObjectClass</i>	RTI Object class type.
<i>theObjectInstanceName</i>	Object instance name.

**Trick Job Class:** *scheduled*

Definition at line 2463 of file Manager.cpp.

References `execution_control`, `TrickHLA::ExecutionControlBase::get_unregistered_object()`, `obj_count`, `objects`, and `TrickHLA::StringUtilities::to_wstring()`.

Referenced by `discover_object_instance()`.

**7.34.3.22 get\_unregistered\_remote\_object()**

```
Object * Manager::get_unregistered_remote_object (
    RTI1516_NAMESPACE::ObjectClassHandle const & theObjectClass ) [private]
```

Returns the first object that is remotely owned, has the same Object-Class, is not registered, and does not have an Object Instance Name associated with it.

**Returns**

The associated `TrickHLA::Object` instance; otherwise NULL.

**Parameters**

<i>theObjectClass</i>	RTI Object class type.
-----------------------	------------------------

**Trick Job Class:** *scheduled*

Definition at line 2494 of file Manager.cpp.

References execution\_control, TrickHLA::ExecutionControlBase::get\_unregistered\_remote\_object(), obj\_count, and objects.

Referenced by discover\_object\_instance().

#### 7.34.3.23 grant\_pull\_request()

```
void Manager::grant_pull_request ( ) [private]
```

Grant any request to pull the ownership.

**Trick Job Class:** *scheduled*

Definition at line 2602 of file Manager.cpp.

References TrickHLA::Object::grant\_pull\_request(), obj\_count, and objects.

Referenced by process\_ownership().

#### 7.34.3.24 has\_federate\_been\_restored()

```
bool TrickHLA::Manager::has_federate_been_restored ( ) const [inline]
```

Check if the federate has been restored.

**Returns**

True if the federate has been restored.

Definition at line 412 of file Manager.hh.

References federate\_has\_been\_restored.

#### 7.34.3.25 initialization\_execution\_control()

```
void TrickHLA::Manager::initialization_execution_control ( ) [private]
```

Initializes the federation execution control scheme, which must occur after the [TrickHLA::Federate](#) and [TrickHLA::FedAmb](#) has been initialized.

#### 7.34.3.26 initialize()

```
void Manager::initialize ( )
```

Initializes the federate using the a multiphase initialization scheme, which must occur after the [Federate](#) and [FedAmb](#) have been initialized.

**Trick Job Class:** *initialization*

Definition at line 150 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_FULL\_TRACE, TrickHLA::DEBUG\_LEVEL\_NO\_TRACE, execution\_control, federate, TrickHLA::DebugHandler::get\_debug\_level\_as\_int(), inter\_count, interactions, mgr\_initialized, obj\_count, objects, THLA\_ENDL, THLA\_NEWLINE, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD. Referenced by [TrickHLA::Federate::pre\\_multiphase\\_initialization\(\)](#).

#### 7.34.3.27 initiate\_federation\_save()

```
void Manager::initiate_federation_save ( const char * file_name ) [private]
```

Tell the federate to initiate a save announce with the user-supplied checkpoint name set for the current frame.

**Parameters**

<i>file_name</i>	Checkpoint file name.
------------------	-----------------------

Trigger federation save, at current time or user-specified time...

NOTE: These routines do not coordinate a federation save via interactions so make these internal routines so that the user does not accidentally call them and mess things up.

Definition at line 2686 of file Manager.cpp.

References [federate](#), [TrickHLA::Federate::initiate\\_save\\_announce\(\)](#), [initiated\\_a\\_federation\\_save](#), and [TrickHLA::Federate::set\\_checkpoint\\_file\\_name\(\)](#).

Referenced by [IMSim::ExecutionControl::start\\_federation\\_save\\_at\\_scenario\\_time\(\)](#).

**7.34.3.28 is\_execution\_configuration\_used()**

```
bool TrickHLA::Manager::is_execution_configuration_used ( ) [inline]
Test is an execution configuration object is used.
```

**Returns**

True if an execution configuration object is used.

Definition at line 428 of file Manager.hh.

References [execution\\_control](#), and [TrickHLA::ExecutionControlBase::is\\_execution\\_configuration\\_used\(\)](#).

Referenced by [request\\_data\\_update\(\)](#), [set\\_all\\_object\\_instance\\_handles\\_by\\_name\(\)](#), [setup\\_all\\_ref\\_attributes\(\)](#), [setup\\_preferred\\_order\\_with\\_RTI\(\)](#), and [wait\\_on\\_registration\\_of\\_required\\_objects\(\)](#).

**7.34.3.29 is\_late\_joining\_federate()**

```
bool TrickHLA::Manager::is_late_joining_federate ( ) const [inline]
Check if this is a late joining federate.
```

**Returns**

True if the is a late joining federate.

Definition at line 156 of file Manager.hh.

References [execution\\_control](#), and [TrickHLA::ExecutionControlBase::is\\_late\\_joine\(\)](#).

Referenced by [IMSim::ExecutionControl::check\\_pause\(\)](#), [TrickHLA::ExecutionControlBase::clear\\_multiphase\\_init\\_sync\\_points\(\)](#), [receive\\_init\\_data\(\)](#), [SpaceFOM::ExecutionControl::receive\\_init\\_root\\_ref\\_frame\(\)](#), [send\\_init\\_data\(\)](#), [SpaceFOM::ExecutionControl::send\\_init\\_root\\_ref\\_frame\(\)](#), and [wait\\_for\\_init\\_sync\\_point\(\)](#).

**7.34.3.30 is\_restore\_federate()**

```
bool TrickHLA::Manager::is_restore_federate ( ) const [inline], [private]
Check to see if this is a restored federate.
```

Definition at line 484 of file Manager.hh.

References [restore\\_federate](#).

**7.34.3.31 is\_RTI\_ready()**

```
bool Manager::is_RTI_ready (
    const char * method_name )
```

Checks to make sure the RTI is ready by making sure the [TrickHLA::Federate](#) and [TrickHLA::FedAmb](#) exist and the RTI handles are initialized.

**Returns**

True if the RTI is ready, false otherwise.

**Parameters**

<i>method_name</i>	The method/function name.
--------------------	---------------------------

Definition at line 2654 of file Manager.cpp.

References federate, get\_RTI\_ambassador(), THLA\_NEWLINE, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, T←RICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by publish(), subscribe(), unpublish(), and unsubscribe().

**7.34.3.32 is\_shutdown()**

bool TrickHLA::Manager::is\_shutdown ( ) const [inline]  
Check if federate is shutting down.

**Returns**

True if the manager is shutting down the federate.

Definition at line 142 of file Manager.hh.

References shutdown\_called.

Referenced by TrickHLA::Interaction::remove(), and TrickHLA::Object::remove().

**7.34.3.33 is\_this\_a\_rejoining\_federate()**

bool Manager::is\_this\_a\_rejoining\_federate ( )  
Checks if any object discoveries have occurred.

**Returns**

True if this is a rejoining federate.

If they have, true is returned if the 'create HLA instance' object was discovered. If no discoveries took place or if the required 'create HLA instance' object was not discovered, false is returned. **Trick Job Class:** *initialization*

Definition at line 3016 of file Manager.cpp.

References obj\_count, objects, and rejoining\_federate.

**7.34.3.34 mark\_object\_as\_deleted\_from\_federation()**

void Manager::mark\_object\_as\_deleted\_from\_federation ( RTI1516\_NAMESPACE::ObjectInstanceHandle const & *instance\_id* )

Identifies the object as deleted from the RTI.

**Parameters**

<i>instance_id</i>	HLA object instance handle.
--------------------	-----------------------------

Definition at line 2535 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER,

execution\_control, TrickHLA::Object::get\_name(), get\_trickhla\_object(), TrickHLA::ExecutionControlBase::mark\_object\_as\_deleted\_from\_federation(), TrickHLA::Object::remove\_object\_instance(), TrickHLA::DebugHandler::should\_print(), THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

#### 7.34.3.35 object\_instance\_name\_reservation\_failed()

```
void Manager::object_instance_name_reservation_failed (
    std::wstring const & obj_instance_name )
```

The object instance name reservation failed for the given name.

Parameters

<code>obj_instance_name</code>	Object instance name.
--------------------------------	-----------------------

**Trick Job Class:** *initialization*

Definition at line 781 of file Manager.cpp.

References execution\_control, obj\_count, TrickHLA::ExecutionControlBase::object\_instance\_name\_reservation\_failed(), objects, THLA\_ENDL, THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_wstring().

#### 7.34.3.36 object\_instance\_name\_reservation\_succeeded()

```
void Manager::object_instance_name_reservation_succeeded (
    std::wstring const & obj_instance_name )
```

The object instance name reservation succeeded for the given name.

Parameters

<code>obj_instance_name</code>	Object instance name.
--------------------------------	-----------------------

**Trick Job Class:** *initialization*

Definition at line 754 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, execution\_control, TrickHLA::Object::get\_name(), get\_trickhla\_object(), TrickHLA::ExecutionControlBase::object\_instance\_name\_reservation\_succeeded(), TrickHLA::Object::set\_name\_registered(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

#### 7.34.3.37 operator=()

```
Manager& TrickHLA::Manager::operator= (
    const Manager & rhs ) [private]
```

Assignment operator for Manager class.

This assignment operator is private to prevent inadvertent copies.

#### 7.34.3.38 process\_deleted\_objects()

```
void Manager::process_deleted_objects ( )
```

Scheduled method used as a callback to identify if any objects were deleted from the RTI.

**Trick Job Class:** *logging*

Definition at line 2563 of file Manager.cpp.

References execution\_control, obj\_count, objects, TrickHLA::Object::process\_deleted\_object(), and TrickHLA::ExecutionControlBase::process\_deleted\_objects().

### 7.34.3.39 process\_interactions()

```
void Manager::process_interactions ( )
```

Process the received interactions.

**Trick Job Class:** *scheduled*

Definition at line 2226 of file Manager.cpp.

References clear\_interactions(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ItemQueue::empty(), execution\_control, TrickHLA::Interaction::extract\_data(), TrickHLA::ItemQueue::front(), TrickHLA::InteractionItem::index, inter\_count, TrickHLA::InteractionItem::interaction\_type, interactions, interactions\_queue, TrickHLA::Interaction::is\_subscribe(), TrickHLA::ItemQueue::pop(), TrickHLA::Interaction::process\_interaction(), TrickHLA::ExecutionControlBase::process\_mode\_interaction(), TrickHLA::DebugHandler::should\_print(), THLA\_ENDL, THLA\_NEWLINE, and TrickHLA::TRICKHLA\_MANAGER\_USER\_DEFINED\_INTERACTION.

### 7.34.3.40 process\_ownership()

```
void Manager::process_ownership ( )
```

Process the ownership requests.

**Trick Job Class:** *scheduled*

Definition at line 2518 of file Manager.cpp.

References grant\_pull\_request(), pull\_ownership(), push\_ownership(), and release\_ownership().

### 7.34.3.41 provide\_attribute\_update()

```
void Manager::provide_attribute_update (
    RTI1516_NAMESPACE::ObjectInstanceHandle const & theObject,
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes )
```

Requesting an attribute value update for the given object instance and attributes.

#### Parameters

<i>theObject</i>	HLA object instance handle.
<i>theAttributes</i>	HLA attribute handle set.

**Trick Job Class:** *scheduled*

Definition at line 2077 of file Manager.cpp.

References execution\_control, get\_trickhla\_object(), TrickHLA::ExecutionControlBase::provide\_attribute\_update(), and TrickHLA::Object::provide\_attribute\_update().

### 7.34.3.42 publish()

```
void Manager::publish ( )
```

Publishes [Object](#) & [Interaction](#) classes and their member data.

**Trick Job Class:** *initialization*

Definition at line 1445 of file Manager.cpp.

References execution\_control, inter\_count, interactions, is\_RTI\_ready(), obj\_count, objects, TrickHLA::ExecutionControlBase::publish(), TrickHLA::Interaction::publish\_interaction(), and TrickHLA::Object::publish\_object\_attributes(). Referenced by publish\_and\_subscribe().

#### 7.34.3.43 publish\_and\_subscribe()

```
void Manager::publish_and_subscribe ( )
Publish and Subscribe to Object and Interaction classes and their member data.
```

**Trick Job Class:** *initialization*

Definition at line 1611 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, publish(), TrickHLA::DebugHandler::should\_print(), subscribe(), and THLA\_NEWLINE.

Referenced by SpaceFOM::ExecutionControl::early\_joine\_hla\_init\_process(), SpaceFOM::ExecutionControl::late\_joine\_hla\_init\_process(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.34.3.44 pull\_ownership()

```
void Manager::pull_ownership ( ) [private]
Pull ownership from the other federates if the pull ownership flag has been enabled.
```

**Trick Job Class:** *scheduled*

Definition at line 2582 of file Manager.cpp.

References obj\_count, objects, and TrickHLA::Object::pull\_ownership().

Referenced by process\_ownership().

#### 7.34.3.45 pull\_ownership\_upon\_rejoin()

```
void Manager::pull_ownership_upon_rejoin ( ) [private]
Pull ownership from the other federates when this federate has rejoined the Federation.
```

**Trick Job Class:** *scheduled*

Definition at line 2896 of file Manager.cpp.

References obj\_count, objects, and TrickHLA::Object::pull\_ownership\_upon\_rejoin().

Referenced by IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.34.3.46 push\_ownership()

```
void Manager::push_ownership ( ) [private]
Push ownership to the other federates if the push ownership flag has been enabled.
```

**Trick Job Class:** *scheduled*

Definition at line 2592 of file Manager.cpp.

References obj\_count, objects, and TrickHLA::Object::push\_ownership().

Referenced by process\_ownership().

#### 7.34.3.47 receive\_cyclic\_data()

```
void Manager::receive_cyclic_data (
    double current_time )
```

Handle the received cyclic data.

##### Parameters

<i>current_time</i>	Current time.
---------------------	---------------

If the object is owned remotely, this function copies its internal data into simulation object and marks the object as

"unchanged". This data was deposited by the reflect callback and marked as "changed". By marking it as unchanged, we avoid copying the same data over and over. If the object is locally owned, we shouldn't be receiving any remote data anyway and if we were to – bogusly – copy it to the internal byte buffer, we'd continually reset our local simulation. **Trick Job Class: *scheduled***

Definition at line 2199 of file Manager.cpp.

References `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `determine_job_cycle_time()`, `execution_control`, `job_cycle_time`, `obj_count`, `objects`, `TrickHLA::ExecutionControlBase::receive_cyclic_data()`, `TrickHLA::Object::receive_cyclic_data()`, `TrickHLA::DebugHandler::should_print()`, and `THLA_NEWLINE`.

#### 7.34.3.48 `receive_init_data()` [1/2]

```
void Manager::receive_init_data ( )
```

Wait to receive all the initialization data that is marked as required.

**Trick Job Class: *initialization***

Definition at line 410 of file Manager.cpp.

References `TrickHLA::Federate::check_for_shutdown_with_termination()`, `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `execution_control`, `federate`, `TrickHLA::Federate::is_execution_member()`, `is_late_joining_federate()`, `TrickHLA::Object::is_required()`, `obj_count`, `objects`, `TrickHLA::Object::receive_init_data()`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, and `TrickHLA::ExecutionControlBase::wait_on_init_data()`.

#### 7.34.3.49 `receive_init_data()` [2/2]

```
void Manager::receive_init_data (
    const char * instance_name )
```

Wait to receive the initialization data for the specified object instance name.

**Returns**

Name of object instance name to receive data for.

**Trick Job Class: *initialization***

Definition at line 498 of file Manager.cpp.

References `TrickHLA::Object::any_remotely_owned_subscribed_init_attribute()`, `TrickHLA::Federate::check_for_shutdown_with_termination()`, `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `execution_control`, `federate`, `get_trickhla_object()`, `TrickHLA::Object::is_changed()`, `TrickHLA::Federate::is_execution_member()`, `is_late_joining_federate()`, `TrickHLA::Object::is_required()`, `TrickHLA::Object::receive_init_data()`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_wstring()`, and `TrickHLA::ExecutionControlBase::wait_on_init_data()`.

#### 7.34.3.50 `receive_interaction()`

```
void Manager::receive_interaction (
    RTI1516_NAMESPACE::InteractionClassHandle const & theInteraction,
    RTI1516_NAMESPACE::ParameterHandleValueMap const & theParameterValues,
    RTI1516_USERDATA const & theUserSuppliedTag,
    RTI1516_NAMESPACE::LogicalTime const & theTime,
    bool received_as_TSO )
```

Process all received interactions by calling in turn each interaction handler that is subscribed to the interaction.

**Parameters**

<code>theInteraction</code>	Interaction handle.
-----------------------------	---------------------

**Parameters**

<i>theParameterValues</i>	Parameter values.
<i>theUserSuppliedTag</i>	Users tag.
<i>theTime</i>	HLA time for the interaction.
<i>received_as_TSO</i>	True if interaction was received by RTI as TSO.

**Trick Job Class:** *scheduled*

Definition at line 2286 of file Manager.cpp.

References `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `execution_control`, `TrickHLA::Int64Time::getDoubleTime()`, `inter_count`, `interactions`, `interactions_queue`, `TrickHLA::ItemQueue::push()`, `TrickHLA::ExecutionControlBase::receive_interaction()`, `TrickHLA::Int64Time::setTo()`, `TrickHLA::DebugHandler::should_print()`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, and `TrickHLA::TRICKHLA_MANAGER::USER_DEFINED_INTERACTION`.

**7.34.3.51 register\_objects\_with\_RTI()**

```
void Manager::register_objects_with_RTI ( )
```

Creates an RTI object instance and registers it with the RTI, but only for the objects that are locally owned.

**Trick Job Class:** *initialization*

Definition at line 1673 of file Manager.cpp.

References `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `execution_control`, `TrickHLA::Object::get_instance_handle()`, `obj_count`, `object_map`, `objects`, `TrickHLA::Object::register_object_with_RTI()`, `TrickHLA::ExecutionControlBase::register_objects_with_RTI()`, `TrickHLA::DebugHandler::should_print()`, and `THLA_NEWLINE`.

Referenced by `SpaceFOM::ExecutionControl::early_joiner_hla_init_process()`, `SpaceFOM::ExecutionControl::late_joiner_hla_init_process()`, `TrickHLA::ExecutionControl::pre_multi_phase_init_processes()`, `DSES::ExecutionControl::pre_multi_phase_init_processes()`, `DIS::ExecutionControl::pre_multi_phase_init_processes()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

**7.34.3.52 release\_ownership()**

```
void Manager::release_ownership ( ) [private]
```

Release ownership if we have a request to divest.

**Trick Job Class:** *scheduled*

Definition at line 2612 of file Manager.cpp.

References `obj_count`, `objects`, and `TrickHLA::Object::release_ownership()`.Referenced by `process_ownership()`.**7.34.3.53 request\_data\_update() [1/2]**

```
void Manager::request_data_update (
    const char * instance_name )
```

Request an update to the object attributes for the given object instance name.

**Parameters**

<i>instance_name</i>	Object instance name.
----------------------	-----------------------

**Trick Job Class:** *initialization*

Definition at line 738 of file Manager.cpp.

References `request_data_update()`, and `TrickHLA::StringUtilities::to_wstring()`.

#### 7.34.3.54 `request_data_update()` [2/2]

```
void Manager::request_data_update (
    std::wstring const & instance_name )
```

Request an update to the object attributes for the given object instance name.

##### Parameters

<code>instance_name</code>	<code>Object</code> instance name.
----------------------------	------------------------------------

##### Trick Job Class: *initialization*

Definition at line 705 of file `Manager.cpp`.

References `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `get_execution_configuration()`, `get_trickhla_object()`, `is_execution_configuration_used()`, `TrickHLA::Object::request_attribute_value_update()`, `TrickHLA::DebugHandler::should_print()`, `THLA_NEWLINE`, and `TrickHLA::StringUtilities::to_wstring()`.

Referenced by `SpaceFOM::ExecutionControl::late_joiner_hla_init_process()`, `IMSim::ExecutionControl::pre_multi_phase_init_processes()`, and `request_data_update()`.

#### 7.34.3.55 `reserve_object_names_with_RTI()`

```
void Manager::reserve_object_names_with_RTI ( )
```

Reserve the RTI object instance names with the RTI, but only for the objects that are locally owned.

##### Trick Job Class: *initialization*

Definition at line 1624 of file `Manager.cpp`.

References `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `obj_count`, `objects`, `TrickHLA::Object::reserve_object_name_with_RTI()`, `TrickHLA::DebugHandler::should_print()`, and `THLA_NEWLINE`.

Referenced by `SpaceFOM::ExecutionControl::early_joiner_hla_init_process()`, `SpaceFOM::ExecutionControl::late_joiner_hla_init_process()`, `TrickHLA::ExecutionControl::pre_multi_phase_init_processes()`, `DSES::ExecutionControl::pre_multi_phase_init_processes()`, `DIS::ExecutionControl::pre_multi_phase_init_processes()`, and `IMSim::ExecutionControl::pre_multi_phase_init_processes()`.

#### 7.34.3.56 `reset_mgr_initialized()`

```
void TrickHLA::Manager::reset_mgr_initialized ( ) [inline]
```

Reset the manager as initialized.

Definition at line 404 of file `Manager.hh`.

References `federate_has_been_restored`, and `mgr_initialized`.

Referenced by `TrickHLA::Federate::post_restore()`.

#### 7.34.3.57 `restart_initialization()`

```
void Manager::restart_initialization ( )
```

Perform initialization after a checkpoint restart.

Definition at line 215 of file `Manager.cpp`.

References `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `execution_control`, `federate`, `TrickHLA::Federate::initialize_MOM_handles()`, `inter_count`, `interactions`, `TrickHLA::ExecutionControlBase::is_master()`, `mgr_initialized`, `obj_count`, `objects`, `TrickHLA::Federate::restart_initialization()`.

restore\_interactions(), TrickHLA::Object::restore\_ownership\_transfer\_checkpointed\_data(), TrickHLA::Federate::set\_all\_federate\_MOM\_instance\_handles\_by\_name(), set\_all\_object\_instance\_handles\_by\_name(), TrickHLA::ExecutionControlBase::set\_master(), setup\_all\_ref\_attributes(), setup\_all\_RTI\_handles(), TrickHLA::DebugHandler::should\_print(), THLA\_ENDL, THLA\_NEWLINE, and TrickHLA::Federate::wait\_for\_required\_federates\_to\_join(). Referenced by IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.34.3.58 restore\_interactions()

void Manager::restore\_interactions ( ) [private]

Encodes checkpoint [InteractionItem](#) linear arrays back into the main interaction queue.

Definition at line 2856 of file Manager.cpp.

References check\_interactions, check\_interactions\_count, debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::InteractionItem::index, TrickHLA::InteractionItem::interaction\_type, interactions\_queue, TrickHLA::InteractionItem::order\_is\_TSO, TrickHLA::InteractionItem::parm\_items, TrickHLA::InteractionItem::parm\_items\_count, TrickHLA::ItemQueue::push(), TrickHLA::InteractionItem::restore\_queue(), TrickHLA::DebugHandler::should\_print(), THLA\_NEWLINE, TrickHLA::InteractionItem::time, trick\_MM, TrickHLA::InteractionItem::user\_supplied\_tag, and TrickHLA::InteractionItem::user\_supplied\_tag\_size.

Referenced by TrickHLA::Federate::post\_restore(), and restart\_initialization().

#### 7.34.3.59 send\_cyclic\_data()

void Manager::send\_cyclic\_data ( double current\_time )

Send cyclic data to remote federates.

Parameters

<i>current_time</i>	Current time.
---------------------	---------------

**Trick Job Class:** *scheduled*

Definition at line 2168 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, determine\_job\_cycle\_time(), job\_cycle\_time, obj\_count, objects, TrickHLA::Object::send\_cyclic\_data(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

#### 7.34.3.60 send\_init\_data() [1/2]

void Manager::send\_init\_data ( )

Sends all the initialization data.

**Trick Job Class:** *initialization*

Definition at line 307 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, is\_late\_joining\_federate(), obj\_count, objects, TrickHLA::Object::send\_init\_data(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

#### 7.34.3.61 send\_init\_data() [2/2]

void Manager::send\_init\_data ( const char \* instance\_name )

Sends the initialization data for the specified object instance name.

**Returns**

Name of object instance name to send data for.

**Trick Job Class: *initialization***

Definition at line 347 of file Manager.cpp.

References TrickHLA::Object::any\_locally\_owned\_published\_init\_attribute(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, get\_trickhla\_object(), is\_late\_joining\_federate(), TrickHLA::Object::send\_init\_data(), TrickHLA::DebugHandler::should\_print(), THLA\_ENDL, THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_wstring().

**7.34.3.62 send\_requested\_data()**

```
void Manager::send_requested_data (
    double current_time )
```

Send the attribute value requested data to the remote federates.

**Parameters**

<i>current_time</i>	Current time.
---------------------	---------------

**Trick Job Class: *scheduled***

Definition at line 2141 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_4\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, determine\_job\_cycle\_time(), execution\_control, job\_cycle\_time, obj\_count, objects, TrickHLA::ExecutionControl::Base::send\_requested\_data(), TrickHLA::Object::send\_requested\_data(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

Referenced by DSES::ExecutionControl::check\_freeze\_exit(), and SpaceFOM::ExecutionControl::check\_freeze\_exit().

**7.34.3.63 set\_all\_object\_instance\_handles\_by\_name()**

```
void Manager::set_all_object_instance_handles_by_name ( ) [private]
```

Set all the object instance handles by using the object instance names.

**Trick Job Class: *initialization***

Definition at line 1906 of file Manager.cpp.

References get\_execution\_configuration(), is\_execution\_configuration\_used(), mgr\_initialized, obj\_count, object\_map, objects, set\_object\_instance\_handles\_by\_name(), and THLA\_NEWLINE.

Referenced by TrickHLA::Federate::post\_restore(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and restart\_initialization().

**7.34.3.64 set\_execution\_configuration()**

```
void TrickHLA::Manager::set_execution_configuration (
    ExecutionConfigurationBase * exec_config ) [inline]
```

Set the execution configuration object.

**Returns**

Pointer to the associated execution configuration object.

Definition at line 416 of file Manager.hh.

References execution\_control, and TrickHLA::ExecutionControlBase::set\_execution\_configuration().

### 7.34.3.65 set\_object\_instance\_handles\_by\_name()

```
void Manager::set_object_instance_handles_by_name (
    const int data_obj_count,
    Object * data_objects ) [private]
```

Set object instance handles by using the object instance names.

#### Parameters

<code>data_obj_count</code>	Number of objects.
<code>data_objects</code>	Simulation <a href="#">TrickHLA</a> Objects.

#### Trick Job Class: *initialization*

Definition at line 1931 of file Manager.cpp.

References `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `federate`, `TrickHLA::Object::get_instance_handle()`, `TrickHLA::Object::get_name()`, `get_RTI_ambassador()`, `TrickHLA::Object::is_instance_handle_valid()`, `mgr_initialized`, `object_map`, `RTI1516_EXCEPTION`, `TrickHLA::Object::set_instance_handle()`, `TrickHLA::DebugHandler::should_print()`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, `TrickHLA::StringUtilities::to_wstring()`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `set_all_object_instance_handles_by_name()`.

### 7.34.3.66 setup()

```
void Manager::setup (
    Federate & federate,
    ExecutionControlBase & execution_control )
```

Setup the required class instance associations.

#### Parameters

<code>federate</code>	Associated <a href="#">TrickHLA::Federate</a> class instance.
<code>execution_control</code>	Associated <a href="#">ExecutionControl</a> class instance.

#### Assumptions and Limitations:

- The [TrickHLA::ExecutionControlBase](#) class is actually an abstract class. Therefore, the actual object instance being passed in is an instantiable polymorphic child of the [TrickHLA::ExecutionControlBase](#) class.

#### Trick Job Class: *default\_data*

Definition at line 133 of file Manager.cpp.

References `execution_control`, and `federate`.

Referenced by `TrickHLA::Federate::setup()`.

### 7.34.3.67 setup\_all\_ref\_attributes()

```
void Manager::setup_all_ref_attributes ( )
```

Set up the Trick ref-attributes for the user specified objects, attributes, interactions, and parameters.

#### Trick Job Class: *initialization*

Definition at line 862 of file Manager.cpp.

References `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_LEVEL_9_TRACE`, `TrickHLA::DEBUG_SOURCE_MANAGER`, `get_execution_configuration()`, `is_execution_configuration_used()`, `obj_count`,

object\_map, objects, setup\_interaction\_ref\_attributes(), setup\_object\_ref\_attributes(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

Referenced by TrickHLA::Federate::post\_restore(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), and restart\_initialization().

#### 7.34.3.68 setup\_all\_RTI\_handles()

```
void Manager::setup_all_RTI_handles ( )
```

Sets the RTI run-time type IDs/handles for the object, attributes, interactions, and parameters.

**Trick Job Class:** *initialization*

Definition at line 1019 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, execution\_control, inter\_count, interactions, obj\_count, objects, TrickHLA::ExecutionControlBase::setup\_interaction\_RTI\_handles(), setup\_interaction\_RTI\_handles(), TrickHLA::ExecutionControlBase::setup\_object\_RTI\_handles(), setup\_object\_RTI\_handles(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

Referenced by SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), TrickHLA::Federate::post\_restore(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), and restart\_initialization().

#### 7.34.3.69 setup\_checkpoint()

```
void Manager::setup_checkpoint ( )
```

Setup the checkpoint data structures.

**Trick Job Class:** *initialization*

Definition at line 2726 of file Manager.cpp.

References execution\_control, TrickHLA::Object::mark\_required(), obj\_count, objects, TrickHLA::ExecutionControlBase::setup\_checkpoint(), setup\_checkpoint\_interactions(), and TrickHLA::Object::setup\_ownership\_transfer\_checkpointed\_data().

Referenced by TrickHLA::Federate::setup\_checkpoint().

#### 7.34.3.70 setup\_checkpoint\_interactions()

```
void Manager::setup_checkpoint_interactions ( ) [private]
```

Decodes interactions queue into an [InteractionItem](#) linear array.

Definition at line 2746 of file Manager.cpp.

References check\_interactions, check\_interactions\_count, TrickHLA::InteractionItem::checkpoint\_queue(), clear\_interactions(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::ItemQueue::empty(), TrickHLA::ItemQueue::front(), TrickHLA::InteractionItem::index, TrickHLA::InteractionItem::interaction\_type, interactions\_queue, TrickHLA::ItemQueue::lock(), TrickHLA::ItemQueue::next(), TrickHLA::InteractionItem::order\_is\_TSO, TrickHLA::InteractionItem::parm\_items, TrickHLA::InteractionItem::parm\_items\_count, TrickHLA::ItemQueue::rewind(), TrickHLA::DebugHandler::should\_print(), TrickHLA::ItemQueue::size(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::InteractionItem::time, trick\_MM, TrickHLA::ItemQueue::unlock(), TrickHLA::InteractionItem::user\_supplied\_tag, and TrickHLA::InteractionItem::user\_supplied\_tag\_size.

Referenced by setup\_checkpoint().

#### 7.34.3.71 setup\_interaction\_ref\_attributes()

```
void Manager::setup_interaction_ref_attributes ( ) [private]
```

Set up the Trick ref-attributes for the user specified interactions and parameters.

**Trick Job Class:** *initialization*

Definition at line 958 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_9\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, execution\_control, TrickHLA::Interaction::get\_FOM\_name(), TrickHLA::Interaction::get\_parameter\_count(), TrickHLA::Interaction::get\_parameters(), TrickHLA::Parameter::get\_trick\_name(), TrickHLA::Parameter::initialize(), TrickHLA::Interaction::initialize(), inter\_count, interactions, mgr\_initialized, TrickHLA::Parameter::set\_debug\_level(), TrickHLA::ExecutionControlBase::setup\_interaction\_ref\_attributes(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

Referenced by setup\_all\_ref\_attributes().

#### 7.34.3.72 setup\_interaction\_RTI\_handles()

```
void Manager::setup_interaction_RTI_handles (
    const int interactions_counter,
    Interaction * in_interactions )
```

Sets the RTI run-time type IDs/handles for the specified interactions and parameters.

##### Parameters

<i>interactions_counter</i>	Number of interactions.
<i>in_interactions</i>	Simulation <a href="#">TrickHLA Interaction</a> objects.

**Trick Job Class:** *initialization*

Definition at line 1249 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_9\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, federate, TrickHLA::Interaction::get\_class\_handle(), TrickHLA::Parameter::get\_FOM\_name(), TrickHLA::Interaction::get\_FOM\_name(), TrickHLA::Interaction::get\_parameter\_count(), TrickHLA::Interaction::get\_parameters(), get\_RTI\_ambassador(), mgr\_initialized, RTI1516\_EXCEPTION, TrickHLA::Interaction::set\_class\_handle(), TrickHLA::Parameter::set\_parameter\_handle(), TrickHLA::DebugHandler::should\_print(), THLA-ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by setup\_all\_RTI\_handles(), and SpaceFOM::ExecutionControl::setup\_interaction\_RTI\_handles().

#### 7.34.3.73 setup\_object\_ref\_attributes()

```
void Manager::setup_object_ref_attributes (
    const int data_obj_count,
    Object * data_objects ) [private]
```

Set up the Trick ref-attributes for the user specified objects and attributes.

##### Parameters

<i>data_obj_count</i>	Number of objects.
<i>data_objects</i>	Simulation <a href="#">TrickHLA::Objects</a> .

**Trick Job Class:** *initialization*

Definition at line 896 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_9\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, TrickHLA::Object::get\_attribute\_count(), TrickHLA::Object::get\_attributes(), TrickHLA::Object::get\_FOM\_name(), TrickHLA::Object::get\_name(), TrickHLA::Attribute::get\_trick\_name(), TrickHLA::

Attribute::initialize(), TrickHLA::Object::initialize(), TrickHLA::Object::is\_create\_HLA\_instance(), mgr\_initialized, TrickHLA::Attribute::set\_debug\_level(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE. Referenced by setup\_all\_ref\_attributes().

#### 7.34.3.74 setup\_object\_RTI\_handles()

```
void Manager::setup_object_RTI_handles (
    const int data_obj_count,
    Object * data_objects )
```

Sets the RTI run-time type IDs/handles for the object and attributes.

##### Parameters

<code>data_obj_count</code>	Number of objects.
<code>data_objects</code>	Simulation <a href="#">TrickHLA</a> Objects.

##### Trick Job Class: *initialization*

Definition at line 1044 of file Manager.cpp.

References TrickHLA::Object::build\_attribute\_map(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_9\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, federate, TrickHLA::Object::get\_attribute\_count(), TrickHLA::Object::get\_attributes(), TrickHLA::Object::get\_class\_handle(), TrickHLA::Attribute::get\_FOM\_name(), TrickHLA::Object::get\_FOM\_name(), TrickHLA::Object::get\_name(), get\_RTI\_ambassador(), mgr\_initialized, RTI1516\_EXCEPTION, TrickHLA::Attribute::set\_attribute\_handle(), TrickHLA::Object::set\_class\_handle(), TrickHLA::DebugHandler::should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by setup\_all\_RTI\_handles(), TrickHLA::ExecutionControl::setup\_object\_RTI\_handles(), and SpaceFOM::ExecutionControl::setup\_object\_RTI\_handles().

#### 7.34.3.75 setup\_preferred\_order\_with\_RTI()

```
void Manager::setup_preferred_order_with_RTI ( )
```

Setup the preferred order (TSO or RO) for all the object attributes and interactions.

##### Trick Job Class: *initialization*

Definition at line 1699 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, get\_execution\_configuration(), inter\_count, interactions, is\_execution\_configuration\_used(), obj\_count, objects, TrickHLA::Interaction::setup\_preferred\_order\_with\_RTI(), TrickHLA::Object::setup\_preferred\_order\_with\_RTI(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

Referenced by TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.34.3.76 should\_print()

```
bool TrickHLA::Manager::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const [inline]
```

Determine if the verbose debug comments should be printed to the console.

**Returns**

Returns true if the requested message should print level.

**Parameters**

<i>level</i>	Debug level of the incoming message.
<i>code</i>	Source code association of the incoming messages.

Definition at line 437 of file Manager.hh.

References debug\_handler, and TrickHLA::DebugHandler::should\_print().

Referenced by TrickHLA::FedAmb::should\_print(), TrickHLA::Interaction::should\_print(), TrickHLA::Object::should\_print(), and shutdown().

**7.34.3.77 shutdown()**

```
void Manager::shutdown ( )
```

Shutdown the federate by shutting down time management, resigning from the federation, and then attempt to destroy the federation.

**Trick Job Class:** *shutdown*

Definition at line 685 of file Manager.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, should\_print(), shutdown\_called, and THLA\_NEWLINE.

Referenced by SpaceFOM::ExecutionControl::check\_for\_shutdown\_with\_termination(), and TrickHLA::Federate::shutdown().

**7.34.3.78 start\_federation\_save()**

```
void Manager::start_federation_save (
    const char * file_name )
```

Start the federation save as soon as possible.

**Parameters**

<i>file_name</i>	Checkpoint file name.
------------------	-----------------------

Definition at line 2695 of file Manager.cpp.

References start\_federation\_save\_at\_scenario\_time().

**7.34.3.79 start\_federation\_save\_at\_scenario\_time()**

```
void Manager::start_federation_save_at_scenario_time (
    double freeze_scenario_time,
    const char * file_name )
```

Start the Federation save at the specified scenario time.

**Parameters**

<i>freeze_scenario_time</i>	Scenario time to freeze.
<i>file_name</i>	Checkpoint file name.

Definition at line 2712 of file Manager.cpp.

References `execution_control`, and `TrickHLA::ExecutionControlBase::start_federation_save_at_scenario_time()`.

Referenced by `start_federation_save()`, and `start_federation_save_at_sim_time()`.

#### 7.34.3.80 `start_federation_save_at_sim_time()`

```
void Manager::start_federation_save_at_sim_time (
    double freeze_sim_time,
    const char * file_name )
```

Start the Federation save at the specified simulation time.

##### Parameters

<code>freeze_sim_time</code>	Simulation time to freeze.
<code>file_name</code>	Checkpoint file name.

Definition at line 2701 of file Manager.cpp.

References `TrickHLA::ExecutionControlBase::convert_sim_time_to_scenario_time()`, `execution_control`, and `start_federation_save_at_scenario_time()`.

#### 7.34.3.81 `subscribe()`

```
void Manager::subscribe ( )
```

Subscribe to [Object](#) and [Interaction](#) classes and their member data.

##### Trick Job Class: *initialization*

Definition at line 1527 of file Manager.cpp.

References `execution_control`, `inter_count`, `interactions`, `is_RTI_ready()`, `obj_count`, `objects`, `TrickHLA::ExecutionControlBase::subscribe()`, `TrickHLA::Interaction::subscribe_to_interaction()`, and `TrickHLA::Object::subscribe_to_object_attributes()`.

Referenced by `publish_and_subscribe()`.

#### 7.34.3.82 `unpublish()`

```
void Manager::unpublish ( )
```

Unpublish the [Object](#) & [Interaction](#) classes.

##### Trick Job Class: *initialization*

Definition at line 1472 of file Manager.cpp.

References `execution_control`, `TrickHLA::Interaction::get_class_handle()`, `TrickHLA::Object::get_class_handle()`, `inter_count`, `interactions`, `is_RTI_ready()`, `obj_count`, `objects`, `TrickHLA::ExecutionControlBase::unpublish()`, `TrickHLA::Object::unpublish_all_object_attributes()`, and `TrickHLA::Interaction::unpublish_interaction()`.

#### 7.34.3.83 `unsubscribe()`

```
void Manager::unsubscribe ( )
```

Unsubscribe from the [Object](#) and [Interaction](#) classes.

##### Trick Job Class: *initialization*

Definition at line 1554 of file Manager.cpp.

References `execution_control`, `TrickHLA::Interaction::get_class_handle()`, `TrickHLA::Object::get_class_handle()`, `inter_count`, `interactions`, `is_RTI_ready()`, `obj_count`, `objects`, `TrickHLA::ExecutionControlBase::unsubscribe()`, `TrickHLA::Object::unsubscribe_all_object_attributes()`, and `TrickHLA::Interaction::unsubscribe_from_interaction()`.

#### 7.34.3.84 wait\_for\_init\_sync\_point()

```
void Manager::wait_for_init_sync_point (
    const char * sync_point_label )
```

Achieve then wait for the federation to become synchronized for the specified sync-point label.

##### Parameters

<code>sync_point_label</code>	Name of the synchronization point label.
-------------------------------	--

##### Trick Job Class: *initialization*

Definition at line 621 of file Manager.cpp.

References TrickHLA::SyncPntListBase::achieve\_and\_wait\_for\_synchronization(), TrickHLA::SyncPntListBase::contains(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, execution\_control, federate, get\_execution\_control(), get\_RTI\_ambassador(), is\_late\_joining\_federate(), TrickHLA::DebugHandler::should\_print(), THLA\_ENDL, THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_wstring().

#### 7.34.3.85 wait\_on\_discovery\_of\_objects()

```
void Manager::wait_on_discovery_of_objects ( )
```

Waits on the discovery of object instances.

Calling this function will block until object instances have been discovered. **Trick Job Class: *initialization***

Definition at line 2910 of file Manager.cpp.

References TrickHLA::Federate::check\_for\_shutdown\_with\_termination(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, federate, TrickHLA::Federate::is\_execution\_member(), TrickHLA::Object::is\_instance\_handle\_valid(), obj\_count, objects, TrickHLA::DebugHandler::should\_print(), THLA\_ENDL, and THLA\_NEWLINE.

Referenced by IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.34.3.86 wait\_on\_registration\_of\_required\_objects()

```
void Manager::wait_on_registration_of_required_objects ( )
```

Calling this function will block until all the required object instances in the Federation have been registered. **Trick Job Class: *initialization***

Definition at line 1727 of file Manager.cpp.

References TrickHLA::Federate::check\_for\_shutdown\_with\_termination(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, federate, get\_execution\_configuration(), TrickHLA::Object::get\_FOM\_name(), TrickHLA::Object::get\_instance\_handle(), TrickHLA::Object::get\_name(), is\_execution\_configuration\_used(), TrickHLA::Federate::is\_execution\_member(), TrickHLA::Object::is\_instance\_handle\_valid(), TrickHLA::Object::is\_required(), obj\_count, object\_map, objects, TrickHLA::DebugHandler::should\_print(), THLA\_ENDL, THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

Referenced by SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.34.3.87 wait\_on\_reservation\_of\_object\_names()

```
void Manager::wait_on_reservation_of_object_names ( )
```

Waits on the reservation of the RTI object instance names for the locally owned objects.

Calling this function will block until all the object instances names for the locally owned objects have been reserved.

##### Trick Job Class: *initialization*

Definition at line 1643 of file Manager.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_MANAGER, obj\_count, objects, TrickHLA::DebugHandler::should\_print(), THLA\_NEWLINE, and TrickHLA::Object::wait\_on\_object\_name\_reservation().

Referenced by SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

## 7.34.4 Friends And Related Function Documentation

### 7.34.4.1 Federate

```
friend class Federate [friend]
```

Definition at line 98 of file Manager.hh.

### 7.34.4.2 init\_attrTrickHLA\_\_Manager

```
void init_attrTrickHLA__Manager ( ) [friend]
```

### 7.34.4.3 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 93 of file Manager.hh.

## 7.34.5 Field Documentation

### 7.34.5.1 check\_interactions

```
InteractionItem* TrickHLA::Manager::check_interactions [private]
```

#### Units: –

checkpoint-able version of interactions\_queue

Definition at line 453 of file Manager.hh.

Referenced by clear\_interactions(), dump\_interactions(), restore\_interactions(), and setup\_checkpoint\_interactions().

### 7.34.5.2 check\_interactions\_count

```
int TrickHLA::Manager::check_interactions_count [private]
```

#### Units: –

Number of checkpointed interactions

Definition at line 452 of file Manager.hh.

Referenced by clear\_interactions(), dump\_interactions(), restore\_interactions(), and setup\_checkpoint\_interactions().

### 7.34.5.3 debug\_handler

```
DebugHandler TrickHLA::Manager::debug_handler
```

#### Units: –

Decides whether to print any debug messages.

Definition at line 110 of file Manager.hh.

Referenced by `determine_job_cycle_time()`, `discover_object_instance()`, `get_debug_handler()`, `initialize()`, `mark_object_as_deleted_from_federation()`, `object_instance_name_reservation_succeeded()`, `process_interactions()`, `publish_and_subscribe()`, `receive_cyclic_data()`, `receive_init_data()`, `receive_interaction()`, `register_objects_with_RTI()`, `request_data_update()`, `reserve_object_names_with_RTI()`, `restart_initialization()`, `restore_interactions()`, `send_cyclic_data()`, `send_init_data()`, `send_requested_data()`, `set_object_instance_handles_by_name()`, `setup_all_ref_attributes()`, `setup_all_RTI_handles()`, `setup_checkpoint_interactions()`, `setup_interaction_ref_attributes()`, `setup_interaction_RTI_handles()`, `setup_object_ref_attributes()`, `setup_object_RTI_handles()`, `setup_preferred_order_with_RTI()`, `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`, `should_print()`, `wait_for_init_sync_point()`, `wait_on_discovery_of_objects()`, `wait_on_registration_of_required_objects()`, and `wait_on_reservation_of_object_names()`.

#### 7.34.5.4 execution\_control

`ExecutionControlBase* TrickHLA::Manager::execution_control` [private]

**Units:** –

Execution control object.

This has to point to an allocated execution control class that inherits from the `ExecutionControlBase` interface class. For instance `SRFOM::ExecutionControl`.

Definition at line 469 of file Manager.hh.

Referenced by `clear_init_sync_points()`, `get_execution_configuration()`, `get_execution_control()`, `get_trickhla_object()`, `get_unregistered_object()`, `get_unregistered_remote_object()`, `initialize()`, `is_execution_configuration_used()`, `is_late_joining_federate()`, `mark_object_as_deleted_from_federation()`, `object_instance_name_reservation_failed()`, `object_instance_name_reservation_succeeded()`, `process_deleted_objects()`, `process_interactions()`, `provide_attribute_update()`, `publish()`, `receive_cyclic_data()`, `receive_init_data()`, `receive_interaction()`, `register_objects_with_RTI()`, `restart_initialization()`, `send_requested_data()`, `set_execution_configuration()`, `setup()`, `setup_all_RTI_handles()`, `setup_checkpoint()`, `setup_interaction_ref_attributes()`, `start_federation_save_at_scenario_time()`, `start_federation_save_at_sim_time()`, `subscribe()`, `unsubscribe()`, and `wait_for_init_sync_point()`.

#### 7.34.5.5 federate

`Federate* TrickHLA::Manager::federate` [private]

**Units:** –

Associated `TrickHLA Federate`.

Definition at line 467 of file Manager.hh.

Referenced by `determine_job_cycle_time()`, `discover_object_instance()`, `get_fed_lookahead()`, `get_federate()`, `get_granted_fed_time()`, `get_granted_time()`, `get_RTI_ambassador()`, `initialize()`, `initiate_federation_save()`, `is_RTI_ready()`, `receive_init_data()`, `restart_initialization()`, `set_object_instance_handles_by_name()`, `setup()`, `setup_interaction_RTI_handles()`, `setup_object_RTI_handles()`, `wait_for_init_sync_point()`, `wait_on_discovery_of_objects()`, `wait_on_registration_of_required_objects()`, and `~Manager()`.

#### 7.34.5.6 federate\_has\_been\_restored

`bool TrickHLA::Manager::federate_has_been_restored` [private]

**Data I/O:** \*\*

`Federate` has been restored. do not reserve the object names again!

Definition at line 465 of file Manager.hh.

Referenced by `has_federate_been_restored()`, and `reset_mgr_initialized()`.

#### 7.34.5.7 initiated\_a\_federation\_save

`bool TrickHLA::Manager::initiated_a_federation_save`

**Data I/O: \*\***

did this manager initiate the federation save?  
Definition at line 114 of file Manager.hh.  
Referenced by `initiate_federation_save()`.

#### 7.34.5.8 inter\_count

```
int TrickHLA::Manager::inter_count
```

**Units: –**

Number of [TrickHLA](#) Interactions.  
Definition at line 107 of file Manager.hh.  
Referenced by `get_interaction_count()`, `initialize()`, `process_interactions()`, `publish()`, `receive_interaction()`, `restart_initialization()`, `setup_all_RTI_handles()`, `setup_interaction_ref_attributes()`, `setup_preferred_order_with_RTI()`, `subscribe()`, `unpublish()`, and `unsubscribe()`.

#### 7.34.5.9 interactions

```
Interaction* TrickHLA::Manager::interactions
```

**Units: –**

Array of [TrickHLA](#) Interactions.  
Definition at line 108 of file Manager.hh.  
Referenced by `get_interactions()`, `initialize()`, `process_interactions()`, `publish()`, `receive_interaction()`, `restart_initialization()`, `setup_all_RTI_handles()`, `setup_interaction_ref_attributes()`, `setup_preferred_order_with_RTI()`, `subscribe()`, `unpublish()`, and `unsubscribe()`.

#### 7.34.5.10 interactions\_queue

```
ItemQueue TrickHLA::Manager::interactions_queue [private]
```

**Data I/O: \*\***

Interactions queue.  
Definition at line 450 of file Manager.hh.  
Referenced by `process_interactions()`, `receive_interaction()`, `restore_interactions()`, and `setup_checkpoint_interactions()`.

#### 7.34.5.11 job\_cycle\_time

```
double TrickHLA::Manager::job_cycle_time [private]
```

Definition at line 455 of file Manager.hh.

Referenced by `determine_job_cycle_time()`, `receive_cyclic_data()`, `send_cyclic_data()`, and `send_requested_data()`.

#### 7.34.5.12 mgr\_initialized

```
bool TrickHLA::Manager::mgr_initialized [private]
```

**Units: –**

Internal flag to indicate [Manager](#) is initialized.  
Definition at line 461 of file Manager.hh.  
Referenced by `initialize()`, `reset_mgr_initialized()`, `restart_initialization()`, `set_all_object_instance_handles_by_name()`, `set_object_instance_handles_by_name()`, `setup_interaction_ref_attributes()`, `setup_interaction_RTI_handles()`, `setup_object_ref_attributes()`, and `setup_object_RTI_handles()`.

### 7.34.5.13 obj\_count

```
int TrickHLA::Manager::obj_count
```

**Units:** –

Number of [TrickHLA](#) Objects.

Definition at line 104 of file Manager.hh.

Referenced by `determine_job_cycle_time()`, `get_object_count()`, `get_trickhla_object()`, `get_unregistered_object()`, `get_unregistered_remote_object()`, `grant_pull_request()`, `initialize()`, `is_this_a_rejoining_federate()`, `object_instance_name_reservation_failed()`, `process_deleted_objects()`, `publish()`, `pull_ownership()`, `pull_ownership_upon_rejoin()`, `push_ownership()`, `receive_cyclic_data()`, `receive_init_data()`, `register_objects_with_RTI()`, `release_ownership()`, `reserve_object_names_with_RTI()`, `restart_initialization()`, `send_cyclic_data()`, `send_init_data()`, `send_requested_data()`, `set_all_object_instance_handles_by_name()`, `setup_all_ref_attributes()`, `setup_all_RTI_handles()`, `setup_checkpoint()`, `setup_preferred_order_with_RTI()`, `subscribe()`, `unpublish()`, `unsubscribe()`, `wait_on_discovery_of_objects()`, `wait_on_registration_of_required_objects()`, and `wait_on_reservation_of_object_names()`.

### 7.34.5.14 object\_map

```
ObjectInstanceMap TrickHLA::Manager::object_map [private]
```

**Data I/O:** \*\*

Map of all the Objects this federate uses, the Key is the object instance-handle.

Definition at line 463 of file Manager.hh.

Referenced by `add_object_to_map()`, `discover_object_instance()`, `get_trickhla_object()`, `register_objects_with_RTI()`, `set_all_object_instance_handles_by_name()`, `set_object_instance_handles_by_name()`, `setup_all_ref_attributes()`, `wait_on_registration_of_required_objects()`, and `~Manager()`.

### 7.34.5.15 objects

```
Object* TrickHLA::Manager::objects
```

**Units:** –

Array of [TrickHLA](#) object.

Definition at line 105 of file Manager.hh.

Referenced by `determine_job_cycle_time()`, `get_objects()`, `get_trickhla_object()`, `get_unregistered_object()`, `get_unregistered_remote_object()`, `grant_pull_request()`, `initialize()`, `is_this_a_rejoining_federate()`, `object_instance_name_reservation_failed()`, `process_deleted_objects()`, `publish()`, `pull_ownership()`, `pull_ownership_upon_rejoin()`, `push_ownership()`, `receive_cyclic_data()`, `receive_init_data()`, `register_objects_with_RTI()`, `release_ownership()`, `reserve_object_names_with_RTI()`, `restart_initialization()`, `send_cyclic_data()`, `send_init_data()`, `send_requested_data()`, `set_all_object_instance_handles_by_name()`, `setup_all_ref_attributes()`, `setup_all_RTI_handles()`, `setup_checkpoint()`, `setup_preferred_order_with_RTI()`, `subscribe()`, `unpublish()`, `unsubscribe()`, `wait_on_discovery_of_objects()`, `wait_on_registration_of_required_objects()`, and `wait_on_reservation_of_object_names()`.

### 7.34.5.16 rejoicing\_federate

```
bool TrickHLA::Manager::rejoining_federate [private]
```

**Units:** –

Internal flag to indicate if the federate is rejoicing the federation.

Definition at line 457 of file Manager.hh.

Referenced by `is_this_a_rejoining_federate()`.

### 7.34.5.17 restore\_determined

```
bool TrickHLA::Manager::restore_determined [private]
```

**Data I/O: \*\***

Internal flag to indicate that the restore status has been determined.

Definition at line 458 of file Manager.hh.

Referenced by IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate().

**7.34.5.18 restore\_federate**

```
bool TrickHLA::Manager::restore_federate [private]
```

**Data I/O: \*\***

Internal flag to indicate if the federate is to be restored

Definition at line 459 of file Manager.hh.

Referenced by IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate(), and is\_restore\_federate().

**7.34.5.19 restore\_federation**

```
bool TrickHLA::Manager::restore_federation
```

**Data I/O: \*i****Units: -**

flag indicating whether to trigger the restore

Definition at line 112 of file Manager.hh.

**7.34.5.20 restore\_file\_name**

```
char* TrickHLA::Manager::restore_file_name
```

**Data I/O: \*i****Units: -**

file name, which will be the label name

Definition at line 113 of file Manager.hh.

**7.34.5.21 shutdown\_called**

```
bool TrickHLA::Manager::shutdown_called [private]
```

**Units: -**

Flag to indicate that shutdown has been called.

Definition at line 448 of file Manager.hh.

Referenced by is\_shutdown(), and shutdown().

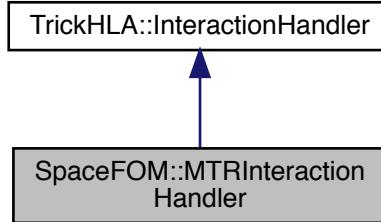
The documentation for this class was generated from the following files:

- [Manager.hh](#)
- [Manager.cpp](#)

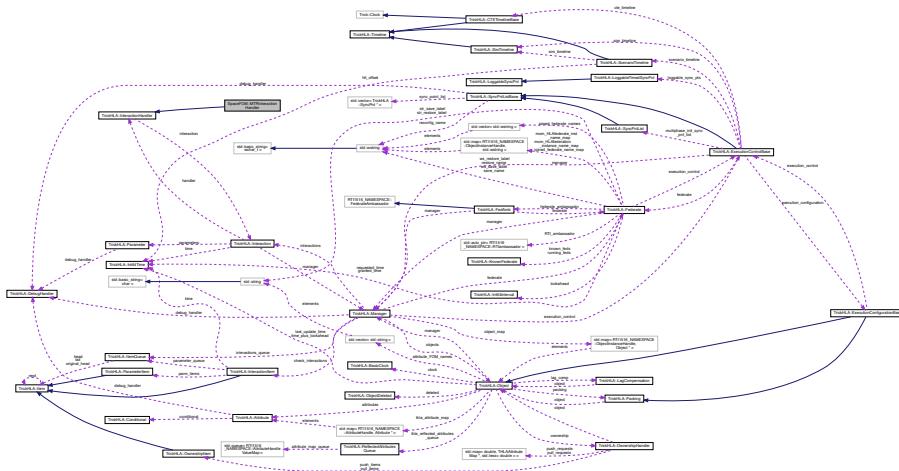
**7.35 SpaceFOM::MTRInteractionHandler Class Reference**

```
#include <MTRInteractionHandler.hh>
```

Inheritance diagram for SpaceFOM::MTRInteractionHandler:



Collaboration diagram for SpaceFOM::MTRInteractionHandler:



## Public Member Functions

- **MTRInteractionHandler (TrickHLA::Federate \*fed)**  
*Initialization constructor for the [SpaceFOM MTRInteractionHandler](#) class.*
- **virtual ~MTRInteractionHandler ()**  
*Destructor for the [SpaceFOM MTRInteractionHandler](#) class.*
- **virtual void send\_interaction (MTREnum mode\_request)**  
*Send the HLA interaction.*
- **virtual void receive\_interaction (RTI1516\_USERDATA const &the\_user\_supplied\_tag)**  
*Receive the HLA interaction.*
- **virtual void set\_name (const char \*new\_name)**  
*Set the associated name for this interaction handler.*
- **int16\_t \* get\_address\_of\_interaction\_mode ()**  
*Get the address of the MTR interaction mode transition state.*

## Data Fields

- `char * name`  
**Units:** –  
*Federation instance name for this interaction.*
- `MTREnum mtr_mode`  
**Units:** –  
*Requested mode transition state.*
- `int16_t mtr_mode_int`  
**Units:** –  
*Requested mode transition state (integer version).*

## Protected Attributes

- `double scenario_time`  
**Units:** *s*  
*Scenario time when MTR was sent/received.*
- `double sim_time`  
**Units:** *s*  
*Simulation time when MTR was sent/received.*
- `double cte_time`  
**Units:** *s*  
*CTE time when MTR was sent/received, if CTE used.*
- `double granted_time`  
**Units:** *s*  
*HLA granted time when MTR was sent/received.*
- `int send_cnt`  
**Units:** *count*  
*The number of times an interaction is sent.*
- `int receive_cnt`  
**Units:** *count*  
*The number of times an interaction was received.*

## Private Member Functions

- `MTRInteractionHandler ()`  
*Default constructor for `MTRInteractionHandler` class.*
- `MTRInteractionHandler (const MTRInteractionHandler &rhs)`  
*Copy constructor for `MTRInteractionHandler` class.*
- `MTRInteractionHandler & operator= (const MTRInteractionHandler &rhs)`  
*Assignment operator for `MTRInteractionHandler` class.*

## Friends

- `class InputProcessor`
- `void init_attrSpaceFOM__MTRInteractionHandler ()`

### 7.35.1 Detailed Description

Definition at line 60 of file MTRInteractionHandler.hh.

## 7.35.2 Constructor & Destructor Documentation

### 7.35.2.1 MTRInteractionHandler() [1/3]

```
MTRInteractionHandler::MTRInteractionHandler (
    TrickHLA::Federate * fed ) [explicit]
Initialization constructor for the SpaceFOM MTRInteractionHandler class.
```

#### Parameters

<i>fed</i>	<a href="#">TrickHLA::Federate</a> associated with this <a href="#">MTRInteractionHandler</a> instance.
------------	---

#### Trick Job Class: *initialization*

Definition at line 57 of file [MTRInteractionHandler.cpp](#).

### 7.35.2.2 ~MTRInteractionHandler()

```
MTRInteractionHandler::~MTRInteractionHandler ( ) [virtual]
Destructor for the SpaceFOM MTRInteractionHandler class.
```

#### Trick Job Class: *shutdown*

Definition at line 75 of file [MTRInteractionHandler.cpp](#).

References name, and [trick\\_MM](#).

### 7.35.2.3 MTRInteractionHandler() [2/3]

```
SpaceFOM::MTRInteractionHandler::MTRInteractionHandler ( ) [private]
Default constructor for MTRInteractionHandler class.
This constructor is private to prevent instantiation without an associated TrickHLA::Federate.
```

### 7.35.2.4 MTRInteractionHandler() [3/3]

```
SpaceFOM::MTRInteractionHandler::MTRInteractionHandler (
    const MTRInteractionHandler & rhs ) [private]
Copy constructor for MTRInteractionHandler class.
This constructor is private to prevent inadvertent copies.
```

## 7.35.3 Member Function Documentation

### 7.35.3.1 get\_address\_of\_interaction\_mode()

```
int16_t* SpaceFOM::MTRInteractionHandler::get_address_of_interaction_mode ( ) [inline]
Get the address of the MTR interaction mode transition state.
```

#### Returns

Address of the MTR interaction mode transition state.

Definition at line 98 of file [MTRInteractionHandler.hh](#).

References [mtr\\_mode\\_int](#).

Referenced by [SpaceFOM::ExecutionControl::setup\\_interaction\\_ref\\_attributes\(\)](#).

### 7.35.3.2 operator=( )

```
MTRInteractionHandler& SpaceFOM::MTRInteractionHandler::operator= (
    const MTRInteractionHandler & rhs ) [private]
```

Assignment operator for `MTRInteractionHandler` class.

This assignment operator is private to prevent inadvertent copies.

### 7.35.3.3 receive\_interaction()

```
void MTRInteractionHandler::receive_interaction (
    RTI1516_USERDATA const & the_user_supplied_tag ) [virtual]
```

Receive the HLA interaction.

#### Parameters

<code>the_user_supplied_tag</code>	User supplied interaction tag.
------------------------------------	--------------------------------

Reimplemented from [TrickHLA::InteractionHandler](#).

Definition at line 193 of file `MTRInteractionHandler.cpp`.

References `cte_time`, `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_SOURCE_INTERACTION`, `TrickHLA::ExecutionControlBase::does_cte_timeline_exist()`, `TrickHLA::InteractionHandler::get_cte_time()`, `TrickHLA::Federate::get_execution_control()`, `TrickHLA::Interaction::get_federate()`, `TrickHLA::Federate::get_granted_time()`, `TrickHLA::InteractionHandler::get_scenario_time()`, `TrickHLA::InteractionHandler::get_sim_time()`, `granted_time`, `TrickHLA::InteractionHandler::interaction`, `SpaceFOM::ExecutionControl::is_mtr_valid()`, `SpaceFOM::mtr_enum_to_string()`, `SpaceFOM::mtr_int16_to_enum()`, `mtr_mode`, `mtr_mode_int`, `name`, `receive_cnt`, `scenario_time`, `TrickHLA::ExecutionControlBase::set_mode_transition_requested()`, `SpaceFOM::ExecutionControl::set_pending_mtr()`, `TrickHLA::InteractionHandler::should_print()`, `sim_time`, `THLA_ENDL`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, and `TrickHLA::Int64Interval::toMicroseconds()`.

### 7.35.3.4 send\_interaction()

```
void MTRInteractionHandler::send_interaction (
    MTREnum mode_request ) [virtual]
```

Send the HLA interaction.

#### Parameters

<code>mode_request</code>	Requested mode transition.
---------------------------	----------------------------

#### Assumptions and Limitations:

- Timestamp Order or Receive Order is determined at compile time.

Definition at line 105 of file `MTRInteractionHandler.cpp`.

References `cte_time`, `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_SOURCE_INTERACTION`, `TrickHLA::ExecutionControlBase::does_cte_timeline_exist()`, `TrickHLA::InteractionHandler::get_cte_time()`, `TrickHLA::Federate::get_execution_control()`, `TrickHLA::Interaction::get_federate()`, `TrickHLA::Federate::get_granted_time()`, `TrickHLA::InteractionHandler::get_scenario_time()`, `TrickHLA::InteractionHandler::get_sim_time()`, `granted_time`, `TrickHLA::InteractionHandler::interaction`, `SpaceFOM::mtr_enum_to_int16()`, `SpaceFOM::mtr_enum_to_string()`, `mtr_mode`, `mtr_mode_int`, `name`, `RTI1516_USERDATA`, `scenario_time`, `send_cnt`, `TrickHLA::InteractionHandler::send_interaction()`, `TrickHLA::InteractionHandler::should_print()`, `sim_time`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, and `TrickHLA::Int64Interval::toMicroseconds()`.

Referenced by `SpaceFOM::ExecutionControl::send_MTR_interaction()`.

### 7.35.3.5 `set_name()`

```
void MTRInteractionHandler::set_name (
    const char * new_name ) [virtual]
```

Set the associated name for this interaction handler.

#### Parameters

<code>new_name</code>	Associated name.
-----------------------	------------------

#### Trick Job Class: `default_data`

Definition at line 87 of file MTRInteractionHandler.cpp.

References name, and trick\_MM.

Referenced by SpaceFOM::ExecutionControl::setup\_interaction\_ref\_attributes().

## 7.35.4 Friends And Related Function Documentation

### 7.35.4.1 `init_attrSpaceFOM__MTRInteractionHandler`

```
void init_attrSpaceFOM__MTRInteractionHandler ( ) [friend]
```

### 7.35.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 68 of file MTRInteractionHandler.hh.

## 7.35.5 Field Documentation

### 7.35.5.1 `cte_time`

```
double SpaceFOM::MTRInteractionHandler::cte_time [protected]
```

#### Units: s

CTE time when MTR was sent/received, if CTE used.

Definition at line 108 of file MTRInteractionHandler.hh.

Referenced by receive\_interaction(), and send\_interaction().

### 7.35.5.2 `granted_time`

```
double SpaceFOM::MTRInteractionHandler::granted_time [protected]
```

#### Units: s

HLA granted time when MTR was sent/received.

Definition at line 109 of file MTRInteractionHandler.hh.

Referenced by receive\_interaction(), and send\_interaction().

### 7.35.5.3 `mtr_mode`

[MTREnum](#) SpaceFOM::MTRInteractionHandler::mtr\_mode

**Units:** –

Requested mode transition state.

Definition at line 102 of file MTRInteractionHandler.hh.

Referenced by receive\_interaction(), and send\_interaction().

#### 7.35.5.4 mtr\_mode\_int

```
int16_t SpaceFOM::MTRInteractionHandler::mtr_mode_int
```

**Units:** –

Requested mode transition state (integer version).

Definition at line 103 of file MTRInteractionHandler.hh.

Referenced by get\_address\_of\_interaction\_mode(), receive\_interaction(), and send\_interaction().

#### 7.35.5.5 name

```
char* SpaceFOM::MTRInteractionHandler::name
```

**Units:** –

Federation instance name for this interaction.

Definition at line 101 of file MTRInteractionHandler.hh.

Referenced by receive\_interaction(), send\_interaction(), set\_name(), and ~MTRInteractionHandler().

#### 7.35.5.6 receive\_cnt

```
int SpaceFOM::MTRInteractionHandler::receive_cnt [protected]
```

**Units:** count

The number of times an interaction was received.

Definition at line 112 of file MTRInteractionHandler.hh.

Referenced by receive\_interaction().

#### 7.35.5.7 scenario\_time

```
double SpaceFOM::MTRInteractionHandler::scenario_time [protected]
```

**Units:** s

Scenario time when MTR was sent/received.

Definition at line 106 of file MTRInteractionHandler.hh.

Referenced by receive\_interaction(), and send\_interaction().

#### 7.35.5.8 send\_cnt

```
int SpaceFOM::MTRInteractionHandler::send_cnt [protected]
```

**Units:** count

The number of times an interaction is sent.

Definition at line 111 of file MTRInteractionHandler.hh.

Referenced by send\_interaction().

#### 7.35.5.9 sim\_time

```
double SpaceFOM::MTRInteractionHandler::sim_time [protected]
```

**Units:** s

Simulation time when MTR was sent/received.

Definition at line 107 of file MTRInteractionHandler.hh.

Referenced by receive\_interaction(), and send\_interaction().

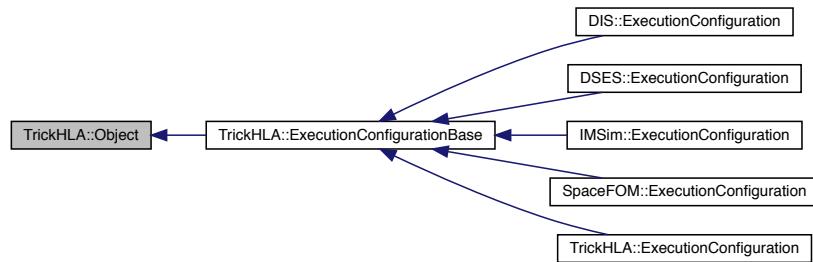
The documentation for this class was generated from the following files:

- [MTRInteractionHandler.hh](#)
- [MTRInteractionHandler.cpp](#)

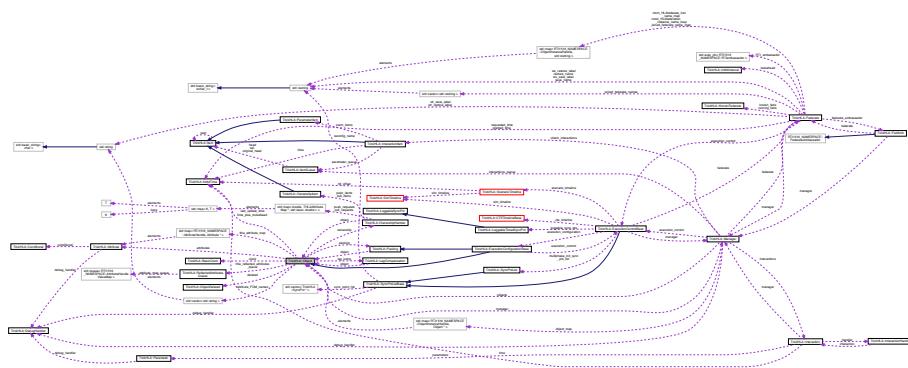
## 7.36 TrickHLA::Object Class Reference

```
#include <Object.hh>
```

Inheritance diagram for TrickHLA::Object:



Collaboration diagram for TrickHLA::Object:



## Public Member Functions

- [Object \(\)](#)  
*Default constructor for the [TrickHLA Object](#) class.*
- [virtual ~Object \(\)](#)  
*Destructor for the [TrickHLA Object](#) class.*
- [virtual void initialize \(Manager \\*manager\)](#)  
*Initializes the [TrickHLA Object](#).*
- [Federate \\* get\\_federate \(\)](#)

- Gets the a pointer to our federate.
  - void `publish_object_attributes ()`  
Publishes *Object* attributes.
  - void `unpublish_all_object_attributes ()`  
Unpublishes all object attributes.
  - void `subscribe_to_object_attributes ()`  
Subscribe to *Object* attributes.
  - void `unsubscribe_all_object_attributes ()`  
Unsubscribe from all the *Object* attributes.
  - void `reserve_object_name_with_RTI ()`  
Reserves a unique object instance name with the RTI.
- void `wait_on_object_name_reservation ()`  
Waits on the reservation of the HLA object instance name with the RTI.
- void `register_object_with_RTI ()`  
Creates an HLA object instance and registers it with the RTI, but only if we own it locally.
- void `wait_on_object_registration ()`  
Waits on the registration of this HLA object instance with the RTI.
- void `setup_preferred_order_with_RTI ()`  
Setup the preferred order for the locally owned attributes.
- void `set_core_job_cycle_time (double cycle_time)`  
Sets the core job cycle time used by the multi-rate attributes.
- void `remove_object_instance ()`  
Marks this object as deleted from the RTI and sets all attributes as non-local.
- bool `is_object_deleted_from_RTI () const`  
Check if object has been deleted from the RTI.
- void `process_deleted_object ()`  
This object instance has been deleted from the RTI, so process the delete action by calling the users delete notification callback.
- void `send_requested_data (double current_time, double cycle_time)`  
Send the requested data update that came from another federate requesting an attribute value update.
- void `send_cyclic_data (double current_time, double cycle_time)`  
Send cyclic data to remote HLA federates.
- void `receive_cyclic_data (double current_time, double cycle_time)`  
Handle the received cyclic data.
- void `send_init_data ()`  
Send initialization data to remote HLA federates.
- void `receive_init_data ()`  
Receive initialization data from remote Federates.
- void `request_attribute_value_update ()`  
Request an update to the attributes for this object.
- void `provide_attribute_update (RTI1516_NAMESPACE::AttributeHandleSet const &theAttributes)`  
Requesting an attribute update for the specified attributes.
- void `extract_data (RTI1516_NAMESPACE::AttributeHandleValueMap &theAttributes)`  
This function extracts the new attribute values.
- void `remove ()`  
Remove this object instance from the RTI/Federation.
- void `release_ownership ()`

- void [pull\\_ownership \(\)](#)

*This function releases ownership of the attributes for this object.*
- void [pull\\_ownership\\_upon\\_rejoin \(\)](#)

*This function pulls ownership for this object.*
- void [grant\\_pull\\_request \(\)](#)

*This function pulls ownership for all published attributes when the federate rejoins an already running federation.*
- void [push\\_ownership \(\)](#)

*This function grants a pull request for this object.*
- void [grant\\_push\\_request \(\)](#)

*This function pushes ownership for this object.*
- void [grant\\_push\\_request \(\)](#)

*This function grants a previously "recorded" push request for this object.*
- void [grant\\_push\\_request\\_pthread \(\)](#)

*This function starts a thread to service the push request grant.*
- void [negotiated\\_attribute\\_ownership\\_divestiture \(RTI1516\\_NAMESPACE::AttributeHandleSet \\*attr\\_hdl\\_set\)](#)

*This function handles the RTI call for the negotiated attribute ownership divestiture.*
- void [setup\\_ownership\\_transfer\\_checkpointed\\_data \(\)](#)

*Setup the checkpoint data structures.*
- void [restore\\_ownership\\_transfer\\_checkpointed\\_data \(\)](#)

*If an ownership\_transfer object has been created by the user, trigger its restore() method is re-establish the pull / push AttributeOwnershipMaps.*
- const char \* [get\\_name \(\) const](#)

*Get the object instance name.*
- const std::string [get\\_name\\_string \(\) const](#)

*Get the object instance name as a C++ string.*
- bool [is\\_name\\_required \(\) const](#)

*Check if an object instance name is required.*
- bool [is\\_name\\_registered \(\) const](#)

*Check if an object instance name is registered.*
- void [set\\_name\\_registered \(\)](#)

*Set the name registration status as true (registered).*
- void [set\\_name\\_unregistered \(\)](#)

*Set the name registration status as false (not registered).*
- const char \* [get\\_FOM\\_name \(\) const](#)

*Get the FOM name for this object.*
- RTI1516\_NAMESPACE::ObjectClassHandle [get\\_class\\_handle \(\) const](#)

*Get the HLA [Object](#) class handle for this object.*
- void [set\\_class\\_handle \(RTI1516\\_NAMESPACE::ObjectClassHandle id\)](#)

*Set the HLA [Object](#) class handle for this object.*
- bool [is\\_instance\\_handle\\_valid \(\) const](#)

*Check if the HLA [Object](#) instance handle is valid for this object.*
- RTI1516\_NAMESPACE::ObjectInstanceHandle [get\\_instance\\_handle \(\) const](#)

*Get the HLA [Object](#) instance handle for this object instance.*
- void [set\\_instance\\_handle \(RTI1516\\_NAMESPACE::ObjectInstanceHandle id\)](#)

*Set the HLA [Object](#) instance handle for this object instance.*
- void [set\\_instance\\_handle\\_and\\_name \(RTI1516\\_NAMESPACE::ObjectInstanceHandle id, std::wstring const &instance\\_name\)](#)

*Set the HLA [Object](#) instance handle and name for this object instance.*

- `bool is_create_HLA_instance () const`  
*Check if the object instance has been created.*
- `void set_create_HLA_instance (bool create)`  
*Set the object instance creation status.*
- `bool is_required () const`  
*Check if the object is a required object instance.*
- `void mark_required ()`  
*Mark this object instance as required.*
- `void stop_publishing_attributes ()`  
*Stops publishing data for the object attributes by setting the attribute publish state to false.*
- `void stop_subscribing_attributes ()`  
*Stops subscribing to object attribute data by setting the attribute subscribe state to false.*
- `bool is_attribute_update_requested () const`  
*Check an attribute update has been requested for an attribute associated with this object instance.*
- `bool any_attribute_published ()`  
*Determines if any attribute is published.*
- `bool any_attribute_subscribed ()`  
*Determines if any attribute is being subscribed to.*
- `bool any_locally_owned_attribute ()`  
*Determines if any attribute is locally owned.*
- `bool any_locally_owned_published_attribute ()`  
*Determines if any attribute is locally owned and published.*
- `bool any_locally_owned_published_attribute (const DataUpdateEnum attr_config)`  
*Determines if any attribute is locally owned and published for the given attribute configuration.*
- `bool any_locally_owned_published_cyclic_data_ready_attribute ()`  
*Determines if any attribute is locally owned, published, and has a cycle-time that is ready for a cyclic send.*
- `bool any_locally_owned_published_init_attribute ()`  
*Determines if any attribute update is locally owned and published at initialization.*
- `bool any_locally_owned_published_requested_attribute ()`  
*Determines if any attribute update is locally owned and published for the attribute value update.*
- `bool any_remotely_owned_subscribed_attribute ()`  
*Determines if any attribute is remotely owned and subscribed to.*
- `bool any_remotely_owned_subscribed_attribute (const DataUpdateEnum attr_config)`  
*Determines if any attribute specified is remotely owned and subscribed to for the given attribute configuration.*
- `bool any_remotely_owned_subscribed_cyclic_attribute ()`  
*Determines if any cyclically updated attributes are remotely owned and subscribed.*
- `bool any_remotely_owned_subscribed_init_attribute ()`  
*Determines if any initialization updated attributes are remotely owned and subscribed.*
- `void mark_all_attributes_as_nonlocal ()`  
*Turn off local flag in all attributes.*
- `bool is_changed ()`  
*Used by the `TrickHLA` extension to determine if the object data changed.*
- `void mark_changed ()`  
*Mark the data as changed, and notify any waiting thread that there is a change.*
- `void mark_unchanged ()`  
*Mark the data as unchanged, and clear the change flag for all the attributes as well.*
- `void set_to_unblocking_cyclic_reads ()`

- `void notify_attribute_ownership_changed ()`

*Set to unblocking cyclic reads and notify any waiting threads.*
- `void set_lag_compensation_type (LagCompensationEnum lag_type)`

*Notify any waiting threads of a change in attribute ownership, which could affect blocking reads.*
- `LagCompensationEnum get_lag_compensation_type () const`

*Set the Lag Compensation type for object attribute updates.*
- `void set_divest_requested (bool request)`

*Get the current Lag Compensation type for this object instance attribute updates.*
- `void set_pull_requested (bool request)`

*Set the ownership divestiture requested flag.*
- `const Int64Time & get_last_update_time () const`

*Set ownership pull requested flag.*
- `Get the last update time.`
- `Int64Interval get_fed_lookahead () const`

*Return a copy of the federate's lookahead time.*
- `Int64Time get_granted_fed_time () const`

*Get the currently granted federation HLA logical time.*
- `void set_last_update_time (RTI1516_NAMESPACE::LogicalTime const &time)`

*Set the last update time.*
- `const Int64Time & get_update_time_plus_lookahead ()`

*Gets the updated time plus the lookahead time.*
- `double get_granted_time () const`

*Get the granted HLA time.*
- `Attribute * get_attribute (RTI1516_NAMESPACE::AttributeHandle attr_handle)`

*Gets the attribute for the given HLA Attribute-Handle.*
- `Attribute * get_attribute (std::string attr_FOM_name)`

*Gets the attribute for the given FOM name.*
- `Attribute * get_attribute (const char *attr_FOM_name)`

*Gets the attribute for the given FOM name.*
- `int get_attribute_count () const`

*Get the count of the number of attributes associated with this object.*
- `Attribute * get_attributes ()`

*Get the list of attributes associated with this object.*
- `void build_attribute_map ()`

*Build the attribute map, which will be used for quickly looking up an attribute given its AttributeHandle.*
- `VectorOfStrings get_attribute_FOM_names () const`

*Get the attribute FOM names.*
- `void pack_requested_attribute_buffers ()`

*Pack the attributes that were part of the attribute value request into the buffer that is used for sending the encoded attribute through the RTI.*
- `void pack_attribute_buffers (const DataUpdateEnum attr_config)`

*Pack the attributes for the given configuration into the buffer that is used for sending the encoded attribute through the RTI.*
- `void unpack_attribute_buffers (const DataUpdateEnum attr_config)`

*Unpack the buffer back into the attributes that have the given configuration.*
- `void pack_cyclic_attribute_buffers ()`

*Copy the cyclic attribute values to the buffer for each attribute.*
- `void unpack_cyclic_attribute_buffers ()`

- void [pack\\_init\\_attribute\\_buffers \(\)](#)

*Copy the packed buffer contents back to each cyclic attribute.*
- void [unpack\\_init\\_attribute\\_buffers \(\)](#)

*Copy the dynamic initialization attribute values to the buffer for each attribute.*
- void [lock \(\)](#)

*Lock the thread mutex.*
- void [unlock \(\)](#)

*Unlock the thread mutex.*
- void [ownership\\_lock \(\)](#)

*Lock the ownership thread mutex.*
- void [ownership\\_unlock \(\)](#)

*Unlock the ownership thread mutex.*
- bool [should\\_print \(const DebugLevelEnum &level, const DebugSourceEnum &code\) const](#)

*Determine if the verbose debug comments should be printed to the console.*
- void [create\\_requested\\_attribute\\_set \(\)](#)

*Create a name value pair set, aka attribute handle value pair, for the attributes that were requested for this object.*
- void [create\\_attribute\\_set \(const DataUpdateEnum required\\_config\)](#)

*Create a name value pair set, aka attribute handle value pair, for the attributes of this object.*

## Data Fields

- bool [data\\_changed](#)

**Units:** –  
*Flag to indicate data changes.*
- char \* [name](#)

**Units:** –  
*Object Instance Name.*
- bool [name\\_required](#)

**Units:** –  
*True (default) to require an object instance name be specified by you, or false to use the instance name automatically assigned by the RTI.*
- char \* [FOM\\_name](#)

**Units:** –  
*FOM name for the object.*
- bool [create\\_HLA\\_instance](#)

**Units:** –  
*Set to true to create an HLA named instance of this object.*
- bool [required](#)

**Units:** –  
*Flag indicating object is required at federation start ( default: true )*
- bool [blocking\\_cyclic\\_read](#)

**Units:** –  
*True to block in receive\_cyclic\_data for data to be received.*
- int [attr\\_count](#)

**Units:** –  
*Number of object attributes.*
- Attribute \* [attributes](#)

**Units:** –  
*Array of object attributes.*

- `LagCompensation * lag_comp`

**Units:** –  
*Lag compensation object.*
- `LagCompensationEnum lag_comp_type`

**Units:** –  
*Type of lag compensation.*
- `Packing * packing`

**Units:** –  
*Data pack/unpack object.*
- `OwnershipHandler * ownership`

**Units:** –  
*Manages attribute ownership.*
- `ObjectDeleted * deleted`

**Units:** –  
*Object Deleted callback object.*
- `bool object_deleted_from_RTI`

**Units:** –  
*Flag that is true when this object has been deleted from the RTI.*

## Protected Member Functions

- `RTI1516_NAMESPACE::RTIambassador * get_RTI_ambassador ()`  
*Gets the RTI Ambassador.*

## Protected Attributes

- `pthread_mutex_t mutex`

**Data I/O:** \*\*  
*Mutex to lock thread over critical code sections.*
- `pthread_mutex_t ownership_mutex`

**Data I/O:** \*\*  
*Mutex to lock thread over attribute ownership code sections.*
- `BasicClock clock`

**Units:** –  
*Clock time object.*
- `bool name_registered`

**Units:** –  
*True if the object instance name is registered.*
- `bool changed`

**Units:** –  
*Flag indicating the data has changed.*
- `bool attr_update_requested`

**Units:** –  
*Flag to indicate an attribute updated was requested by another federate.*
- `bool removed_instance`

**Units:** –  
*Flag to indicate if object instance was removed from RTI.*
- `bool any_attribute_FOM_specified_order`

**Units:** –  
*True if any attribute is the FOM specified order.*
- `bool any_attribute_timestamp_order`

- Units:** –  
*True if any attribute is timestamp order.*
- RTI1516\_NAMESPACE::ObjectClassHandle `class_handle`

**Data I/O:** \*\*  
*HLA Object Class handle.*
  - RTI1516\_NAMESPACE::ObjectInstanceHandle `instance_handle`

**Data I/O:** \*\*  
*HLA Object Instance handle.*
  - Int64Time `last_update_time`

**Data I/O:** \*\*  
*Last update time.*
  - Int64Time `time_plus_lookahead`

**Data I/O:** \*\*  
*Time plus lookahead.*
  - bool `pull_requested`

**Units:** –  
*Has someone asked to own us?*
  - bool `divest_requested`

**Units:** –  
*Are we releasing ownership?*
  - VectorOfStrings `attribute_FOM_names`

**Data I/O:** \*\*  
*String array containing the Attribute FOM names.*
  - Manager \* `manager`

**Units:** –  
*Reference to the TrickHLA Manager.*
  - RTI1516\_NAMESPACE::RTIambassador \* `rti_ambassador`

**Data I/O:** \*\*  
*Reference to the RTI ambassador.*
  - RTI1516\_NAMESPACE::AttributeHandleValueMap \* `attribute_values_map`

**Data I/O:** \*\*  
*Map of attributes that will be sent as an update to other federates.*
  - ReflectedAttributesQueue `thla_reflected_attributes_queue`

**Data I/O:** \*\*  
*Queue of reflected attributes.*
  - AttributeMap `thla_attribute_map`

**Data I/O:** \*\*  
*Map of the Attribute's, key is the AttributeHandle.*

## Private Member Functions

- void `set_name` (const char \*`new_name`)
 

*Sets the new value of the name attribute.*
- void `set_name_and_mark_changed` (const char \*`new_name`)
 

*Set the name of the object and mark it as changed.*
- Object (const Object &`rhs`)
 

*Copy constructor for Object class.*
- Object & `operator=` (const Object &`rhs`)
 

*Assignment operator for Object class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_\\_Object\(\)](#)

### 7.36.1 Detailed Description

Definition at line 103 of file Object.hh.

### 7.36.2 Constructor & Destructor Documentation

#### 7.36.2.1 [Object\(\)](#) [1/2]

`Object::Object ( )`

Default constructor for the [TrickHLA Object](#) class.

**Trick Job Class:** *initialization*

Definition at line 84 of file Object.cpp.

References attribute\_values\_map, mutex, and ownership\_mutex.

#### 7.36.2.2 [~Object\(\)](#)

`Object::~Object ( ) [virtual]`

Destructor for the [TrickHLA Object](#) class.

Frees memory allocated, and remove this object from the federation execution. **Trick Job Class:** *shutdown*

Definition at line 141 of file Object.cpp.

References attribute\_values\_map, mutex, name, ownership\_mutex, remove(), removed\_instance, set\_to\_unblocking←\_cyclic\_reads(), and thla\_attribute\_map.

#### 7.36.2.3 [Object\(\)](#) [2/2]

`TrickHLA::Object::Object (`  
    `const Object & rhs ) [private]`

Copy constructor for [Object](#) class.

This constructor is private to prevent inadvertent copies.

### 7.36.3 Member Function Documentation

#### 7.36.3.1 [any\\_attribute\\_published\(\)](#)

`bool Object::any_attribute_published ( )`

Determines if any attribute is published.

##### Returns

True for any published attribute.

Definition at line 3915 of file Object.cpp.

References attr\_count, and attributes.

Referenced by `publish_object_attributes()`, and `unpublish_all_object_attributes()`.

### 7.36.3.2 any\_attribute\_subscribed()

```
bool Object::any_attribute_subscribed ( )
```

Determines if any attribute is being subscribed to.

#### Returns

True any subscribed attribute.

Definition at line 3925 of file Object.cpp.

References attr\_count, and attributes.

Referenced by subscribe\_to\_object\_attributes(), and unsubscribe\_all\_object\_attributes().

### 7.36.3.3 any\_locally\_owned\_attribute()

```
bool Object::any_locally_owned_attribute ( )
```

Determines if any attribute is locally owned.

#### Returns

True for any locally owned attribute.

Definition at line 3935 of file Object.cpp.

References attr\_count, and attributes.

Referenced by setup\_preferred\_order\_with\_RTI().

### 7.36.3.4 any\_locally\_owned\_published\_attribute() [1/2]

```
bool Object::any_locally_owned_published_attribute ( )
```

Determines if any attribute is locally owned and published.

#### Returns

True for any locally owned and published attribute.

Definition at line 3945 of file Object.cpp.

References attr\_count, and attributes.

Referenced by any\_locally\_owned\_published\_init\_attribute().

### 7.36.3.5 any\_locally\_owned\_published\_attribute() [2/2]

```
bool Object::any_locally_owned_published_attribute (
```

<code>const DataUpdateEnum attr_config</code>
---

Determines if any attribute is locally owned and published for the given attribute configuration.

#### Returns

True for any locally owned and published attribute.

#### Parameters

<code>attr_config</code>	Attribute configuration.
--------------------------	--------------------------

Definition at line 3955 of file Object.cpp.

References attr\_count, and attributes.

### 7.36.3.6 any\_locally\_owned\_published\_cyclic\_data\_ready\_attribute()

```
bool Object::any_locally_owned_published_cyclic_data_ready_attribute ( )
```

Determines if any attribute is locally owned, published, and has a cycle-time that is ready for a cyclic send.

#### Returns

True for any locally owned and published attribute with a sub-rate that is ready.

Definition at line 3966 of file Object.cpp.

References attr\_count, attributes, and TrickHLA::CONFIG\_CYCLIC.

Referenced by send\_cyclic\_data().

### 7.36.3.7 any\_locally\_owned\_published\_init\_attribute()

```
bool TrickHLA::Object::any_locally_owned_published_init_attribute ( ) [inline]
```

Determines if any attribute update is locally owned and published at initialization.

#### Returns

True for any locally owned and published initialization attribute.

Definition at line 443 of file Object.hh.

References any\_locally\_owned\_published\_attribute(), and TrickHLA::CONFIG\_INITIALIZE.

Referenced by TrickHLA::ExecutionControlBase::send\_execution\_configuration(), TrickHLA::Manager::send\_init\_data(), send\_init\_data(), and SpaceFOM::ExecutionControl::send\_root\_ref\_frame().

### 7.36.3.8 any\_locally\_owned\_published\_requested\_attribute()

```
bool Object::any_locally_owned_published_requested_attribute ( )
```

Determines if any attribute update is locally owned and published for the attribute value update.

#### Returns

True for any locally owned and published attribute.

Definition at line 3980 of file Object.cpp.

References attr\_count, and attributes.

Referenced by send\_requested\_data().

### 7.36.3.9 any\_remotely\_owned\_subscribed\_attribute() [1/2]

```
bool Object::any_remotely_owned_subscribed_attribute ( )
```

Determines if any attribute is remotely owned and subscribed to.

#### Returns

True for any remotely owned and subscribed attribute.

Definition at line 3990 of file Object.cpp.

References attr\_count, and attributes.

Referenced by any\_remotely\_owned\_subscribed\_cyclic\_attribute(), any\_remotely\_owned\_subscribed\_init\_attribute(), and request\_attribute\_value\_update().

**7.36.3.10 any\_remotely\_owned\_subscribed\_attribute() [2/2]**

```
bool Object::any_remotely_owned_subscribed_attribute (
    const DataUpdateEnum attr_config )
```

Determines if any attribute specified is remotely owned and subscribed to for the given attribute configuration.

**Returns**

True for any remotely owned, subscribed attribute.

**Parameters**

<i>attr_config</i>	Attribute configuration.
--------------------	--------------------------

Definition at line 4000 of file Object.cpp.

References attr\_count, and attributes.

**7.36.3.11 any\_remotely\_owned\_subscribed\_cyclic\_attribute()**

```
bool TrickHLA::Object::any_remotely_owned_subscribed_cyclic_attribute ( ) [inline]
```

Determines if any cyclically updated attributes are remotely owned and subscribed.

**Returns**

True if there are any remotely owned, subscribed cyclically updated attributes.

Definition at line 467 of file Object.hh.

References any\_remotely\_owned\_subscribed\_attribute(), and TrickHLA::CONFIG\_CYCLIC.  
Referenced by receive\_cyclic\_data().

**7.36.3.12 any\_remotely\_owned\_subscribed\_init\_attribute()**

```
bool TrickHLA::Object::any_remotely_owned_subscribed_init_attribute ( ) [inline]
```

Determines if any initialization updated attributes are remotely owned and subscribed.

**Returns**

True if there are any remotely owned, subscribed updated attributes at initialization.

Definition at line 476 of file Object.hh.

References any\_remotely\_owned\_subscribed\_attribute(), and TrickHLA::CONFIG\_INITIALIZE.

Referenced by TrickHLA::ExecutionControlBase::receive\_execution\_configuration(), TrickHLA::Manager::receive\_init\_data(), receive\_init\_data(), SpaceFOM::ExecutionControl::receive\_root\_ref\_frame(), TrickHLA::ExecutionConfigurationBase::wait\_on\_update(), DSES::ExecutionConfiguration::wait\_on\_update(), DIS::ExecutionConfiguration::wait\_on\_update(), IMSim::ExecutionConfiguration::wait\_on\_update(), and SpaceFOM::ExecutionConfiguration::wait\_on\_update().

**7.36.3.13 build\_attribute\_map()**

```
void Object::build_attribute_map ( )
```

Build the attribute map, which will be used for quickly looking up an attribute given its AttributeHandle.

Definition at line 3866 of file Object.cpp.

References attr\_count, attributes, TrickHLA::Attribute::get\_attribute\_handle(), and thla\_attribute\_map.

Referenced by TrickHLA::Manager::setup\_object\_RTI\_handles().

### 7.36.3.14 create\_attribute\_set()

```
AttributeHandleValueMap * Object::create_attribute_set (
    const DataUpdateEnum required_config )
```

Create a name value pair set, aka attribute handle value pair, for the attributes of this object.

#### Parameters

<code>required_config</code>	Attribute configuration required in order to send data.
------------------------------	---

#### Trick Job Class: *scheduled*

Definition at line 2808 of file Object.cpp.

References attr\_count, attribute\_values\_map, attributes, TrickHLA::CONFIG\_CYCLIC, TrickHLA::DEBUG\_LEVEL\_7\_←  
TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, TrickHLA::Attribute::get\_attribute\_handle(), TrickHLA::Attribute::get←  
\_attribute\_value(), get\_FOM\_name(), get\_name(), TrickHLA::Attribute::set\_update\_requested(), should\_print(), and T←  
HLA\_NEWLINE.

Referenced by send\_cyclic\_data(), and send\_init\_data().

### 7.36.3.15 create\_requested\_attribute\_set()

```
AttributeHandleValueMap * Object::create_requested_attribute_set ( )
```

Create a name value pair set, aka attribute handle value pair, for the attributes that were requested for this object.

#### Trick Job Class: *scheduled*

Definition at line 2762 of file Object.cpp.

References attr\_count, attribute\_values\_map, attributes, TrickHLA::DEBUG\_LEVEL\_7\_TRACE, TrickHLA::DEBUG\_←  
SOURCE\_OBJECT, TrickHLA::Attribute::get\_attribute\_handle(), TrickHLA::Attribute::get\_attribute\_value(), get\_FOM←  
\_name(), TrickHLA::Attribute::set\_update\_requested(), should\_print(), and THLA\_NEWLINE.

Referenced by send\_requested\_data().

### 7.36.3.16 extract\_data()

```
void Object::extract_data (
    RTI1516_NAMESPACE::AttributeHandleValueMap & theAttributes )
```

This function extracts the new attribute values.

#### Parameters

<code>theAttributes</code>	Attributes data.
----------------------------	------------------

This routine is called by the federate ambassador when new attribute values come in for this object. **Trick Job Class: *scheduled***

Definition at line 2896 of file Object.cpp.

References data\_changed, TrickHLA::DEBUG\_LEVEL\_7\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, TrickHLA←  
::Attribute::extract\_data(), FOM\_name, get\_attribute(), mark\_changed(), name, should\_print(), THLA\_NEWLINE, and  
TrickHLA::StringUtilities::to\_string().

Referenced by is\_changed(), and TrickHLA::FedAmb::reflectAttributeValues().

### 7.36.3.17 get\_attribute() [1/3]

```
Attribute * Object::get_attribute (
    const char * attr_FOM_name )
```

Gets the attribute for the given FOM name.

**Returns**

Associated [TrickHLA::Attribute](#).

**Parameters**

<i>attr_FOM_name</i>	Attribute FOM name.
----------------------	---------------------

Definition at line 3888 of file Object.cpp.

References attr\_count, attributes, and [get\\_FOM\\_name\(\)](#).

**7.36.3.18 get\_attribute() [2/3]**

```
Attribute * Object::get_attribute (
    RTI1516_NAMESPACE::AttributeHandle attr_handle )
```

Gets the attribute for the given HLA Attribute-Handle.

**Returns**

Associated [TrickHLA::Attribute](#).

**Parameters**

<i>attr_handle</i>	Attribute ID.
--------------------	---------------

Definition at line 3874 of file Object.cpp.

References thla\_attribute\_map.

Referenced by [TrickHLA::FedAmb::attributeOwnershipAcquisitionNotification\(\)](#), [extract\\_data\(\)](#), [TrickHLA::OwnershipHandler::get\\_attribute\(\)](#), [get\\_attribute\(\)](#), [grant\\_pull\\_request\(\)](#), [release\\_ownership\(\)](#), [TrickHLA::FedAmb::requestAttributeOwnershipAssumption\(\)](#), [TrickHLA::FedAmb::requestAttributeOwnershipRelease\(\)](#), and [TrickHLA::FedAmb::requestDivestitureConfirmation\(\)](#).

**7.36.3.19 get\_attribute() [3/3]**

```
Attribute * Object::get_attribute (
    std::string attr_FOM_name )
```

Gets the attribute for the given FOM name.

**Returns**

Associated [TrickHLA::Attribute](#).

**Parameters**

<i>attr_FOM_name</i>	Attribute FOM name.
----------------------	---------------------

Definition at line 3882 of file Object.cpp.

References [get\\_attribute\(\)](#).

### 7.36.3.20 `get_attribute_count()`

```
int TrickHLA::Object::get_attribute_count ( ) const [inline]
Get the count of the number of attributes associated with this object.
```

#### Returns

The number of attributes associated with this object.

Definition at line 579 of file Object.hh.

References attr\_count.

Referenced by TrickHLA::OwnershipHandler::get\_attribute\_count(), TrickHLA::Manager::setup\_object\_ref\_attributes(), and TrickHLA::Manager::setup\_object\_RTI\_handles().

### 7.36.3.21 `get_attribute_FOM_names()`

```
VectorOfStrings TrickHLA::Object::get_attribute_FOM_names ( ) const [inline]
Get the attribute FOM names.
```

#### Returns

A vector of strings containing the attribute FOM names.

Definition at line 591 of file Object.hh.

References attribute\_FOM\_names.

Referenced by TrickHLA::OwnershipHandler::get\_attribute\_FOM\_names().

### 7.36.3.22 `get_attributes()`

```
Attribute* TrickHLA::Object::get_attributes ( ) [inline]
Get the list of attributes associated with this object.
```

#### Returns

The array of attributes associated with this object.

Definition at line 583 of file Object.hh.

References attributes.

Referenced by TrickHLA::Manager::setup\_object\_ref\_attributes(), and TrickHLA::Manager::setup\_object\_RTI\_handles().

### 7.36.3.23 `get_class_handle()`

```
RTI1516_NAMESPACE::ObjectClassHandle TrickHLA::Object::get_class_handle ( ) const [inline]
Get the HLA Object class handle for this object.
```

#### Returns

The HLA ObjectClassHandle for this object.

Definition at line 353 of file Object.hh.

References class\_handle.

Referenced by TrickHLA::ExecutionControlBase::get\_unregisteredException(), TrickHLA::ExecutionControlBase::get\_unregisteredException(), TrickHLA::Manager::setup\_object\_RTI\_handles(), TrickHLA::Manager::unpublish(), and TrickHLA::Manager::unsubscribe().

**7.36.3.24 get\_fed\_lookahead()**

```
Int64Interval Object::get_fed_lookahead ( ) const
Return a copy of the federate's lookahead time.
```

**Returns**

Lookahead time interval.

If the manager does not exist, -1.0 seconds is returned.

Definition at line 4119 of file Object.cpp.

References TrickHLA::Manager::get\_fed\_lookahead(), and manager.

**7.36.3.25 get\_federate()**

```
Federate * Object::get_federate ( )
```

Gets a pointer to our federate.

**Returns**

Pointer to [TrickHLA::Federate](#) instance.

Definition at line 415 of file Object.cpp.

References TrickHLA::Manager::get\_federate(), and manager.

Referenced by TrickHLA::Packing::get\_cte\_time(), TrickHLA::OwnershipHandler::get\_cte\_time(), get\_RTI\_ambassador(), TrickHLA::Packing::get\_scenario\_time(), TrickHLA::OwnershipHandler::get\_scenario\_time(), get\_update\_time\_← plus\_lookahead(), DIS::ExecutionConfiguration::pack(), DSES::ExecutionConfiguration::pack(), IMSim::Execution← Configuration::pack(), SpaceFOM::ExecutionConfiguration::pack(), pull\_ownership(), pull\_ownership\_upon\_rejoin(), push\_ownership(), receive\_cyclic\_data(), remove(), send\_cyclic\_data(), send\_init\_data(), send\_requested← data(), SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes(), DSES::ExecutionConfiguration::unpack(), DIS← ::ExecutionConfiguration::unpack(), IMSim::ExecutionConfiguration::unpack(), SpaceFOM::ExecutionConfiguration← ::unpack(), wait\_on\_object\_name\_reservation(), wait\_on\_object\_registration(), TrickHLA::ExecutionConfiguration← Base::wait\_on\_registration(), TrickHLA::ExecutionConfigurationBase::wait\_on\_update(), DIS::ExecutionConfiguration← ::wait\_on\_update(), DSES::ExecutionConfiguration::wait\_on\_update(), IMSim::ExecutionConfiguration::wait\_on← update(), and SpaceFOM::ExecutionConfiguration::wait\_on\_update().

**7.36.3.26 get\_FOM\_name()**

```
const char* TrickHLA::Object::get_FOM_name ( ) const [inline]
```

Get the FOM name for this object.

**Returns**

The FOM name for this object.

Definition at line 349 of file Object.hh.

References FOM\_name.

Referenced by TrickHLA::FedAmb::attributeOwnershipAcquisitionNotification(), create\_attribute\_set(), create← requested\_attribute\_set(), get\_attribute(), TrickHLA::OwnershipHandler::get\_object\_FOM\_name(), grant\_pull← request(), initialize(), mark\_all\_attributes\_as\_nonlocal(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init← processes(), pull\_ownership(), pull\_ownership\_upon\_rejoin(), push\_ownership(), release\_ownership(), TrickHLA::← FedAmb::requestAttributeOwnershipAssumption(), TrickHLA::FedAmb::requestAttributeOwnershipRelease(), TrickHL← A::FedAmb::requestDivestitureConfirmation(), TrickHLA::Manager::setup\_object\_ref\_attributes(), TrickHLA::Manager← ::setup\_object\_RTI\_handles(), setup\_preferred\_order\_with\_RTI(), TrickHLA::ExecutionConfigurationBase::wait\_on← registration(), and TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects().

### 7.36.3.27 get\_granted\_fed\_time()

```
Int64Time Object::get_granted_fed_time ( ) const
Get the currently granted federation HLA logical time.
```

#### Returns

A copy of the granted HLA logical time.

If the manager does not exist, MAX\_LOGICAL\_TIME\_SECONDS is returned.

Definition at line 4132 of file Object.cpp.

References TrickHLA::Manager::get\_granted\_fed\_time(), manager, and TrickHLA::MAX\_LOGICAL\_TIME\_SECONDS. Referenced by receive\_cyclic\_data().

### 7.36.3.28 get\_granted\_time()

```
double Object::get_granted_time ( ) const
Get the granted HLA time.
```

#### Returns

Granted time.

Definition at line 4111 of file Object.cpp.

References TrickHLA::Manager::get\_granted\_time(), and manager.

### 7.36.3.29 get\_instance\_handle()

```
RTI1516_NAMESPACE::ObjectInstanceHandle TrickHLA::Object::get_instance_handle ( ) const [inline]
Get the HLA Object instance handle for this object instance.
```

#### Returns

The HLA ObjectInstanceHandle for this object instance.

Definition at line 365 of file Object.hh.

References instance\_handle.

Referenced by TrickHLA::Manager::add\_object\_to\_map(), TrickHLA::Manager::discover\_object\_instance(), TrickHLA::ExecutionControlBase::mark\_object\_as\_deleted\_from\_federation(), TrickHLA::ExecutionControlBase::provide\_attribute\_update(), TrickHLA::Manager::register\_objects\_with\_RTI(), TrickHLA::Manager::set\_object\_instance\_handles\_by\_name(), TrickHLA::ExecutionConfigurationBase::wait\_on\_registration(), and TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects().

### 7.36.3.30 get\_lag\_compensation\_type()

```
LagCompensationEnum TrickHLA::Object::get_lag_compensation_type ( ) const [inline]
Get the current Lag Compensation type for this object instance attribute updates.
```

#### Returns

The current Lag Compensation type.

Definition at line 525 of file Object.hh.

References lag\_comp\_type.

**7.36.3.31 get\_last\_update\_time()**

```
const Int64Time& TrickHLA::Object::get_last_update_time ( ) const [inline]
Get the last update time.
```

**Returns**

The last HLA logical time update value.

Definition at line 537 of file Object.hh.

References last\_update\_time.

**7.36.3.32 get\_name()**

```
const char* TrickHLA::Object::get_name ( ) const [inline]
Get the object instance name.
```

**Returns**

Object instance name.

Definition at line 327 of file Object.hh.

References name.

Referenced by TrickHLA::FedAmb::attributeOwnershipAcquisitionNotification(), create\_attribute\_set(), TrickHLA< Model>::SineObjectDeleted::deleted(), TrickHLA::Manager::discover\_object\_instance(), TrickHLA::OwnershipHandler< :>::get\_object\_name(), TrickHLA::ExecutionControlBase::get\_unregistered\_remote\_object(), grant\_pull\_request(), grant\_push\_request(), SpaceFOM::ExecutionControl::late\_joinder\_hla\_init\_process(), mark\_all\_attributes\_as< nonlocal>(), TrickHLA::ExecutionControlBase::mark\_object\_as\_deleted\_from\_federation(), TrickHLA::Manager::mark< \_object\_as\_deleted\_from\_federation>(), negotiated\_attribute\_ownership\_divestiture(), TrickHLA::ExecutionControl< Base>::object\_instance\_name\_reservation\_succeeded(), TrickHLA::Manager::object\_instance\_name\_reservation< \_succeeded>(), DIS::ExecutionConfiguration::pack(), DSES::ExecutionConfiguration::pack(), IMSim::Execution< Configuration>::pack(), SpaceFOM::ExecutionConfiguration::pack(), IMSim::ExecutionControl::pre\_multi\_phase< init\_processes>(), TrickHLA::ExecutionConfiguration::print\_execution\_configuration(), DIS::ExecutionConfiguration< ::print\_execution\_configuration>(), DSES::ExecutionConfiguration::print\_execution\_configuration(), IMSim::Execution< Configuration>::print\_execution\_configuration(), SpaceFOM::ExecutionConfiguration::print\_execution\_configuration(), process\_deleted\_object(), provide\_attribute\_update(), publish\_object\_attributes(), pull\_ownership(), pull\_ownership< \_upon\_rejoin>(), push\_ownership(), receive\_cyclic\_data(), receive\_init\_data(), TrickHLA::FedAmb::reflectAttribute< Values>(), register\_object\_with\_RTI(), release\_ownership(), remove(), remove\_object\_instance(), request< attribute\_value\_update>(), TrickHLA::FedAmb::requestAttributeOwnershipAssumption(), TrickHLA::FedAmb::request< AttributeOwnershipRelease>(), TrickHLA::FedAmb::requestDivestitureConfirmation(), reserve\_object\_name\_with\_RT< I>(), restore\_ownership\_transfer\_checkpointed\_data(), send\_cyclic\_data(), send\_init\_data(), SpaceFOM::Execution< Control>::send\_init\_root\_ref\_frame(), send\_requested\_data(), TrickHLA::Manager::set\_object\_instance\_handles\_by< name>(), TrickHLA::Manager::setup\_object\_ref\_attributes(), TrickHLA::Manager::setup\_object\_RTI\_handles(), setup< ownership\_transfer\_checkpointed\_data>(), setup\_preferred\_order\_with\_RTI(), subscribe\_to\_object\_attributes(), DIS::ExecutionConfiguration::unpack(), DSES::ExecutionConfiguration::unpack(), IMSim::ExecutionConfiguration::unpack(), SpaceFOM::ExecutionConfiguration::unpack(), unpublish\_all\_object\_attributes(), unsubscribe\_all\_object\_attributes(), wait\_on\_object\_name\_reservation(), wait\_on\_object\_registration(), TrickHLA::ExecutionConfigurationBase::wait\_on< registration>(), and TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects().

**7.36.3.33 get\_name\_string()**

```
const std::string TrickHLA::Object::get_name_string ( ) const [inline]
Get the object instance name as a C++ string.
```

**Returns**

The object instance name as a C++ string.

Definition at line 331 of file Object.hh.

References name.

Referenced by TrickHLAModel::SinePacking::pack(), and TrickHLAModel::SinePacking::unpack().

**7.36.3.34 get\_RTI\_ambassador()**

```
RTI1516_NAMESPACE::RTIambassador * Object::get_RTI_ambassador ( ) [protected]
```

Gets the RTI Ambassador.

**Returns**

Pointer to associated HLA RTIambassador instance.

Definition at line 420 of file Object.cpp.

References get\_federate(), TrickHLA::Federate::get\_RTI\_ambassador(), rti\_ambassador, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by grant\_pull\_request(), grant\_push\_request(), negotiated\_attribute\_ownership\_divestiture(), publish\_object\_attributes(), pull\_ownership(), pull\_ownership\_upon\_rejoin(), push\_ownership(), register\_object\_with\_RTI(), release\_ownership(), remove(), request\_attribute\_value\_update(), reserve\_object\_name\_with\_RTI(), send\_cyclic\_data(), send\_init\_data(), send\_requested\_data(), setup\_preferred\_order\_with\_RTI(), subscribe\_to\_object\_attributes(), unpublish\_all\_object\_attributes(), and unsubscribe\_all\_object\_attributes().

**7.36.3.35 get\_update\_time\_plus\_lookahead()**

```
const Int64Time & Object::get_update_time_plus_lookahead ( )
```

Gets the updated time plus the lookahead time.

**Returns**

Logical time.

Definition at line 4096 of file Object.cpp.

References get\_federate(), TrickHLA::Federate::get\_lookahead(), last\_update\_time, THLA\_NEWLINE, and time\_plus\_lookahead.

Referenced by remove(), send\_cyclic\_data(), and send\_requested\_data().

**7.36.3.36 grant\_pull\_request()**

```
void Object::grant_pull_request ( )
```

This function grants a pull request for this object.

**Trick Job Class:** *scheduled*

Definition at line 3309 of file Object.cpp.

References attr\_count, attributes, TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_attribute(), TrickHLA::Attribute::get\_FOM\_name(), get\_FOM\_name(), get\_name(), get\_RTI\_ambassador(), instance\_handle, is\_instance\_handle\_valid(), TrickHLA::Attribute::mark\_remotely\_owned(), pull\_requested, RTI1516\_EXCEPTION, TrickHLA::Attribute::set\_pull\_requested(), should\_print(), THLA\_NEWLINE, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD. Referenced by TrickHLA::Manager::grant\_pull\_request().

### 7.36.3.37 grant\_push\_request()

```
void Object::grant_push_request ( )
```

This function grants a previously "recorded" push request for this object.

**Trick Job Class:** *scheduled*

Definition at line 3448 of file Object.cpp.

References attr\_count, attributes, TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_name(), get\_RTI\_ambassador(), instance\_handle, is\_instance\_handle\_valid(), lock(), RTI1516\_EXCEPTION, RTI1516\_USERDATA, TrickHLA::Attribute::set\_push\_requested(), should\_print(), THLA\_NEWLINE, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD, and unlock().

Referenced by grant\_push\_pthread\_function().

### 7.36.3.38 grant\_push\_request\_pthread()

```
void Object::grant_push_request_pthread ( )
```

This function starts a thread to service the push request grant.

**Trick Job Class:** *scheduled*

Definition at line 3432 of file Object.cpp.

References grant\_push\_pthread\_function(), and THLA\_NEWLINE.

Referenced by TrickHLA::FedAmb::requestAttributeOwnershipAssumption().

### 7.36.3.39 initialize()

```
void Object::initialize (
    Manager * trickhla_mgr )  [virtual]
```

Initializes the [TrickHLA Object](#).

#### Parameters

<i>trickhla_mgr</i>	The <a href="#">TrickHLA::Manager</a> instance.
---------------------	---

**Trick Job Class:** *initialization*

Definition at line 186 of file Object.cpp.

References any\_attribute\_FOM\_specified\_order, any\_attribute\_timestamp\_order, attr\_count, attribute\_FOM\_names, attributes, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, deleted, FOM\_name, get\_FOM\_name(), TrickHLA::Packing::initialize\_callback(), TrickHLA::OwnershipHandler::initialize\_callback(), TrickHLA::LagCompensation::initialize\_callback(), is\_create\_HLA\_instance(), is\_name\_required(), lag\_comp, lag\_comp\_type, TrickHLA::LAG\_COMPENSATION\_FIRST\_VALUE, TrickHLA::LAG\_COMPENSATION\_LAST\_VALUE, TrickHLA::LAG\_COMPENSATION\_NONE, manager, name, ownership, packing, set\_name(), should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::TRANSPORT\_SPECIFIED\_IN\_FOM, TrickHLA::TRANSPORT\_TIMESTAMP\_ORDER, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::ExecutionConfiguration::configure(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), TrickHLA::Manager::setup\_object\_ref\_attributes(), and SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

### 7.36.3.40 is\_attribute\_update\_requested()

```
bool TrickHLA::Object::is_attribute_update_requested ( ) const [inline]
```

Check an attribute update has been requested for an attribute associated with this object instance.

**Returns**

True is an attribute update has been requested.

Definition at line 411 of file Object.hh.

References attr\_update\_requested.

Referenced by DSES::ExecutionControl::check\_freeze\_exit(), and SpaceFOM::ExecutionControl::check\_freeze\_exit().

**7.36.3.41 is\_changed()**

```
bool TrickHLA::Object::is_changed ( ) [inline]
```

Used by the [TrickHLA](#) extension to determine if the object data changed.

**Returns**

True if object data has changed.

Definition at line 487 of file Object.hh.

References changed, TrickHLA::ReflectedAttributesQueue::empty(), extract\_data(), TrickHLA::ReflectedAttributesQueue::front(), TrickHLA::ReflectedAttributesQueue::pop(), and thla\_reflected\_attributes\_queue.

Referenced by receive\_cyclic\_data(), TrickHLA::ExecutionControlBase::receive\_execution\_configuration(), TrickHLA::Manager::receive\_init\_data(), receive\_init\_data(), SpaceFOM::ExecutionControl::receive\_root\_ref\_frame(), TrickHLA::ExecutionConfigurationBase::wait\_on\_update(), DIS::ExecutionConfiguration::wait\_on\_update(), DSES::ExecutionConfiguration::wait\_on\_update(), IMSim::ExecutionConfiguration::wait\_on\_update(), and SpaceFOM::ExecutionConfiguration::wait\_on\_update().

**7.36.3.42 is\_create\_HLA\_instance()**

```
bool TrickHLA::Object::is_create_HLA_instance ( ) const [inline]
```

Check if the object instance has been created.

**Returns**

True if the object instance has been created.

Definition at line 386 of file Object.hh.

References create\_HLA\_instance.

Referenced by TrickHLA::ExecutionControlBase::get\_unregistered\_remote\_object(), initialize(), register\_object\_with\_RTI(), remove(), reserve\_object\_name\_with\_RTI(), TrickHLA::Manager::setup\_object\_ref\_attributes(), setup\_preferred\_order\_with\_RTI(), and wait\_on\_object\_name\_reservation().

**7.36.3.43 is\_instance\_handle\_valid()**

```
bool TrickHLA::Object::is_instance_handle_valid ( ) const [inline]
```

Check if the HLA [Object](#) instance handle is valid for this object.

**Returns**

True if the HLA ObjectInstanceHandle is valid for the object instance.

Definition at line 361 of file Object.hh.

References instance\_handle.

Referenced by TrickHLA::Manager::add\_object\_to\_map(), TrickHLA::ExecutionControlBase::get\_unregistered\_object(), TrickHLA::ExecutionControlBase::get\_unregistered\_remote\_object(), grant\_pull\_request(), grant\_push\_request(), process\_deleted\_object(), pull\_ownership(), pull\_ownership\_upon\_rejoin(), push\_ownership(), receive\_cyclic\_data(), register\_object\_with\_RTI(), release\_ownership(), remove(), remove\_object\_instance(), reserve\_object\_name\_with\_RTI(), TrickHLA::Manager::set\_object\_instance\_handles\_by\_name(), TrickHLA::ExecutionControlBase::setup\_checkpoint(), setup\_preferred\_order\_with\_RTI(), TrickHLA::Manager::wait\_on\_discovery\_of\_objects(),

wait\_on\_object\_name\_reservation(), wait\_on\_object\_registration(), TrickHLA::ExecutionConfigurationBase::wait\_on\_registration(), and TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects().

#### 7.36.3.44 is\_name\_registered()

```
bool TrickHLA::Object::is_name_registered ( ) const [inline]
```

Check if an object instance name is registered.

##### Returns

True is the object instance name is registered.

Definition at line 339 of file Object.hh.

References name\_registered.

#### 7.36.3.45 is\_name\_required()

```
bool TrickHLA::Object::is_name_required ( ) const [inline]
```

Check if an object instance name is required.

##### Returns

True is the object instance name is required.

Definition at line 335 of file Object.hh.

References name\_required.

Referenced by TrickHLA::ExecutionControlBase::get\_unregistered\_remote\_object(), initialize(), register\_object\_with\_RTI(), reserve\_object\_name\_with\_RTI(), and wait\_on\_object\_name\_reservation().

#### 7.36.3.46 is\_object\_deleted\_from\_RTI()

```
bool TrickHLA::Object::is_object_deleted_from_RTI ( ) const [inline]
```

Check if object has been deleted from the RTI.

##### Returns

True if object has been deleted from the RTI.

Definition at line 214 of file Object.hh.

References object\_deleted\_from\_RTI.

Referenced by TrickHLA::ExecutionControlBase::process\_deleted\_objects().

#### 7.36.3.47 is\_required()

```
bool TrickHLA::Object::is_required ( ) const [inline]
```

Check if the object is a required object instance.

##### Returns

True if this is a required object instance.

Definition at line 395 of file Object.hh.

References required.

Referenced by TrickHLA::Manager::receive\_init\_data(), TrickHLA::ExecutionConfigurationBase::wait\_on\_registration(), and TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects().

### 7.36.3.48 lock()

```
void TrickHLA::Object::lock ( ) [inline]
```

Lock the thread mutex.

Definition at line 621 of file Object.hh.

References mutex.

Referenced by grant\_push\_request(), and TrickHLA::FedAmb::requestAttributeOwnershipAssumption().

### 7.36.3.49 mark\_all\_attributes\_as\_nonlocal()

```
void Object::mark_all_attributes_as_nonlocal ( )
```

Turn off local flag in all attributes.

Turns off the local flag for all attributes in this object. This is needed so when an object pushed the ownership of its attributes and disappears before pulling the ownership back or the object resigns from the federation, the object which received the ownership does not have delete privilege (that disappeared with the declaring object) so it must not continue to publish the deleted object's data. This also applies to attributes of an object that resigns from the federation.

Definition at line 605 of file Object.cpp.

References attr\_count, attributes, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_FOM\_name(), get\_name(), TrickHLA::Attribute::get\_trick\_name(), instance\_handle, TrickHLA::Attribute::is\_locally\_owned(), should\_print(), THLA\_ENDL, TrickHLA::StringUtilities::to\_string(), and TrickHLA::Attribute::unmark\_locally\_owned().

Referenced by remove\_object\_instance().

### 7.36.3.50 mark\_changed()

```
void Object::mark_changed ( )
```

Mark the data as changed, and notify any waiting thread that there is a change.

Definition at line 4063 of file Object.cpp.

References changed.

Referenced by extract\_data(), receive\_cyclic\_data(), and set\_name\_and\_mark\_changed().

### 7.36.3.51 mark\_required()

```
void TrickHLA::Object::mark_required ( ) [inline]
```

Mark this object instance as required.

Definition at line 398 of file Object.hh.

Referenced by DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init\_processes(), TrickHLA::Manager::setup\_checkpoint(), and TrickHLA::ExecutionControlBase::setup\_checkpoint().

### 7.36.3.52 mark\_unchanged()

```
void Object::mark_unchanged ( )
```

Mark the data as unchanged, and clear the change flag for all the attributes as well.

Definition at line 4077 of file Object.cpp.

References attr\_count, attributes, changed, and TrickHLA::Attribute::mark\_unchanged().

Referenced by receive\_cyclic\_data(), and receive\_init\_data().

**7.36.3.53 negotiated\_attribute\_ownership\_divestiture()**

```
void Object::negotiated_attribute_ownership_divestiture (
    RTI1516_NAMESPACE::AttributeHandleSet * attr_hdl_set )
```

This function handles the RTI call for the negotiated attribute ownership divestiture.

**Parameters**

<i>attr_hdl_set</i>	Attributes.
---------------------	-------------

**Trick Job Class: *scheduled***

Definition at line 3567 of file Object.cpp.

References get\_name(), get\_RTI\_ambassador(), instance\_handle, RTI1516\_EXCEPTION, RTI1516\_USERDATA, T←HLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA←\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by ownership\_divestiture\_pthread\_function().

**7.36.3.54 notify\_attribute\_ownership\_changed()**

```
void Object::notify_attribute_ownership_changed ( )
```

Notify any waiting threads of a change in attribute ownership, which could affect blocking reads.

Definition at line 4054 of file Object.cpp.

Referenced by TrickHLA::FedAmb::attributeOwnershipAcquisitionNotification().

**7.36.3.55 operator=()**

```
Object& TrickHLA::Object::operator= (
    const Object & rhs ) [private]
```

Assignment operator for **Object** class.

This assignment operator is private to prevent inadvertent copies.

**7.36.3.56 ownership\_lock()**

```
void TrickHLA::Object::ownership_lock ( ) [inline]
```

Lock the ownership thread mutex.

Definition at line 627 of file Object.hh.

References ownership\_mutex.

Referenced by pull\_ownership(), pull\_ownership\_upon\_rejoin(), push\_ownership(), and release\_ownership().

**7.36.3.57 ownership\_unlock()**

```
void TrickHLA::Object::ownership_unlock ( ) [inline]
```

Unlock the ownership thread mutex.

Definition at line 630 of file Object.hh.

References ownership\_mutex.

Referenced by pull\_ownership(), pull\_ownership\_upon\_rejoin(), push\_ownership(), and release\_ownership().

**7.36.3.58 pack\_attribute\_buffers()**

```
void Object::pack_attribute_buffers (
    const DataUpdateEnum attr_config )
```

Pack the attributes for the given configuration into the buffer that is used for sending the encoded attribute through the RTI.

#### Parameters

<i>attr_config</i>	Attribute configuration.
--------------------	--------------------------

Definition at line 4020 of file Object.cpp.

References attr\_count, attributes, and TrickHLA::Attribute::pack\_attribute\_buffer().

Referenced by pack\_cyclic\_attribute\_buffers(), and pack\_init\_attribute\_buffers().

### 7.36.3.59 pack\_cyclic\_attribute\_buffers()

```
void TrickHLA::Object::pack_cyclic_attribute_buffers ( ) [inline]
```

Copy the cyclic attribute values to the buffer for each attribute.

Definition at line 609 of file Object.hh.

References TrickHLA::CONFIG\_CYCLIC, and pack\_attribute\_buffers().

Referenced by send\_cyclic\_data().

### 7.36.3.60 pack\_init\_attribute\_buffers()

```
void TrickHLA::Object::pack_init_attribute_buffers ( ) [inline]
```

Copy the dynamic initialization attribute values to the buffer for each attribute.

Definition at line 615 of file Object.hh.

References TrickHLA::CONFIG\_INITIALIZE, and pack\_attribute\_buffers().

Referenced by send\_init\_data().

### 7.36.3.61 pack\_requested\_attribute\_buffers()

```
void Object::pack_requested_attribute_buffers ( )
```

Pack the attributes that were part of the attribute value request into the buffer that is used for sending the encoded attribute through the RTI.

Definition at line 4011 of file Object.cpp.

References attr\_count, attributes, and TrickHLA::Attribute::pack\_attribute\_buffer().

Referenced by send\_requested\_data().

### 7.36.3.62 process\_deleted\_object()

```
void Object::process_deleted_object ( )
```

This object instance has been deleted from the RTI, so process the delete action by calling the users delete notification callback.

Definition at line 573 of file Object.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, TrickHLA::ObjectDeleted::deleted(), deleted, get\_name(), instance\_handle, is\_instance\_handle\_valid(), object\_deleted\_from\_RTI, should\_print(), THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

Referenced by TrickHLA::ExecutionControlBase::process\_deleted\_objects(), and TrickHLA::Manager::process\_deleted\_objects().

### 7.36.3.63 provide\_attribute\_update()

```
void Object::provide_attribute_update (
    RTI1516_NAMESPACE::AttributeHandleSet const & theAttributes )
```

Requesting an attribute update for the specified attributes.

#### Parameters

<code>theAttributes</code>	The specified attributes.
----------------------------	---------------------------

#### Trick Job Class: *scheduled*

Definition at line 1528 of file Object.cpp.

References attr\_count, attr\_update\_requested, attributes, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEB←UG\_SOURCE\_OBJECT, TrickHLA::Attribute::get\_attribute\_handle(), get\_name(), TrickHLA::Attribute::set\_update←requested(), should\_print(), and THLA\_NEWLINE.

Referenced by TrickHLA::ExecutionControlBase::provide\_attribute\_update(), and TrickHLA::Manager::provide←attribute\_update().

### 7.36.3.64 publish\_object\_attributes()

```
void Object::publish_object_attributes ( )
```

Publishes [Object](#) attributes.

#### Trick Job Class: *initialization*

Definition at line 650 of file Object.cpp.

References any\_attribute\_published(), attr\_count, attributes, class\_handle, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_name(), get\_RTI\_ambassador(), RTI1516\_EXCEPTION, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA←A\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by SpaceFOM::ExecutionControl::publish(), and TrickHLA::Manager::publish().

### 7.36.3.65 pull\_ownership()

```
void Object::pull_ownership ( )
```

This function pulls ownership for this object.

#### Trick Job Class: *scheduled*

Definition at line 3133 of file Object.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, TrickHLA::Attribute::get←attribute\_handle(), TrickHLA::Federate::get\_execution\_control(), get\_federate(), TrickHLA::Attribute::get\_FOM\_name(), get\_FOM\_name(), get\_name(), get\_RTI\_ambassador(), TrickHLA::ExecutionControlBase::get\_sim\_time(), TrickHLA←A::Int64Time::getDoubleTime(), instance\_handle, is\_instance\_handle\_valid(), TrickHLA::Attribute::is\_publish(), TrickHLA::Attribute::is\_remotely\_owned(), TrickHLA::Federate::is\_time\_management\_enabled(), ownership, ownership←lock(), ownership\_unlock(), TrickHLA::OwnershipHandler::pull\_requests, RTI1516\_EXCEPTION, RTI1516\_USERDA←TA, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL←WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::Manager::pull\_ownership().

### 7.36.3.66 pull\_ownership\_upon\_rejoin()

```
void Object::pull_ownership_upon_rejoin ( )
```

This function pulls ownership for all published attributes when the federate rejoins an already running federation.

Definition at line 4142 of file Object.cpp.

References attr\_count, attributes, TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, TrickHLA::Attribute::get\_attribute\_handle(), get\_federate(), get\_FOM\_name(), get\_name(), get\_RTI\_ambassador(), instance\_handle, TrickHLA::Federate::is\_execution\_member(), is\_instance\_handle\_valid(), TrickHLA::Attribute::is\_locally\_owned(), ownership\_lock(), ownership\_unlock(), RTI1516\_EXCEPTION, RTI1516\_USERDATA, should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WO\_RD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD, and TrickHLA::Attribute::unmark\_locally\_owned().

Referenced by TrickHLA::Manager::pull\_ownership\_upon\_rejoin().

### 7.36.3.67 push\_ownership()

```
void Object::push_ownership ()
```

This function pushes ownership for this object.

**Trick Job Class:** *scheduled*

Definition at line 3636 of file Object.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, TrickHLA::Attribute::get\_attribute\_handle(), TrickHLA::Federate::get\_execution\_control(), get\_federate(), TrickHLA::Attribute::get\_FOM\_name(), get\_FOM\_name(), get\_name(), get\_RTI\_ambassador(), TrickHLA::ExecutionControlBase::get\_sim\_time(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::DivestThreadArgs::handle\_set, is\_instance\_handle\_valid(), TrickHLA::Attribute::is\_locally\_owned(), TrickHLA::Federate::is\_time\_management\_enabled(), ownership, ownership\_divestiture\_pthread\_function(), ownership\_lock(), ownership\_unlock(), TrickHLA::OwnershipHandler::push\_requests, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::DivestThreadArgs::trick\_hla\_obj, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::Manager::push\_ownership().

### 7.36.3.68 receive\_cyclic\_data()

```
void Object::receive_cyclic_data (
    double current_time,
    double cycle_time )
```

Handle the received cyclic data.

#### Parameters

<i>current_time</i>	Current time in seconds.
<i>cycle_time</i>	Cycle time between calls to this function in seconds.

If the object is owned remotely, this function copies its internal data into simulation object and marks the object as "unchanged". This data was deposited by the reflect callback and marked as "changed". By marking it as unchanged, we avoid copying the same data over and over. If the object is locally owned, we shouldn't be receiving any remote data anyway and if we were to – bogusly – copy it to the internal byte buffer, we'd continually reset our local simulation. **Trick Job Class:** *scheduled* THLA\_USLEEP\_DELAY\_FOR\_SPIN\_LOCK

Definition at line 2217 of file Object.cpp.

References any\_remotely\_owned\_subscribed\_cyclic\_attribute(), blocking\_cyclic\_read, clock, get\_federate(), get\_granted\_fed\_time(), TrickHLA::Federate::get\_granted\_fed\_time(), get\_name(), TrickHLA::BasicClock::get\_time(), TrickHLA::Int64Time::getDoubleTime(), is\_changed(), is\_instance\_handle\_valid(), lag\_comp, lag\_comp\_type, TrickHLA::LAG\_COMPENSATION\_RECEIVE\_SIDE, mark\_changed(), mark\_unchanged(), packing, TrickHLA::LagCompensation::receive\_lag\_compensation(), THLA\_NEWLINE, TrickHLA::Packing::unpack(), and unpack\_cyclic\_attribute\_buffers().

Referenced by TrickHLA::Manager::receive\_cyclic\_data().

### 7.36.3.69 receive\_init\_data()

```
void Object::receive_init_data ( )
```

Receive initialization data from remote Federates.

If the object is owned remotely, this function copies its internal data into simulation object and marks the object as "unchanged". This data was deposited by the reflect callback and marked as "changed". By marking it as unchanged, we avoid copying the same data over and over. If the object is locally owned, we shouldn't be receiving any remote data anyway and if we were to – bogusly – copy it to the internal byte buffer, we'd continually reset our local simulation. **Trick Job Class:** *scheduled*

Definition at line 2722 of file Object.cpp.

References any\_remotely\_owned\_subscribed\_init\_attribute(), get\_name(), is\_changed(), mark\_unchanged(), packing, THLA\_NEWLINE, TrickHLA::Packing::unpack(), and unpack\_init\_attribute\_buffers().

Referenced by DSES::ExecutionControl::check\_freeze\_exit(), SpaceFOM::ExecutionControl::check\_freeze\_exit(), TrickHLA::ExecutionControlBase::receive\_cyclic\_data(), TrickHLA::ExecutionControlBase::receive\_execution\_configuration(), TrickHLA::Manager::receive\_init\_data(), SpaceFOM::ExecutionControl::receive\_root\_ref\_frame(), TrickHLA::ExecutionConfigurationBase::wait\_on\_update(), DSES::ExecutionConfiguration::wait\_on\_update(), DIS::ExecutionConfiguration::wait\_on\_update(), IMSim::ExecutionConfiguration::wait\_on\_update(), and SpaceFOM::ExecutionConfiguration::wait\_on\_update().

### 7.36.3.70 register\_object\_with\_RTI()

```
void Object::register_object_with_RTI ( )
```

Creates an HLA object instance and registers it with the RTI, but only if we own it locally.

**Trick Job Class:** *initialization*

Definition at line 1117 of file Object.cpp.

References class\_handle, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_name(), get\_RTI\_ambassador(), instance\_handle, TrickHLA::StringUtilities::ip\_strdup\_wstring(), is\_create\_HLA\_instance(), is\_instance\_handle\_valid(), is\_name\_required(), RTI1516\_EXCEPTION, set\_name(), should\_print(), THLA\_ENDL, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::ExecutionControlBase::register\_objects\_with\_RTI(), and TrickHLA::Manager::register\_objects\_with\_RTI().

### 7.36.3.71 release\_ownership()

```
void Object::release_ownership ( )
```

This function releases ownership of the attributes for this object.

**Trick Job Class:** *scheduled*

Definition at line 2947 of file Object.cpp.

References attr\_count, attributes, TrickHLA::OwnershipHandler::clear\_checkpoint(), TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, divest\_requested, get\_attribute(), TrickHLA::Attribute::get\_FOM\_name(), get\_FOM\_name(), get\_name(), get\_RTI\_ambassador(), instance\_handle, is\_instance\_handle\_valid(), TrickHLA::Attribute::mark\_remotely\_owned(), ownership, ownership\_lock(), ownership\_unlock(), RTI1516\_EXCEPTION, RTI1516\_USERDATA, TrickHLA::Attribute::set\_divest\_requested(), should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::Manager::release\_ownership().

### 7.36.3.72 remove()

```
void Object::remove ( )
```

Remove this object instance from the RTI/Federation.

Called from the virtual destructor. **Trick Job Class:** *shutdown*

Definition at line 443 of file Object.cpp.

References TrickHLA::Int64Time::get(), get\_federate(), get\_name(), get\_RTI\_ambassador(), get\_update\_time\_← plus\_loookahead(), TrickHLA::Federate::in\_time\_regulating\_state(), instance\_handle, is\_create\_HLA\_instance(), is← \_instance\_handle\_valid(), TrickHLA::Manager::is\_shutdown(), manager, removed\_instance, RTI1516\_EXCEPTION, RTI1516\_USERDATA, THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONT← ROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::ExecutionControlBase::remove\_execution\_configuration(), and ~Object().

### 7.36.3.73 remove\_object\_instance()

```
void Object::remove_object_instance ()
```

Marks this object as deleted from the RTI and sets all attributes as non-local.

Definition at line 538 of file Object.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_name(), instance\_← handle, is\_instance\_handle\_valid(), mark\_all\_attributes\_as\_nonlocal(), object\_deleted\_from\_RTI, required, set\_to\_← unblocking\_cyclic\_reads(), should\_print(), THLA\_NEWLINE, and TrickHLA::StringUtilities::to\_string().

Referenced by TrickHLA::ExecutionControlBase::mark\_object\_as\_deleted\_from\_federation(), and TrickHLA::← Manager::mark\_object\_as\_deleted\_from\_federation().

### 7.36.3.74 request\_attribute\_value\_update()

```
void Object::request_attribute_value_update ()
```

Request an update to the attributes for this object.

**Trick Job Class:** *scheduled*

Definition at line 1448 of file Object.cpp.

References any\_remotely\_owned\_subscribed\_attribute(), attr\_count, attributes, get\_name(), get\_RTI\_ambassador(), instance\_handle, RTI1516\_EXCEPTION, RTI1516\_USERDATA, THLA\_NEWLINE, TrickHLA::StringUtilities::to← string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::Manager::request\_data\_update().

### 7.36.3.75 reserve\_object\_name\_with\_RTI()

```
void Object::reserve_object_name_with_RTI ()
```

Reserves a unique object instance name with the RTI.

**Trick Job Class:** *initialization*

Definition at line 966 of file Object.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_name(), get\_RTI\_← ambassador(), is\_create\_HLA\_instance(), is\_instance\_handle\_valid(), is\_name\_required(), RTI1516\_EXCEPTION, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TrickHLA::StringUtilities::to\_wstring(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by DSES::ExecutionControl::determine\_federation\_master(), SpaceFOM::ExecutionControl::early\_joine← \_hla\_init\_process(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), and TrickHLA::Manager::reserve\_← object\_names\_with\_RTI().

### 7.36.3.76 restore\_ownership\_transfer\_checkpointed\_data()

```
void Object::restore_ownership_transfer_checkpointed_data ()
```

If an ownership\_transfer object has been created by the user, trigger it's restore() method is re-establish the pull / push AttributeOwnershipMaps.

Definition at line 3829 of file Object.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_name(), ownership, TrickHLA::OwnershipHandler::restore\_requests(), should\_print(), and THLA\_NEWLINE.

Referenced by TrickHLA::Federate::post\_restore(), and TrickHLA::Manager::restart\_initialization().

### 7.36.3.77 send\_cyclic\_data()

```
void Object::send_cyclic_data (
    double current_time,
    double cycle_time )
```

Send cyclic data to remote HLA federates.

#### Parameters

<i>current_time</i>	Current time in seconds.
<i>cycle_time</i>	Cycle time between calls to this function in seconds.

#### Trick Job Class: *scheduled*

Definition at line 1878 of file Object.cpp.

References any\_attribute\_FOM\_specified\_order, any\_attribute\_timestamp\_order, any\_locally\_owned\_published\_cyclic\_data\_ready\_attribute(), attribute\_values\_map, TrickHLA::CONFIG\_CYCLIC, create\_attribute\_set(), TrickHLA::DEBUG\_LEVEL\_7\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, TrickHLA::Int64Time::get(), get\_federate(), TrickHLA::Federate::get\_granted\_fed\_time(), TrickHLA::Federate::get\_lookahead(), get\_name(), get\_RTI\_ambassador(), get\_update\_time\_plus\_lookahead(), TrickHLA::Int64Interval::getDoubleTime(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Int64Interval::getTimeInMicros(), TrickHLA::Int64Time::getTimeInMicros(), TrickHLA::Federate::in\_time\_regulating\_state(), instance\_handle, TrickHLA::Federate::is\_zero\_lookahead\_time(), lag\_comp, lag\_comp\_type, TrickHLA::LAG\_COMPENSATION\_SEND\_SIDE, last\_update\_time, TrickHLA::Packing::pack(), pack\_cyclic\_attribute\_buffers(), packing, RTI1516\_EXCEPTION, RTI1516\_USERDATA, TrickHLA::LagCompensation::send\_lag\_compensation(), set\_last\_update\_time(), TrickHLA::Int64Time::setTo(), should\_print(), TrickHLA::Federate::should\_publish\_data(), THLA\_NEWLINE, time\_plus\_lookahead, TrickHLA::StringUtilities::to\_string(), TrickHLA::Int64Interval::toMicroseconds(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::Manager::send\_cyclic\_data().

### 7.36.3.78 send\_init\_data()

```
void Object::send_init_data ( )
```

Send initialization data to remote HLA federates.

#### Trick Job Class: *scheduled*

Definition at line 2483 of file Object.cpp.

References any\_locally\_owned\_published\_init\_attribute(), attribute\_values\_map, TrickHLA::CONFIG\_INITIALIZE, create\_attribute\_set(), TrickHLA::DEBUG\_LEVEL\_7\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, TrickHLA::Int64Time::get(), get\_federate(), TrickHLA::Federate::get\_granted\_fed\_time(), TrickHLA::Federate::get\_lookahead(), get\_name(), get\_RTI\_ambassador(), TrickHLA::Int64Interval::getDoubleTime(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Int64Interval::getTimeInMicros(), TrickHLA::Int64Time::getTimeInMicros(), instance\_handle, last\_update\_time, TrickHLA::Packing::pack(), pack\_init\_attribute\_buffers(), packing, RTI1516\_EXCEPTION, RTI1516\_USERDATA, set\_last\_update\_time(), should\_print(), TrickHLA::Federate::should\_publish\_data(), THLA\_NEWLINE, time\_plus\_lookahead, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by SpaceFOM::ExecutionControl::enter\_freeze(), SpaceFOM::ExecutionControl::epoch\_and\_root\_frame

\_discovery\_process(), SpaceFOM::ExecutionControl::exit\_freeze(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), IMSim::ExecutionControl::process\_mode\_transition\_request(), DSES::ExecutionControl::run\_mode\_transition(), DIS::ExecutionControl::run\_mode\_transition(), IMSim::ExecutionControl::run\_mode\_transition(), SpaceFOM::ExecutionControl::run\_mode\_transition(), TrickHLA::ExecutionControlBase::send\_execution\_configuration(), TrickHLA::Manager::send\_init\_data(), SpaceFOM::ExecutionControl::send\_root\_ref\_frame(), DSES::ExecutionControl::shutdown\_mode\_announce(), DIS::ExecutionControl::shutdown\_mode\_announce(), IMSim::ExecutionControl::shutdown\_mode\_announce(), and SpaceFOM::ExecutionControl::shutdown\_mode\_announce().

### 7.36.3.79 send\_requested\_data()

```
void Object::send_requested_data (
    double current_time,
    double cycle_time )
```

Send the requested data update that came from another federate requesting an attribute value update.

#### Parameters

<i>current_time</i>	Current time in seconds.
<i>cycle_time</i>	Cycle time between calls to this function in seconds.

#### Trick Job Class: *scheduled*

Definition at line 1561 of file Object.cpp.

References any\_attribute\_FOM\_specified\_order, any\_attribute\_timestamp\_order, any\_locally\_owned\_published\_requested\_attribute(), attr\_update\_requested, attribute\_values\_map, create\_requested\_attribute\_set(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_LEVEL\_7\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, TrickHLA::A::Int64Time::get(), get\_federate(), TrickHLA::Federate::get\_granted\_fed\_time(), TrickHLA::Federate::get\_lookahead(), get\_name(), get\_RTI\_ambassador(), get\_update\_time\_plus\_lookahead(), TrickHLA::Int64Interval::getDoubleTime(), TrickHLA::Int64Time::getDoubleTime(), TrickHLA::Int64Interval::getTimelnMicros(), TrickHLA::Int64Time::getTimelnMicros(), TrickHLA::Federate::in\_time\_regulating\_state(), instance\_handle, TrickHLA::Federate::is\_zero\_lookahead\_time(), lag\_comp, lag\_comp\_type, TrickHLA::LAG\_COMPENSATION\_SEND\_SIDE, last\_update\_time, TrickHLA::Packing::pack(), pack\_requested\_attribute\_buffers(), packing, RTI1516\_EXCEPTION, RTI1516\_USERDATA, TrickHLA::A::LagCompensation::send\_lag\_compensation(), set\_last\_update\_time(), TrickHLA::Int64Time::setTo(), should\_print(), TrickHLA::Federate::should\_publish\_data(), THLA\_NEWLINE, time\_plus\_lookahead, TrickHLA::StringUtilities::to\_string(), TrickHLA::Int64Interval::toMicroseconds(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::Manager::send\_requested\_data(), and TrickHLA::ExecutionControlBase::send\_requested\_data().

### 7.36.3.80 set\_class\_handle()

```
void TrickHLA::Object::set_class_handle (
    RTI1516_NAMESPACE::ObjectClassHandle id ) [inline]
```

Set the HLA [Object](#) class handle for this object.

#### Parameters

<i>id</i>	The HLA ObjectClassHandle for this object.
-----------	--

Definition at line 357 of file Object.hh.

References class\_handle.

Referenced by TrickHLA::Manager::setup\_object\_RTI\_handles().

#### 7.36.3.81 set\_core\_job\_cycle\_time()

```
void Object::set_core_job_cycle_time (  
    double cycle_time )
```

Sets the core job cycle time used by the multi-rate attributes.

##### Parameters

<i>cycle_time</i>	The core job cycle time in seconds.
-------------------	-------------------------------------

Definition at line 3840 of file Object.cpp.

References attr\_count, attributes, and TrickHLA::Attribute::determine\_cycle\_ratio().

Referenced by TrickHLA::Manager::determine\_job\_cycle\_time().

#### 7.36.3.82 set\_create\_HLA\_instance()

```
void TrickHLA::Object::set_create_HLA_instance (   
    bool create ) [inline]
```

Set the object instance creation status.

##### Parameters

<i>create</i>	The associated object instance creation status; True if the object has been created, otherwise False.
---------------	---

Definition at line 391 of file Object.hh.

References create\_HLA\_instance.

Referenced by TrickHLA::ExecutionConfigurationBase::reset\_ownership\_states(), and TrickHLA::ExecutionConfigurationBase::set\_master().

#### 7.36.3.83 set\_divest\_requested()

```
void TrickHLA::Object::set_divest_requested (   
    bool request ) [inline]
```

Set the ownership divestiture requested flag.

##### Parameters

<i>request</i>	The desired divestiture request state.
----------------	--

Definition at line 529 of file Object.hh.

References divest\_requested.

Referenced by TrickHLA::FedAmb::requestDivestitureConfirmation().

#### 7.36.3.84 set\_instance\_handle()

```
void TrickHLA::Object::set_instance_handle (   
    RTI1516_NAMESPACE::ObjectInstanceHandle id ) [inline]
```

Set the HLA [Object](#) instance handle for this object instance.

## Parameters

<i>id</i>	The HLA ObjectInstanceHandle for this object instance.
-----------	--

Definition at line 369 of file Object.hh.

References `instance_handle`.

Referenced by `set_instance_handle_and_name()`, and `TrickHLA::Manager::set_object_instance_handles_by_name()`.

### 7.36.3.85 `set_instance_handle_and_name()`

```
void TrickHLA::Object::set_instance_handle_and_name (
    RTI1516_NAMESPACE::ObjectInstanceHandle id,
    std::wstring const & instance_name ) [inline]
```

Set the HLA [Object](#) instance handle and name for this object instance.

## Parameters

<i>id</i>	The HLA ObjectInstanceHandle for this object instance.
<i>instance_name</i>	The associated object instance name.

Definition at line 374 of file Object.hh.

References `set_instance_handle()`, `set_name()`, `set_name_registered()`, and `TrickHLA::StringUtilities::to_string()`.

Referenced by `TrickHLA::Manager::discover_object_instance()`.

### 7.36.3.86 `set_lag_compensation_type()`

```
void TrickHLA::Object::set_lag_compensation_type (
    LagCompensationEnum lag_type ) [inline]
```

Set the Lag Compensation type for object attribute updates.

## Parameters

<i>lag_type</i>	Desired lag compensation type.
-----------------	--------------------------------

Definition at line 517 of file Object.hh.

Referenced by `SpaceFOM::ExecutionConfiguration::configure()`.

### 7.36.3.87 `set_last_update_time()`

```
void TrickHLA::Object::set_last_update_time (
    RTI1516_NAMESPACE::LogicalTime const & time ) [inline]
```

Set the last update time.

## Parameters

<i>time</i>	The last HLA logical time update value.
-------------	---

Definition at line 549 of file Object.hh.

References `last_update_time`, and `TrickHLA::Int64Time::setTo()`.

Referenced by `TrickHLA::FedAmb::reflectAttributeValues()`, `send_cyclic_data()`, `send_init_data()`, and `send_`

requested\_data()).

### 7.36.3.88 set\_name()

```
void Object::set_name (
    const char * new_name ) [private]
```

Sets the new value of the name attribute.

#### Parameters

<i>new_name</i>	New name for the object instance.
-----------------	-----------------------------------

Definition at line 3848 of file Object.cpp.

References name.

Referenced by initialize(), register\_object\_with\_RTI(), set\_instance\_handle\_and\_name(), and set\_name\_and\_mark\_changed().

### 7.36.3.89 set\_name\_and\_mark\_changed()

```
void TrickHLA::Object::set_name_and_mark_changed (
    const char * new_name ) [inline], [private]
```

Set the name of the object and mark it as changed.

#### Parameters

<i>new_name</i>	The new name of the object.
-----------------	-----------------------------

Definition at line 721 of file Object.hh.

References mark\_changed(), and set\_name().

### 7.36.3.90 set\_name\_registered()

```
void TrickHLA::Object::set_name_registered ( ) [inline]
```

Set the name registration status as true (registered).

Definition at line 342 of file Object.hh.

References name\_registered.

Referenced by TrickHLA::ExecutionControlBase::object\_instance\_name\_reservation\_succeeded(), TrickHLA::Manager::object\_instance\_name\_reservation\_succeeded(), and set\_instance\_handle\_and\_name().

### 7.36.3.91 set\_name\_unregistered()

```
void TrickHLA::Object::set_name_unregistered ( ) [inline]
```

Set the name registration status as false (not registered).

Definition at line 345 of file Object.hh.

References name\_registered.

### 7.36.3.92 set\_pull\_requested()

```
void TrickHLA::Object::set_pull_requested (
    bool request ) [inline]
```

Set ownership pull requested flag.

#### Parameters

<code>request</code>	The desired pull request state.
----------------------	---------------------------------

Definition at line 533 of file Object.hh.

References `pull_requested`.

Referenced by `TrickHLA::FedAmb::requestAttributeOwnershipRelease()`.

#### 7.36.3.93 `set_to_unblocking_cyclic_reads()`

`void Object::set_to_unblocking_cyclic_reads ( )`

Set to unblocking cyclic reads and notify any waiting threads.

Definition at line 4040 of file Object.cpp.

References `blocking_cyclic_read`.

Referenced by `remove_object_instance()`, and `~Object()`.

#### 7.36.3.94 `setup_ownership_transfer_checkpointed_data()`

`void Object::setup_ownership_transfer_checkpointed_data ( )`

Setup the checkpoint data structures.

Definition at line 3818 of file Object.cpp.

References `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_OBJECT`, `get_name()`, `ownership`, `TrickHLA::OwnershipHandler::setup_checkpoint_requests()`, `should_print()`, and `THLA_NEWLINE`.

Referenced by `TrickHLA::Manager::setup_checkpoint()`, and `TrickHLA::ExecutionControlBase::setup_checkpoint()`.

#### 7.36.3.95 `setup_preferred_order_with_RTI()`

`void Object::setup_preferred_order_with_RTI ( )`

Setup the preferred order for the locally owned attributes.

**Trick Job Class:** *scheduled*

Definition at line 1311 of file Object.cpp.

References `any_locally_owned_attribute()`, `attr_count`, `attributes`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_OBJECT`, `get_FOM_name()`, `get_name()`, `get_RTI_ambassador()`, `TrickHLA::Attribute::get_trick_name()`, `instance_handle`, `is_create_HLA_instance()`, `is_instance_handle_valid()`, `RTI1516_EXCEPTION`, `should_print()`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, `TrickHLA::TRANSPORT_RECEIVE_ORDER`, `TrickHLA::TRANSPORT_TIMESTAMP_ORDER`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `TrickHLA::Manager::setup_preferred_order_with_RTI()`.

#### 7.36.3.96 `should_print()`

```
bool Object::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const
```

Determine if the verbose debug comments should be printed to the console.

#### Returns

Returns true if the requested message should print level.

**Parameters**

<i>level</i>	Debug level of the incoming message.
<i>code</i>	Source code association of the incoming messages.

Definition at line 405 of file Object.cpp.

References manager, and TrickHLA::Manager::should\_print().

Referenced by create\_attribute\_set(), create\_requested\_attribute\_set(), extract\_data(), grant\_pull\_request(), grant\_push\_request(), initialize(), mark\_all\_attributes\_as\_nonlocal(), process\_deleted\_object(), provide\_attribute\_update(), publish\_object\_attributes(), pull\_ownership(), pull\_ownership\_upon\_rejoin(), push\_ownership(), register\_object\_with\_RTI(), release\_ownership(), remove\_object\_instance(), reserve\_object\_name\_with\_RTI(), restore\_ownership\_transfer\_checkpointed\_data(), send\_cyclic\_data(), send\_init\_data(), send\_requested\_data(), setup\_ownership\_transfer\_checkpointed\_data(), setup\_preferred\_order\_with\_RTI(), TrickHLA::LagCompensation::should\_print(), subscribe\_to\_object\_attributes(), unpublish\_all\_object\_attributes(), unsubscribe\_all\_object\_attributes(), wait\_on\_object\_name\_reservation(), and wait\_on\_object\_registration().

**7.36.3.97 stop\_publishing\_attributes()**

```
void Object::stop_publishing_attributes ( )
```

Stops publishing data for the object attributes by setting the attribute publish state to false.

Definition at line 3901 of file Object.cpp.

References attr\_count, attributes, and TrickHLA::Attribute::set\_publish().

**7.36.3.98 stop\_subscribing\_attributes()**

```
void Object::stop_subscribing_attributes ( )
```

Stops subscribing to object attribute data by setting the attribute subscribe state to false.

Definition at line 3908 of file Object.cpp.

References attr\_count, attributes, and TrickHLA::Attribute::set\_subscribe().

**7.36.3.99 subscribe\_to\_object\_attributes()**

```
void Object::subscribe_to_object_attributes ( )
```

Subscribe to [Object](#) attributes.

**Trick Job Class:** *initialization*

Definition at line 805 of file Object.cpp.

References any\_attribute\_subscribed(), attr\_count, attributes, class\_handle, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_name(), get\_RTI\_ambassador(), RTI1516\_EXCEPTION, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::subscribe(), IMSim::ExecutionControl::subscribe(), and TrickHLA::Manager::subscribe().

**7.36.3.100 unlock()**

```
void TrickHLA::Object::unlock ( ) [inline]
```

Unlock the thread mutex.

Definition at line 624 of file Object.hh.

References mutex.

Referenced by `grant_push_request()`, and `TrickHLA::FedAmb::requestAttributeOwnershipAssumption()`.

#### 7.36.3.101 `unpack_attribute_buffers()`

```
void Object::unpack_attribute_buffers (
    const DataUpdateEnum attr_config )
```

Unpack the buffer back into the attributes that have the given configuration.

##### Parameters

<code>attr_config</code>	<code>Attribute</code> configuration.
--------------------------	---------------------------------------

Definition at line 4030 of file `Object.cpp`.

References `attr_count`, `attributes`, and `TrickHLA::Attribute::unpack_attribute_buffer()`.

Referenced by `unpack_cyclic_attribute_buffers()`, and `unpack_init_attribute_buffers()`.

#### 7.36.3.102 `unpack_cyclic_attribute_buffers()`

```
void TrickHLA::Object::unpack_cyclic_attribute_buffers ( ) [inline]
```

Copy the packed buffer contents back to each cyclic attribute.

Definition at line 612 of file `Object.hh`.

References `TrickHLA::CONFIG_CYCLIC`, and `unpack_attribute_buffers()`.

Referenced by `receive_cyclic_data()`.

#### 7.36.3.103 `unpack_init_attribute_buffers()`

```
void TrickHLA::Object::unpack_init_attribute_buffers ( ) [inline]
```

Copy the packed buffer contents back to each dynamic initialization attribute.

Definition at line 618 of file `Object.hh`.

References `TrickHLA::CONFIG_INITIALIZE`, and `unpack_attribute_buffers()`.

Referenced by `receive_init_data()`.

#### 7.36.3.104 `unpublish_all_object_attributes()`

```
void Object::unpublish_all_object_attributes ( )
```

Unpublishes all object attributes.

**Trick Job Class:** *initialization*

Definition at line 732 of file `Object.cpp`.

References `any_attribute_published()`, `class_handle`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBUG_SOURCE_OBJECT`, `get_name()`, `get_RTI_ambassador()`, `RTI1516_EXCEPTION`, `should_print()`, `THLA_NEWLINE`, `TrickHLA::StringUtilities::to_string()`, `TRICKHLA_RESTORE_FPU_CONTROL_WORD`, `TRICKHLA_SAVE_FPU_CONTROL_WORD`, and `TRICKHLA_VALIDATE_FPU_CONTROL_WORD`.

Referenced by `SpaceFOM::ExecutionControl::unpublish()`, `IMSim::ExecutionControl::unpublish()`, and `TrickHLA::Manager::unpublish()`.

#### 7.36.3.105 `unsubscribe_all_object_attributes()`

```
void Object::unsubscribe_all_object_attributes ( )
```

Unsubscribe from all the `Object` attributes.

**Trick Job Class:** *initialization*

Definition at line 897 of file Object.cpp.

References any\_attribute\_subscribed(), class\_handle, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_name(), get\_RTI\_ambassador(), RTI1516\_EXCEPTION, should\_print(), THLA\_NEWLINE, TrickHLA::StringUtilities::to\_string(), TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by SpaceFOM::ExecutionControl::unsubscribe(), IMSim::ExecutionControl::unsubscribe(), and TrickHLA::Manager::unsubscribe().

### 7.36.3.106 wait\_on\_object\_name\_reservation()

```
void Object::wait_on_object_name_reservation ( )
```

Waits on the reservation of the HLA object instance name with the RTI.

Calling this function will block until the object instance name is reserved, but only if the object is locally owned. **Trick Job Class:** *initialization*

Definition at line 1055 of file Object.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, get\_federate(), get\_name(), is\_create\_HLA\_instance(), TrickHLA::Federate::is\_execution\_member(), is\_instance\_handle\_valid(), is\_name\_required(), name\_registered, should\_print(), THLA\_ENDL, and THLA\_NEWLINE.

Referenced by DSES::ExecutionControl::determine\_federation\_master(), SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), and TrickHLA::Manager::wait\_on\_reservation\_of\_object\_names().

### 7.36.3.107 wait\_on\_object\_registration()

```
void Object::wait_on_object_registration ( )
```

Waits on the registration of this HLA object instance with the RTI.

Calling this function will block until the object instance is registered. **Trick Job Class:** *initialization*

Definition at line 1267 of file Object.cpp.

References TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_OBJECT, FOM\_name, get\_federate(), get\_name(), TrickHLA::Federate::is\_execution\_member(), is\_instance\_handle\_valid(), should\_print(), THLA\_ENDL, and THLA\_NEWLINE.

## 7.36.4 Friends And Related Function Documentation

### 7.36.4.1 init\_attrTrickHLA\_\_Object

```
void init_attrTrickHLA__Object ( ) [friend]
```

### 7.36.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 110 of file Object.hh.

## 7.36.5 Field Documentation

### 7.36.5.1 any\_attribute\_FOM\_specified\_order

```
bool TrickHLA::Object::any_attribute_FOM_specified_order [protected]
```

**Units:** –

True if any attribute is the FOM specified order.

Definition at line 688 of file Object.hh.

Referenced by initialize(), send\_cyclic\_data(), and send\_requested\_data().

### 7.36.5.2 any\_attribute\_timestamp\_order

```
bool TrickHLA::Object::any_attribute_timestamp_order [protected]
```

**Units:** –

True if any attribute is timestamp order.

Definition at line 689 of file Object.hh.

Referenced by initialize(), TrickHLA::ExecutionConfigurationBase::reset\_preferred\_order(), send\_cyclic\_data(), and send\_requested\_data().

### 7.36.5.3 attr\_count

```
int TrickHLA::Object::attr_count
```

**Units:** –

Number of object attributes.

Definition at line 136 of file Object.hh.

Referenced by any\_attribute\_published(), any\_attribute\_subscribed(), any\_locally\_owned\_attribute(), any\_locally\_owned\_published\_attribute(), any\_locally\_owned\_published\_cyclic\_data\_ready\_attribute(), any\_locally\_owned\_published\_requested\_attribute(), any\_remotely\_owned\_subscribed\_attribute(), build\_attribute\_map(), TrickHLA::ExecutionConfiguration::configure\_attributes(), DIS::ExecutionConfiguration::configure\_attributes(), DSES::ExecutionConfiguration::configure\_attributes(), IMSim::ExecutionConfiguration::configure\_attributes(), SpaceFOM::ExecutionConfiguration::configure\_attributes(), create\_attribute\_set(), create\_requested\_attribute\_set(), SpaceFOM::RefFrameBase::default\_data(), get\_attribute(), get\_attribute\_count(), grant\_pull\_request(), grant\_push\_request(), initialize(), mark\_all\_attributes\_as\_nonlocal(), mark\_unchanged(), pack\_attribute\_buffers(), pack\_requested\_attribute\_buffers(), provide\_attribute\_update(), publish\_object\_attributes(), pull\_ownership\_upon\_rejoin(), release\_ownership(), request\_attribute\_value\_update(), TrickHLA::ExecutionConfigurationBase::reset\_ownership\_states(), TrickHLA::ExecutionConfigurationBase::reset\_preferred\_order(), set\_core\_job\_cycle\_time(), TrickHLA::ExecutionConfigurationBase::set\_master(), setup\_preferred\_order\_with\_RTI(), SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes(), stop\_publishing\_attributes(), stop\_subscribing\_attributes(), subscribe\_to\_object\_attributes(), and unpack\_attribute\_buffers().

### 7.36.5.4 attr\_update\_requested

```
bool TrickHLA::Object::attr_update_requested [protected]
```

**Units:** –

Flag to indicate an attribute updated was requested by another federate.

Definition at line 684 of file Object.hh.

Referenced by is\_attribute\_update\_requested(), provide\_attribute\_update(), and send\_requested\_data().

### 7.36.5.5 attribute\_FOM\_names

```
VectorOfStrings TrickHLA::Object::attribute_FOM_names [protected]
```

**Data I/O:** \*\*

String array containing the [Attribute](#) FOM names.

Definition at line 700 of file Object.hh.

Referenced by `get_attribute_FOM_names()`, and `initialize()`.

### 7.36.5.6 attribute\_values\_map

`RTI1516_NAMESPACE::AttributeHandleValueMap* TrickHLA::Object::attribute_values_map [protected]`

**Data I/O:** \*\*

Map of attributes that will be sent as an update to other federates.

Definition at line 707 of file Object.hh.

Referenced by `create_attribute_set()`, `create_requested_attribute_set()`, `Object()`, `send_cyclic_data()`, `send_init_data()`, `send_requested_data()`, and `~Object()`.

### 7.36.5.7 attributes

`Attribute* TrickHLA::Object::attributes`

**Units:** –

Array of object attributes.

Definition at line 137 of file Object.hh.

Referenced by `any_attribute_published()`, `any_attribute_subscribed()`, `any_locally_owned_attribute()`, `any_locally_owned_published_attribute()`, `any_locally_owned_published_cyclic_data_ready_attribute()`, `any_locally_owned_published_requested_attribute()`, `any_remotely_owned_subscribed_attribute()`, `build_attribute_map()`, `TrickHLA::ExecutionConfiguration::configure_attributes()`, `DIS::ExecutionConfiguration::configure_attributes()`, `DSES::ExecutionConfiguration::configure_attributes()`, `IMSim::ExecutionConfiguration::configure_attributes()`, `SpaceFO::M::ExecutionConfiguration::configure_attributes()`, `create_attribute_set()`, `create_requested_attribute_set()`, `get_attribute()`, `get_attributes()`, `grant_pull_request()`, `grant_push_request()`, `initialize()`, `mark_all_attributes_as_nonlocal()`, `mark_unchanged()`, `pack_attribute_buffers()`, `pack_requested_attribute_buffers()`, `provide_attribute_update()`, `publish_object_attributes()`, `pull_ownership_upon_rejoin()`, `release_ownership()`, `request_attribute_value_update()`, `TrickHLA::ExecutionConfigurationBase::reset_ownership_states()`, `TrickHLA::ExecutionConfigurationBase::reset_preferred_order()`, `set_core_job_cycle_time()`, `TrickHLA::ExecutionConfigurationBase::set_master()`, `setup_preferred_order_with_RTI()`, `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`, `stop_publishing_attributes()`, `stop_subscribing_attributes()`, `subscribe_to_object_attributes()`, and `unpack_attribute_buffers()`.

### 7.36.5.8 blocking\_cyclic\_read

`bool TrickHLA::Object::blocking_cyclic_read`

**Units:** –

True to block in `receive_cyclic_data` for data to be received.

Definition at line 134 of file Object.hh.

Referenced by `receive_cyclic_data()`, `set_to_unblocking_cyclic_reads()`, and `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`.

### 7.36.5.9 changed

`bool TrickHLA::Object::changed [protected]`

**Units:** –

Flag indicating the data has changed.

Definition at line 682 of file Object.hh.

Referenced by `is_changed()`, `mark_changed()`, and `mark_unchanged()`.

### 7.36.5.10 class\_handle

RTI1516\_NAMESPACE::ObjectClassHandle TrickHLA::Object::class\_handle [protected]

**Data I/O:** \*\*

HLA [Object](#) Class handle.

Definition at line 691 of file Object.hh.

Referenced by `get_class_handle()`, `publish_object_attributes()`, `register_object_with_RTI()`, `set_class_handle()`, `subscribe_to_object_attributes()`, `unpublish_all_object_attributes()`, and `unsubscribe_all_object_attributes()`.

### 7.36.5.11 clock

BasicClock TrickHLA::Object::clock [protected]

**Units:** –

Clock time object.

Definition at line 678 of file Object.hh.

Referenced by `receive_cyclic_data()`.

### 7.36.5.12 create\_HLA\_instance

bool TrickHLA::Object::create\_HLA\_instance

**Units:** –

Set to true to create an HLA named instance of this object.

Definition at line 130 of file Object.hh.

Referenced by `is_create_HLA_instance()`, `set_create_HLA_instance()`, and `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`.

### 7.36.5.13 data\_changed

bool TrickHLA::Object::data\_changed

**Units:** –

Flag to indicate data changes.

Definition at line 121 of file Object.hh.

Referenced by `extract_data()`, and `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`.

### 7.36.5.14 deleted

[ObjectDeleted](#)\* TrickHLA::Object::deleted

**Units:** –

Object Deleted callback object.

Definition at line 146 of file Object.hh.

Referenced by `initialize()`, `process_deleted_object()`, and `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`.

### 7.36.5.15 divest\_requested

bool TrickHLA::Object::divest\_requested [protected]

**Units:** –

Are we releasing ownership?

Definition at line 698 of file Object.hh.

Referenced by `release_ownership()`, and `set_divest_requested()`.

### 7.36.5.16 FOM\_name

`char* TrickHLA::Object::FOM_name`

**Units:** –

FOM name for the object.

Definition at line 128 of file Object.hh.

Referenced by `TrickHLA::ExecutionConfiguration::configure_attributes()`, `DIS::ExecutionConfiguration::configure_attributes()`, `DSES::ExecutionConfiguration::configure_attributes()`, `IMSim::ExecutionConfiguration::configure_attributes()`, `SpaceFOM::ExecutionConfiguration::configure_attributes()`, `extract_data()`, `get_FOM_name()`, `initialize()`, `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`, and `wait_on_object_registration()`.

### 7.36.5.17 instance\_handle

`RTI1516_NAMESPACE::ObjectInstanceHandle TrickHLA::Object::instance_handle [protected]`

**Data I/O:** \*\*

HLA `Object` Instance handle.

Definition at line 692 of file Object.hh.

Referenced by `get_instance_handle()`, `grant_pull_request()`, `grant_push_request()`, `is_instance_handle_valid()`, `mark_all_attributes_as_nonlocal()`, `negotiated_attribute_ownership_divestiture()`, `process_deleted_object()`, `pull_ownership()`, `pull_ownership_upon_rejoin()`, `register_object_with_RTI()`, `release_ownership()`, `remove()`, `remove_object_instance()`, `request_attribute_value_update()`, `send_cyclic_data()`, `send_init_data()`, `send_requested_data()`, `set_instance_handle()`, and `setup_preferred_order_with_RTI()`.

### 7.36.5.18 lag\_comp

`LagCompensation* TrickHLA::Object::lag_comp`

**Units:** –

Lag compensation object.

Definition at line 139 of file Object.hh.

Referenced by `initialize()`, `receive_cyclic_data()`, `send_cyclic_data()`, `send_requested_data()`, and `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`.

### 7.36.5.19 lag\_comp\_type

`LagCompensationEnum TrickHLA::Object::lag_comp_type`

**Units:** –

Type of lag compensation.

Definition at line 140 of file Object.hh.

Referenced by `get_lag_compensation_type()`, `initialize()`, `receive_cyclic_data()`, `send_cyclic_data()`, `send_requested_data()`, and `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`.

### 7.36.5.20 last\_update\_time

`Int64Time TrickHLA::Object::last_update_time [protected]`

**Data I/O:** \*\*

Last update time.

Definition at line 694 of file Object.hh.

Referenced by `get_last_update_time()`, `get_update_time_plus_lookahead()`, `send_cyclic_data()`, `send_init_data()`, `send_requested_data()`, and `set_last_update_time()`.

### 7.36.5.21 manager

Manager\* TrickHLA::Object::manager [protected]

**Units:** –

Reference to the [TrickHLA Manager](#).

Definition at line 702 of file Object.hh.

Referenced by TrickHLA::ExecutionConfiguration::configure(), SpaceFOM::ExecutionConfiguration::configure(), get\_fed\_lookahead(), get\_federate(), get\_granted\_fed\_time(), get\_granted\_time(), initialize(), remove(), and should\_print().

### 7.36.5.22 mutex

pthread\_mutex\_t TrickHLA::Object::mutex [protected]

**Data I/O:** \*\*

Mutex to lock thread over critical code sections.

Definition at line 670 of file Object.hh.

Referenced by lock(), Object(), unlock(), and ~Object().

### 7.36.5.23 name

char\* TrickHLA::Object::name

**Units:** –

[Object](#) Instance Name.

Definition at line 125 of file Object.hh.

Referenced by SpaceFOM::ExecutionConfiguration::configure(), TrickHLA::ExecutionConfiguration::configure\_attributes(), DSES::ExecutionConfiguration::configure\_attributes(), DIS::ExecutionConfiguration::configure\_attributes(), IMSim::ExecutionConfiguration::configure\_attributes(), SpaceFOM::ExecutionConfiguration::configure\_attributes(), DIS::ExecutionConfiguration::ExecutionConfiguration(), DSES::ExecutionConfiguration::ExecutionConfiguration(), IMSim::ExecutionConfiguration::ExecutionConfiguration(), TrickHLA::ExecutionConfigurationBase::ExecutionConfigurationBase(), extract\_data(), get\_name(), get\_name\_string(), initialize(), set\_name(), DSES::ExecutionConfiguration::set\_root\_frame\_name(), DIS::ExecutionConfiguration::set\_root\_frame\_name(), IMSim::ExecutionConfiguration::set\_root\_frame\_name(), SpaceFOM::ExecutionConfiguration::set\_root\_frame\_name(), SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes(), and ~Object().

### 7.36.5.24 name\_registered

bool TrickHLA::Object::name\_registered [protected]

**Units:** –

True if the object instance name is registered.

Definition at line 680 of file Object.hh.

Referenced by is\_name\_registered(), set\_name\_registered(), set\_name\_unregistered(), and wait\_on\_object\_name\_reservation().

### 7.36.5.25 name\_required

bool TrickHLA::Object::name\_required

**Units:** –

True (default) to require an object instance name be specified by you, or false to use the instance name automatically assigned by the RTI.

Definition at line 126 of file Object.hh.

Referenced by is\_name\_required(), and SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

### 7.36.5.26 object\_deleted\_from\_RTI

bool TrickHLA::Object::object\_deleted\_from\_RTI

**Units:** –

Flag that is true when this object has been deleted from the RTI.

Definition at line 147 of file Object.hh.

Referenced by is\_object\_deleted\_from\_RTI(), process\_deleted\_object(), remove\_object\_instance(), and SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

### 7.36.5.27 ownership

OwnershipHandler\* TrickHLA::Object::ownership

**Units:** –

Manages attribute ownership.

Definition at line 144 of file Object.hh.

Referenced by SpaceFOM::ExecutionConfiguration::configure(), initialize(), pull\_ownership(), push\_ownership(), release\_ownership(), restore\_ownership\_transfer\_checkpointed\_data(), setup\_ownership\_transfer\_checkpointed\_data(), and SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

### 7.36.5.28 ownership\_mutex

pthread\_mutex\_t TrickHLA::Object::ownership\_mutex [protected]

**Data I/O:** \*\*

Mutex to lock thread over attribute ownership code sections.

Definition at line 671 of file Object.hh.

Referenced by Object(), ownership\_lock(), ownership\_unlock(), and ~Object().

### 7.36.5.29 packing

Packing\* TrickHLA::Object::packing

**Units:** –

Data pack/unpack object.

Definition at line 142 of file Object.hh.

Referenced by TrickHLA::ExecutionConfiguration::configure\_attributes(), DIS::ExecutionConfiguration::configure\_attributes(), DSES::ExecutionConfiguration::configure\_attributes(), IMSim::ExecutionConfiguration::configure\_attributes(), SpaceFOM::ExecutionConfiguration::configure\_attributes(), DSES::ExecutionConfiguration::ExecutionConfiguration(), DIS::ExecutionConfiguration::ExecutionConfiguration(), IMSim::ExecutionConfiguration::ExecutionConfiguration(), TrickHLA::ExecutionConfigurationBase::ExecutionConfigurationBase(), initialize(), receive\_cyclic\_data(), receive\_init\_data(), send\_cyclic\_data(), send\_init\_data(), send\_requested\_data(), and SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

### 7.36.5.30 pull\_requested

bool TrickHLA::Object::pull\_requested [protected]

**Units:** –

Has someone asked to own us?

Definition at line 697 of file Object.hh.

Referenced by grant\_pull\_request(), and set\_pull\_requested().

### 7.36.5.31 removed\_instance

```
bool TrickHLA::Object::removed_instance [protected]
```

**Units:** –

Flag to indicate if object instance was removed from RTI.

Definition at line 686 of file Object.hh.

Referenced by `remove()`, and `~Object()`.

### 7.36.5.32 required

```
bool TrickHLA::Object::required
```

**Units:** –

Flag indicating object is required at federation start ( default: true )

Definition at line 132 of file Object.hh.

Referenced by `is_required()`, `remove_object_instance()`, and `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`.

### 7.36.5.33 rti\_ambassador

```
RTI1516_NAMESPACE::RTIambassador* TrickHLA::Object::rti_ambassador [protected]
```

**Data I/O:** \*\*

Reference to the RTI ambassador.

Definition at line 704 of file Object.hh.

Referenced by `get_RTI_ambassador()`.

### 7.36.5.34 thla\_attribute\_map

```
AttributeMap TrickHLA::Object::thla_attribute_map [protected]
```

**Data I/O:** \*\*

Map of the [Attribute](#)'s, key is the AttributeHandle.

Definition at line 712 of file Object.hh.

Referenced by `build_attribute_map()`, `get_attribute()`, and `~Object()`.

### 7.36.5.35 thla\_reflected\_attributes\_queue

```
ReflectedAttributesQueue TrickHLA::Object::thla_reflected_attributes_queue [protected]
```

**Data I/O:** \*\*

Queue of reflected attributes.

Definition at line 710 of file Object.hh.

Referenced by `is_changed()`.

### 7.36.5.36 time\_plus\_lookahead

```
Int64Time TrickHLA::Object::time_plus_lookahead [protected]
```

**Data I/O:** \*\*

Time plus lookahead.

Definition at line 695 of file Object.hh.

Referenced by `get_update_time_plus_lookahead()`, `send_cyclic_data()`, `send_init_data()`, and `send_requested_data()`.

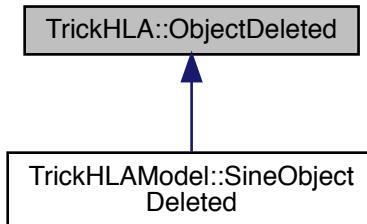
The documentation for this class was generated from the following files:

- [Object.hh](#)

- [Object.cpp](#)

## 7.37 TrickHLA::ObjectDeleted Class Reference

```
#include <ObjectDeleted.hh>
Inheritance diagram for TrickHLA::ObjectDeleted:
```



### Public Member Functions

- [ObjectDeleted \(\)](#)  
*Default constructor for the [TrickHLA ObjectDeleted](#) class.*
- [virtual ~ObjectDeleted \(\)](#)  
*Destructor for the [TrickHLA ObjectDeleted](#) class.*
- [virtual void deleted \(Object \\*theObj\)](#)  
*Mark this specified object as deleted.*

### Friends

- [class InputProcessor](#)
- [void init\\_attrTrickHLA\\_\\_ObjectDeleted \(\)](#)

#### 7.37.1 Detailed Description

Definition at line 46 of file ObjectDeleted.hh.

#### 7.37.2 Constructor & Destructor Documentation

##### 7.37.2.1 ObjectDeleted()

```
TrickHLA::ObjectDeleted::ObjectDeleted ( ) [inline]
Default constructor for the TrickHLA ObjectDeleted class.
Definition at line 63 of file ObjectDeleted.hh.
```

### 7.37.2.2 ~ObjectDeleted()

```
virtual TrickHLA::ObjectDeleted::~ObjectDeleted ( ) [inline], [virtual]
```

Destructor for the [TrickHLA ObjectDeleted](#) class.

Definition at line 65 of file ObjectDeleted.hh.

## 7.37.3 Member Function Documentation

### 7.37.3.1 deleted()

```
virtual void TrickHLA::ObjectDeleted::deleted (   
    Object * theObj ) [inline], [virtual]
```

Mark this specified object as deleted.

Parameters

<i>theObj</i>	Deleted object.
---------------	-----------------

Reimplemented in [TrickHLAModel::SineObjectDeleted](#).

Definition at line 75 of file ObjectDeleted.hh.

Referenced by [TrickHLA::Object::process\\_deleted\\_object\(\)](#).

## 7.37.4 Friends And Related Function Documentation

### 7.37.4.1 init\_attrTrickHLA\_\_ObjectDeleted

```
void init_attrTrickHLA__ObjectDeleted ( ) [friend]
```

### 7.37.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 53 of file ObjectDeleted.hh.

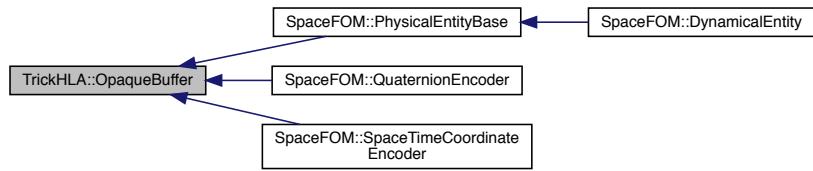
The documentation for this class was generated from the following file:

- [ObjectDeleted.hh](#)

## 7.38 TrickHLA::OpaqueBuffer Class Reference

```
#include <OpaqueBuffer.hh>
```

Inheritance diagram for TrickHLA::OpaqueBuffer:



## Public Member Functions

- **OpaqueBuffer ()**  
*Default constructor for the [TrickHLA OpaqueBuffer](#) class.*
- **virtual ~OpaqueBuffer ()**  
*Destructor for the [TrickHLA OpaqueBuffer](#) class.*
- **void [set\\_byte\\_alignment](#) (unsigned int size)**  
*Set the byte alignment to use for the data in the buffer.*
- **unsigned int [get\\_byte\\_alignment](#) () const**  
*Get the current buffer byte alignment.*
- **size\_t [get\\_capacity](#) () const**  
*Get the current buffer capacity.*
- **void [ensure\\_buffer\\_capacity](#) (size\_t size)**  
*Ensure the buffer has at least the specified capacity.*
- **void [reset\\_push\\_position](#) ()**  
*Reset the push buffer position.*
- **void [reset\\_pull\\_position](#) ()**  
*Reset the pull buffer position.*
- **void [reset\\_buffer\\_positions](#) ()**  
*Reset both the push and pull buffer positions.*
- **void [push\\_to\\_buffer](#) (void \*src, size\_t size)**  
*Push the specified data into the buffer with no encoding.*
- **void [push\\_to\\_buffer](#) (void \*src, size\_t size, [EncodingException](#) encoding)**  
*Push the specified data into the buffer using the specified encoding.*
- **void [pull\\_from\\_buffer](#) (void \*dest, size\_t size)**  
*Pull the specified number of data bytes from the buffer into the specified destination with no encoding.*
- **void [pull\\_from\\_buffer](#) (void \*dest, size\_t size, [EncodingException](#) encoding)**  
*Pull the specified number of data bytes from the buffer into the specified destination and for the specified encoding of the data in the buffer.*

## Data Fields

- **unsigned int [alignment](#)**  
**Units:** –  
*The byte alignment to use for the buffer.*
- **size\_t [push\\_pos](#)**

- **size\_t** [pull\\_pos](#)  
**Units:** –  
*Position to push data to.*
- **size\_t** [capacity](#)  
**Units:** –  
*Capacity of the buffer.*
- **unsigned char \*** [buffer](#)  
**Units:** –  
*Byte buffer.*

## Protected Member Functions

- **void** [push\\_pad\\_to\\_buffer](#) (**size\_t** pad\_size)  
*Push the specified number of pad bytes to the buffer.*
- **void** [pull\\_pad\\_from\\_buffer](#) (**size\_t** pad\_size)  
*Pull the specified number of pad bytes from the buffer.*
- **void** [byteswap\\_buffer\\_copy](#) (**void** \*dest, **void** \*src, **size\_t** size, [EncodingException](#) encoding)  
*Copy the data from the source to the destination and byteswap if needed.*

## Friends

- **class** [InputProcessor](#)
- **void** [init\\_attrTrickHLA\\_OpaqueBuffer](#) ()

### 7.38.1 Detailed Description

Definition at line 46 of file OpaqueBuffer.hh.

### 7.38.2 Constructor & Destructor Documentation

#### 7.38.2.1 OpaqueBuffer()

`OpaqueBuffer::OpaqueBuffer ( )`  
Default constructor for the [TrickHLA OpaqueBuffer](#) class.

**Trick Job Class:** *initialization*

Definition at line 53 of file OpaqueBuffer.cpp.

References [ensure\\_buffer\\_capacity\(\)](#).

#### 7.38.2.2 ~OpaqueBuffer()

`OpaqueBuffer::~OpaqueBuffer ( ) [virtual]`  
Destructor for the [TrickHLA OpaqueBuffer](#) class.  
Frees allocated memory. **Trick Job Class:** *shutdown*  
Definition at line 70 of file OpaqueBuffer.cpp.  
References [buffer](#), [capacity](#), [pull\\_pos](#), and [push\\_pos](#).

### 7.38.3 Member Function Documentation

### 7.38.3.1 byteswap\_buffer\_copy()

```
void OpaqueBuffer::byteswap_buffer_copy (
    void * dest,
    void * src,
    size_t size,
    EncodingEnum encoding ) [protected]
```

Copy the data from the source to the destination and byteswap if needed.

#### Parameters

<i>dest</i>	Destination to put data.
<i>src</i>	Source data.
<i>size</i>	Number of bytes to copy.
<i>encoding</i>	One of ENCODING_TYPE_LITTLE_ENDIAN, ENCODING_TYPE_BIG_ENDIAN, or ENCODING_TYPE_NO_ENCODING.

#### Assumptions and Limitations:

- The destination must be large enough to hold *size* bytes of data.

Definition at line 305 of file OpaqueBuffer.cpp.

References TrickHLA::Utilities::byteswap\_double(), TrickHLA::Utilities::byteswap\_unsigned\_int(), TrickHLA::Utilities::byteswap\_unsigned\_long\_long(), TrickHLA::Utilities::byteswap\_unsigned\_short(), TrickHLA::Utilities::is\_transmission\_byteswap(), and THLA\_ENDIAN.

Referenced by pull\_from\_buffer(), and push\_to\_buffer().

### 7.38.3.2 ensure\_buffer\_capacity()

```
void OpaqueBuffer::ensure_buffer_capacity (
    size_t size )
```

Ensure the buffer has at least the specified capacity.

#### Parameters

<i>size</i>	Requested buffer capacity.
-------------	----------------------------

#### Trick Job Class: *initialization*

Definition at line 111 of file OpaqueBuffer.cpp.

References alignment, buffer, capacity, and THLA\_ENDIAN.

Referenced by OpaqueBuffer(), push\_pad\_to\_buffer(), push\_to\_buffer(), SpaceFOM::QuaternionEncoder::QuaternionEncoder(), and SpaceFOM::SpaceTimeCoordinateEncoder::SpaceTimeCoordinateEncoder().

### 7.38.3.3 get\_byte\_alignment()

```
unsigned int TrickHLA::OpaqueBuffer::get_byte_alignment ( ) const [inline]
```

Get the current buffer byte alignment.

#### Returns

Buffer byte alignment.

Definition at line 73 of file OpaqueBuffer.hh.

References alignment.

#### 7.38.3.4 `get_capacity()`

```
size_t TrickHLA::OpaqueBuffer::get_capacity ( ) const [inline]
Get the current buffer capacity.
```

##### Returns

Buffer capacity.

Definition at line 77 of file OpaqueBuffer.hh.

References capacity.

Referenced by SpaceFOM::SpaceTimeCoordinateEncoder::encode(), and SpaceFOM::QuaternionEncoder::encode().

#### 7.38.3.5 `pull_from_buffer()` [1/2]

```
void TrickHLA::OpaqueBuffer::pull_from_buffer (
    void * dest,
    size_t size ) [inline]
```

Pull the specified number of data bytes from the buffer into the specified destination with no encoding.

##### Parameters

<code>dest</code>	Destination to pull data into from buffer.
<code>size</code>	Size of data in bytes.

Definition at line 115 of file OpaqueBuffer.hh.

References TrickHLA::ENCODING\_NO\_ENCODING.

#### 7.38.3.6 `pull_from_buffer()` [2/2]

```
void OpaqueBuffer::pull_from_buffer (
    void * dest,
    size_t size,
    EncodingEnum encoding )
```

Pull the specified number of data bytes from the buffer into the specified destination and for the specified encoding of the data in the buffer.

##### Parameters

<code>dest</code>	Destination to pull data into from buffer.
<code>size</code>	Size of data in bytes.
<code>encoding</code>	One of ENCODING_TYPE_LITTLE_ENDIAN, ENCODING_TYPE_BIG_ENDIAN, or ENCODING_TYPE_NO_ENCODING.

##### Assumptions and Limitations:

- The destination must be large enough to hold size bytes of data.

Definition at line 202 of file OpaqueBuffer.cpp.

References alignment, buffer, byteswap\_buffer\_copy(), capacity, TrickHLA::ENCODING\_BIG\_ENDIAN, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_NO\_ENCODING, pull\_pad\_from\_buffer(), pull\_pos, THLA\_ENDIAN, and THLA\_NEWLINE.

### 7.38.3.7 pull\_pad\_from\_buffer()

```
void OpaqueBuffer::pull_pad_from_buffer (
    size_t pad_size ) [protected]
```

Pull the specified number of pad bytes from the buffer.

#### Parameters

<i>pad_size</i>	Size of data in bytes.
-----------------	------------------------

Definition at line 278 of file OpaqueBuffer.cpp.

References capacity, pull\_pos, and THLA\_ENDIAN.

Referenced by pull\_from\_buffer().

### 7.38.3.8 push\_pad\_to\_buffer()

```
void OpaqueBuffer::push_pad_to_buffer (
    size_t pad_size ) [protected]
```

Push the specified number of pad bytes to the buffer.

#### Parameters

<i>pad_size</i>	Size of data in bytes.
-----------------	------------------------

Definition at line 252 of file OpaqueBuffer.cpp.

References buffer, capacity, ensure\_buffer\_capacity(), push\_pos, and THLA\_ENDIAN.

Referenced by push\_to\_buffer().

### 7.38.3.9 push\_to\_buffer() [1/2]

```
void TrickHLA::OpaqueBuffer::push_to_buffer (
    void * src,
    size_t size ) [inline]
```

Push the specified data into the buffer with no encoding.

#### Parameters

<i>src</i>	Source of data to push into buffer.
<i>size</i>	Size of data in bytes.

Definition at line 99 of file OpaqueBuffer.hh.

References TrickHLA::ENCODING\_NO\_ENCODING.

### 7.38.3.10 push\_to\_buffer() [2/2]

```
void OpaqueBuffer::push_to_buffer (
```

```
void * src,
size_t size,
EncodingEnum encoding )
```

Push the specified data into the buffer using the specified encoding.

#### Parameters

<i>src</i>	Source of data to push into buffer.
<i>size</i>	Size of data in bytes.
<i>encoding</i>	One of ENCODING_TYPE_LITTLE_ENDIAN, ENCODING_TYPE_BIG_ENDIAN, or ENCODING_TYPE_NO_ENCODING.

Definition at line 148 of file OpaqueBuffer.cpp.

References alignment, buffer, byteswap\_buffer\_copy(), capacity, TrickHLA::ENCODING\_BIG\_ENDIAN, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_NO\_ENCODING, ensure\_buffer\_capacity(), push\_pad\_to\_buffer(), push\_pos, THLA\_ENDIAN, and THLA\_NEWLINE.

#### 7.38.3.11 reset\_buffer\_positions()

```
void TrickHLA::OpaqueBuffer::reset_buffer_positions ( ) [inline]
```

Reset both the push and pull buffer positions.

Definition at line 90 of file OpaqueBuffer.hh.

References reset\_pull\_position(), and reset\_push\_position().

#### 7.38.3.12 reset\_pull\_position()

```
void TrickHLA::OpaqueBuffer::reset_pull_position ( ) [inline]
```

Reset the pull buffer position.

Definition at line 87 of file OpaqueBuffer.hh.

References pull\_pos.

Referenced by reset\_buffer\_positions().

#### 7.38.3.13 reset\_push\_position()

```
void TrickHLA::OpaqueBuffer::reset_push_position ( ) [inline]
```

Reset the push buffer position.

Definition at line 84 of file OpaqueBuffer.hh.

References push\_pos.

Referenced by reset\_buffer\_positions().

#### 7.38.3.14 set\_byte\_alignment()

```
void OpaqueBuffer::set_byte_alignment (
    unsigned int size )
```

Set the byte alignment to use for the data in the buffer.

#### Parameters

<i>size</i>	Byte alignment size.
-------------	----------------------

**Trick Job Class:** *initialization*

Definition at line 86 of file OpaqueBuffer.cpp.

References alignment, and THLA\_ENDIAN.

Referenced by SpaceFOM::QuaternionEncoder::QuaternionEncoder(), and SpaceFOM::SpaceTimeCoordinateEncoder::SpaceTimeCoordinateEncoder().

## 7.38.4 Friends And Related Function Documentation

### 7.38.4.1 init\_attrTrickHLA\_\_OpaqueBuffer

```
void init_attrTrickHLA__OpaqueBuffer ( ) [friend]
```

### 7.38.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 53 of file OpaqueBuffer.hh.

## 7.38.5 Field Documentation

### 7.38.5.1 alignment

```
unsigned int TrickHLA::OpaqueBuffer::alignment
```

**Units:** –

The byte alignment to use for the buffer.

Definition at line 147 of file OpaqueBuffer.hh.

Referenced by ensure\_buffer\_capacity(), get\_byte\_alignment(), pull\_from\_buffer(), push\_to\_buffer(), and set\_byte\_alignment().

### 7.38.5.2 buffer

```
unsigned char* TrickHLA::OpaqueBuffer::buffer
```

**Units:** –

Byte buffer.

Definition at line 153 of file OpaqueBuffer.hh.

Referenced by SpaceFOM::SpaceTimeCoordinateEncoder::decode(), SpaceFOM::QuaternionEncoder::decode(), SpaceFOM::SpaceTimeCoordinateEncoder::encode(), SpaceFOM::QuaternionEncoder::encode(), ensure\_buffer\_capacity(), pull\_from\_buffer(), push\_pad\_to\_buffer(), push\_to\_buffer(), and ~OpaqueBuffer().

### 7.38.5.3 capacity

```
size_t TrickHLA::OpaqueBuffer::capacity
```

**Units:** –

Capacity of the buffer.

Definition at line 151 of file OpaqueBuffer.hh.

Referenced by SpaceFOM::SpaceTimeCoordinateEncoder::decode(), SpaceFOM::QuaternionEncoder::decode(), ensure\_buffer\_capacity(), get\_capacity(), pull\_from\_buffer(), pull\_pad\_from\_buffer(), push\_pad\_to\_buffer(), push\_to\_buffer(), and ~OpaqueBuffer().

#### 7.38.5.4 pull\_pos

```
size_t TrickHLA::OpaqueBuffer::pull_pos
```

**Units:** –

Position to pull data from.

Definition at line 150 of file OpaqueBuffer.hh.

Referenced by pull\_from\_buffer(), pull\_pad\_from\_buffer(), reset\_pull\_position(), and ~OpaqueBuffer().

#### 7.38.5.5 push\_pos

```
size_t TrickHLA::OpaqueBuffer::push_pos
```

**Units:** –

Position to push data to.

Definition at line 149 of file OpaqueBuffer.hh.

Referenced by push\_pad\_to\_buffer(), push\_to\_buffer(), reset\_push\_position(), and ~OpaqueBuffer().

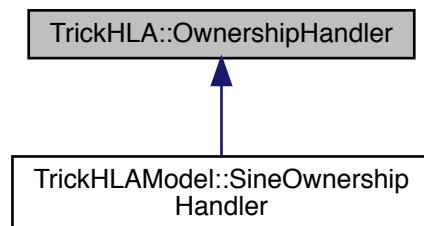
The documentation for this class was generated from the following files:

- [OpaqueBuffer.hh](#)
- [OpaqueBuffer.cpp](#)

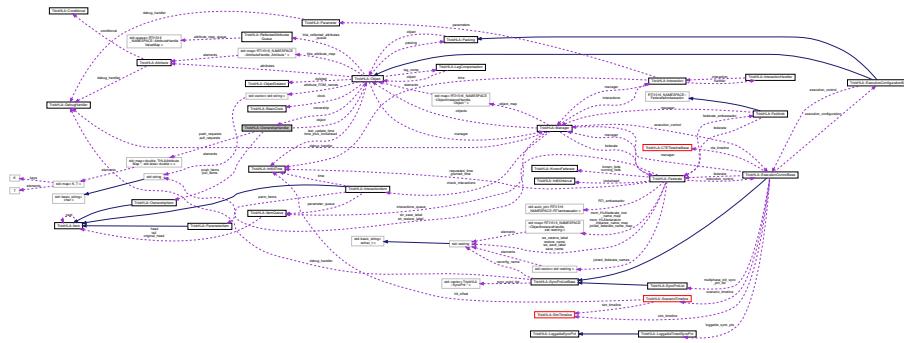
## 7.39 TrickHLA::OwnershipHandler Class Reference

```
#include <OwnershipHandler.hh>
```

Inheritance diagram for TrickHLA::OwnershipHandler:



## Collaboration diagram for TrickHLA::OwnershipHandler:



## Public Member Functions

- **OwnershipHandler ()**  
*Default constructor for the [TrickHLA Attribute](#) class.*
  - virtual **~OwnershipHandler ()**  
*Destructor for the [TrickHLA Attribute](#) class.*
  - void **setup\_checkpoint\_requests ()**  
*Encodes the push and pull attribute ownership maps into checkpoint-able queues.*
  - void **clear\_checkpoint ()**  
*Clears out the push / pull checkpoint-able queues.*
  - void **restore\_requests ()**  
*Decodes the push / pull checkpoint-able queues back into attribute ownership maps.*
  - virtual void **initialize\_callback (Object \*obj)**  
*Initializes the callback to the interaction.*
  - bool **should\_print (const DebugLevelEnum &level, const DebugSourceEnum &code) const**  
*Determine if the verbose debug comments should be printed to the console.*
  - std::string **get\_object\_name () const**  
*Returns the name of the object.*
  - std::string **get\_object\_FOM\_name () const**  
*Returns the FOM name of the object.*
  - int **get\_attribute\_count () const**  
*Returns the number of attributes the object contains.*
  - **VectorOfStrings get\_attribute\_FOM\_names () const**  
*Returns the attribute FOM names of the object.*
  - bool **is\_locally\_owned (const char \*attribute\_FOM\_name)**  
*Query if the attribute is locally owned.*
  - bool **is\_remotely\_owned (const char \*attribute\_FOM\_name)**  
*Query if the attribute is remotely owned.*
  - bool **is\_published (const char \*attribute\_FOM\_name)**  
*Query if the attribute is published.*
  - bool **is\_subscribed (const char \*attribute\_FOM\_name)**  
*Query if the attribute is subscribed.*
  - void **pull\_ownership ()**  
*Pull ownership of all object attributes as soon as possible.*

- void `pull_ownership` (double time)  
*Pull ownership of all object attributes at the specified time.*
- void `pull_ownership` (const char \*attribute\_FOM\_name)  
*Pull ownership of the specified attribute as soon as possible.*
- void `pull_ownership` (const char \*attribute\_FOM\_name, double time)  
*Pull ownership of the specified attribute at the given time.*
- void `push_ownership` ()  
*Push ownership of all the object attributes as soon as possible.*
- void `push_ownership` (double time)  
*Push ownership of all the object attributes at the specified time.*
- void `push_ownership` (const char \*attribute\_FOM\_name)  
*Push ownership of the specified attribute as soon as possible.*
- void `push_ownership` (const char \*attribute\_FOM\_name, double time)  
*Push ownership of the specified attribute at the given time.*
- `Int64Interval get_fed_lookahead` () const  
*Return a copy of the object's lookahead time.*
- `Int64Time get_granted_fed_time` () const  
*Return a copy of the granted HLA logical time.*
- double `get_scenario_time` ()  
*Get the current scenario time.*
- double `get_cte_time` ()  
*Get the current Central Timing Equipment (CTE) time.*

## Protected Member Functions

- `Attribute * get_attribute` (const char \*attribute\_FOM\_name)  
*Returns the attribute for the given attribute FOM name or NULL if an attribute corresponding to the FOM name is not found.*

## Protected Attributes

- `Object * object`  
**Data I/O:** \*\*  
*Reference to the TrickHLA Object.*
- `AttributeOwnershipMap pull_requests`  
**Data I/O:** \*\*  
*Map of pull ownership user requests.*
- `AttributeOwnershipMap push_requests`  
**Data I/O:** \*\*  
*Map of push ownership user requests.*
- `size_t pull_items_cnt`  
**Units:** count  
*Number of pull items*
- `OwnershipItem * pull_items`  
**Units:** –  
*Array of pulled attributes*
- `size_t push_items_cnt`  
**Units:** count  
*Number of push items*
- `OwnershipItem * push_items`  
**Units:** –  
*Array of pushed attributes*

## Private Member Functions

- [OwnershipHandler \(const OwnershipHandler &rhs\)](#)  
*Copy constructor for [OwnershipHandler](#) class.*
- [OwnershipHandler & operator= \(const OwnershipHandler &rhs\)](#)  
*Assignment operator for [OwnershipHandler](#) class.*

## Friends

- class [InputProcessor](#)
- class [Object](#)
- void [init\\_attrTrickHLA\\_OwnershipHandler \(\)](#)

### 7.39.1 Detailed Description

Definition at line 68 of file [OwnershipHandler.hh](#).

### 7.39.2 Constructor & Destructor Documentation

#### 7.39.2.1 [OwnershipHandler\(\) \[1/2\]](#)

`OwnershipHandler::OwnershipHandler ( )`  
Default constructor for the [TrickHLA Attribute](#) class.  
**Trick Job Class:** *initialization*  
Definition at line 62 of file [OwnershipHandler.cpp](#).

#### 7.39.2.2 [~OwnershipHandler\(\)](#)

`OwnershipHandler::~OwnershipHandler ( ) [virtual]`  
Destructor for the [TrickHLA Attribute](#) class.  
**Trick Job Class:** *shutdown*  
Definition at line 76 of file [OwnershipHandler.cpp](#).  
References [clear\\_checkpoint\(\)](#).

#### 7.39.2.3 [OwnershipHandler\(\) \[2/2\]](#)

`TrickHLA::OwnershipHandler::OwnershipHandler (`  
    `const OwnershipHandler & rhs ) [private]`  
Copy constructor for [OwnershipHandler](#) class.  
This constructor is private to prevent inadvertent copies.

### 7.39.3 Member Function Documentation

#### 7.39.3.1 [clear\\_checkpoint\(\)](#)

`void OwnershipHandler::clear_checkpoint ( )`  
Clears out the push / pull checkpoint-able queues.  
Definition at line 176 of file [OwnershipHandler.cpp](#).

References TrickHLA::OwnershipItem::clear(), pull\_items, pull\_items\_cnt, push\_items, and push\_items\_cnt.  
 Referenced by TrickHLA::Object::release\_ownership(), setup\_checkpoint\_requests(), and ~OwnershipHandler().

### 7.39.3.2 get\_attribute()

```
Attribute * OwnershipHandler::get_attribute (
    const char * attribute_FOM_name ) [protected]
```

Returns the attribute for the given attribute FOM name or NULL if an attribute corresponding to the FOM name is not found.

#### Returns

Attribute of the object.

#### Parameters

attribute_FOM_name	Attribute FOM name.
--------------------	---------------------

Definition at line 305 of file OwnershipHandler.cpp.

References TrickHLA::Object::get\_attribute(), and object.

Referenced by is\_locally\_owned(), is\_published(), is\_remotely\_owned(), is\_subscribed(), pull\_ownership(), push\_ownership(), and restore\_requests().

### 7.39.3.3 get\_attribute\_count()

```
int OwnershipHandler::get_attribute_count ( ) const
```

Returns the number of attributes the object contains.

#### Returns

Number of attributes.

Definition at line 295 of file OwnershipHandler.cpp.

References TrickHLA::Object::get\_attribute\_count(), and object.

### 7.39.3.4 get\_attribute\_FOM\_names()

```
VectorOfStrings OwnershipHandler::get_attribute_FOM_names ( ) const
```

Returns the attribute FOM names of the object.

#### Returns

Vector of attribute FOM name strings.

Definition at line 300 of file OwnershipHandler.cpp.

References TrickHLA::Object::get\_attribute\_FOM\_names(), and object.

### 7.39.3.5 get\_cte\_time()

```
double OwnershipHandler::get_cte_time ( )
```

Get the current Central Timing Equipment (CTE) time.

**Returns**

Returns the current CTE time.

Definition at line 586 of file OwnershipHandler.cpp.

References TrickHLA::ExecutionControlBase::does\_cte\_timeline\_exist(), TrickHLA::ExecutionControlBase::get\_cte\_time(), TrickHLA::Federate::get\_execution\_control(), TrickHLA::Object::get\_federate(), TrickHLA::ExecutionControlBase::get\_federate(), and object.

### 7.39.3.6 get\_fed\_lookahead()

```
Int64Interval OwnershipHandler::get_fed_lookahead ( ) const
```

Return a copy of the object's lookahead time.

**Returns**

A copy of the federate's lookahead time

If the manager does not exist, -1.0 seconds is assigned to the returned object.

Definition at line 552 of file OwnershipHandler.cpp.

Referenced by pull\_ownership(), and push\_ownership().

### 7.39.3.7 get\_granted\_fed\_time()

```
Int64Time OwnershipHandler::get_granted_fed_time ( ) const
```

Return a copy of the granted HLA logical time.

**Returns**

A copy of the federation granted time.

If the object does not exist, MAX\_LOGICAL\_TIME\_SECONDS is assigned to the returned object.

Definition at line 566 of file OwnershipHandler.cpp.

References TrickHLA::MAX\_LOGICAL\_TIME\_SECONDS.

Referenced by pull\_ownership(), and push\_ownership().

### 7.39.3.8 get\_object\_FOM\_name()

```
string OwnershipHandler::get_object_FOM_name ( ) const
```

Returns the FOM name of the object.

**Returns**

FOM name of the object.

Definition at line 290 of file OwnershipHandler.cpp.

References TrickHLA::Object::get\_FOM\_name(), and object.

### 7.39.3.9 get\_object\_name()

```
string OwnershipHandler::get_object_name ( ) const
```

Returns the name of the object.

**Returns**

Name of the object.

Definition at line 285 of file OwnershipHandler.cpp.

References TrickHLA::Object::get\_name(), and object.

### 7.39.3.10 get\_scenario\_time()

```
double OwnershipHandler::get_scenario_time ( )
```

Get the current scenario time.

#### Returns

Returns the current scenario time in seconds.

Definition at line 577 of file OwnershipHandler.cpp.

References `TrickHLA::Federate::get_execution_control()`, `TrickHLA::Object::get_federate()`, `TrickHLA::ExecutionControlBase::get_federate()`, `TrickHLA::ExecutionControlBase::get_scenario_time()`, and `object`.

Referenced by `pull_ownership()`, and `push_ownership()`.

### 7.39.3.11 initialize\_callback()

```
void OwnershipHandler::initialize_callback (
    Object * obj ) [virtual]
```

Initializes the callback to the interaction.

#### Parameters

<i>obj</i>	Associated object for this class.
------------	-----------------------------------

Reimplemented in [TrickHLAModel::SineOwnershipHandler](#).

Definition at line 269 of file OwnershipHandler.cpp.

Referenced by `TrickHLA::Object::initialize()`.

### 7.39.3.12 is\_locally\_owned()

```
bool OwnershipHandler::is_locally_owned (
    const char * attribute_FOM_name )
```

Query if the attribute is locally owned.

#### Returns

True if attribute is locally owned; False otherwise.

#### Parameters

<i>attribute_FOM_name</i>	Attribute FOM name.
---------------------------	---------------------

Definition at line 311 of file OwnershipHandler.cpp.

References `get_attribute()`, and `TrickHLA::Attribute::is_locally_owned()`.

### 7.39.3.13 is\_published()

```
bool OwnershipHandler::is_published (
    const char * attribute_FOM_name )
```

Query if the attribute is published.

**Returns**

True if attribute is published; False otherwise.

**Parameters**

<i>attribute_FOM_name</i>	Attribute FOM name.
---------------------------	---------------------

Definition at line 325 of file OwnershipHandler.cpp.

References `get_attribute()`, and `TrickHLA::Attribute::is_publish()`.

**7.39.3.14 `is_remotely_owned()`**

```
bool OwnershipHandler::is_remotely_owned (
    const char * attribute_FOM_name )
```

Query if the attribute is remotely owned.

**Returns**

True if attribute is remotely owned; False otherwise.

**Parameters**

<i>attribute_FOM_name</i>	Attribute FOM name.
---------------------------	---------------------

Definition at line 318 of file OwnershipHandler.cpp.

References `get_attribute()`, and `TrickHLA::Attribute::is_remotely_owned()`.

**7.39.3.15 `is_subscribed()`**

```
bool OwnershipHandler::is_subscribed (
    const char * attribute_FOM_name )
```

Query if the attribute is subscribed.

**Returns**

True if attribute is subscribed; False otherwise.

**Parameters**

<i>attribute_FOM_name</i>	Attribute FOM name.
---------------------------	---------------------

Definition at line 332 of file OwnershipHandler.cpp.

References `get_attribute()`, and `TrickHLA::Attribute::is_subscribe()`.

**7.39.3.16 `operator=()`**

```
OwnershipHandler& TrickHLA::OwnershipHandler::operator= (
    const OwnershipHandler & rhs ) [private]
```

Assignment operator for `OwnershipHandler` class.

This assignment operator is private to prevent inadvertent copies.

### 7.39.3.17 `pull_ownership()` [1/4]

```
void OwnershipHandler::pull_ownership ( )
```

Pull ownership of all object attributes as soon as possible.  
 Definition at line 339 of file OwnershipHandler.cpp.  
 Referenced by `pull_ownership()`.

### 7.39.3.18 `pull_ownership()` [2/4]

```
void OwnershipHandler::pull_ownership (
    const char * attribute_FOM_name )
```

Pull ownership of the specified attribute as soon as possible.

#### Parameters

<code>attribute_FOM_name</code>	Attribute FOM name.
---------------------------------	---------------------

Definition at line 392 of file OwnershipHandler.cpp.  
 References `pull_ownership()`.

### 7.39.3.19 `pull_ownership()` [3/4]

```
void OwnershipHandler::pull_ownership (
    const char * attribute_FOM_name,
    double time )
```

Pull ownership of the specified attribute at the given time.

#### Parameters

<code>attribute_FOM_name</code>	Attribute FOM name.
<code>time</code>	Requested time to pull ownership.

Definition at line 398 of file OwnershipHandler.cpp.

References `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_OWNERSHIP`, `get_attribute()`, `get_fed_lookahead()`, `TrickHLA::Attribute::get_FOM_name()`, `get_granted_fed_time()`, `get_scenario_time()`, `pull_requests`, `should_print()`, and `THLA_NEWLINE`.

### 7.39.3.20 `pull_ownership()` [4/4]

```
void OwnershipHandler::pull_ownership (
    double time )
```

Pull ownership of all object attributes at the specified time.

#### Parameters

<code>time</code>	Requested time to pull ownership.
-------------------	-----------------------------------

Definition at line 344 of file OwnershipHandler.cpp.

References `TrickHLA::DEBUG_LEVEL_3_TRACE`, `TrickHLA::DEBUG_SOURCE_OWNERSHIP`, `get_fed_lookahead()`, `get_granted_fed_time()`, `get_scenario_time()`, `pull_requests`, `should_print()`, and `THLA_NEWLINE`.

**7.39.3.21 push\_ownership() [1/4]**

```
void OwnershipHandler::push_ownership ( )
```

Push ownership of all the object attributes as soon as possible.  
 Definition at line 444 of file OwnershipHandler.cpp.  
 Referenced by push\_ownership().

**7.39.3.22 push\_ownership() [2/4]**

```
void OwnershipHandler::push_ownership (
    const char * attribute_FOM_name )
```

Push ownership of the specified attribute as soon as possible.

**Parameters**

<i>attribute_FOM_name</i>	Attribute FOM name.
---------------------------	---------------------

Definition at line 497 of file OwnershipHandler.cpp.  
 References push\_ownership().

**7.39.3.23 push\_ownership() [3/4]**

```
void OwnershipHandler::push_ownership (
    const char * attribute_FOM_name,
    double time )
```

Push ownership of the specified attribute at the given time.

**Parameters**

<i>attribute_FOM_name</i>	Attribute FOM name.
<i>time</i>	Requested time to push ownership.

Definition at line 503 of file OwnershipHandler.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OWNERSHIP, get\_attribute(), get\_fed\_lookahead(), TrickHLA::Attribute::get\_FOM\_name(), get\_granted\_fed\_time(), get\_scenario\_time(), push\_requests, should\_print(), and THLA\_NEWLINE.

**7.39.3.24 push\_ownership() [4/4]**

```
void OwnershipHandler::push_ownership (
    double time )
```

Push ownership of all the object attributes at the specified time.

**Parameters**

<i>time</i>	Requested time to push ownership.
-------------	-----------------------------------

Definition at line 449 of file OwnershipHandler.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OWNERSHIP, get\_fed\_lookahead(), get\_granted\_fed\_time(), get\_scenario\_time(), push\_requests, should\_print(), and THLA\_NEWLINE.

### 7.39.3.25 `restore_requests()`

```
void OwnershipHandler::restore_requests ( )
```

Decodes the push / pull checkpoint-able queues back into attribute ownership maps.

Definition at line 199 of file OwnershipHandler.cpp.

References TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OWNERSHIP, get\_attribute(), pull\_items, pull\_items\_cnt, pull\_requests, push\_items, push\_items\_cnt, push\_requests, should\_print(), THLA\_NEWLINE, and TrickHLA::OwnershipItem::time.

Referenced by TrickHLA::Object::restore\_ownership\_transfer\_checkpointed\_data().

### 7.39.3.26 `setup_checkpoint_requests()`

```
void OwnershipHandler::setup_checkpoint_requests ( )
```

Encodes the push and pull attribute ownership maps into checkpoint-able queues.

Definition at line 81 of file OwnershipHandler.cpp.

References clear\_checkpoint(), TrickHLA::DEBUG\_LEVEL\_3\_TRACE, TrickHLA::DEBUG\_SOURCE\_OWNERSHIP, TrickHLA::OwnershipItem::FOM\_name, pull\_items, pull\_items\_cnt, pull\_requests, push\_items, push\_items\_cnt, push\_requests, should\_print(), THLA\_NEWLINE, and TrickHLA::OwnershipItem::time.

Referenced by TrickHLA::Object::setup\_ownership\_transfer\_checkpointed\_data().

### 7.39.3.27 `should_print()`

```
bool OwnershipHandler::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const
```

Determine if the verbose debug comments should be printed to the console.

Returns

Returns true if the requested message should print level.

Parameters

<i>level</i>	Debug level of the incoming message.
<i>code</i>	Source code association of the incoming messages.

Definition at line 275 of file OwnershipHandler.cpp.

Referenced by pull\_ownership(), push\_ownership(), restore\_requests(), and setup\_checkpoint\_requests().

## 7.39.4 Friends And Related Function Documentation

### 7.39.4.1 `init_attrTrickHLA_OwnershipHandler`

```
void init_attrTrickHLA_OwnershipHandler ( ) [friend]
```

### 7.39.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 75 of file OwnershipHandler.hh.

### 7.39.4.3 Object

```
friend class Object [friend]
Definition at line 81 of file OwnershipHandler.hh.
```

## 7.39.5 Field Documentation

### 7.39.5.1 object

```
Object* TrickHLA::OwnershipHandler::object [protected]
```

#### Data I/O: \*\*

Reference to the [TrickHLA Object](#).

Definition at line 207 of file OwnershipHandler.hh.

Referenced by `get_attribute()`, `get_attribute_count()`, `get_attribute_FOM_names()`, `get_cte_time()`, `get_object_FOM_name()`, `get_object_name()`, and `get_scenario_time()`.

### 7.39.5.2 pull\_items

```
OwnershipItem* TrickHLA::OwnershipHandler::pull_items [protected]
```

#### Units: –

Array of pulled attributes

Definition at line 213 of file OwnershipHandler.hh.

Referenced by `clear_checkpoint()`, `restore_requests()`, and `setup_checkpoint_requests()`.

### 7.39.5.3 pull\_items\_cnt

```
size_t TrickHLA::OwnershipHandler::pull_items_cnt [protected]
```

#### Units: count

Number of pull items

Definition at line 212 of file OwnershipHandler.hh.

Referenced by `clear_checkpoint()`, `restore_requests()`, and `setup_checkpoint_requests()`.

### 7.39.5.4 pull\_requests

```
AttributeOwnershipMap TrickHLA::OwnershipHandler::pull_requests [protected]
```

#### Data I/O: \*\*

Map of pull ownership user requests.

Definition at line 209 of file OwnershipHandler.hh.

Referenced by `pull_ownership()`, `TrickHLA::Object::pull_ownership()`, `restore_requests()`, and `setup_checkpoint_requests()`.

### 7.39.5.5 push\_items

```
OwnershipItem* TrickHLA::OwnershipHandler::push_items [protected]
```

#### Units: –

Array of pushed attributes

Definition at line 215 of file OwnershipHandler.hh.

Referenced by `clear_checkpoint()`, `restore_requests()`, and `setup_checkpoint_requests()`.

### 7.39.5.6 push\_items\_cnt

```
size_t TrickHLA::OwnershipHandler::push_items_cnt [protected]
```

**Units:** *count*

Number of push items

Definition at line 214 of file OwnershipHandler.hh.

Referenced by `clear_checkpoint()`, `restore_requests()`, and `setup_checkpoint_requests()`.

### 7.39.5.7 push\_requests

```
AttributeOwnershipMap TrickHLA::OwnershipHandler::push_requests [protected]
```

**Data I/O:** \*\*

Map of push ownership user requests.

Definition at line 210 of file OwnershipHandler.hh.

Referenced by `push_ownership()`, `TrickHLA::Object::push_ownership()`, `restore_requests()`, and `setup_checkpoint_requests()`.

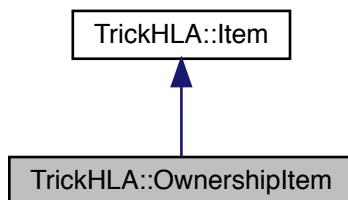
The documentation for this class was generated from the following files:

- [OwnershipHandler.hh](#)
- [OwnershipHandler.cpp](#)

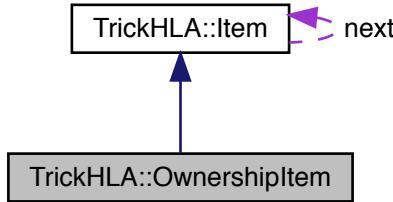
## 7.40 TrickHLA::OwnershipItem Class Reference

```
#include <OwnershipItem.hh>
```

Inheritance diagram for TrickHLA::OwnershipItem:



Collaboration diagram for TrickHLA::OwnershipItem:



## Public Member Functions

- [OwnershipItem \(\)](#)  
*Default constructor for the `TrickHLA OwnershipItem` class.*
- [virtual ~OwnershipItem \(\)](#)  
*Destructor for the `TrickHLA OwnershipItem` class.*
- [void clear \(\)](#)  
*Clear the Trick allocated memory for the FOM name.*

## Data Fields

- [double time](#)  
**Units:** –  
*Federation time when this attribute's ownership is to be transferred.*
- [char \\* FOM\\_name](#)  
**Units:** –  
*FOM name for the attribute.*

## Private Member Functions

- [OwnershipItem \(const OwnershipItem &rhs\)](#)  
*Copy constructor for `OwnershipItem` class.*
- [OwnershipItem & operator= \(const OwnershipItem &rhs\)](#)  
*Assignment operator for `OwnershipItem` class.*

## Friends

- class [InputProcessor](#)
- [void init\\_attrTrickHLA\\_OwnershipItem \(\)](#)

### 7.40.1 Detailed Description

Definition at line 52 of file `OwnershipItem.hh`.

## 7.40.2 Constructor & Destructor Documentation

### 7.40.2.1 OwnershipItem() [1/2]

```
TrickHLA::OwnershipItem::OwnershipItem ( ) [inline]
```

Default constructor for the [TrickHLA OwnershipItem](#) class.

Definition at line 73 of file OwnershipItem.hh.

### 7.40.2.2 ~OwnershipItem()

```
virtual TrickHLA::OwnershipItem::~OwnershipItem ( ) [inline], [virtual]
```

Destructor for the [TrickHLA OwnershipItem](#) class.

Definition at line 75 of file OwnershipItem.hh.

References [clear\(\)](#).

### 7.40.2.3 OwnershipItem() [2/2]

```
TrickHLA::OwnershipItem::OwnershipItem (   
     const OwnershipItem & rhs ) [private]
```

Copy constructor for [OwnershipItem](#) class.

This constructor is private to prevent inadvertent copies.

## 7.40.3 Member Function Documentation

### 7.40.3.1 clear()

```
void TrickHLA::OwnershipItem::clear ( ) [inline]
```

Clear the Trick allocated memory for the FOM name.

Definition at line 78 of file OwnershipItem.hh.

References [FOM\\_name](#).

Referenced by [TrickHLA::OwnershipHandler::clear\\_checkpoint\(\)](#), and [~OwnershipItem\(\)](#).

### 7.40.3.2 operator=()

```
OwnershipItem& TrickHLA::OwnershipItem::operator= (   
     const OwnershipItem & rhs ) [private]
```

Assignment operator for [OwnershipItem](#) class.

This assignment operator is private to prevent inadvertent copies.

## 7.40.4 Friends And Related Function Documentation

### 7.40.4.1 init\_attrTrickHLA\_OwnershipItem

```
void init_attrTrickHLA_OwnershipItem ( ) [friend]
```

#### 7.40.4.2 InputProcessor

```
friend class InputProcessor [friend]
Definition at line 59 of file OwnershipItem.hh.
```

### 7.40.5 Field Documentation

#### 7.40.5.1 FOM\_name

```
char* TrickHLA::OwnershipItem::FOM_name
```

**Units:** –

FOM name for the attribute.

Definition at line 66 of file OwnershipItem.hh.

Referenced by clear(), and TrickHLA::OwnershipHandler::setup\_checkpoint\_requests().

#### 7.40.5.2 time

```
double TrickHLA::OwnershipItem::time
```

**Units:** –

Federation time when this attribute's ownership is to be transferred.

Definition at line 65 of file OwnershipItem.hh.

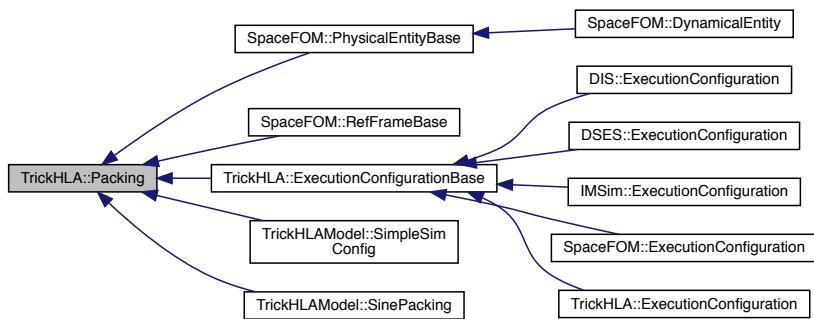
Referenced by TrickHLA::OwnershipHandler::restore\_requests(), and TrickHLA::OwnershipHandler::setup\_checkpoint\_requests().

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

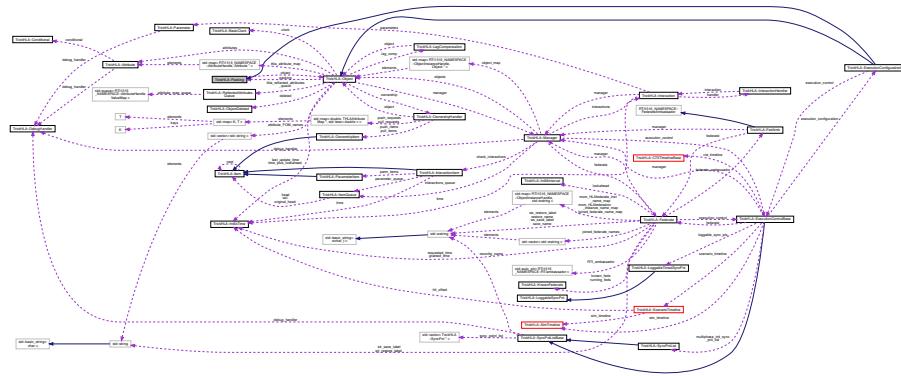
- [OwnershipItem.hh](#)

## 7.41 TrickHLA::Packing Class Reference

```
#include <Packing.hh>
Inheritance diagram for TrickHLA::Packing:
```



Collaboration diagram for TrickHLA::Packing:



## Public Member Functions

- **Packing ()**  
*Default constructor for the [TrickHLA Packing](#) class.*
- **virtual ~Packing ()**  
*Destructor for the [TrickHLA Packing](#) class.*
- **bool should\_print (const DebugLevelEnum &level, const DebugSourceEnum &code) const**  
*Determine if the verbose debug comments should be printed to the console.*
- **Attribute \* get\_attribute (const char \*attr\_FOM\_name)**  
*Get the [Attribute](#) by FOM name.*
- **Attribute \* get\_attribute\_and\_validate (const char \*attr\_FOM\_name)**  
*This function returns the [Attribute](#) for the given attribute FOM name.*
- **double get\_scenario\_time ()**  
*Get the current scenario time.*
- **double get\_cte\_time ()**  
*Get the current Central Timing Equipment (CTE) time.*
- **virtual void initialize\_callback (Object \*obj)**  
*Initialize the callback object to the supplied [Object](#) pointer.*
- **virtual void pack ()=0**  
*Pack the data before being sent.*
- **virtual void unpack ()=0**  
*Unpack the received data. The default.*

## Protected Attributes

- **Object \* object**  
**Data I/O: \*\***  
*Object associated with this packing class.*

## Private Member Functions

- **Packing (const Packing &rhs)**  
*Copy constructor for [Packing](#) class.*
- **Packing & operator= (const Packing &rhs)**  
*Assignment operator for [Packing](#) class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_Packing \(\)](#)

### 7.41.1 Detailed Description

Definition at line 53 of file [Packing.hh](#).

### 7.41.2 Constructor & Destructor Documentation

#### 7.41.2.1 [Packing\(\) \[1/2\]](#)

[Packing::Packing \(\)](#)

Default constructor for the [TrickHLA Packing](#) class.

**Trick Job Class:** *initialization*

Definition at line 56 of file [Packing.cpp](#).

#### 7.41.2.2 [~Packing\(\)](#)

[Packing::~Packing \(\) \[virtual\]](#)

Destructor for the [TrickHLA Packing](#) class.

**Trick Job Class:** *shutdown*

Definition at line 64 of file [Packing.cpp](#).

#### 7.41.2.3 [Packing\(\) \[2/2\]](#)

[TrickHLA::Packing::Packing \(](#)  
    [const Packing & rhs \) \[private\]](#)

Copy constructor for [Packing](#) class.

This constructor is private to prevent inadvertent copies.

### 7.41.3 Member Function Documentation

#### 7.41.3.1 [get\\_attribute\(\)](#)

```
Attribute * Packing::get\_attribute \(  
          const char \* attr\_FOM\_name \)
```

Get the [Attribute](#) by FOM name.

**Returns**

[Attribute](#) for the given name.

**Parameters**

<a href="#">attr_FOM_name</a>	<a href="#">Attribute</a> FOM name.
-------------------------------	-------------------------------------

Definition at line 84 of file [Packing.cpp](#).

Referenced by `get_attribute_and_validate()`, `TrickHLAModel::SinePacking::pack()`, and `TrickHLAModel::SinePacking::unpack()`.

#### 7.41.3.2 `get_attribute_and_validate()`

```
Attribute * Packing::get_attribute_and_validate (
    const char * attr_FOM_name )
```

This function returns the `Attribute` for the given attribute FOM name.

##### Returns

`Attribute` for the given name.

##### Parameters

<code>attr_FOM_name</code>	<code>Attribute</code> FOM name.
----------------------------	----------------------------------

If the attribute is not found then an error message is displayed then `exec-terminate` is called.

Definition at line 94 of file `Packing.cpp`.

References `get_attribute()`, and `THLA_ENDL`.

Referenced by `TrickHLAModel::SinePacking::initialize_callback()`, and `SpaceFOM::RefFrameBase::initialize_callback()`.

#### 7.41.3.3 `get_cte_time()`

```
double Packing::get_cte_time ( )
```

Get the current Central Timing Equipment (CTE) time.

##### Returns

Returns the current CTE time.

Definition at line 135 of file `Packing.cpp`.

References `TrickHLA::ExecutionControlBase::does_cte_timeline_exist()`, `TrickHLA::ExecutionControlBase::get_cte_time()`, `TrickHLA::Federate::get_execution_control()`, `TrickHLA::Object::get_federate()`, `TrickHLA::ExecutionControlBase::get_federate()`, and `object`.

#### 7.41.3.4 `get_scenario_time()`

```
double Packing::get_scenario_time ( )
```

Get the current scenario time.

##### Returns

Returns the current scenario time.

Definition at line 126 of file `Packing.cpp`.

References `TrickHLA::Federate::get_execution_control()`, `TrickHLA::Object::get_federate()`, `TrickHLA::ExecutionControlBase::get_federate()`, `TrickHLA::ExecutionControlBase::get_scenario_time()`, and `object`.

Referenced by `SpaceFOM::RefFrameBase::pack()`.

#### 7.41.3.5 `initialize_callback()`

```
void Packing::initialize_callback (
    Object * obj ) [virtual]
```

Initialize the callback object to the supplied [Object](#) pointer.

**Parameters**

<i>obj</i>	Associated object for this class.
------------	-----------------------------------

Reimplemented in [SpaceFOM::RefFrameBase](#), and [TrickHLAModel::SinePacking](#).

Definition at line 68 of file [Packing.cpp](#).

Referenced by [TrickHLA::Object::initialize\(\)](#), and [SpaceFOM::RefFrameBase::initialize\\_callback\(\)](#).

**7.41.3.6 operator=()**

```
Packing& TrickHLA::Packing::operator= (
    const Packing & rhs ) [private]
```

Assignment operator for [Packing](#) class.

This assignment operator is private to prevent inadvertent copies.

**7.41.3.7 pack()**

```
virtual void TrickHLA::Packing::pack ( ) [pure virtual]
```

Pack the data before being sent.

Implemented in [TrickHLA::ExecutionConfigurationBase](#), [SpaceFOM::RefFrameBase](#), [SpaceFOM::ExecutionConfiguration](#), [IMSim::ExecutionConfiguration](#), [DIS::ExecutionConfiguration](#), [DSES::ExecutionConfiguration](#), [TrickHLAModel::SinePacking](#), [SpaceFOM::PhysicalEntityBase](#), [TrickHLA::ExecutionConfiguration](#), [TrickHLAModel::SimpleSimConfig](#), and [SpaceFOM::DynamicalEntity](#).

Referenced by [TrickHLA::Object::send\\_cyclic\\_data\(\)](#), [TrickHLA::Object::send\\_init\\_data\(\)](#), and [TrickHLA::Object::send\\_requested\\_data\(\)](#).

**7.41.3.8 should\_print()**

```
bool Packing::should_print (
    const DebugLevelEnum & level,
    const DebugSourceEnum & code ) const
```

Determine if the verbose debug comments should be printed to the console.

**Returns**

Returns true if the requested message should print level.

**Parameters**

<i>level</i>	Debug level of the incoming message.
<i>code</i>	Source code association of the incoming messages.

Definition at line 74 of file [Packing.cpp](#).

Referenced by [TrickHLAModel::SimpleSimConfig::pack\(\)](#), [TrickHLA::ExecutionConfiguration::pack\(\)](#), [TrickHLAModel::SinePacking::pack\(\)](#), [DIS::ExecutionConfiguration::pack\(\)](#), [DSES::ExecutionConfiguration::pack\(\)](#), [IMSim::ExecutionConfiguration::pack\(\)](#), [SpaceFOM::ExecutionConfiguration::pack\(\)](#), [TrickHLA::ExecutionConfiguration::print\\_execution\\_configuration\(\)](#), [TrickHLAModel::SimpleSimConfig::unpack\(\)](#), [TrickHLA::ExecutionConfiguration::unpack\(\)](#), [TrickHLAModel::SinePacking::unpack\(\)](#), [DSES::ExecutionConfiguration::unpack\(\)](#), [DIS::ExecutionConfiguration::unpack\(\)](#), [IMSim::ExecutionConfiguration::unpack\(\)](#), and [SpaceFOM::ExecutionConfiguration::unpack\(\)](#).

### 7.41.3.9 `unpack()`

```
virtual void TrickHLA::Packing::unpack ( ) [pure virtual]
```

Unpack the received data. The default.

Implemented in [TrickHLA::ExecutionConfigurationBase](#), [SpaceFOM::RefFrameBase](#), [SpaceFOM::ExecutionConfiguration](#), [IMSim::ExecutionConfiguration](#), [DIS::ExecutionConfiguration](#), [DSES::ExecutionConfiguration](#), [TrickHLAModel::SinePacking](#), [SpaceFOM::PhysicalEntityBase](#), [TrickHLA::ExecutionConfiguration](#), [TrickHLAModel::SimpleSimConfig](#), and [SpaceFOM::DynamicalEntity](#). Referenced by [TrickHLA::Object::receive\\_cyclic\\_data\(\)](#), and [TrickHLA::Object::receive\\_init\\_data\(\)](#).

## 7.41.4 Friends And Related Function Documentation

### 7.41.4.1 `init_attrTrickHLA__Packing`

```
void init_attrTrickHLA__Packing ( ) [friend]
```

### 7.41.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 60 of file `Packing.hh`.

## 7.41.5 Field Documentation

### 7.41.5.1 `object`

```
Object* TrickHLA::Packing::object [protected]
```

**Data I/O: \*\***

[Object](#) associated with this packing class.

Definition at line 113 of file `Packing.hh`.

Referenced by [SpaceFOM::RefFrameBase::default\\_data\(\)](#), [get\\_cte\\_time\(\)](#), [SpaceFOM::RefFrameBase::get\\_object\(\)](#), [get\\_scenario\\_time\(\)](#), [TrickHLAModel::SinePacking::pack\(\)](#), and [TrickHLAModel::SinePacking::unpack\(\)](#).

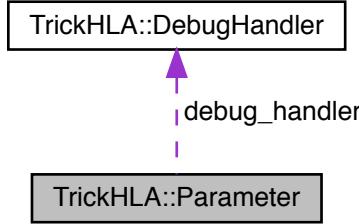
The documentation for this class was generated from the following files:

- [Packing.hh](#)
- [Packing.cpp](#)

## 7.42 TrickHLA::Parameter Class Reference

```
#include <Parameter.hh>
```

Collaboration diagram for TrickHLA::Parameter:



## Public Member Functions

- **Parameter ()**  
*Default constructor for the [TrickHLA Parameter](#) class.*
- **virtual ~Parameter ()**  
*Destructor for the [TrickHLA Parameter](#) class.*
- **void initialize (const char \*interaction\_fom\_name, const int interaction\_index, const int parameter\_index)**  
*Initializes the [TrickHLA Parameter](#) from the trick variable name supplied in the input file.*
- **void initialize (const char \*interaction\_fom\_name, void \*in\_addr, ATTRIBUTES \*in\_attr)**  
*Initializes the [TrickHLA Parameter](#) from the supplied address and ATTRIBUTES of the trick variable.*
- **void complete\_initialization ()**  
*Initializes the [TrickHLA Parameter](#).*
- **void set\_debug\_level (DebugHandler hndlr)**  
*Set the debug handler for this parameter.*
- **const char \* get\_FOM\_name () const**  
*Get the FOM name for this parameter.*
- **void set\_FOM\_name (const char \*in\_name)**  
*Set the FOM name for the parameter.*
- **const char \* get\_trick\_name () const**  
*Get the Trick variable name associated with this parameter.*
- **void set\_encoding (EncodingEnum in\_type)**  
*Set the RTI encoding and based on the new encoding determine if we need to byte-swap.*
- **size\_t calculate\_number\_of\_items ()**  
*Calculate the number of attribute items associated with this parameter.*
- **bool is\_static\_in\_size () const**  
*Determines if the parameter is static in size.*
- **bool is\_byteswap () const**  
*Determines if the parameter has to be byte swapped.*
- **RTI1516\_NAMESPACE::ParameterHandle get\_parameter\_handle () const**  
*Get the HLA [Parameter](#) handle.*
- **void set\_parameter\_handle (RTI1516\_NAMESPACE::ParameterHandle hdl)**  
*Set the associated HLA [Parameter](#) handle.*

- RTI1516\_NAMESPACE::VariableLengthData [get\\_encoded\\_parameter\\_value \(\)](#)  
*Gets the HLA [Parameter](#) Value using the appropriate encoding.*
- void [extract\\_data](#) (const size\_t param\_size, const unsigned char \*param\_data)  
*Extract the data out of the HLA [Parameter](#) Value.*
- bool [is\\_changed](#) () const  
*Check if a parameter value has changed.*
- void [mark\\_changed](#) ()  
*Mark the parameter as having changed.*
- void [mark\\_unchanged](#) ()  
*Mark the parameter as having NOT changed.*
- void [unpack\\_parameter\\_buffer](#) ()  
*Unpack the parameter from the buffer into the trick-variable using the appropriate decoding.*
- void [print\\_buffer](#) () const  
*Prints the contents of buffer used to encode/decode the parameter to the console on standard out.*
- ATTRIBUTES [get\\_ref2\\_attributes](#) () const  
*Get the Trick ATTRIBUTES for this [Parameter](#).*
- EncodingEnum [get\\_rti\\_encoding](#) () const  
*Get the RTI encoding for this [Parameter](#).*

## Data Fields

- char \* [trick\\_name](#)  
**Units:** –  
*Trick name for the attribute.*
- char \* [FOM\\_name](#)  
**Units:** –  
*FOM name for the attribute*
- EncodingEnum [rti\\_encoding](#)  
**Units:** –  
*RTI encoding of the data.*

## Private Member Functions

- void [ensure\\_buffer\\_capacity](#) (size\_t capacity)  
*Ensure the parameter buffer has at least the specified capacity.*
- void [pack\\_parameter\\_buffer](#) ()  
*Pack the parameter into the buffer using the appropriate encoding.*
- size\_t [get\\_parameter\\_size](#) ()  
*Gets the parameter size in bytes.*
- void [calculate\\_size\\_and\\_number\\_of\\_items](#) ()  
*Calculates the parameter size in bytes and the number of items it contains.*
- void [calculate\\_static\\_number\\_of\\_items](#) ()  
*Calculates the number of static items contained by the parameter.*
- bool [is\\_supported\\_parameter\\_type](#) () const  
*Determines if the HLA interaction parameter type is supported given the RTI encoding.*
- void [encode\\_boolean\\_to\\_buffer](#) ()  
*Encode a boolean parameter into the buffer using the HLA boolean data type which is encoded as a HLAinteger32BE.*
- void [decode\\_boolean\\_from\\_buffer](#) () const

- Decode a boolean parameter from the buffer using the HLAboolean data type which is encoded as a HLAinteger32BE.*
- void `encode_logical_time () const`

*Encode the interaction parameter using the HLAlogicalTime 64-bit integer encoding.*
  - void `decode_logical_time ()`

*Decode the interaction parameter that is using the HLAlogicalTime 64-bit integer encoding.*
  - void `encode_opaque_data_to_buffer ()`

*Encode the data as HLA opaque data into the buffer.*
  - void `decode_opaque_data_from_buffer ()`

*Decode the opaque data in the buffer.*
  - void `encode_string_to_buffer ()`

*Encode a string parameter into the buffer using the appropriate encoding.*
  - void `decode_string_from_buffer ()`

*Decode a string from the buffer into the parameter using the appropriate decoding.*
  - void `byteswap_buffer_copy (void *dest, void *src, int type, size_t length, size_t num_bytes) const`

*Copy the data from the source to the destination and byteswap as needed.*
  - `Parameter (const Parameter &rhs)`

*Copy constructor for `Parameter` class.*
  - `Parameter & operator= (const Parameter &rhs)`

*Assignment operator for `Parameter` class.*

## Private Attributes

- unsigned char \* `buffer`

**Units:** –  
*Byte buffer for the attribute value bytes.*
- size\_t `buffer_capacity`

**Units:** –  
*The capacity of the buffer.*
- bool `size_is_static`

**Units:** –  
*Flag to indicate the size of this attribute is static.*
- size\_t `size`

**Units:** –  
*The size of the attribute in bytes.*
- size\_t `num_items`

**Units:** –  
*Number of attribute items, length of the array.*
- bool `value_changed`

**Units:** –  
*Flag to indicate the attribute value changed.*
- unsigned int `HLAtrue`

**Units:** –  
*A 32-bit integer with a value of 1 on a Big Endian computer.*
- bool `byteswap`

**Units:** –  
*Flag to indicate byte-swap before RTI Rx/Tx.*
- void \* `address`

**Data I/O:** \*\*  
*Address of the trick variable*

- ATTRIBUTES \* [attr](#)

**Data I/O:** \*\*  
*ATTRIBUTES of the trick variable*
- char \* [interaction\\_FOM\\_name](#)

**Data I/O:** \*\*  
*Copy of the user-supplied interaction FOM\_name*
- RTI1516\_NAMESPACE::ParameterHandle [param\\_handle](#)

**Data I/O:** \*\*  
*The RTI parameter handle.*
- [DebugHandler debug\\_handler](#)

**Units:** –  
*Prints out multiple debug levels*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_Parameter](#) ()

### 7.42.1 Detailed Description

Definition at line 56 of file Parameter.hh.

### 7.42.2 Constructor & Destructor Documentation

#### 7.42.2.1 Parameter() [1/2]

[Parameter::Parameter](#) ( )  
 Default constructor for the [TrickHLA Parameter](#) class.

**Trick Job Class:** *initialization*

Definition at line 62 of file Parameter.cpp.

References [TrickHLA::Utilities::get\\_endianness\(\)](#), and [HLAtrue](#).

#### 7.42.2.2 ~Parameter()

[Parameter::~Parameter](#) ( ) [virtual]  
 Destructor for the [TrickHLA Parameter](#) class.  
 Frees the Trick allocated memory. **Trick Job Class:** *shutdown*  
 Definition at line 87 of file Parameter.cpp.  
 References [buffer](#), [buffer\\_capacity](#), and [interaction\\_FOM\\_name](#).

#### 7.42.2.3 Parameter() [2/2]

[TrickHLA::Parameter::Parameter](#) (   
     const [Parameter](#) & *rhs* ) [private]  
 Copy constructor for [Parameter](#) class.  
 This constructor is private to prevent inadvertent copies.

### 7.42.3 Member Function Documentation

### 7.42.3.1 byteswap\_buffer\_copy()

```
void Parameter::byteswap_buffer_copy (
    void * dest,
    void * src,
    int type,
    size_t length,
    size_t num_bytes ) const [private]
```

Copy the data from the source to the destination and byteswap as needed.

#### Parameters

<i>dest</i>	Destination to copy data to.
<i>src</i>	Source of the data to byteswap and copy from.
<i>type</i>	The type of the data.
<i>length</i>	The length/number of entries in the source array.
<i>num_bytes</i>	The number of bytes in the source array.

#### Assumptions and Limitations:

- The destination must be large enough to hold num\_bytes of data.
- Only primitive types and static arrays of primitive type are supported for now.

Definition at line 3015 of file Parameter.cpp.

References byteswap, TrickHLA::Utilities::byteswap\_double(), TrickHLA::Utilities::byteswap\_float(), TrickHLA::Utilities::byteswap\_int(), TrickHLA::Utilities::byteswap\_long(), TrickHLA::Utilities::byteswap\_long\_long(), TrickHLA::Utilities::byteswap\_short(), TrickHLA::Utilities::byteswap\_unsigned\_int(), TrickHLA::Utilities::byteswap\_unsigned\_long(), TrickHLA::Utilities::byteswap\_unsigned\_long\_long(), TrickHLA::Utilities::byteswap\_unsigned\_short(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_11\_TRACE, TrickHLA::DEBUG\_SOURCE\_PARAMETER, TrickHLA::ENCODING\_NO\_ENCODING, FOM\_name, rti\_encoding, TrickHLA::DebugHandler::should\_print(), and trick\_name.

Referenced by pack\_parameter\_buffer(), and unpack\_parameter\_buffer().

### 7.42.3.2 calculate\_number\_of\_items()

```
size_t TrickHLA::Parameter::calculate_number_of_items ( ) [inline]
```

Calculate the number of attribute items associated with this parameter.

#### Returns

The number of attribute items associated with this parameter.

Definition at line 148 of file Parameter.hh.

References calculate\_size\_and\_number\_of\_items(), and num\_items.

### 7.42.3.3 calculate\_size\_and\_number\_of\_items()

```
void Parameter::calculate_size_and_number_of_items ( ) [private]
```

Calculates the parameter size in bytes and the number of items it contains.

Definition at line 742 of file Parameter.cpp.

References address, attr, buffer\_capacity, calculate\_static\_number\_of\_items(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_10\_TRACE, TrickHLA::DEBUG\_SOURCE\_PARAMETER, TrickHLA::ENCODING\_NO\_ENCODING, TrickHLA::ENCODING\_OPAQUE\_DATA, FOM\_name, is\_byteswap(), num\_items, rti\_encoding, TrickHLA::DebugHandler::should\_print(), size, size\_is\_static, and trick\_name.

Referenced by calculate\_number\_of\_items(), complete\_initialization(), get\_parameter\_size(), pack\_parameter\_buffer(), and unpack\_parameter\_buffer().

#### 7.42.3.4 calculate\_static\_number\_of\_items()

```
void Parameter::calculate_static_number_of_items ( ) [private]
```

Calculates the number of static items contained by the parameter.

If the parameter is not for an array then a value of one is returned. Otherwise the number of items in the static array are returned.

**Assumptions and Limitations:**

- Only static arrays are supported for now. **Trick Job Class:** *initialization*

Definition at line 882 of file Parameter.cpp.

References attr, and num\_items.

Referenced by calculate\_size\_and\_number\_of\_items(), and unpack\_parameter\_buffer().

#### 7.42.3.5 complete\_initialization()

```
void Parameter::complete_initialization ( )
```

Initializes the **TrickHLA Parameter**.

**Trick Job Class:** *initialization*

Definition at line 226 of file Parameter.cpp.

References address, attr, buffer\_capacity, byteswap, calculate\_size\_and\_number\_of\_items(), debug\_handler, TrickHLA::A::DEBUG\_LEVEL\_9\_TRACE, TrickHLA::DEBUG\_SOURCE\_PARAMETER, TrickHLA::ENCODING\_ASCII\_STRING, TrickHLA::ENCODING\_BIG\_ENDIAN, TrickHLA::ENCODING\_BOOLEAN, TrickHLA::ENCODING\_C\_STRING, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_LOGICAL\_TIME, TrickHLA::ENCODING\_NO\_ENCODING, TrickHLA::ENCODING\_OPAQUE\_DATA, TrickHLA::ENCODING\_UNICODE\_STRING, TrickHLA::ENCODING\_UNKNOWN, ensure\_buffer\_capacity(), FOM\_name, interaction\_FOM\_name, is\_byteswap(), is\_static\_in\_size(), is\_supported\_parameter\_type(), TrickHLA::Utilities::is\_transmission\_byteswap(), num\_items, rti\_encoding, TrickHLA::DebugHandler::should\_print(), size, size\_is\_static, THLA\_ENDIAN, trick\_name, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by initialize().

#### 7.42.3.6 decode\_boolean\_from\_buffer()

```
void Parameter::decode_boolean_from_buffer ( ) const [private]
```

Decode a boolean parameter from the buffer using the HLAboolean data type which is encoded as a HLAinteger32BE.

Definition at line 1252 of file Parameter.cpp.

References address, attr, buffer, and num\_items.

Referenced by unpack\_parameter\_buffer().

#### 7.42.3.7 decode\_logical\_time()

```
void Parameter::decode_logical_time ( ) [private]
```

Decode the interaction parameter that is using the HLAlogicalTime 64-bit integer encoding.

Definition at line 1370 of file Parameter.cpp.

References address, attr, buffer, FOM\_name, TrickHLA::MAX\_VALUE\_IN\_MICROS, TrickHLA::MICROS\_MULTIPLIER, THLA\_ENDIAN, and trick\_name.

Referenced by unpack\_parameter\_buffer().

#### 7.42.3.8 decode\_opaque\_data\_from\_buffer()

```
void Parameter::decode_opaque_data_from_buffer ( ) [private]
```

Decode the opaque data in the buffer.

Definition at line 1525 of file Parameter.cpp.

References address, attr, buffer, decode\_string\_from\_buffer(), FOM\_name, TrickHLA::Utilities::get\_endianness(), size, THLA\_ENDL, THLA\_NEWLINE, and trick\_name.

Referenced by unpack\_parameter\_buffer().

#### 7.42.3.9 decode\_string\_from\_buffer()

```
void Parameter::decode_string_from_buffer ( ) [private]
```

Decode a string from the buffer into the parameter using the appropriate decoding.

Definition at line 2178 of file Parameter.cpp.

References address, buffer, TrickHLA::ENCODING\_ASCII\_STRING, TrickHLA::ENCODING\_C\_STRING, TrickHLA::ENCODING\_NO\_ENCODING, TrickHLA::ENCODING\_OPAQUE\_DATA, TrickHLA::ENCODING\_UNICODE\_STRING, FOM\_name, TrickHLA::Utilities::get\_endianness(), num\_items, rti\_encoding, size, size\_is\_static, THLA\_ENDL, THLA\_NEWLINE, and trick\_name.

Referenced by decode\_opaque\_data\_from\_buffer(), and unpack\_parameter\_buffer().

#### 7.42.3.10 encode\_boolean\_to\_buffer()

```
void Parameter::encode_boolean_to_buffer ( ) [private]
```

Encode a boolean parameter into the buffer using the HLAboolean data type which is encoded as a HLAinteger32BE.

Definition at line 1224 of file Parameter.cpp.

References address, attr, buffer, ensure\_buffer\_capacity(), HLAtrue, and num\_items.

Referenced by pack\_parameter\_buffer().

#### 7.42.3.11 encode\_logical\_time()

```
void Parameter::encode_logical_time ( ) const [private]
```

Encode the interaction parameter using the HLAlogicalTime 64-bit integer encoding.

Definition at line 1276 of file Parameter.cpp.

References address, attr, buffer, FOM\_name, TrickHLA::MAX\_LOGICAL\_TIME\_SECONDS, TrickHLA::MAX\_VALUE\_IN\_MICROS, TrickHLA::MICROS\_MULTIPLIER, THLA\_ENDL, and trick\_name.

Referenced by pack\_parameter\_buffer().

#### 7.42.3.12 encode\_opaque\_data\_to\_buffer()

```
void Parameter::encode_opaque_data_to_buffer ( ) [private]
```

Encode the data as HLA opaque data into the buffer.

Definition at line 1457 of file Parameter.cpp.

References address, attr, buffer, encode\_string\_to\_buffer(), ensure\_buffer\_capacity(), TrickHLA::Utilities::get\_endianness(), and size.

Referenced by pack\_parameter\_buffer().

#### 7.42.3.13 encode\_string\_to\_buffer()

```
void Parameter::encode_string_to_buffer ( ) [private]
```

Encode a string parameter into the buffer using the appropriate encoding.

Definition at line 1652 of file Parameter.cpp.

References address, buffer, TrickHLA::ENCODING\_ASCII\_STRING, TrickHLA::ENCODING\_C\_STRING, TrickHLA::ENCODING\_NO\_ENCODING, TrickHLA::ENCODING\_OPAQUE\_DATA, TrickHLA::ENCODING\_UNICODE\_STRING, ensure\_buffer\_capacity(), FOM\_name, TrickHLA::Utilities::get\_endianness(), num\_items, rti\_encoding, size, THLA-ENDL, and trick\_name.

Referenced by encode\_opaque\_data\_to\_buffer(), and pack\_parameter\_buffer().

#### 7.42.3.14 ensure\_buffer\_capacity()

```
void Parameter::ensure_buffer_capacity (
    size_t capacity ) [private]
```

Ensure the parameter buffer has at least the specified capacity.

##### Parameters

<i>capacity</i>	Desired capacity of the buffer in bytes.
-----------------	--

Definition at line 711 of file Parameter.cpp.

References buffer, buffer\_capacity, FOM\_name, and THLA-ENDL.

Referenced by complete\_initialization(), encode\_boolean\_to\_buffer(), encode\_opaque\_data\_to\_buffer(), encode\_string\_to\_buffer(), extract\_data(), and pack\_parameter\_buffer().

#### 7.42.3.15 extract\_data()

```
void Parameter::extract_data (
    const size_t param_size,
    const unsigned char * param_data )
```

Extract the data out of the HLA [Parameter](#) Value.

##### Parameters

<i>param_size</i>	<a href="#">Parameter</a> data size.
<i>param_data</i>	<a href="#">Parameter</a> data.

Definition at line 522 of file Parameter.cpp.

References attr, buffer, debug\_handler, TrickHLA::DEBUG\_LEVEL\_11\_TRACE, TrickHLA::DEBUG\_LEVEL\_7\_TRACE, TrickHLA::DEBUG\_SOURCE\_PARAMETER, TrickHLA::ENCODING\_BOOLEAN, TrickHLA::ENCODING\_LOGICAL\_TIME, TrickHLA::ENCODING\_NO\_ENCODING, TrickHLA::ENCODING\_OPAQUE\_DATA, TrickHLA::ENCODING\_UNICODE\_STRING, ensure\_buffer\_capacity(), FOM\_name, get\_parameter\_size(), interaction\_FOM\_name, mark\_changed(), print\_buffer(), rti\_encoding, TrickHLA::DebugHandler::should\_print(), size, size\_is\_static, THLA-ENDL, T-  
HLA\_NEWLINE, and trick\_name.

Referenced by TrickHLA::Interaction::extract\_data().

#### 7.42.3.16 get\_encoded\_parameter\_value()

```
VariableLengthData Parameter::get_encoded_parameter_value ( )
```

Gets the HLA [Parameter](#) Value using the appropriate encoding.

**Returns**

Encoded parameter value.

Definition at line 509 of file Parameter.cpp.

References buffer, TrickHLA::ENCODING\_BOOLEAN, pack\_parameter\_buffer(), rti\_encoding, and size.

Referenced by TrickHLA::Interaction::send().

**7.42.3.17 get\_FOM\_name()**

```
const char* TrickHLA::Parameter::get_FOM_name ( ) const [inline]
```

Get the FOM name for this parameter.

**Returns**

The FOM name of this parameter.

Definition at line 116 of file Parameter.hh.

References FOM\_name.

Referenced by TrickHLA::Interaction::extract\_data(), print\_buffer(), IMSim::ExecutionControl::setup\_interaction\_ref\_attributes(), SpaceFOM::ExecutionControl::setup\_interaction\_ref\_attributes(), and TrickHLA::Manager::setup\_interaction\_RTI\_handles().

**7.42.3.18 get\_parameter\_handle()**

```
RTI1516_NAMESPACE::ParameterHandle TrickHLA::Parameter::get_parameter_handle ( ) const [inline]
```

Get the HLA [Parameter](#) handle.

**Returns**

The HLA [Parameter](#) handle.

Definition at line 164 of file Parameter.hh.

References param\_handle.

Referenced by TrickHLA::Interaction::send().

**7.42.3.19 get\_parameter\_size()**

```
size_t Parameter::get_parameter_size ( ) [private]
```

Gets the parameter size in bytes.

**Returns**

The size in bytes of the parameter.

If the parameter is static in size it uses a cached size value otherwise the size is calculated.

Definition at line 849 of file Parameter.cpp.

References calculate\_size\_and\_number\_of\_items(), size, and size\_is\_static.

Referenced by extract\_data().

**7.42.3.20 get\_ref2\_attributes()**

```
ATTRIBUTES TrickHLA::Parameter::get_ref2_attributes ( ) const [inline]
```

Get the Trick [ATTRIBUTES](#) for this [Parameter](#).

**Returns**

A copy of the Trick ATTRIBUTES structure for this [Parameter](#).

Definition at line 200 of file Parameter.hh.

References attr.

**7.42.3.21 get\_rti\_encoding()**

`EncodingEnum TrickHLA::Parameter::get_rti_encoding () const [inline]`  
Get the RTI encoding for this [Parameter](#).

**Returns**

The encoding type for this [Parameter](#).

Definition at line 204 of file Parameter.hh.

References rti\_encoding.

**7.42.3.22 get\_trick\_name()**

`const char* TrickHLA::Parameter::get_trick_name () const [inline]`  
Get the Trick variable name associated with this parameter.

**Returns**

The Trick variable name associated with this parameter.

Definition at line 133 of file Parameter.hh.

References trick\_name.

Referenced by TrickHLA::Manager::setup\_interaction\_ref\_attributes().

**7.42.3.23 initialize() [1/2]**

```
void Parameter::initialize (
    const char * interaction_fom_name,
    const int interaction_index,
    const int parameter_index )
```

Initializes the [TrickHLA Parameter](#) from the trick variable name supplied in the input file.

**Parameters**

<code>interaction_fom_name</code>	The FOM name of the parent interaction.
<code>interaction_index</code>	The array index to the parent <a href="#">TrickHLA::Interaction</a> .
<code>parameter_index</code>	The array index to this <a href="#">Parameter</a> .

**Trick Job Class: initialization**

Definition at line 108 of file Parameter.cpp.

References address, attr, complete\_initialization(), TrickHLA::ENCODING\_FIRST\_VALUE, TrickHLA::ENCODING\_L← AST\_VALUE, FOM\_name, interaction\_FOM\_name, rti\_encoding, THLA\_ENDL, trick\_name, and TRICKHLA\_VALID← ATE\_FPU\_CONTROL\_WORD.

Referenced by IMSim::ExecutionControl::setup\_interaction\_ref\_attributes(), SpaceFOM::ExecutionControl::setup← interaction\_ref\_attributes(), and TrickHLA::Manager::setup\_interaction\_ref\_attributes().

#### 7.42.3.24 initialize() [2/2]

```
void Parameter::initialize (
    const char * interaction_fom_name,
    void * in_addr,
    ATTRIBUTES * in_attr )
```

Initializes the [TrickHLA Parameter](#) from the supplied address and ATTRIBUTES of the trick variable.

##### Parameters

<i>interaction_fom_name</i>	FOM name of the interaction.
<i>in_addr</i>	Address of the trick variable.
<i>in_attr</i>	ATTRIBUTES of the trick variable.

##### Trick Job Class: *initialization*

Definition at line 193 of file Parameter.cpp.

References address, attr, complete\_initialization(), interaction\_FOM\_name, and THLA\_ENDL.

#### 7.42.3.25 is\_byteswap()

```
bool TrickHLA::Parameter::is_byteswap ( ) const [inline]
```

Determines if the parameter has to be byte swapped.

##### Returns

True if parameter is byte swapped.

Definition at line 160 of file Parameter.hh.

References byteswap.

Referenced by calculate\_size\_and\_number\_of\_items(), complete\_initialization(), pack\_parameter\_buffer(), print← buffer(), and unpack\_parameter\_buffer().

#### 7.42.3.26 is\_changed()

```
bool TrickHLA::Parameter::is_changed ( ) const [inline]
```

Check if a parameter value has changed.

##### Returns

True if a parameter value has changed; False otherwise.

Definition at line 182 of file Parameter.hh.

References value\_changed.

#### 7.42.3.27 is\_static\_in\_size()

```
bool Parameter::is_static_in_size ( ) const
```

Determines if the parameter is static in size.

##### Returns

True if parameter size is static.

Definition at line 857 of file Parameter.cpp.

References attr, and is\_supported\_parameter\_type().

Referenced by complete\_initialization().

**7.42.3.28 is\_supported\_parameter\_type()**

```
bool Parameter::is_supported_parameter_type ( ) const [private]
Determines if the HLA interaction parameter type is supported given the RTI encoding.
```

**Returns**

True if supported, false otherwise.

**Assumptions and Limitations:**

- Only primitive types and static arrays of primitive type are supported for now.

Definition at line 3188 of file Parameter.cpp.

References attr, TrickHLA::ENCODING\_ASCII\_STRING, TrickHLA::ENCODING\_BIG\_ENDIAN, TrickHLA::ENCODING\_G\_BOOLEAN, TrickHLA::ENCODING\_C\_STRING, TrickHLA::ENCODING\_LITTLE\_ENDIAN, TrickHLA::ENCODING\_LOGICAL\_TIME, TrickHLA::ENCODING\_NO\_ENCODING, TrickHLA::ENCODING\_OPAQUE\_DATA, TrickHLA::ENCODING\_UNICODE\_STRING, TrickHLA::ENCODING\_UNKNOWN, and rti\_encoding.

Referenced by complete\_initialization(), and is\_static\_in\_size().

**7.42.3.29 mark\_changed()**

```
void TrickHLA::Parameter::mark_changed ( ) [inline]
```

Mark the parameter as having changed.

Definition at line 185 of file Parameter.hh.

References value\_changed.

Referenced by extract\_data().

**7.42.3.30 mark\_unchanged()**

```
void TrickHLA::Parameter::mark_unchanged ( ) [inline]
```

Mark the parameter as having NOT changed.

Definition at line 188 of file Parameter.hh.

References value\_changed.

Referenced by TrickHLA::Interaction::mark\_unchanged().

**7.42.3.31 operator=()**

```
Parameter& TrickHLA::Parameter::operator= (
    const Parameter & rhs ) [private]
```

Assignment operator for Parameter class.

This assignment operator is private to prevent inadvertent copies.

**7.42.3.32 pack\_parameter\_buffer()**

```
void Parameter::pack_parameter_buffer ( ) [private]
```

Pack the parameter into the buffer using the appropriate encoding.

Definition at line 898 of file Parameter.cpp.

References address, attr, buffer, buffer\_capacity, byteswap\_buffer\_copy(), calculate\_size\_and\_number\_of\_items(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_10\_TRACE, TrickHLA::DEBUG\_LEVEL\_11\_TRACE, TrickHLA::DEBU\_G\_SOURCE\_PARAMETER, encode\_boolean\_to\_buffer(), encode\_logical\_time(), encode\_opaque\_data\_to\_buffer(), encode\_string\_to\_buffer(), TrickHLA::ENCODING\_BOOLEAN, TrickHLA::ENCODING\_LOGICAL\_TIME, TrickHLA::ENCODING\_OPAQUE\_DATA, ensure\_buffer\_capacity(), FOM\_name, is\_byteswap(), num\_items, print\_buffer(), rti\_encoding, TrickHLA::DebugHandler::should\_print(), size, size\_is\_static, and trick\_name.

Referenced by `get_encoded_parameter_value()`.

#### 7.42.3.33 `print_buffer()`

`void Parameter::print_buffer ( ) const`

Prints the contents of buffer used to encode/decode the parameter to the console on standard out.

Definition at line 3258 of file `Parameter.cpp`.

References `attr`, `buffer`, `TrickHLA::Utilities::byteswap_double()`, `get_FOM_name()`, `is_byteswap()`, `num_items`, and `size`. Referenced by `extract_data()`, `pack_parameter_buffer()`, and `unpack_parameter_buffer()`.

#### 7.42.3.34 `set_debug_level()`

`void TrickHLA::Parameter::set_debug_level (`  
`DebugHandler hndlr ) [inline]`

Set the debug handler for this parameter.

##### Parameters

<code>hndlr</code>	The <code>TrickHLA::DebugHandler</code> instance.
--------------------	---

Definition at line 112 of file `Parameter.hh`.

References `debug_handler`, and `TrickHLA::DebugHandler::set()`.

Referenced by `IMSim::ExecutionControl::setup_interaction_ref_attributes()`, `SpaceFOM::ExecutionControl::setup_interaction_ref_attributes()`, and `TrickHLA::Manager::setup_interaction_ref_attributes()`.

#### 7.42.3.35 `set_encoding()`

`void TrickHLA::Parameter::set_encoding (`  
`EncodingEnum in_type ) [inline]`

Set the RTI encoding and based on the new encoding determine if we need to byte-swap.

##### Parameters

<code>in_type</code>	The encoding type for this parameter.
----------------------	---------------------------------------

Definition at line 138 of file `Parameter.hh`.

References `byteswap`, `TrickHLA::Utilities::is_transmission_byteswap()`, and `rti_encoding`.

Referenced by `IMSim::ExecutionControl::setup_interaction_ref_attributes()`, and `SpaceFOM::ExecutionControl::setup_interaction_ref_attributes()`.

#### 7.42.3.36 `set_FOM_name()`

`void TrickHLA::Parameter::set_FOM_name (`  
`const char * in_name ) [inline]`

Set the FOM name for the parameter.

##### Parameters

<code>in_name</code>	The FOM name for the parameter.
----------------------	---------------------------------

Definition at line 120 of file Parameter.hh.

References FOM\_name.

Referenced by IMSim::ExecutionControl::setup\_interaction\_ref\_attributes(), and SpaceFOM::ExecutionControl::setup\_interaction\_ref\_attributes().

#### 7.42.3.37 set\_parameter\_handle()

```
void TrickHLA::Parameter::set_parameter_handle (
    RTI1516_NAMESPACE::ParameterHandle hdl ) [inline]
```

Set the associated HLA [Parameter](#) handle.

##### Parameters

<i>hdl</i>	The associated HLA <a href="#">Parameter</a> handle.
------------	--

Definition at line 168 of file Parameter.hh.

References param\_handle.

Referenced by TrickHLA::Manager::setup\_interaction\_RTI\_handles().

#### 7.42.3.38 unpack\_parameter\_buffer()

```
void Parameter::unpack_parameter_buffer ( )
```

Unpack the parameter from the buffer into the trick-variable using the appropriate decoding.

Definition at line 1078 of file Parameter.cpp.

References address, attr, buffer, buffer\_capacity, byteswap\_buffer\_copy(), calculate\_size\_and\_number\_of\_items(), calculate\_static\_number\_of\_items(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_10\_TRACE, TrickHLA::DEBUG\_LEVEL\_11\_TRACE, TrickHLA::DEBUG\_SOURCE\_PARAMETER, decode\_boolean\_from\_buffer(), decode\_logical\_time(), decode\_opaque\_data\_from\_buffer(), decode\_string\_from\_buffer(), TrickHLA::ENCODING\_BOOLEAN, TrickHLA::ENCODING\_LOGICAL\_TIME, TrickHLA::ENCODING\_OPAQUE\_DATA, FOM\_name, is\_byteswap(), num\_items, print\_buffer(), rti\_encoding, TrickHLA::DebugHandler::should\_print(), size, size\_is\_static, and trick\_name.

Referenced by TrickHLA::Interaction::process\_interaction().

### 7.42.4 Friends And Related Function Documentation

#### 7.42.4.1 init\_attrTrickHLA\_Parameter

```
void init_attrTrickHLA_Parameter ( ) [friend]
```

#### 7.42.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 63 of file Parameter.hh.

### 7.42.5 Field Documentation

#### 7.42.5.1 address

```
void* TrickHLA::Parameter::address [private]
```

**Data I/O: \*\***

Address of the trick variable

Definition at line 221 of file Parameter.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), complete\_initialization(), decode\_boolean\_from\_buffer(), decode\_logical\_time(), decode\_opaque\_data\_from\_buffer(), decode\_string\_from\_buffer(), encode\_boolean\_to\_buffer(), encode\_logical\_time(), encode\_opaque\_data\_to\_buffer(), encode\_string\_to\_buffer(), initialize(), pack\_parameter\_buffer(), and unpack\_parameter\_buffer().

#### 7.42.5.2 attr

```
ATTRIBUTES* TrickHLA::Parameter::attr [private]
```

**Data I/O: \*\***

ATTRIBUTES of the trick variable

Definition at line 222 of file Parameter.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), calculate\_static\_number\_of\_items(), complete\_initialization(), decode\_boolean\_from\_buffer(), decode\_logical\_time(), decode\_opaque\_data\_from\_buffer(), encode\_boolean\_to\_buffer(), encode\_logical\_time(), encode\_opaque\_data\_to\_buffer(), extract\_data(), get\_ref2\_attributes(), initialize(), is\_static\_in\_size(), is\_supported\_parameter\_type(), pack\_parameter\_buffer(), print\_buffer(), and unpack\_parameter\_buffer().

#### 7.42.5.3 buffer

```
unsigned char* TrickHLA::Parameter::buffer [private]
```

**Units: -**

Byte buffer for the attribute value bytes.

Definition at line 207 of file Parameter.hh.

Referenced by decode\_boolean\_from\_buffer(), decode\_logical\_time(), decode\_opaque\_data\_from\_buffer(), decode\_string\_from\_buffer(), encode\_boolean\_to\_buffer(), encode\_logical\_time(), encode\_opaque\_data\_to\_buffer(), encode\_string\_to\_buffer(), ensure\_buffer\_capacity(), extract\_data(), get\_encoded\_parameter\_value(), pack\_parameter\_buffer(), print\_buffer(), unpack\_parameter\_buffer(), and ~Parameter().

#### 7.42.5.4 buffer\_capacity

```
size_t TrickHLA::Parameter::buffer_capacity [private]
```

**Units: -**

The capacity of the buffer.

Definition at line 208 of file Parameter.hh.

Referenced by calculate\_size\_and\_number\_of\_items(), complete\_initialization(), ensure\_buffer\_capacity(), pack\_parameter\_buffer(), unpack\_parameter\_buffer(), and ~Parameter().

#### 7.42.5.5 byteswap

```
bool TrickHLA::Parameter::byteswap [private]
```

**Units: -**

Flag to indicate byte-swap before RTI Rx/Tx.

Definition at line 219 of file Parameter.hh.

Referenced by byteswap\_buffer\_copy(), complete\_initialization(), is\_byteswap(), and set\_encoding().

#### 7.42.5.6 debug\_handler

`DebugHandler TrickHLA::Parameter::debug_handler [private]`

**Units:** –

Prints out multiple debug levels

Definition at line 227 of file Parameter.hh.

Referenced by `byteswap_buffer_copy()`, `calculate_size_and_number_of_items()`, `complete_initialization()`, `extract_data()`, `pack_parameter_buffer()`, `set_debug_level()`, and `unpack_parameter_buffer()`.

#### 7.42.5.7 FOM\_name

`char* TrickHLA::Parameter::FOM_name`

**Units:** –

FOM name for the attribute

Definition at line 73 of file Parameter.hh.

Referenced by `byteswap_buffer_copy()`, `calculate_size_and_number_of_items()`, `complete_initialization()`, `decode_logical_time()`, `decode_opaque_data_from_buffer()`, `decode_string_from_buffer()`, `encode_logical_time()`, `encode_string_to_buffer()`, `ensure_buffer_capacity()`, `extract_data()`, `get_FOM_name()`, `initialize()`, `pack_parameter_buffer()`, `set_FOM_name()`, and `unpack_parameter_buffer()`.

#### 7.42.5.8 HLAture

`unsigned int TrickHLA::Parameter::HLAture [private]`

**Units:** –

A 32-bit integer with a value of 1 on a Big Endian computer.

Definition at line 217 of file Parameter.hh.

Referenced by `encode_boolean_to_buffer()`, and `Parameter()`.

#### 7.42.5.9 interaction\_FOM\_name

`char* TrickHLA::Parameter::interaction_FOM_name [private]`

**Data I/O:** \*\*

Copy of the user-supplied interaction FOM\_name

Definition at line 223 of file Parameter.hh.

Referenced by `complete_initialization()`, `extract_data()`, `initialize()`, and `~Parameter()`.

#### 7.42.5.10 num\_items

`size_t TrickHLA::Parameter::num_items [private]`

**Units:** –

Number of attribute items, length of the array.

Definition at line 213 of file Parameter.hh.

Referenced by `calculate_number_of_items()`, `calculate_size_and_number_of_items()`, `calculate_static_number_of_items()`, `complete_initialization()`, `decode_boolean_from_buffer()`, `decode_string_from_buffer()`, `encode_boolean_to_buffer()`, `encode_string_to_buffer()`, `pack_parameter_buffer()`, `print_buffer()`, and `unpack_parameter_buffer()`.

#### 7.42.5.11 param\_handle

`RTI1516_NAMESPACE::ParameterHandle TrickHLA::Parameter::param_handle [private]`

**Data I/O: \*\***

The RTI parameter handle.

Definition at line 225 of file Parameter.hh.

Referenced by `get_parameter_handle()`, and `set_parameter_handle()`.

**7.42.5.12 rti\_encoding**

```
EncodingEnum TrickHLA::Parameter::rti_encoding
```

**Units: –**

RTI encoding of the data.

Definition at line 75 of file Parameter.hh.

Referenced by `byteswap_buffer_copy()`, `calculate_size_and_number_of_items()`, `complete_initialization()`, `decode_string_from_buffer()`, `encode_string_to_buffer()`, `extract_data()`, `get_encoded_parameter_value()`, `get_rti_encoding()`, `initialize()`, `is_supported_parameter_type()`, `pack_parameter_buffer()`, `set_encoding()`, and `unpack_parameter_buffer()`.

**7.42.5.13 size**

```
size_t TrickHLA::Parameter::size [private]
```

**Units: –**

The size of the attribute in bytes.

Definition at line 212 of file Parameter.hh.

Referenced by `calculate_size_and_number_of_items()`, `complete_initialization()`, `decode_opaque_data_from_buffer()`, `decode_string_from_buffer()`, `encode_opaque_data_to_buffer()`, `encode_string_to_buffer()`, `extract_data()`, `get_encoded_parameter_value()`, `get_parameter_size()`, `pack_parameter_buffer()`, `print_buffer()`, and `unpack_parameter_buffer()`.

**7.42.5.14 size\_is\_static**

```
bool TrickHLA::Parameter::size_is_static [private]
```

**Units: –**

Flag to indicate the size of this attribute is static.

Definition at line 210 of file Parameter.hh.

Referenced by `calculate_size_and_number_of_items()`, `complete_initialization()`, `decode_string_from_buffer()`, `extract_data()`, `get_parameter_size()`, `pack_parameter_buffer()`, and `unpack_parameter_buffer()`.

**7.42.5.15 trick\_name**

```
char* TrickHLA::Parameter::trick_name
```

**Units: –**

Trick name for the attribute.

Definition at line 72 of file Parameter.hh.

Referenced by `byteswap_buffer_copy()`, `calculate_size_and_number_of_items()`, `complete_initialization()`, `decode_logical_time()`, `decode_opaque_data_from_buffer()`, `decode_string_from_buffer()`, `encode_logical_time()`, `encode_string_to_buffer()`, `extract_data()`, `get_trick_name()`, `initialize()`, `pack_parameter_buffer()`, and `unpack_parameter_buffer()`.

**7.42.5.16 value\_changed**

```
bool TrickHLA::Parameter::value_changed [private]
```

**Units:** –

Flag to indicate the attribute value changed.

Definition at line 215 of file Parameter.hh.

Referenced by `is_changed()`, `mark_changed()`, and `mark_unchanged()`.

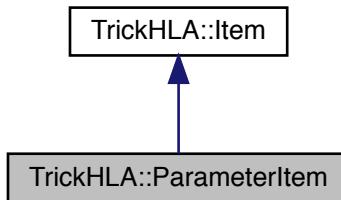
The documentation for this class was generated from the following files:

- [Parameter.hh](#)
- [Parameter.cpp](#)

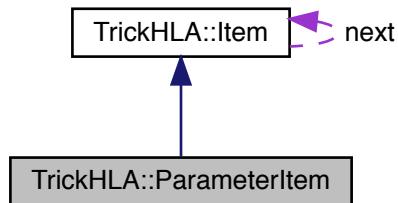
## 7.43 TrickHLA::ParameterItem Class Reference

```
#include <ParameterItem.hh>
```

Inheritance diagram for TrickHLA::ParameterItem:



Collaboration diagram for TrickHLA::ParameterItem:



## Public Member Functions

- [ParameterItem \(\)](#)  
*Default constructor for the `TrickHLA ParameterItem` class.*
- [ParameterItem \(int parameter\\_index, const RTI1516\\_NAMESPACE::VariableLengthData \\*param\\_value\)](#)  
*Initialization constructor for the `TrickHLA ParameterItem` class.*
- [virtual ~ParameterItem \(\)](#)

*Destructor for the [TrickHLA ParameterItem](#) class.*

- void [clear](#) ()
 

*Frees allocated memory.*

## Data Fields

- int [index](#)

**Units:** –  
*Index to the applicable parameter.*
- size\_t [size](#)

**Units:** –  
*Number of bytes in the parameter data.*
- unsigned char \* [data](#)

**Units:** –  
*Parameter data.*

## Private Member Functions

- [ParameterItem](#) (const [ParameterItem](#) &rhs)
 

*Copy constructor for [ParameterItem](#) class.*
- [ParameterItem](#) & [operator=](#) (const [ParameterItem](#) &rhs)
 

*Assignment operator for [ParameterItem](#) class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_ParameterItem](#) ()

### 7.43.1 Detailed Description

Definition at line 50 of file ParameterItem.hh.

### 7.43.2 Constructor & Destructor Documentation

#### 7.43.2.1 ParameterItem() [1/3]

`ParameterItem::ParameterItem ( )`  
 Default constructor for the [TrickHLA ParameterItem](#) class.

**Trick Job Class:** *initialization*

Definition at line 49 of file ParameterItem.cpp.

#### 7.43.2.2 ParameterItem() [2/3]

```
ParameterItem::ParameterItem (
    int parameter_index,
    const RTI1516_NAMESPACE::VariableLengthData * param_value )
```

Initialization constructor for the [TrickHLA ParameterItem](#) class.

#### Parameters

<i>parameter_index</i>	<a href="#">Parameter</a> index.
------------------------	----------------------------------

**Parameters**

<code>param_value</code>	HLA RTI <a href="#">Parameter</a> value.
--------------------------	--

**Trick Job Class:** *initialization*

Definition at line 59 of file ParameterItem.cpp.

References data, and size.

**7.43.2.3 ~ParameterItem()**`ParameterItem::~ParameterItem ( ) [virtual]`Destructor for the [TrickHLA ParameterItem](#) class.**Trick Job Class:** *shutdown*

Definition at line 81 of file ParameterItem.cpp.

References clear().

**7.43.2.4 ParameterItem() [3/3]**`TrickHLA::ParameterItem::ParameterItem (`  
    `const ParameterItem & rhs ) [private]`Copy constructor for [ParameterItem](#) class.

This constructor is private to prevent inadvertent copies.

**7.43.3 Member Function Documentation****7.43.3.1 clear()**`void ParameterItem::clear ( )`

Frees allocated memory.

Definition at line 86 of file ParameterItem.cpp.

References data, index, and size.

Referenced by `TrickHLA::InteractionItem::clear_parm_items()`, and `~ParameterItem()`.**7.43.3.2 operator=()**`ParameterItem& TrickHLA::ParameterItem::operator= (`  
    `const ParameterItem & rhs ) [private]`Assignment operator for [ParameterItem](#) class.

This assignment operator is private to prevent inadvertent copies.

**7.43.4 Friends And Related Function Documentation****7.43.4.1 init\_attrTrickHLA\_\_ParameterItem**`void init_attrTrickHLA__ParameterItem ( ) [friend]`

#### 7.43.4.2 InputProcessor

```
friend class InputProcessor [friend]
Definition at line 57 of file ParameterItem.hh.
```

### 7.43.5 Field Documentation

#### 7.43.5.1 data

```
unsigned char* TrickHLA::ParameterItem::data
```

**Units:** –

[Parameter](#) data.

Definition at line 85 of file ParameterItem.hh.

Referenced by [TrickHLA::InteractionItem::checkpoint\\_queue\(\)](#), [clear\(\)](#), [TrickHLA::Interaction::extract\\_data\(\)](#), [ParameterItem\(\)](#), and [TrickHLA::InteractionItem::restore\\_queue\(\)](#).

#### 7.43.5.2 index

```
int TrickHLA::ParameterItem::index
```

**Units:** –

Index to the applicable parameter.

Definition at line 82 of file ParameterItem.hh.

Referenced by [TrickHLA::InteractionItem::checkpoint\\_queue\(\)](#), [clear\(\)](#), [TrickHLA::Manager::dump\\_interactions\(\)](#), [TrickHLA::Interaction::extract\\_data\(\)](#), and [TrickHLA::InteractionItem::restore\\_queue\(\)](#).

#### 7.43.5.3 size

```
size_t TrickHLA::ParameterItem::size
```

**Units:** –

Number of bytes in the parameter data.

Definition at line 84 of file ParameterItem.hh.

Referenced by [TrickHLA::InteractionItem::checkpoint\\_queue\(\)](#), [clear\(\)](#), [TrickHLA::Manager::dump\\_interactions\(\)](#), [TrickHLA::Interaction::extract\\_data\(\)](#), [ParameterItem\(\)](#), and [TrickHLA::InteractionItem::restore\\_queue\(\)](#).

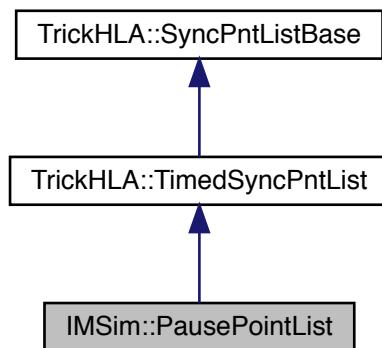
The documentation for this class was generated from the following files:

- [ParameterItem.hh](#)
- [ParameterItem.cpp](#)

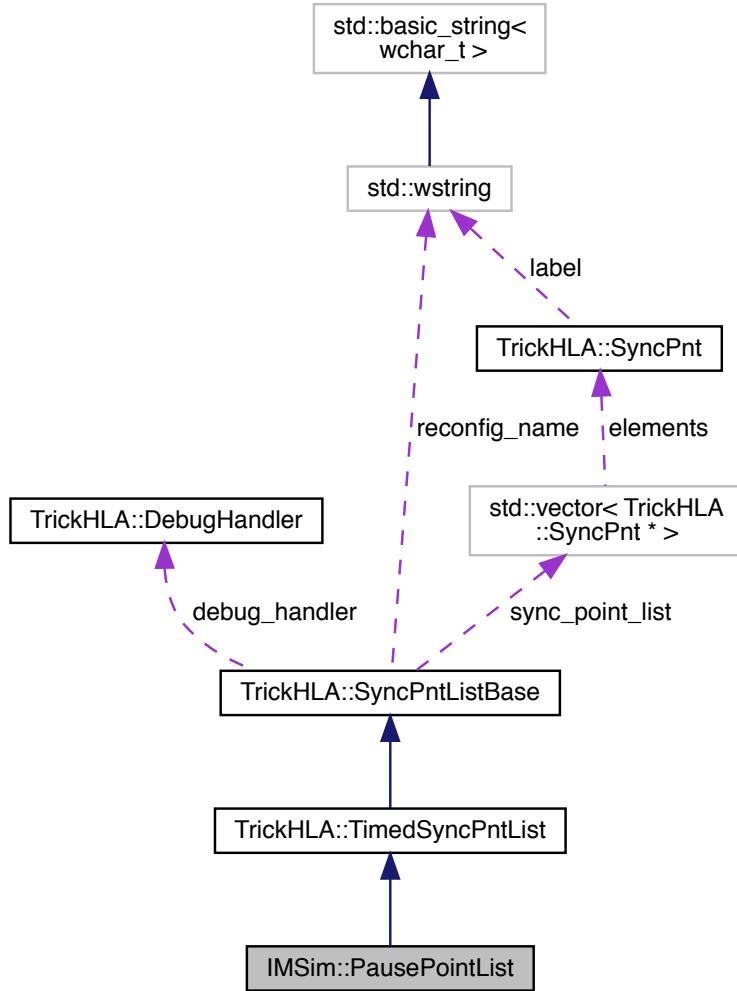
## 7.44 IMSim::PausePointList Class Reference

```
#include <PausePointList.hh>
```

Inheritance diagram for IMSim::PausePointList:



Collaboration diagram for IMSim::PausePointList:



## Public Member Functions

- `PausePointList ()`  
*Default constructor for the `TrickHLA PausePointList` class.*
- `virtual ~PausePointList ()`  
*Destructor for the `TrickHLA PausePointList` class.*
- `virtual bool clear_sync_pnt (std::wstring const &label)`  
*Clear the given synchronization point label.*
- `void check_state ()`  
*Check the state of the sync-points to determine if the state needs to go to Freeze or Run.*
- `PausePointStateEnum get_state () const`
- `void clear_state ()`

- `bool should_run () const`
- `bool should_freeze () const`
- `bool should_exit () const`
- `bool should_restart () const`
- `bool should_reconfig () const`
- `std::wstring to_string ()`

*Returns a wide string representing the state of the synchronization points.*

- `virtual void print_sync_pnts ()`

*Dumps synchronization point information to the screen.*

## Private Member Functions

- `PausePointList (const PausePointList &rhs)`  
*Copy constructor for `PausePointList` class.*
- `PausePointList & operator= (const PausePointList &rhs)`  
*Assignment operator for `PausePointList` class.*

## Private Attributes

- `PausePointStateEnum state`

*Units:* –

*State of the pause points.*

## Friends

- `class InputProcessor`
- `void init_attrTrickHLA_PausePointList ()`

## Additional Inherited Members

### 7.44.1 Detailed Description

Definition at line 58 of file IMSim/PausePointList.hh.

### 7.44.2 Constructor & Destructor Documentation

#### 7.44.2.1 PausePointList() [1/2]

`PausePointList::PausePointList ( )`

Default constructor for the `TrickHLA PausePointList` class.

**Trick Job Class:** *initialization*

Definition at line 57 of file IMSim/PausePointList.cpp.

#### 7.44.2.2 ~PausePointList()

`virtual IMSim::PausePointList::~PausePointList ( ) [inline], [virtual]`

Destructor for the `TrickHLA PausePointList` class.

Definition at line 78 of file IMSim/PausePointList.hh.

### 7.44.2.3 PausePointList() [2/2]

```
IMSim::PausePointList::PausePointList (
    const PausePointList & rhs ) [private]
```

Copy constructor for `PausePointList` class.

This constructor is private to prevent inadvertent copies.

## 7.44.3 Member Function Documentation

### 7.44.3.1 check\_state()

```
void PausePointList::check_state ( )
```

Check the state of the sync-points to determine if the state needs to go to Freeze or Run.

Definition at line 94 of file `IMSim/PausePointList.cpp`.

References `TrickHLA::SyncPntListBase::lock_read_only()`, `IMSim::PAUSE_POINT_STATE_EXIT`, `IMSim::PAUSE_`

`POINT_STATE_FREEZE`, `IMSim::PAUSE_POINT_STATE_RECONFIG`, `IMSim::PAUSE_POINT_STATE_RESTART`,

`IMSim::PAUSE_POINT_STATE_RUN`, `IMSim::PAUSE_POINT_STATE_UNKNOWN`, `state`, `TrickHLA::SYNC_PNT_`

`STATE_ACHIEVED`, `TrickHLA::SyncPntListBase::sync_point_list`, and `TrickHLA::SyncPntListBase::unlock_read_only()`.

Referenced by `IMSim::ExecutionControl::check_freeze_exit()`.

### 7.44.3.2 clear\_state()

```
void IMSim::PausePointList::clear_state ( ) [inline]
```

Definition at line 91 of file `IMSim/PausePointList.hh`.

References `IMSim::PAUSE_POINT_STATE_UNKNOWN`, and `state`.

### 7.44.3.3 clear\_sync\_pnt()

```
bool PausePointList::clear_sync_pnt (
    std::wstring const & label ) [virtual]
```

Clear the given synchronization point label.

#### Returns

True if synchronization point is cleared.

#### Parameters

<code>label</code>	The synchronization point label.
--------------------	----------------------------------

Reimplemented from `TrickHLA::SyncPntListBase`.

Definition at line 62 of file `IMSim/PausePointList.cpp`.

References `TrickHLA::SyncPnt::get_label()`, `TrickHLA::SyncPnt::get_state()`, `TrickHLA::SyncPntListBase::lock_read_`

`write()`, `IMSim::PAUSE_POINT_STATE_EXIT`, `IMSim::PAUSE_POINT_STATE_RECONFIG`, `IMSim::PAUSE_`

`POINT_STATE_RESTART`, `TrickHLA::SyncPntListBase::reconfig_name`, `state`, `TrickHLA::SYNC_PNT_`

`STATE_ACHIEVED`, `TrickHLA::SyncPntListBase::sync_point_list`, and `TrickHLA::SyncPntListBase::unlock_read_write()`.

Referenced by `IMSim::ExecutionControl::clear_pause()`.

#### 7.44.3.4 `get_state()`

```
PausePointStateEnum IMSim::PausePointList::get_state ( ) const [inline]
```

Definition at line 90 of file IMSim/PausePointList.hh.

References state.

#### 7.44.3.5 `operator=()`

```
PausePointList& IMSim::PausePointList::operator= (
    const PausePointList & rhs ) [private]
```

Assignment operator for `PausePointList` class.

This assignment operator is private to prevent inadvertent copies.

#### 7.44.3.6 `print_sync_pnts()`

```
void PausePointList::print_sync_pnts ( ) [virtual]
```

Dumps synchronization point information to the screen.

Reimplemented from `TrickHLA::TimedSyncPntList`.

Definition at line 188 of file IMSim/PausePointList.cpp.

References `TrickHLA::TimedSyncPnt::get_time()`, `TrickHLA::Int64Time::getDoubleTime()`, `TrickHLA::SyncPntList::Base::lock_read_only()`, `TrickHLA::SyncPntListBase::sync_point_list`, and `TrickHLA::SyncPntListBase::unlock_read_only()`.

#### 7.44.3.7 `should_exit()`

```
bool IMSim::PausePointList::should_exit ( ) const [inline]
```

Definition at line 95 of file IMSim/PausePointList.hh.

References `IMSim::PAUSE_POINT_STATE_EXIT`, and state.

Referenced by `IMSim::ExecutionControl::check_freeze_exit()`.

#### 7.44.3.8 `should_freeze()`

```
bool IMSim::PausePointList::should_freeze ( ) const [inline]
```

Definition at line 94 of file IMSim/PausePointList.hh.

References `IMSim::PAUSE_POINT_STATE_FREEZE`, and state.

#### 7.44.3.9 `should_reconfig()`

```
bool IMSim::PausePointList::should_reconfig ( ) const [inline]
```

Definition at line 97 of file IMSim/PausePointList.hh.

References `IMSim::PAUSE_POINT_STATE_RECONFIG`, and state.

Referenced by `IMSim::ExecutionControl::check_freeze_exit()`.

#### 7.44.3.10 `should_restart()`

```
bool IMSim::PausePointList::should_restart ( ) const [inline]
```

Definition at line 96 of file IMSim/PausePointList.hh.

References `IMSim::PAUSE_POINT_STATE_RESTART`, and state.

Referenced by `IMSim::ExecutionControl::check_freeze_exit()`.

#### 7.44.3.11 `should_run()`

```
bool IMSim::PausePointList::should_run ( ) const [inline]
```

Definition at line 93 of file IMSim/PausePointList.hh.

References IMSim::PAUSE\_POINT\_STATE\_RUN, and state.

Referenced by IMSim::ExecutionControl::check\_freeze\_exit().

#### 7.44.3.12 `to_string()`

```
wstring PausePointList::to_string ( ) [virtual]
```

Returns a wide string representing the state of the synchronization points.

##### Returns

String summary of synchronization points.

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 132 of file IMSim/PausePointList.cpp.

References TrickHLA::SyncPntListBase::lock\_read\_only(), IMSim::PAUSE\_POINT\_STATE\_ACKNOWLEDGED, IMSim::PAUSE\_POINT\_STATE\_ERROR, IMSim::PAUSE\_POINT\_STATE\_EXIT, IMSim::PAUSE\_POINT\_STATE\_PENDING, IMSim::PAUSE\_POINT\_STATE\_RECONFIG, IMSim::PAUSE\_POINT\_STATE\_RESTART, IMSim::PAUSE\_POINT\_STATE\_RUN, IMSim::PAUSE\_POINT\_STATE\_UNKNOWN, state, TrickHLA::SyncPntListBase::sync\_point\_list, TrickHLA::SyncPnt::to\_string(), and TrickHLA::SyncPntListBase::unlock\_read\_only().

### 7.44.4 Friends And Related Function Documentation

#### 7.44.4.1 `init_attrTrickHLA__PausePointList`

```
void init_attrTrickHLA__PausePointList ( ) [friend]
```

#### 7.44.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 65 of file IMSim/PausePointList.hh.

### 7.44.5 Field Documentation

#### 7.44.5.1 `state`

```
PausePointStateEnum IMSim::PausePointList::state [private]
```

##### Units: –

State of the pause points.

Definition at line 108 of file IMSim/PausePointList.hh.

Referenced by `check_state()`, `clear_state()`, `clear_sync_pnt()`, `get_state()`, `should_exit()`, `should_freeze()`, `should_reconfig()`, `should_restart()`, `should_run()`, and `to_string()`.

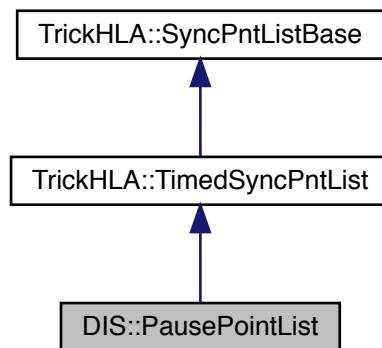
The documentation for this class was generated from the following files:

- [IMSim/PausePointList.hh](#)
- [IMSim/PausePointList.cpp](#)

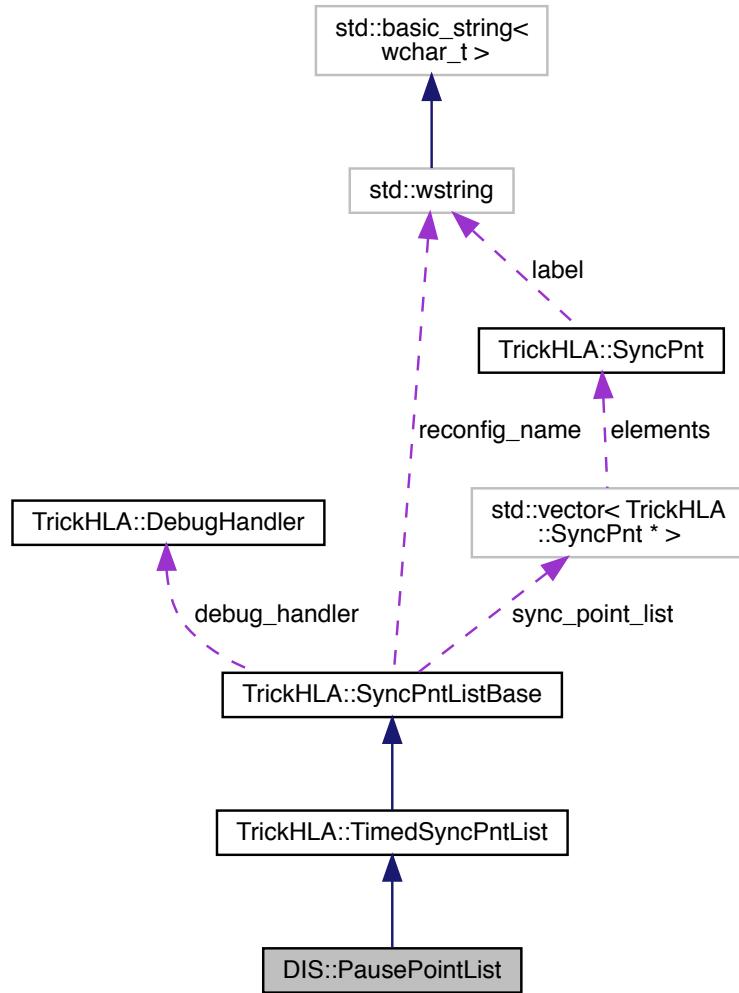
## 7.45 DIS::PausePointList Class Reference

```
#include <PausePointList.hh>
```

Inheritance diagram for DIS::PausePointList:



## Collaboration diagram for DIS::PausePointList:



## Public Member Functions

- [PausePointList \(\)](#)  
*Default constructor for the `TrickHLA` `PausePointList` class.*
  - [virtual ~PausePointList \(\)](#)  
*Destructor for the `TrickHLA` `PausePointList` class.*
  - [virtual bool clear\\_sync\\_pnt \(std::wstring const &label\)](#)  
*Clear the given synchronization point label.*
  - [void check\\_state \(\)](#)  
*Check the state of the sync-points to determine if the state needs to go to Freeze or Run.*
  - [PausePointStateEnum get\\_state \(\) const](#)
  - [void clear\\_state \(\)](#)

- `bool should_run () const`
- `bool should_freeze () const`
- `bool should_exit () const`
- `bool should_restart () const`
- `bool should_reconfig () const`
- `std::wstring to_string ()`

*Returns a wide string representing the state of the synchronization points.*

- `virtual void print_sync_pnts ()`

*Dumps synchronization point information to the screen.*

## Private Member Functions

- `PausePointList (const PausePointList &rhs)`  
*Copy constructor for `PausePointList` class.*
- `PausePointList & operator= (const PausePointList &rhs)`  
*Assignment operator for `PausePointList` class.*

## Private Attributes

- `PausePointStateEnum state`

*Units:* –

*State of the pause points.*

## Friends

- `class InputProcessor`
- `void init_attrTrickHLA_PausePointList ()`

## Additional Inherited Members

### 7.45.1 Detailed Description

Definition at line 58 of file DIS/PausePointList.hh.

### 7.45.2 Constructor & Destructor Documentation

#### 7.45.2.1 PausePointList() [1/2]

`PausePointList::PausePointList ( )`

Default constructor for the `TrickHLA PausePointList` class.

**Trick Job Class:** *initialization*

Definition at line 57 of file DIS/PausePointList.cpp.

#### 7.45.2.2 ~PausePointList()

`virtual DIS::PausePointList::~PausePointList ( ) [inline], [virtual]`

Destructor for the `TrickHLA PausePointList` class.

Definition at line 78 of file DIS/PausePointList.hh.

### 7.45.2.3 PausePointList() [2/2]

```
DIS::PausePointList::PausePointList (
    const PausePointList & rhs ) [private]
```

Copy constructor for [PausePointList](#) class.

This constructor is private to prevent inadvertent copies.

## 7.45.3 Member Function Documentation

### 7.45.3.1 check\_state()

```
void PausePointList::check_state ( )
```

Check the state of the sync-points to determine if the state needs to go to Freeze or Run.

Definition at line 94 of file [DIS/PausePointList.cpp](#).

References [TrickHLA::SyncPntListBase::lock\\_read\\_only\(\)](#), [DIS::PAUSE\\_POINT\\_STATE\\_EXIT](#), [DIS::PAUSE\\_POINT\\_STATE\\_FREEZE](#), [DIS::PAUSE\\_POINT\\_STATE\\_RECONFIG](#), [DIS::PAUSE\\_POINT\\_STATE\\_RESTART](#), [DIS::PAUSE\\_POINT\\_STATE\\_RUN](#), [DIS::PAUSE\\_POINT\\_STATE\\_UNKNOWN](#), [state](#), [TrickHLA::SYNC\\_PNT\\_STATE\\_ACHIEVED](#), [TrickHLA::SyncPntListBase::sync\\_point\\_list](#), and [TrickHLA::SyncPntListBase::unlock\\_read\\_only\(\)](#).

### 7.45.3.2 clear\_state()

```
void DIS::PausePointList::clear_state ( ) [inline]
```

Definition at line 91 of file [DIS/PausePointList.hh](#).

References [DIS::PAUSE\\_POINT\\_STATE\\_UNKNOWN](#), and [state](#).

### 7.45.3.3 clear\_sync\_pnt()

```
bool PausePointList::clear_sync_pnt (
    std::wstring const & label ) [virtual]
```

Clear the given synchronization point label.

#### Returns

True if synchronization point is cleared.

#### Parameters

<i>label</i>	The synchronization point label.
--------------	----------------------------------

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 62 of file [DIS/PausePointList.cpp](#).

References [TrickHLA::SyncPnt::get\\_label\(\)](#), [TrickHLA::SyncPnt::get\\_state\(\)](#), [TrickHLA::SyncPntListBase::lock\\_read\\_write\(\)](#), [DIS::PAUSE\\_POINT\\_STATE\\_EXIT](#), [DIS::PAUSE\\_POINT\\_STATE\\_RECONFIG](#), [DIS::PAUSE\\_POINT\\_STATE\\_RESTART](#), [TrickHLA::SyncPntListBase::reconfig\\_name](#), [state](#), [TrickHLA::SYNC\\_PNT\\_STATE\\_ACHIEVED](#), [TrickHLA::SyncPntListBase::sync\\_point\\_list](#), and [TrickHLA::SyncPntListBase::unlock\\_read\\_write\(\)](#).

### 7.45.3.4 get\_state()

```
PausePointStateEnum DIS::PausePointList::get_state ( ) const [inline]
```

Definition at line 90 of file [DIS/PausePointList.hh](#).

References state.

#### 7.45.3.5 operator=()

```
PausePointList& DIS::PausePointList::operator= (
    const PausePointList & rhs ) [private]
```

Assignment operator for [PausePointList](#) class.

This assignment operator is private to prevent inadvertent copies.

#### 7.45.3.6 print\_sync\_pnts()

```
void PausePointList::print_sync_pnts ( ) [virtual]
```

Dumps synchronization point information to the screen.

Reimplemented from [TrickHLA::TimedSyncPntList](#).

Definition at line 188 of file [DIS/PausePointList.cpp](#).

References [TrickHLA::TimedSyncPnt::get\\_time\(\)](#), [TrickHLA::Int64Time::getDoubleTime\(\)](#), [TrickHLA::SyncPntList::Base::lock\\_read\\_only\(\)](#), [TrickHLA::SyncPntListBase::sync\\_point\\_list](#), and [TrickHLA::SyncPntListBase::unlock\\_read\\_only\(\)](#).

#### 7.45.3.7 should\_exit()

```
bool DIS::PausePointList::should_exit ( ) const [inline]
```

Definition at line 95 of file [DIS/PausePointList.hh](#).

References [DIS::PAUSE\\_POINT\\_STATE\\_EXIT](#), and state.

#### 7.45.3.8 should\_freeze()

```
bool DIS::PausePointList::should_freeze ( ) const [inline]
```

Definition at line 94 of file [DIS/PausePointList.hh](#).

References [DIS::PAUSE\\_POINT\\_STATE\\_FREEZE](#), and state.

#### 7.45.3.9 should\_reconfig()

```
bool DIS::PausePointList::should_reconfig ( ) const [inline]
```

Definition at line 97 of file [DIS/PausePointList.hh](#).

References [DIS::PAUSE\\_POINT\\_STATE\\_RECONFIG](#), and state.

#### 7.45.3.10 should\_restart()

```
bool DIS::PausePointList::should_restart ( ) const [inline]
```

Definition at line 96 of file [DIS/PausePointList.hh](#).

References [DIS::PAUSE\\_POINT\\_STATE\\_RESTART](#), and state.

#### 7.45.3.11 should\_run()

```
bool DIS::PausePointList::should_run ( ) const [inline]
```

Definition at line 93 of file [DIS/PausePointList.hh](#).

References [DIS::PAUSE\\_POINT\\_STATE\\_RUN](#), and state.

### 7.45.3.12 to\_string()

```
wstring PausePointList::to_string ( ) [virtual]
Returns a wide string representing the state of the synchronization points.
```

#### Returns

String summary of synchronization points.

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 132 of file DIS/PausePointList.cpp.

References TrickHLA::SyncPntListBase::lock\_read\_only(), DIS::PAUSE\_POINT\_STATE\_ACKNOWLEDGED, DIS::PAUSE\_POINT\_STATE\_ERROR, DIS::PAUSE\_POINT\_STATE\_EXIT, DIS::PAUSE\_POINT\_STATE\_PENDING, DIS::PAUSE\_POINT\_STATE\_RECONFIG, DIS::PAUSE\_POINT\_STATE\_RESTART, DIS::PAUSE\_POINT\_STATE\_RUN, DIS::PAUSE\_POINT\_STATE\_UNKNOWN, state, TrickHLA::SyncPntListBase::sync\_point\_list, TrickHLA::SyncPnt::to\_string(), and TrickHLA::SyncPntListBase::unlock\_read\_only().

## 7.45.4 Friends And Related Function Documentation

### 7.45.4.1 init\_attrTrickHLA\_\_PausePointList

```
void init_attrTrickHLA__PausePointList ( ) [friend]
```

### 7.45.4.2 InputProcessor

```
friend class InputProcessor [friend]
Definition at line 65 of file DIS/PausePointList.hh.
```

## 7.45.5 Field Documentation

### 7.45.5.1 state

```
PausePointStateEnum DIS::PausePointList::state [private]
```

#### Units: -

State of the pause points.

Definition at line 108 of file DIS/PausePointList.hh.

Referenced by [check\\_state\(\)](#), [clear\\_state\(\)](#), [clear\\_sync\\_pnt\(\)](#), [get\\_state\(\)](#), [should\\_exit\(\)](#), [should\\_freeze\(\)](#), [should\\_reconfig\(\)](#), [should\\_restart\(\)](#), [should\\_run\(\)](#), and [to\\_string\(\)](#).

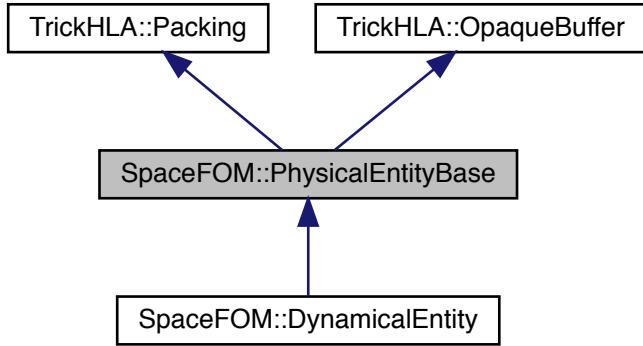
The documentation for this class was generated from the following files:

- [DIS/PausePointList.hh](#)
- [DIS/PausePointList.cpp](#)

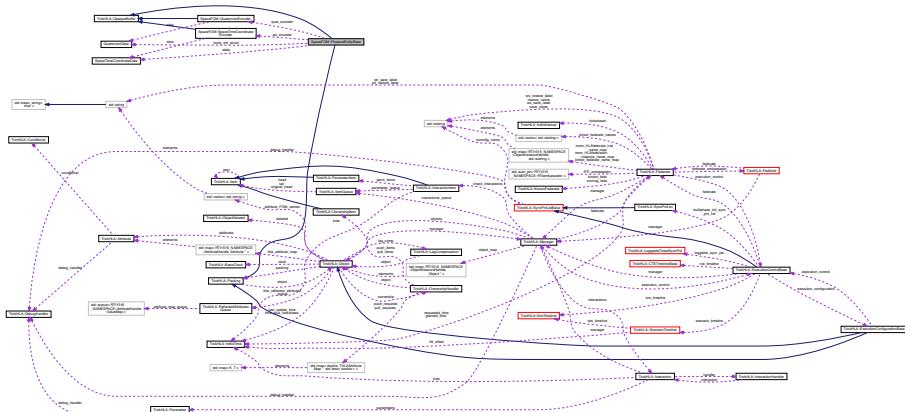
## 7.46 SpaceFOM::PhysicalEntityBase Class Reference

```
#include <PhysicalEntityBase.hh>
```

Inheritance diagram for SpaceFOM::PhysicalEntityBase:



Collaboration diagram for SpaceFOM::PhysicalEntityBase:



## Public Member Functions

- `PhysicalEntityBase ()`
- `virtual ~PhysicalEntityBase ()`
- `void initialize ()`
- `virtual void set_name (const char *name)`
- `virtual const char * get_name ()`
- `virtual void set_type (const char *type)`
- `virtual const char * get_type ()`
- `virtual void set_status (const char *status)`
- `virtual const char * get_status ()`
- `virtual void set_parent_ref_frame (const char *parent_ref_frame)`
- `virtual const char * get_parent_ref_frame ()`
- `double get_time ()`

- virtual void `pack ()`  
*Pack the data before being sent.*
- virtual void `unpack ()`  
*Unpack the received data. The default.*

## Protected Attributes

- `char * name`  
**Units:** –  
*Name of this entity(required).*
- `char * type`  
**Units:** –  
*True underlying type for this entity(optional).*
- `char * status`  
**Units:** –  
*Status string for this entity (optional).*
- `char * parent_ref_frame`  
**Units:** –  
*Name of this entity's parent frame(required).*
- `double accel [3]`  
**Units:**  $m/s^2$   
*Vehicle inertial acceleration (optional).*
- `double rot_accel [3]`  
**Units:**  $rad/s^2$   
*Angular body accels, body referenced (optional).*
- `double cm [3]`  
**Units:**  $m$   
*Center of mass location in vehicle structural frame (required).*
- `SpaceTimeCoordinateData & state`  
**Units:** –  
*SpaceTimeCoordinate from encoder (required).*
- `QuaternionData & body_wrt_struct`  
**Units:** –  
*Attitude quaternion for body frame w.r.t. structural frame.(optional)*
- `SpaceTimeCoordinateEncoder stc_encoder`  
**Units:** –  
*Entity state encoder.*
- `QuaternionEncoder quat_encoder`  
**Units:** –  
*Attitude quaternion encoder.*

## Private Member Functions

- `PhysicalEntityBase (const PhysicalEntityBase &)`
- `PhysicalEntityBase & operator= (const PhysicalEntityBase &)`

## Friends

- class `InputProcessor`
- void `init_attrSpaceFOM__PhysicalEntityBase ()`

## Additional Inherited Members

### 7.46.1 Detailed Description

Definition at line 63 of file PhysicalEntityBase.hh.

### 7.46.2 Constructor & Destructor Documentation

#### 7.46.2.1 PhysicalEntityBase() [1/2]

```
PhysicalEntityBase::PhysicalEntityBase ( )
```

**Trick Job Class:** *initialization*

Definition at line 59 of file PhysicalEntityBase.cpp.

References accel, cm, and rot\_accel.

#### 7.46.2.2 ~PhysicalEntityBase()

```
PhysicalEntityBase::~PhysicalEntityBase ( ) [virtual]
```

**Trick Job Class:** *shutdown*

Definition at line 75 of file PhysicalEntityBase.cpp.

References name, parent\_ref\_frame, status, trick\_MM, and type.

#### 7.46.2.3 PhysicalEntityBase() [2/2]

```
SpaceFOM::PhysicalEntityBase::PhysicalEntityBase ( 
    const PhysicalEntityBase & ) [private]
```

### 7.46.3 Member Function Documentation

#### 7.46.3.1 get\_name()

```
virtual const char* SpaceFOM::PhysicalEntityBase::get_name ( ) [inline], [virtual]
```

Definition at line 85 of file PhysicalEntityBase.hh.

References name.

#### 7.46.3.2 get\_parent\_ref\_frame()

```
virtual const char* SpaceFOM::PhysicalEntityBase::get_parent_ref_frame ( ) [inline], [virtual]
```

Definition at line 94 of file PhysicalEntityBase.hh.

References parent\_ref\_frame.

#### 7.46.3.3 get\_status()

```
virtual const char* SpaceFOM::PhysicalEntityBase::get_status ( ) [inline], [virtual]
```

Definition at line 91 of file PhysicalEntityBase.hh.

References status.

#### 7.46.3.4 `get_time()`

```
double SpaceFOM::PhysicalEntityBase::get_time ( ) [inline]
```

Definition at line 96 of file PhysicalEntityBase.hh.

References state, and SpaceTimeCoordinateData::time.

#### 7.46.3.5 `get_type()`

```
virtual const char* SpaceFOM::PhysicalEntityBase::get_type ( ) [inline], [virtual]
```

Definition at line 88 of file PhysicalEntityBase.hh.

References type.

#### 7.46.3.6 `initialize()`

```
void PhysicalEntityBase::initialize ( )
```

**Trick Job Class:** *initialization*

Definition at line 98 of file PhysicalEntityBase.cpp.

References name, parent\_ref\_frame, status, THLA\_ENDL, trick\_MM, and type.

#### 7.46.3.7 `operator=()`

```
PhysicalEntityBase& SpaceFOM::PhysicalEntityBase::operator= ( const PhysicalEntityBase & ) [private]
```

#### 7.46.3.8 `pack()`

```
void PhysicalEntityBase::pack ( ) [virtual]
```

Pack the data before being sent.

Implements [TrickHLA::Packing](#).

Reimplemented in [SpaceFOM::DynamicalEntity](#).

Definition at line 179 of file PhysicalEntityBase.cpp.

References SpaceFOM::SpaceTimeCoordinateEncoder::encode(), SpaceFOM::QuaternionEncoder::encode(), quat\_← encoder, and stc\_encoder.

Referenced by [SpaceFOM::DynamicalEntity::pack\(\)](#).

#### 7.46.3.9 `set_name()`

```
void PhysicalEntityBase::set_name ( const char * name ) [virtual]
```

Definition at line 142 of file PhysicalEntityBase.cpp.

References name, and trick\_MM.

#### 7.46.3.10 `set_parent_ref_frame()`

```
void PhysicalEntityBase::set_parent_ref_frame ( const char * parent_ref_frame ) [virtual]
```

Definition at line 169 of file PhysicalEntityBase.cpp.

References parent\_ref\_frame, and trick\_MM.

#### 7.46.3.11 set\_status()

```
void PhysicalEntityBase::set_status (
    const char * status ) [virtual]
```

Definition at line 160 of file PhysicalEntityBase.cpp.

References status, and trick\_MM.

#### 7.46.3.12 set\_type()

```
void PhysicalEntityBase::set_type (
    const char * type ) [virtual]
```

Definition at line 151 of file PhysicalEntityBase.cpp.

References trick\_MM, and type.

#### 7.46.3.13 unpack()

```
void PhysicalEntityBase::unpack ( ) [virtual]
```

Unpack the received data. The default.

Implements [TrickHLA::Packing](#).

Reimplemented in [SpaceFOM::DynamicalEntity](#).

Definition at line 185 of file PhysicalEntityBase.cpp.

References SpaceFOM::SpaceTimeCoordinateEncoder::decode(), SpaceFOM::QuaternionEncoder::decode(), quat\_← encoder, and stc\_encoder.

Referenced by [SpaceFOM::DynamicalEntity::unpack\(\)](#).

### 7.46.4 Friends And Related Function Documentation

#### 7.46.4.1 init\_attrSpaceFOM\_\_PhysicalEntityBase

```
void init_attrSpaceFOM__PhysicalEntityBase ( ) [friend]
```

#### 7.46.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 70 of file PhysicalEntityBase.hh.

### 7.46.5 Field Documentation

#### 7.46.5.1 accel

```
double SpaceFOM::PhysicalEntityBase::accel[3] [protected]
```

**Units:** m/s<sup>2</sup>

Vehicle inertial acceleration (optional).

Definition at line 106 of file PhysicalEntityBase.hh.

Referenced by [PhysicalEntityBase\(\)](#).

### 7.46.5.2 body\_wrt\_struct

`QuaternionData& SpaceFOM::PhysicalEntityBase::body_wrt_struct [protected]`

**Units:** –

Attitude quaternion for body frame w.r.t. structural frame.(optional)

Definition at line 111 of file PhysicalEntityBase.hh.

### 7.46.5.3 cm

`double SpaceFOM::PhysicalEntityBase::cm[3] [protected]`

**Units:** *m*

Center of mass location in vehicle structural frame (required).

Definition at line 108 of file PhysicalEntityBase.hh.

Referenced by PhysicalEntityBase().

### 7.46.5.4 name

`char* SpaceFOM::PhysicalEntityBase::name [protected]`

**Units:** –

Name of this entity(required).

Definition at line 102 of file PhysicalEntityBase.hh.

Referenced by get\_name(), initialize(), set\_name(), and ~PhysicalEntityBase().

### 7.46.5.5 parent\_ref\_frame

`char* SpaceFOM::PhysicalEntityBase::parent_ref_frame [protected]`

**Units:** –

Name of this entity's parent frame(required).

Definition at line 105 of file PhysicalEntityBase.hh.

Referenced by get\_parent\_ref\_frame(), initialize(), set\_parent\_ref\_frame(), and ~PhysicalEntityBase().

### 7.46.5.6 quat\_encoder

`QuaternionEncoder SpaceFOM::PhysicalEntityBase::quat_encoder [protected]`

**Units:** –

Attitude quaternion encoder.

Definition at line 115 of file PhysicalEntityBase.hh.

Referenced by pack(), and unpack().

### 7.46.5.7 rot\_accel

`double SpaceFOM::PhysicalEntityBase::rot_accel[3] [protected]`

**Units:** *rad/s2*

Angular body accels, body referenced (optional).

Definition at line 107 of file PhysicalEntityBase.hh.

Referenced by PhysicalEntityBase().

### 7.46.5.8 state

`SpaceTimeCoordinateData& SpaceFOM::PhysicalEntityBase::state [protected]`

**Units:** –

SpaceTimeCoordinate from encoder (required).  
 Definition at line 110 of file PhysicalEntityBase.hh.  
 Referenced by get\_time().

**7.46.5.9 status**

```
char* SpaceFOM::PhysicalEntityBase::status [protected]
```

**Units:** –

Status string for this entity (optional).  
 Definition at line 104 of file PhysicalEntityBase.hh.  
 Referenced by get\_status(), initialize(), set\_status(), and ~PhysicalEntityBase().

**7.46.5.10 stc\_encoder**

```
SpaceTimeCoordinateEncoder SpaceFOM::PhysicalEntityBase::stc_encoder [protected]
```

**Units:** –

Entity state encoder.  
 Definition at line 114 of file PhysicalEntityBase.hh.  
 Referenced by pack(), and unpack().

**7.46.5.11 type**

```
char* SpaceFOM::PhysicalEntityBase::type [protected]
```

**Units:** –

True underlying type for this entity(optional).  
 Definition at line 103 of file PhysicalEntityBase.hh.  
 Referenced by get\_type(), initialize(), set\_type(), and ~PhysicalEntityBase().  
 The documentation for this class was generated from the following files:

- [PhysicalEntityBase.hh](#)
- [PhysicalEntityBase.cpp](#)

**7.47 QuaternionData Struct Reference**

```
#include <QuaternionData.h>
```

**Data Fields**

- double **scalar**

**Units:** –

*Attitude quaternion scalar.*

- double **vector** [3]

**Units:** –

*Attitude quaternion vector.*

**7.47.1 Detailed Description**

Definition at line 37 of file QuaternionData.h.

## 7.47.2 Field Documentation

### 7.47.2.1 scalar

```
double QuaternionData::scalar
```

**Units:** –

Attitude quaternion scalar.

Definition at line 39 of file QuaternionData.h.

### 7.47.2.2 vector

```
double QuaternionData::vector[3]
```

**Units:** –

Attitude quaternion vector.

Definition at line 40 of file QuaternionData.h.

Referenced by SpaceFOM::QuaternionEncoder::QuaternionEncoder().

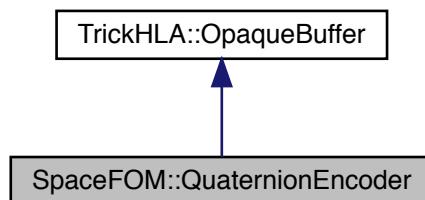
The documentation for this struct was generated from the following file:

- [QuaternionData.h](#)

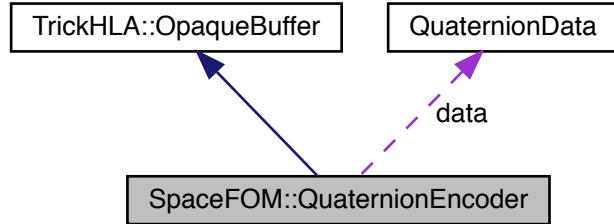
## 7.48 SpaceFOM::QuaternionEncoder Class Reference

```
#include <QuaternionEncoder.hh>
```

Inheritance diagram for SpaceFOM::QuaternionEncoder:



Collaboration diagram for SpaceFOM::QuaternionEncoder:



## Public Member Functions

- [QuaternionEncoder \(\)](#)  
*Default constructor for the [SpaceFOM QuaternionEncoder](#) class.*
- [void encode \(\)](#)  
*Encode the quaternion data for sending out.*
- [void decode \(\)](#)  
*Decode the quaternion space/time coordinate data.*
- [QuaternionData & get\\_data \(\)](#)  
*Get the quaternion data.*

## Protected Attributes

- [QuaternionData data](#)  
**Units:** –  
*Reference frame transmission data.*
- [rti1516e::HLAfloat64LE scalar\\_encoder](#)  
**Data I/O:** \*\*  
*Quaternion scalar encoder*
- [rti1516e::HLAfloat64LE vector \[3\]](#)  
**Data I/O:** \*\*  
*HLAfloat64LE quaternion vector*
- [rti1516e::HLAfixedArray vector\\_encoder](#)  
**Data I/O:** \*\*  
*Quaternion vector encoder*
- [rti1516e::HLAfixedRecord encoder](#)  
**Data I/O:** \*\*  
*Attitude quaternion encoder*

## Private Member Functions

- [QuaternionEncoder \(const QuaternionEncoder &\)](#)  
*Copy constructor for [QuaternionEncoder](#) class.*
- [QuaternionEncoder & operator= \(const QuaternionEncoder &\)](#)  
*Assignment operator for [QuaternionEncoder](#) class.*

## Friends

- class `InputProcessor`
- void `init_attrSpaceFOM__QuaternionEncoder()`

## Additional Inherited Members

### 7.48.1 Detailed Description

Definition at line 51 of file QuaternionEncoder.hh.

### 7.48.2 Constructor & Destructor Documentation

#### 7.48.2.1 QuaternionEncoder() [1/2]

`QuaternionEncoder::QuaternionEncoder()`  
Default constructor for the `SpaceFOM QuaternionEncoder` class.

**Trick Job Class:** *initialization*

Definition at line 47 of file QuaternionEncoder.cpp.

References `data`, `encoder`, `TrickHLA::OpaqueBuffer::ensure_buffer_capacity()`, `scalar_encoder`, `TrickHLA::OpaqueBuffer::set_byte_alignment()`, `QuaternionData::vector`, `vector_encoder`.

#### 7.48.2.2 QuaternionEncoder() [2/2]

`SpaceFOM::QuaternionEncoder::QuaternionEncoder(`  
    `const QuaternionEncoder & )` [private]  
Copy constructor for `QuaternionEncoder` class.

This constructor is private to prevent inadvertent copies.

### 7.48.3 Member Function Documentation

#### 7.48.3.1 decode()

`void QuaternionEncoder::decode()`  
Decode the quaternion space/time coordinate data.

**Trick Job Class:** *scheduled*

Definition at line 105 of file QuaternionEncoder.cpp.

References `TrickHLA::OpaqueBuffer::buffer`, `TrickHLA::OpaqueBuffer::capacity`, and `encoder`.  
Referenced by `SpaceFOM::PhysicalEntityBase::unpack()`.

#### 7.48.3.2 encode()

`void QuaternionEncoder::encode()`  
Encode the quaternion data for sending out.

**Trick Job Class:** *scheduled*

Definition at line 81 of file QuaternionEncoder.cpp.

References `TrickHLA::OpaqueBuffer::buffer`, `encoder`, and `TrickHLA::OpaqueBuffer::get_capacity()`.  
Referenced by `SpaceFOM::PhysicalEntityBase::pack()`.

### 7.48.3.3 `get_data()`

```
QuaternionData& SpaceFOM::QuaternionEncoder::get_data ( ) [inline]  
Get the quaternion data.
```

#### Returns

A reference to the [QuaternionData](#).

Definition at line 74 of file QuaternionEncoder.hh.  
References data.

### 7.48.3.4 `operator=()`

```
QuaternionEncoder& SpaceFOM::QuaternionEncoder::operator= (   
    const QuaternionEncoder & ) [private]
```

Assignment operator for [QuaternionEncoder](#) class.

This assignment operator is private to prevent inadvertent copies.

## 7.48.4 Friends And Related Function Documentation

### 7.48.4.1 `init_attrSpaceFOM__QuaternionEncoder`

```
void init_attrSpaceFOM__QuaternionEncoder ( ) [friend]
```

### 7.48.4.2 `InputProcessor`

```
friend class InputProcessor [friend]  
Definition at line 58 of file QuaternionEncoder.hh.
```

## 7.48.5 Field Documentation

### 7.48.5.1 `data`

```
QuaternionData SpaceFOM::QuaternionEncoder::data [protected]
```

#### Units: –

Reference frame transmission data.

Definition at line 77 of file QuaternionEncoder.hh.

Referenced by `get_data()`, and `QuaternionEncoder()`.

### 7.48.5.2 `encoder`

```
rti1516e::HLAfixedRecord SpaceFOM::QuaternionEncoder::encoder [protected]
```

#### Data I/O: \*\*

Attitude quaternion encoder

Definition at line 83 of file QuaternionEncoder.hh.

Referenced by `decode()`, `encode()`, and `QuaternionEncoder()`.

#### 7.48.5.3 scalar\_encoder

```
rti1516e::HLAfloat64LE SpaceFOM::QuaternionEncoder::scalar_encoder [protected]
```

**Data I/O:** \*\*

Quaternion scalar encoder

Definition at line 80 of file QuaternionEncoder.hh.

Referenced by QuaternionEncoder().

#### 7.48.5.4 vector

```
rti1516e::HLAfloat64LE SpaceFOM::QuaternionEncoder::vector[3] [protected]
```

**Data I/O:** \*\*

HLAfloat64LE quaternion vector

Definition at line 81 of file QuaternionEncoder.hh.

Referenced by QuaternionEncoder().

#### 7.48.5.5 vector\_encoder

```
rti1516e::HLAfixedArray SpaceFOM::QuaternionEncoder::vector_encoder [protected]
```

**Data I/O:** \*\*

Quaternion vector encoder

Definition at line 82 of file QuaternionEncoder.hh.

Referenced by QuaternionEncoder().

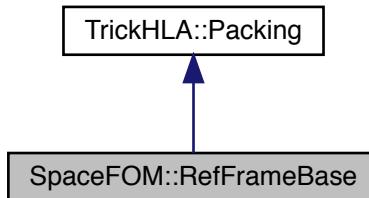
The documentation for this class was generated from the following files:

- [QuaternionEncoder.hh](#)
- [QuaternionEncoder.cpp](#)

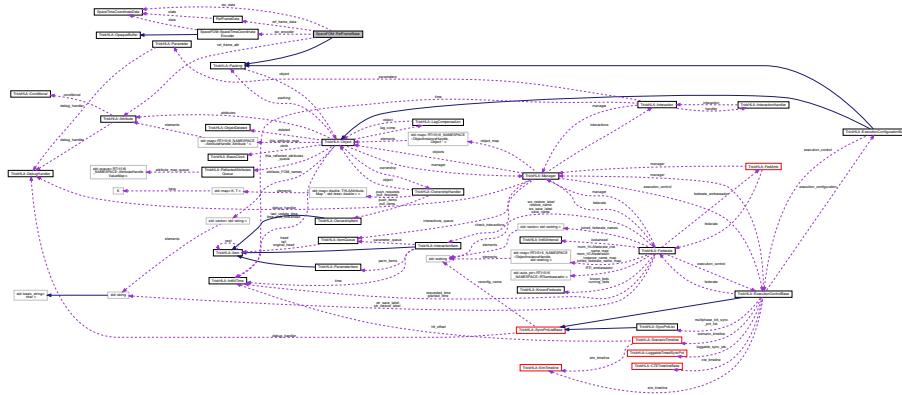
## 7.49 SpaceFOM::RefFrameBase Class Reference

```
#include <RefFrameBase.hh>
```

Inheritance diagram for SpaceFOM::RefFrameBase:



Collaboration diagram for SpaceFOM::RefFrameBase:



## Public Member Functions

- [RefFrameBase \(\)](#)  
*Default constructor for the SpaceFOM RefFrameBase class.*
- [virtual ~RefFrameBase \(\)](#)  
*Destructor for the SpaceFOM RefFrameBase class.*
- [virtual void default\\_data \(SpaceFOM::ExecutionControl \\*execution\\_control, TrickHLA::Object \\*object, const char \\*sim\\_obj\\_name, const char \\*ref\\_frame\\_obj\\_name, const char \\*ref\\_frame\\_name, bool publishes\)](#)  
*Sets up the attributes for a reference frame using default values.*
- [void initialize \(RefFrameData \\*ref\\_frame\\_data\\_ptr\)](#)  
*Set the reference to the reference frame data.*
- [virtual void initialize\\_callback \(TrickHLA::Object \\*obj\)](#)  
*Initialization callback as part of the TrickHLA::Packing functions.*
- [virtual void set\\_name \(const char \\*new\\_name\)](#)  
*Access function to set the HLA federation instance name for the reference frame.*
- [virtual const char \\* get\\_name \(\)](#)  
*Access function to get the HLA federation instance name for the reference frame.*
- [virtual void set\\_parent\\_name \(const char \\*name\)](#)  
*Access function to set the HLA federation instance name for the parent reference frame.*
- [virtual const char \\* get\\_parent\\_name \(\)](#)  
*Access function to get the HLA federation instance name for the parent reference frame.*
- [virtual void pack \(\)](#)  
*Called to pack the data before the data is sent to the RTI.*
- [virtual void unpack \(\)](#)  
*Called to unpack the data after data is received from the RTI.*
- [virtual TrickHLA::Object \\* get\\_object \(\)](#)

## Data Fields

- [bool debug](#)  
**Units:** –  
*Debug output flag.*

## Protected Attributes

- bool `initialized`

**Units:** –  
*Initialization indication flag.*
- `RefFrameData * ref_frame_data`

**Units:** –  
*Reference frame data.*
- `TrickHLA::Attribute * ref_frame_attr`

**Data I/O:** \*\*  
*Reference Frame Attribute.*
- `char * name`

**Units:** –  
*Name of the reference frame.*
- `char * parent_name`

**Units:** –  
*Name of this frames parent frame.*
- `SpaceTimeCoordinateEncoder stc_encoder`

**Units:** –  
*Encoder.*
- `SpaceTimeCoordinateData & stc_data`

**Units:** –  
*Encoder data.*

## Private Member Functions

- `RefFrameBase (const RefFrameBase &)`  
*Copy constructor for `RefFrameBase` class.*
- `RefFrameBase & operator= (const RefFrameBase &)`  
*Assignment operator for `RefFrameBase` class.*
- `char * allocate_input_string (const char *c_string)`  
*Uses Trick memory allocation routines to allocate a new string that is input file compliant.*
- `char * allocate_input_string (std::string cpp_string)`  
*Uses Trick memory allocation routines to allocate a new string that is input file compliant.*

## Friends

- class `InputProcessor`
- void `init_attrSpaceFOM__RefFrameBase ()`

### 7.49.1 Detailed Description

Definition at line 68 of file `RefFrameBase.hh`.

### 7.49.2 Constructor & Destructor Documentation

**7.49.2.1 RefFrameBase() [1/2]**

```
RefFrameBase::RefFrameBase ( )
Default constructor for the SpaceFOM RefFrameBase class.
Trick Job Class: initialization
Definition at line 64 of file RefFrameBase.cpp.
```

**7.49.2.2 ~RefFrameBase()**

```
RefFrameBase::~RefFrameBase ( ) [virtual]
Destructor for the SpaceFOM RefFrameBase class.
Trick Job Class: shutdown
Definition at line 79 of file RefFrameBase.cpp.
References name, parent_name, and trick_MM.
```

**7.49.2.3 RefFrameBase() [2/2]**

```
SpaceFOM::RefFrameBase::RefFrameBase (
    const RefFrameBase & ) [private]
Copy constructor for RefFrameBase class.
This constructor is private to prevent inadvertent copies.
```

**7.49.3 Member Function Documentation****7.49.3.1 allocate\_input\_string() [1/2]**

```
char * RefFrameBase::allocate_input_string (
    const char * c_string ) [private]
Uses Trick memory allocation routines to allocate a new string that is input file compliant.
Trick Job Class: default_data
Definition at line 377 of file RefFrameBase.cpp.
Referenced by default_data().
```

**7.49.3.2 allocate\_input\_string() [2/2]**

```
char * RefFrameBase::allocate_input_string (
    std::string cpp_string ) [private]
Uses Trick memory allocation routines to allocate a new string that is input file compliant.
Trick Job Class: default_data
Definition at line 391 of file RefFrameBase.cpp.
```

**7.49.3.3 default\_data()**

```
void RefFrameBase::default_data (
    SpaceFOM::ExecutionControl * execution_control,
    TrickHLA::Object * object,
    const char * sim_obj_name,
    const char * ref_frame_obj_name,
```

```
const char * ref_frame_name,
bool publishes ) [virtual]
```

Sets up the attributes for a reference frame using default values.

#### Parameters

<code>execution_control</code>	<code>SpaceFOM::ExecutionControl</code> associated with this federate.
<code>object</code>	<code>TrickHLA::Object</code> associated with this reference frame.
<code>sim_obj_name</code>	Name of SimObject containing this reference frame.
<code>ref_frame_obj_name</code>	Name of the ReferenceFrame object in the SimObject.
<code>ref_frame_name</code>	Name of the ReferenceFrame instance.
<code>publishes</code>	Does this federate publish this reference frame.

These can be overridden in the input file. **Trick Job Class:** `default_data`

Definition at line 95 of file RefFrameBase.cpp.

References `allocate_input_string()`, `TrickHLA::Object::attr_count`, `TrickHLA::CONFIG_CYCLIC`, `TrickHLA::CONFIG_INITIALIZE`, `TrickHLA::ENCODING_OPAQUE_DATA`, `TrickHLA::ENCODING_UNICODE_STRING`, `TrickHLA::Packing::object`, `SpaceFOM::ExecutionControl::root_ref_frame`, and `trick_MM`.

#### 7.49.3.4 `get_name()`

```
virtual const char* SpaceFOM::RefFrameBase::get_name ( ) [inline], [virtual]
```

Access function to get the HLA federation instance name for the reference frame.

#### Returns

Object instance name for this reference frame.

Definition at line 120 of file RefFrameBase.hh.

References `name`.

Referenced by `SpaceFOM::ExecutionControl::epoch_and_root_frame_discovery_process()`, and `SpaceFOM::ExecutionControl::receive_init_root_ref_frame()`.

#### 7.49.3.5 `get_object()`

```
virtual TrickHLA::Object* SpaceFOM::RefFrameBase::get_object ( ) [inline], [virtual]
```

Definition at line 139 of file RefFrameBase.hh.

References `TrickHLA::Packing::object`.

Referenced by `SpaceFOM::ExecutionControl::receive_root_ref_frame()`, and `SpaceFOM::ExecutionControl::send_root_ref_frame()`.

#### 7.49.3.6 `get_parent_name()`

```
virtual const char* SpaceFOM::RefFrameBase::get_parent_name ( ) [inline], [virtual]
```

Access function to get the HLA federation instance name for the parent reference frame.

#### Returns

Object instance name for the parent reference frame.

Definition at line 128 of file RefFrameBase.hh.

References `parent_name`.

### 7.49.3.7 initialize()

```
void RefFrameBase::initialize (
    RefFrameData * ref_frame_data_ptr )
```

Set the reference to the reference frame data.

#### Parameters

<code>ref_frame_data_ptr</code>	Pointer to the <a href="#">RefFrameData</a> instance.
---------------------------------	---

#### Trick Job Class: *initialization*

Definition at line 160 of file RefFrameBase.cpp.

References initialized, name, parent\_name, ref\_frame\_data, THLA\_ENDL, and trick\_MM.

### 7.49.3.8 initialize\_callback()

```
void RefFrameBase::initialize_callback (
    TrickHLA::Object * obj ) [virtual]
```

Initialization callback as part of the [TrickHLA::Packing](#) functions.

#### Parameters

<code>obj</code>	Object associated with this packing class.
------------------	--

From the [TrickHLA::Packing](#) class. We override this function so that we can initialize references to the [TrickHLA::Attribute](#)'s that are used in the unpack function to handle attribute ownership and different attribute data rates.

Use the initialize callback function as a way to setup [TrickHLA::Attribute](#) references which are use to determine ownership or if data for an attribute was received.

#### Trick Job Class: *initialization*

Reimplemented from [TrickHLA::Packing](#).

Definition at line 213 of file RefFrameBase.cpp.

References TrickHLA::Packing::get\_attribute\_and\_validate(), TrickHLA::Packing::initialize\_callback(), and ref\_frame\_← attr.

### 7.49.3.9 operator=( )

```
RefFrameBase& SpaceFOM::RefFrameBase::operator= (
    const RefFrameBase & ) [private]
```

Assignment operator for [RefFrameBase](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.49.3.10 pack()

```
void RefFrameBase::pack ( ) [virtual]
```

Called to pack the data before the data is sent to the RTI.

Implements [TrickHLA::Packing](#).

Definition at line 249 of file RefFrameBase.cpp.

References SpaceTimeCoordinateData::ang\_vel, debug, SpaceFOM::SpaceTimeCoordinateEncoder::encode(), TrickHLA::Packing::get\_scenario\_time(), initialized, name, parent\_name, SpaceTimeCoordinateData::pos, Space← TimeCoordinateData::quat\_scalar, SpaceTimeCoordinateData::quat\_vector, ref\_frame\_data, RefFrameData::state, stc\_data, stc\_encoder, SpaceTimeCoordinateData::time, and SpaceTimeCoordinateData::vel.

### 7.49.3.11 set\_name()

```
void RefFrameBase::set_name (
    const char * new_name ) [virtual]
```

Access function to set the HLA federation instance name for the reference frame.

#### Parameters

<i>new_name</i>	Object instance name for this reference frame.
-----------------	--

**Trick Job Class:** *initialization*

Definition at line 228 of file RefFrameBase.cpp.

References name, and trick\_MM.

### 7.49.3.12 set\_parent\_name()

```
void RefFrameBase::set_parent_name (
    const char * name ) [virtual]
```

Access function to set the HLA federation instance name for the parent reference frame.

#### Parameters

<i>name</i>	Object instance name for the parent reference frame.
-------------	--

**Trick Job Class:** *initialization*

Definition at line 240 of file RefFrameBase.cpp.

References parent\_name, and trick\_MM.

### 7.49.3.13 unpack()

```
void RefFrameBase::unpack ( ) [virtual]
```

Called to unpack the data after data is received from the RTI.

Implements [TrickHLA::Packing](#).

Definition at line 301 of file RefFrameBase.cpp.

References SpaceTimeCoordinateData::ang\_vel, debug, SpaceFOM::SpaceTimeCoordinateEncoder::decode(), initialized, TrickHLA::Attribute::is\_received(), RefFrameData::name, name, RefFrameData::parent\_name, parent\_name, SpaceTimeCoordinateData::pos, SpaceTimeCoordinateData::quat\_scalar, SpaceTimeCoordinateData::quat\_vector, ref\_frame\_attr, ref\_frame\_data, RefFrameData::state, stc\_data, stc\_encoder, SpaceTimeCoordinateData::time, and SpaceTimeCoordinateData::vel.

## 7.49.4 Friends And Related Function Documentation

### 7.49.4.1 init\_attrSpaceFOM\_RefFrameBase

```
void init_attrSpaceFOM_RefFrameBase ( ) [friend]
```

### 7.49.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 76 of file RefFrameBase.hh.

## 7.49.5 Field Documentation

### 7.49.5.1 debug

```
bool SpaceFOM::RefFrameBase::debug
```

**Units:** –

Debug output flag.

Definition at line 142 of file RefFrameBase.hh.

Referenced by pack(), and unpack().

### 7.49.5.2 initialized

```
bool SpaceFOM::RefFrameBase::initialized [protected]
```

**Units:** –

Initialization indication flag.

Definition at line 145 of file RefFrameBase.hh.

Referenced by initialize(), pack(), and unpack().

### 7.49.5.3 name

```
char* SpaceFOM::RefFrameBase::name [protected]
```

**Units:** –

Name of the reference frame.

Definition at line 149 of file RefFrameBase.hh.

Referenced by get\_name(), initialize(), pack(), set\_name(), unpack(), and ~RefFrameBase().

### 7.49.5.4 parent\_name

```
char* SpaceFOM::RefFrameBase::parent_name [protected]
```

**Units:** –

Name of this frames parent frame.

Definition at line 150 of file RefFrameBase.hh.

Referenced by get\_parent\_name(), initialize(), pack(), set\_parent\_name(), unpack(), and ~RefFrameBase().

### 7.49.5.5 ref\_frame\_attr

```
TrickHLA::Attribute* SpaceFOM::RefFrameBase::ref_frame_attr [protected]
```

**Data I/O:** \*\*

Reference Frame Attribute.

Definition at line 147 of file RefFrameBase.hh.

Referenced by initialize\_callback(), and unpack().

### 7.49.5.6 ref\_frame\_data

```
RefFrameData* SpaceFOM::RefFrameBase::ref_frame_data [protected]
```

**Units:** –

Reference frame data.

Definition at line 146 of file RefFrameBase.hh.

Referenced by initialize(), pack(), and unpack().

#### 7.49.5.7 stc\_data

`SpaceTimeCoordinateData& SpaceFOM::RefFrameBase::stc_data [protected]`

**Units:** –

Encoder data.

Definition at line 154 of file RefFrameBase.hh.

Referenced by pack(), and unpack().

#### 7.49.5.8 stc\_encoder

`SpaceTimeCoordinateEncoder SpaceFOM::RefFrameBase::stc_encoder [protected]`

**Units:** –

Encoder.

Definition at line 153 of file RefFrameBase.hh.

Referenced by pack(), and unpack().

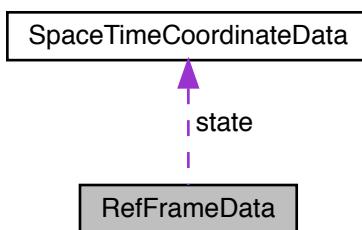
The documentation for this class was generated from the following files:

- [RefFrameBase.hh](#)
- [RefFrameBase.cpp](#)

## 7.50 RefFrameData Struct Reference

#include <RefFrameData.h>

Collaboration diagram for RefFrameData:



## Data Fields

- `char * name`

**Units:** –  
*Name of the reference frame.*
- `char * parent_name`

**Units:** –  
*Name of this frames parent frame.*
- `SpaceTimeCoordinateData state`

**Units:** –  
Space time coordinate state.

### 7.50.1 Detailed Description

Definition at line 39 of file RefFrameData.h.

### 7.50.2 Field Documentation

#### 7.50.2.1 name

char\* RefFrameData::name

**Units:** –

Name of the reference frame.

Definition at line 41 of file RefFrameData.h.

Referenced by SpaceFOM::RefFrameBase::unpack().

#### 7.50.2.2 parent\_name

char\* RefFrameData::parent\_name

**Units:** –

Name of this frames parent frame.

Definition at line 42 of file RefFrameData.h.

Referenced by SpaceFOM::RefFrameBase::unpack().

#### 7.50.2.3 state

SpaceTimeCoordinateData RefFrameData::state

**Units:** –

Space time coordinate state.

Definition at line 44 of file RefFrameData.h.

Referenced by SpaceFOM::RefFrameBase::pack(), and SpaceFOM::RefFrameBase::unpack().

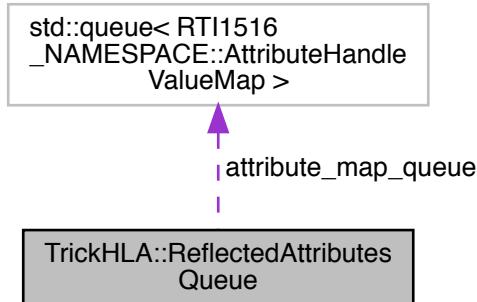
The documentation for this struct was generated from the following file:

- [RefFrameData.h](#)

## 7.51 TrickHLA::ReflectedAttributesQueue Class Reference

```
#include <ReflectedAttributesQueue.hh>
```

Collaboration diagram for TrickHLA::ReflectedAttributesQueue:



## Public Member Functions

- [ReflectedAttributesQueue \(\)](#)  
*Default constructor for the `TrickHLA ReflectedAttributesQueue` class.*
- [virtual ~ReflectedAttributesQueue \(\)](#)  
*Destructor for the `TrickHLA ReflectedAttributesQueue` class.*
- [bool empty \(\)](#)  
*Determine if the queue is empty.*
- [void push \(const RTI1516\\_NAMESPACE::AttributeHandleValueMap &theAttributes\)](#)  
*Push the attributes onto the queue.*
- [void pop \(\)](#)  
*Pop the front value off the queue and the destructor for the value will be called.*
- [const RTI1516\\_NAMESPACE::AttributeHandleValueMap & front \(\)](#)  
*Get the first/oldest item in the queue.*
- [void clear \(\)](#)  
*Clear the queue of all values.*

## Data Fields

- [pthread\\_mutex\\_t queue\\_mutex](#)  
**Data I/O:** \*\*  
*Mutex to lock thread over critical code sections.*
- [HLAAttributeMapQueue attribute\\_map\\_queue](#)  
**Data I/O:** \*\*  
*Queue of AttributeHandleValueMap from attribute reflections.*

## Private Member Functions

- [ReflectedAttributesQueue \(const ReflectedAttributesQueue &rhs\)](#)  
*Copy constructor for `ReflectedAttributesQueue` class.*
- [ReflectedAttributesQueue & operator= \(const ReflectedAttributesQueue &rhs\)](#)  
*Assignment operator for `ReflectedAttributesQueue` class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_ReflectedAttributesQueue \(\)](#)

### 7.51.1 Detailed Description

Definition at line 48 of file ReflectedAttributesQueue.hh.

### 7.51.2 Constructor & Destructor Documentation

#### 7.51.2.1 ReflectedAttributesQueue() [1/2]

`ReflectedAttributesQueue::ReflectedAttributesQueue ( )`  
Default constructor for the [TrickHLA ReflectedAttributesQueue](#) class.

**Trick Job Class:** *initialization*

Definition at line 40 of file ReflectedAttributesQueue.cpp.

References queue\_mutex.

#### 7.51.2.2 ~ReflectedAttributesQueue()

`ReflectedAttributesQueue::~ReflectedAttributesQueue ( ) [virtual]`  
Destructor for the [TrickHLA ReflectedAttributesQueue](#) class.  
Frees the Trick allocated memory. **Trick Job Class:** *shutdown*  
Definition at line 51 of file ReflectedAttributesQueue.cpp.  
References attribute\_map\_queue, and queue\_mutex.

#### 7.51.2.3 ReflectedAttributesQueue() [2/2]

`TrickHLA::ReflectedAttributesQueue::ReflectedAttributesQueue (`  
    `const ReflectedAttributesQueue & rhs ) [private]`  
Copy constructor for [ReflectedAttributesQueue](#) class.  
This constructor is private to prevent inadvertent copies.

### 7.51.3 Member Function Documentation

#### 7.51.3.1 clear()

`void ReflectedAttributesQueue::clear ( )`  
Clear the queue of all values.  
Definition at line 93 of file ReflectedAttributesQueue.cpp.  
References attribute\_map\_queue, and queue\_mutex.

#### 7.51.3.2 empty()

`bool ReflectedAttributesQueue::empty ( )`  
Determine if the queue is empty.

**Returns**

True if queue is empty, False otherwise.

Definition at line 62 of file ReflectedAttributesQueue.cpp.

References attribute\_map\_queue, and queue\_mutex.

Referenced by TrickHLA::Object::is\_changed().

**7.51.3.3 front()**

```
const AttributeHandleValueMap & ReflectedAttributesQueue::front ( )
```

Get the first/oldest item in the queue.

**Returns**

The first/oldest item in the queue.

Definition at line 85 of file ReflectedAttributesQueue.cpp.

References attribute\_map\_queue, and queue\_mutex.

Referenced by TrickHLA::Object::is\_changed().

**7.51.3.4 operator=()**

```
ReflectedAttributesQueue& TrickHLA::ReflectedAttributesQueue::operator= ( 
    const ReflectedAttributesQueue & rhs )  [private]
```

Assignment operator for ReflectedAttributesQueue class.

This assignment operator is private to prevent inadvertent copies.

**7.51.3.5 pop()**

```
void ReflectedAttributesQueue::pop ( )
```

Pop the front value off the queue and the destructor for the value will be called.

Definition at line 78 of file ReflectedAttributesQueue.cpp.

References attribute\_map\_queue, and queue\_mutex.

Referenced by TrickHLA::Object::is\_changed().

**7.51.3.6 push()**

```
void ReflectedAttributesQueue::push (
```

```
    const RTI1516_NAMESPACE::AttributeHandleValueMap & theAttributes )
```

Push the attributes onto the queue.

**Parameters**

<i>theAttributes</i>	The reflected attributes.
----------------------	---------------------------

Definition at line 70 of file ReflectedAttributesQueue.cpp.

References attribute\_map\_queue, and queue\_mutex.

**7.51.4 Friends And Related Function Documentation**

#### 7.51.4.1 `init_attrTrickHLA__ReflectedAttributesQueue`

```
void init_attrTrickHLA__ReflectedAttributesQueue ( ) [friend]
```

#### 7.51.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
Definition at line 55 of file ReflectedAttributesQueue.hh.
```

### 7.51.5 Field Documentation

#### 7.51.5.1 `attribute_map_queue`

`HLAAttributeMapQueue` `TrickHLA::ReflectedAttributesQueue::attribute_map_queue`

##### **Data I/O: \*\***

Queue of AttributeHandleValueMap from attribute reflections.

Definition at line 68 of file ReflectedAttributesQueue.hh.

Referenced by `clear()`, `empty()`, `front()`, `pop()`, `push()`, and `~ReflectedAttributesQueue()`.

#### 7.51.5.2 `queue_mutex`

`pthread_mutex_t` `TrickHLA::ReflectedAttributesQueue::queue_mutex`

##### **Data I/O: \*\***

Mutex to lock thread over critical code sections.

Definition at line 61 of file ReflectedAttributesQueue.hh.

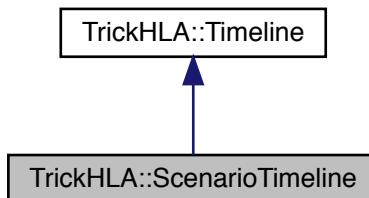
Referenced by `clear()`, `empty()`, `front()`, `pop()`, `push()`, `ReflectedAttributesQueue()`, and `~ReflectedAttributesQueue()`.

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

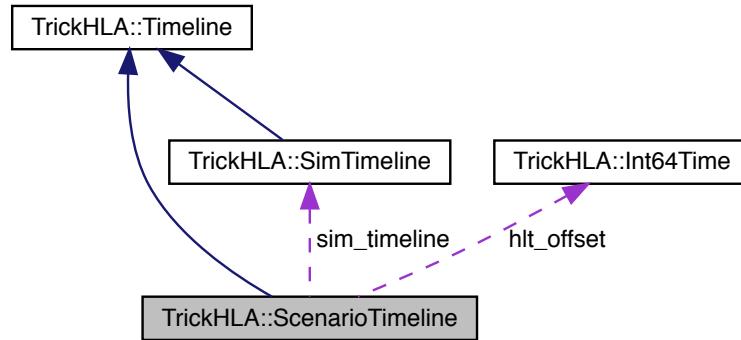
- [ReflectedAttributesQueue.hh](#)
- [ReflectedAttributesQueue.cpp](#)

## 7.52 TrickHLA::ScenarioTimeline Class Reference

```
#include <ScenarioTimeline.hh>
Inheritance diagram for TrickHLA::ScenarioTimeline:
```



Collaboration diagram for TrickHLA::ScenarioTimeline:



## Public Member Functions

- `ScenarioTimeline (SimTimeline &sim_timeline, double tt0=0.0, double st0=0.0)`  
*Initialization constructor for the `TrickHLA ScenarioTimeline` class.*
- `virtual ~ScenarioTimeline ()`  
*Destructor for the `TrickHLA ScenarioTimeline` class.*
- `virtual double get_time ()`  
*Get the current scenario time.*
- `virtual double compute_simulation_time (double scenario_time)`  
*Compute a simulation time from a given scenario time.*
- `virtual double time_from_simulation_time (double sim_time)`  
*Compute a scenario time from a given simulation time.*
- `virtual Int64Time compute_HLT (double scenario_time)`  
*Compute a HLA Logical Time (HLT) from a given scenario time.*
- `virtual double time_from_HLT (Int64Time hlt)`  
*Compute a scenario time from and given HLA Logical Time (HLT).*
- `virtual double get_sim_offset ()`  
*Get the offset of the simulation time line from the scenario timeline epoch.*
- `virtual void set_sim_offset (double st0)`  
*Set the offset of the simulation time line from the scenario timeline epoch.*
- `virtual Int64Time get_HLT_offset ()`  
*Get the offset of the HLA Logical Time (HLT) timeline from the scenario timeline.*
- `virtual void set_HLT_offset (Int64Time hlt0)`  
*Set the offset of the HLA Logical Time (HLT) timeline from the scenario timeline.*

## Protected Attributes

- `SimTimeline & sim_timeline`  
**Data I/O:** \*\*  
*Reference to simulation timeline.*
- `double sim_offset`
- `Int64Time hlt_offset`

## Private Member Functions

- `ScenarioTimeline (const ScenarioTimeline &rhs)`  
`Copy constructor for ScenarioTimeline class.`
- `ScenarioTimeline & operator= (const ScenarioTimeline &rhs)`  
`Assignment operator for ScenarioTimeline class.`

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA_ScenarioTimeline ()`

### 7.52.1 Detailed Description

Definition at line 57 of file ScenarioTimeline.hh.

### 7.52.2 Constructor & Destructor Documentation

#### 7.52.2.1 ScenarioTimeline() [1/2]

```
ScenarioTimeline::ScenarioTimeline (
    SimTimeline & sim_timeline,
    double tt0 = 0.0,
    double st0 = 0.0 ) [explicit]
```

Initialization constructor for the `TrickHLA ScenarioTimeline` class.

#### Parameters

<code>sim_timeline</code>	Simulation time line.
<code>tt0</code>	Scenario time epoch in TT seconds TJD format.
<code>st0</code>	Simulation starting time offset from epoch.

#### Trick Job Class: initialization

Definition at line 43 of file ScenarioTimeline.cpp.

#### 7.52.2.2 ~ScenarioTimeline()

```
ScenarioTimeline::~ScenarioTimeline () [virtual]
```

Destructor for the `TrickHLA ScenarioTimeline` class.

#### Trick Job Class: shutdown

Definition at line 58 of file ScenarioTimeline.cpp.

#### 7.52.2.3 ScenarioTimeline() [2/2]

```
TrickHLA::ScenarioTimeline::ScenarioTimeline (
    const ScenarioTimeline & rhs ) [private]
```

Copy constructor for `ScenarioTimeline` class.

This constructor is private to prevent inadvertent copies.

### 7.52.3 Member Function Documentation

#### 7.52.3.1 compute\_HLT()

```
Int64Time ScenarioTimeline::compute_HLT (
    double scenario_time ) [virtual]
```

Compute a HLA Logical Time (HLT) from a given scenario time.

##### Returns

HLT in microseconds.

##### Parameters

<i>scenario_time</i>	Desired scenario time.
----------------------	------------------------

Definition at line 75 of file ScenarioTimeline.cpp.

References TrickHLA::Timeline::epoch, and hlt\_offset.

#### 7.52.3.2 compute\_simulation\_time()

```
double ScenarioTimeline::compute_simulation_time (
    double scenario_time ) [virtual]
```

Compute a simulation time from a given scenario time.

##### Returns

Simulation time in seconds.

##### Parameters

<i>scenario_time</i>	Desired scenario time in seconds.
----------------------	-----------------------------------

Definition at line 63 of file ScenarioTimeline.cpp.

References TrickHLA::Timeline::epoch, and sim\_offset.

Referenced by DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::set\_next\_execution\_control\_mode(), SpaceFOM::ExecutionControl::set\_next\_execution\_control\_mode(), DIS::ExecutionControl::set\_next\_execution\_control\_mode(), and IMSim::ExecutionControl::set\_next\_execution\_control\_mode().

#### 7.52.3.3 get\_HLT\_offset()

```
virtual Int64Time TrickHLA::ScenarioTimeline::get_HLT_offset ( ) [inline], [virtual]
```

Get the offset of the HLA Logical Time (HLT) timeline from the scenario timeline.

##### Returns

Offset in microseconds.

Definition at line 128 of file ScenarioTimeline.hh.

References hlt\_offset.

#### 7.52.3.4 `get_sim_offset()`

```
virtual double TrickHLA::ScenarioTimeline::get_sim_offset ( ) [inline], [virtual]
Get the offset of the simulation time line from the scenario timeline epoch.
```

##### Returns

Offset time in seconds.

Definition at line 118 of file ScenarioTimeline.hh.

References sim\_offset.

Referenced by SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), and IMSim::ExecutionControl::process\_mode\_transition\_request().

#### 7.52.3.5 `get_time()`

```
double ScenarioTimeline::get_time ( ) [virtual]
```

Get the current scenario time.

##### Returns

Current scenario time in seconds.

Implements [TrickHLA::Timeline](#).

Definition at line 88 of file ScenarioTimeline.cpp.

References TrickHLA::Timeline::epoch, TrickHLA::SimTimeline::get\_time(), sim\_offset, and sim\_timeline.

Referenced by TrickHLA::ExecutionControlBase::get\_scenario\_time(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), IMSim::ExecutionConfiguration::pack(), SpaceFOM::ExecutionConfiguration::pack(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), IMSim::ExecutionControl::process\_mode\_transition\_request(), and SpaceFOM::ExecutionConfiguration::unpack().

#### 7.52.3.6 `operator=( )`

```
ScenarioTimeline& TrickHLA::ScenarioTimeline::operator= (
    const ScenarioTimeline & rhs ) [private]
```

Assignment operator for `ScenarioTimeline` class.

This assignment operator is private to prevent inadvertent copies.

#### 7.52.3.7 `set_HTL_offset()`

```
virtual void TrickHLA::ScenarioTimeline::set_HTL_offset (
    Int64Time hlt0 ) [inline], [virtual]
```

Set the offset of the HLA Logical Time (HLT) timeline from the scenario timeline.

##### Parameters

<code>hlt0</code>	Desired offset in microseconds.
-------------------	---------------------------------

Definition at line 133 of file ScenarioTimeline.hh.

References hlt\_offset.

#### 7.52.3.8 set\_sim\_offset()

```
virtual void TrickHLA::ScenarioTimeline::set_sim_offset (
    double st0 ) [inline], [virtual]
```

Set the offset of the simulation time line from the scenario timeline epoch.

##### Parameters

<i>st0</i>	The offset time in seconds.
------------	-----------------------------

Definition at line 123 of file ScenarioTimeline.hh.

References sim\_offset.

Referenced by SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes().

#### 7.52.3.9 time\_from\_HLT()

```
double ScenarioTimeline::time_from_HLT (
    Int64Time hlt ) [virtual]
```

Compute a scenario time from and given HLA Logical Time (HLT).

##### Returns

Scenario time in seconds.

##### Parameters

<i>hlt</i>	Desired HLT in microseconds.
------------	------------------------------

Definition at line 82 of file ScenarioTimeline.cpp.

References TrickHLA::Timeline::epoch, TrickHLA::Int64Time::getDoubleTime(), and hlt\_offset.

#### 7.52.3.10 time\_from\_simulation\_time()

```
double ScenarioTimeline::time_from_simulation_time (
    double sim_time ) [virtual]
```

Compute a scenario time from a given simulation time.

##### Returns

Scenario time in seconds.

##### Parameters

<i>sim_time</i>	Desired simulation time.
-----------------	--------------------------

Definition at line 69 of file ScenarioTimeline.cpp.

References TrickHLA::Timeline::epoch, and sim\_offset.

## 7.52.4 Friends And Related Function Documentation

### 7.52.4.1 init\_attrTrickHLA\_\_ScenarioTimeline

```
void init_attrTrickHLA__ScenarioTimeline ( ) [friend]
```

### 7.52.4.2 InputProcessor

```
friend class InputProcessor [friend]  
Definition at line 64 of file ScenarioTimeline.hh.
```

## 7.52.5 Field Documentation

### 7.52.5.1 hlt\_offset

```
Int64Time TrickHLA::ScenarioTimeline::hlt_offset [protected]
```

**Units:** us

The offset of the HLA Logical Time (HLT) timeline from the scenario timeline epoch.

Definition at line 145 of file ScenarioTimeline.hh.

Referenced by compute\_HLT(), get\_HLT\_offset(), set\_HLT\_offset(), and time\_from\_HLT().

### 7.52.5.2 sim\_offset

```
double TrickHLA::ScenarioTimeline::sim_offset [protected]
```

**Units:** s

The offset of the simulation timeline from the scenario timeline epoch. For early joiners, this number will usually be 0.0. However, for late joiners, this will provide the starting offset of the simulation with respect to the original start of the federation execution.

Definition at line 138 of file ScenarioTimeline.hh.

Referenced by compute\_simulation\_time(), get\_sim\_offset(), get\_time(), set\_sim\_offset(), and time\_from\_simulation\_time().

### 7.52.5.3 sim\_timeline

```
SimTimeline& TrickHLA::ScenarioTimeline::sim_timeline [protected]
```

**Data I/O:** \*\*

Reference to simulation timeline.

Definition at line 136 of file ScenarioTimeline.hh.

Referenced by get\_time().

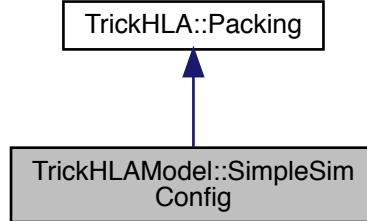
The documentation for this class was generated from the following files:

- [ScenarioTimeline.hh](#)
- [ScenarioTimeline.cpp](#)

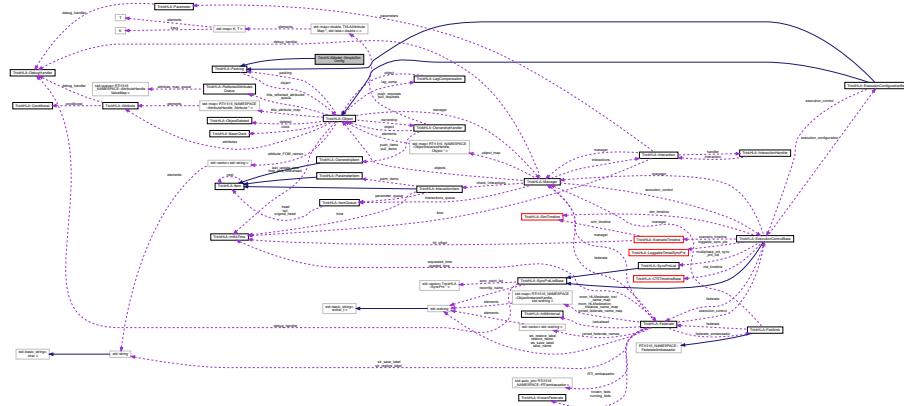
## 7.53 TrickHLAModel::SimpleSimConfig Class Reference

```
#include <SimpleSimConfig.hh>
```

Inheritance diagram for TrickHLAModel::SimpleSimConfig:



Collaboration diagram for TrickHLAModel::SimpleSimConfig:



## Public Member Functions

- **SimpleSimConfig ()**  
*Default constructor for the [TrickHLAModel SimpleSimConfig](#) class.*
- **virtual ~SimpleSimConfig ()**  
*Destructor for the [TrickHLAModel SimpleSimConfig](#) class.*
- **void initialize (int known\_feds\_count, TrickHLA::KnownFederate \*known\_feds)**  
*Initialize the simulation configuration and build the list of federates based on the known federates.*
- **virtual void pack ()**  
*Called to pack the data before the data is sent to the RTI.*
- **virtual void unpack ()**  
*Called to unpack the data after data is received from the RTI.*

## Data Fields

- **double run\_duration**  
**Units:** s  
*The run duration of the simulation.*

- long long `run_duration_microsec`  
**Units:** *us*  
*The run duration in microseconds.*
- int `num_federates`  
**Units:** –  
*Number of required federates.*
- char \* `required_federates`  
**Units:** –  
*Comma-separated list of required federates.*
- char \* `owner`  
**Units:** –  
*Federates name publishing the object.*

## Private Member Functions

- `SimpleSimConfig (const SimpleSimConfig &rhs)`  
*Copy constructor for `SimpleSimConfig` class.*
- `SimpleSimConfig & operator= (const SimpleSimConfig &rhs)`  
*Assignment operator for `SimpleSimConfig` class.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLAModel__SimpleSimConfig ()`

## Additional Inherited Members

### 7.53.1 Detailed Description

Definition at line 43 of file SimpleSimConfig.hh.

### 7.53.2 Constructor & Destructor Documentation

#### 7.53.2.1 `SimpleSimConfig()` [1/2]

`SimpleSimConfig::SimpleSimConfig ( )`  
Default constructor for the `TrickHLAModel SimpleSimConfig` class.  
**Trick Job Class:** *initialization*  
Definition at line 59 of file SimpleSimConfig.cpp.

#### 7.53.2.2 `~SimpleSimConfig()`

`SimpleSimConfig::~SimpleSimConfig ( ) [virtual]`  
Destructor for the `TrickHLAModel SimpleSimConfig` class.  
**Trick Job Class:** *shutdown*  
Definition at line 72 of file SimpleSimConfig.cpp.  
References `owner`, and `required_federates`.

### 7.53.2.3 SimpleSimConfig() [2/2]

```
TrickHLAModel::SimpleSimConfig::SimpleSimConfig (
    const SimpleSimConfig & rhs ) [private]
```

Copy constructor for [SimpleSimConfig](#) class.

This constructor is private to prevent inadvertent copies.

## 7.53.3 Member Function Documentation

### 7.53.3.1 initialize()

```
void SimpleSimConfig::initialize (
    int known_feds_count,
    TrickHLA::KnownFederate * known_feds )
```

Initialize the simulation configuration and build the list of federates based on the known federates.

Parameters

<code>known_feds_count</code>	Number of known federates.
<code>known_feds</code>	Array of known federates.

**Trick Job Class:** *initialization*

Definition at line 93 of file SimpleSimConfig.cpp.

References TrickHLA::KnownFederate::name, num\_federates, and required\_federates.

### 7.53.3.2 operator=()

```
SimpleSimConfig& TrickHLAModel::SimpleSimConfig::operator= (
    const SimpleSimConfig & rhs ) [private]
```

Assignment operator for [SimpleSimConfig](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.53.3.3 pack()

```
void SimpleSimConfig::pack ( ) [virtual]
```

Called to pack the data before the data is sent to the RTI.

Implements [TrickHLA::Packing](#).

Definition at line 127 of file SimpleSimConfig.cpp.

References TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, TrickHLA::MAX\_LOGICAL\_TIME\_SECONDS, TrickHLA::MAX\_VALUE\_IN\_MICROS, TrickHLA::MICROS\_MULTIPLIER, num\_federates, owner, required\_federates, run\_duration, run\_duration\_microsec, and TrickHLA::Packing::should\_print().

### 7.53.3.4 unpack()

```
void SimpleSimConfig::unpack ( ) [virtual]
```

Called to unpack the data after data is received from the RTI.

Implements [TrickHLA::Packing](#).

Definition at line 177 of file SimpleSimConfig.cpp.

References TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, TrickHLA::MICROS\_MULTIPLIER, num\_federates, owner, required\_federates, run\_duration, run\_duration\_microsec, and TrickHLA::Packing::should\_print().

## 7.53.4 Friends And Related Function Documentation

### 7.53.4.1 `init_attrTrickHLAModel__SimpleSimConfig`

```
void init_attrTrickHLAModel__SimpleSimConfig ( ) [friend]
```

### 7.53.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 50 of file SimpleSimConfig.hh.

## 7.53.5 Field Documentation

### 7.53.5.1 `num_federates`

```
int TrickHLAModel::SimpleSimConfig::num_federates
```

**Units:** –

Number of required federates.

Definition at line 59 of file SimpleSimConfig.hh.

Referenced by `initialize()`, `pack()`, and `unpack()`.

### 7.53.5.2 `owner`

```
char* TrickHLAModel::SimpleSimConfig::owner
```

**Units:** –

Federates name publishing the object.

Definition at line 62 of file SimpleSimConfig.hh.

Referenced by `pack()`, `unpack()`, and `~SimpleSimConfig()`.

### 7.53.5.3 `required_federates`

```
char* TrickHLAModel::SimpleSimConfig::required_federates
```

**Units:** –

Comma-separated list of required federates.

Definition at line 60 of file SimpleSimConfig.hh.

Referenced by `initialize()`, `pack()`, `unpack()`, and `~SimpleSimConfig()`.

### 7.53.5.4 `run_duration`

```
double TrickHLAModel::SimpleSimConfig::run_duration
```

**Units:** s

The run duration of the simulation.

Definition at line 56 of file SimpleSimConfig.hh.

Referenced by `pack()`, and `unpack()`.

### 7.53.5.5 run\_duration\_microsec

```
long long TrickHLAModel::SimpleSimConfig::run_duration_microsec
```

**Units:** us

The run duration in microseconds.

Definition at line 57 of file SimpleSimConfig.hh.

Referenced by pack(), and unpack().

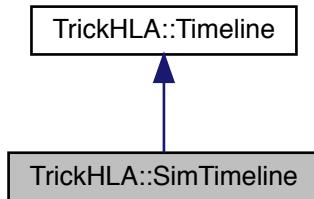
The documentation for this class was generated from the following files:

- [SimpleSimConfig.hh](#)
- [SimpleSimConfig.cpp](#)

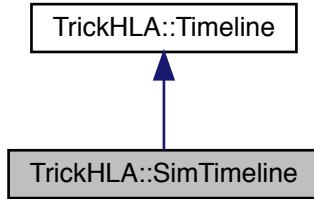
## 7.54 TrickHLA::SimTimeline Class Reference

```
#include <SimTimeline.hh>
```

Inheritance diagram for TrickHLA::SimTimeline:



Collaboration diagram for TrickHLA::SimTimeline:



### Public Member Functions

- [SimTimeline \(\)](#)  
*Default constructor for the [TrickHLA SimTimeline](#) class.*
- [virtual ~SimTimeline \(\)](#)

*Destructor for the [TrickHLA SimTimeline](#) class.*

- [virtual double get\\_time \(\)](#)

*Get the current time for this timeline in seconds.*

## Private Member Functions

- [SimTimeline \(const \[SimTimeline\]\(#\) &rhs\)](#)  
*Copy constructor for [SimTimeline](#) class.*
- [SimTimeline & operator= \(const \[SimTimeline\]\(#\) &rhs\)](#)  
*Assignment operator for [SimTimeline](#) class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLA\\_SimTimeline \(\)](#)

## Additional Inherited Members

### 7.54.1 Detailed Description

Definition at line 43 of file [SimTimeline.hh](#).

### 7.54.2 Constructor & Destructor Documentation

#### 7.54.2.1 [SimTimeline\(\)](#) [1/2]

`SimTimeline::SimTimeline ( )`

Default constructor for the [TrickHLA SimTimeline](#) class.

**Trick Job Class:** *initialization*

Definition at line 41 of file [SimTimeline.cpp](#).

#### 7.54.2.2 [~SimTimeline\(\)](#)

`SimTimeline::~SimTimeline ( ) [virtual]`

Destructor for the [TrickHLA SimTimeline](#) class.

**Trick Job Class:** *shutdown*

Definition at line 48 of file [SimTimeline.cpp](#).

#### 7.54.2.3 [SimTimeline\(\)](#) [2/2]

`TrickHLA::SimTimeline::SimTimeline (`  
    `const SimTimeline & rhs ) [private]`

Copy constructor for [SimTimeline](#) class.

This constructor is private to prevent inadvertent copies.

### 7.54.3 Member Function Documentation

### 7.54.3.1 `get_time()`

```
double SimTimeline::get_time ( ) [virtual]
Get the current time for this timeline in seconds.
```

#### Returns

Returns the current timeline time in seconds.

Implements [TrickHLA::Timeline](#).

Definition at line 52 of file SimTimeline.cpp.

Referenced by TrickHLA::ExecutionControlBase::get\_sim\_time(), TrickHLA::ScenarioTimeline::get\_time(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSES::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), and IMSim::ExecutionControl::process\_mode\_transition\_request().

### 7.54.3.2 `operator=()`

```
SimTimeline& TrickHLA::SimTimeline::operator= (
    const SimTimeline & rhs ) [private]
```

Assignment operator for [SimTimeline](#) class.

This assignment operator is private to prevent inadvertent copies.

## 7.54.4 Friends And Related Function Documentation

### 7.54.4.1 `init_attrTrickHLA__SimTimeline`

```
void init_attrTrickHLA__SimTimeline ( ) [friend]
```

### 7.54.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 50 of file SimTimeline.hh.

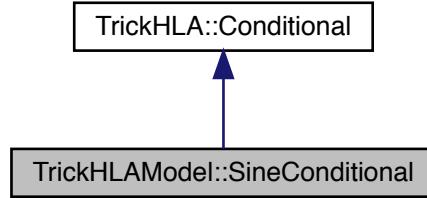
The documentation for this class was generated from the following files:

- [SimTimeline.hh](#)
- [SimTimeline.cpp](#)

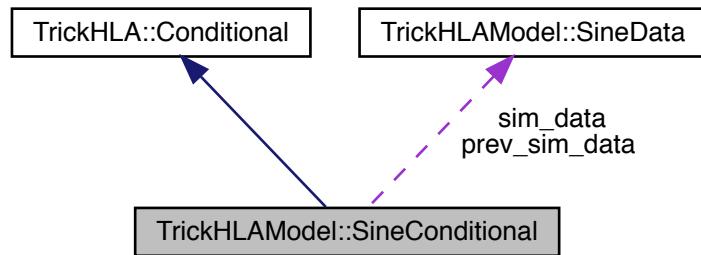
## 7.55 TrickHLAModel::SineConditional Class Reference

```
#include <SineConditional.hh>
```

Inheritance diagram for TrickHLAModel::SineConditional:



Collaboration diagram for TrickHLAModel::SineConditional:



## Public Member Functions

- [SineConditional \(\)](#)  
*Default constructor for the `TrickHLAModel SineConditional` class.*
- [virtual ~SineConditional \(\)](#)  
*Destructor for the `TrickHLAModel SineConditional` class.*
- [void initialize \(SineData \\*data, const char \\*attr\\_FOM\\_name\)](#)  
*Initializes the `sim_data` to the supplied.*
- [virtual bool should\\_send \(TrickHLA::Attribute \\*attr\)](#)  
*Determines if the attribute has changed and returns the truth of that determination.*

## Private Member Functions

- [int convert\\_FOM\\_name\\_to\\_pos \(const char \\*attr\\_FOM\\_name\)](#)  
*Determines the supplied name's position in the `SineData` structure.*

## Private Attributes

- `SineData * sim_data`

**Units:** –  
*pointer to the data to reflect in this cycle*
- `SineData prev_sim_data`

**Units:** –  
*copy of the data we previously reflected*
- `int attr_pos`

**Units:** –  
*attribute position in SineData*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLAModel__SineConditional ()`

### 7.55.1 Detailed Description

Definition at line 48 of file SineConditional.hh.

### 7.55.2 Constructor & Destructor Documentation

#### 7.55.2.1 SineConditional()

`SineConditional::SineConditional ( )`  
 Default constructor for the `TrickHLAModel SineConditional` class.  
**Trick Job Class:** *initialization*  
 Definition at line 52 of file SineConditional.cpp.

#### 7.55.2.2 ~SineConditional()

`SineConditional::~SineConditional ( ) [virtual]`  
 Destructor for the `TrickHLAModel SineConditional` class.  
**Trick Job Class:** *shutdown*  
 Definition at line 63 of file SineConditional.cpp.

### 7.55.3 Member Function Documentation

#### 7.55.3.1 convert\_FOM\_name\_to\_pos()

```
int SineConditional::convert_FOM_name_to_pos (
    const char * attr_FOM_name ) [private]
```

Determines the supplied name's position in the `SineData` structure.

#### Returns

position in `SineData`.

## Parameters

<code>attr_FOM_name</code>	FOM name of the attribute.
----------------------------	----------------------------

If a match does not exist or a empty string was supplied, -1 is returned. **Trick Job Class:** *scheduled*  
Definition at line 169 of file SineConditional.cpp.

Referenced by `initialize()`, and `should_send()`.

### 7.55.3.2 `initialize()`

```
void SineConditional::initialize (
    SineData * data,
    const char * attr_FOM_name )
```

Initializes the `sim_data` to the supplied.

## Parameters

<code>data</code>	External simulation data.
<code>attr_FOM_name</code>	FOM name of the attribute to track when changed.

**Trick Job Class:** *initialization*

Definition at line 71 of file SineConditional.cpp.

References `attr_pos`, `convert_FOM_name_to_pos()`, `prev_sim_data`, and `sim_data`.

### 7.55.3.3 `should_send()`

```
bool SineConditional::should_send (
    TrickHLA::Attribute * attr ) [virtual]
```

Determines if the attribute has changed and returns the truth of that determination.

## Returns

True if value should be sent.

## Parameters

<code>attr</code>	Attribute to check.
-------------------	---------------------

**Trick Job Class:** *scheduled*

Reimplemented from [TrickHLA::Conditional](#).

Definition at line 89 of file SineConditional.cpp.

References `attr_pos`, `convert_FOM_name_to_pos()`, `TrickHLAModel::SineData::get_amplitude()`, `TrickHLAModel::SineData::get_derivative()`, `TrickHLA::Attribute::get_FOM_name()`, `TrickHLAModel::SineData::get_frequency()`, `TrickHLAModel::SineData::get_name()`, `TrickHLAModel::SineData::get_phase()`, `TrickHLAModel::SineData::get_time()`, `TrickHLAModel::SineData::get_tolerance()`, `TrickHLAModel::SineData::get_value()`, `prev_sim_data`, and `sim_data`.

## 7.55.4 Friends And Related Function Documentation

#### 7.55.4.1 `init_attrTrickHLAModel__SineConditional`

```
void init_attrTrickHLAModel__SineConditional ( ) [friend]
```

#### 7.55.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
Definition at line 55 of file SineConditional.hh.
```

### 7.55.5 Field Documentation

#### 7.55.5.1 `attr_pos`

```
int TrickHLAModel::SineConditional::attr_pos [private]
```

**Units:** –

attribute position in [SineData](#)

Definition at line 90 of file SineConditional.hh.

Referenced by `initialize()`, and `should_send()`.

#### 7.55.5.2 `prev_sim_data`

```
SineData TrickHLAModel::SineConditional::prev_sim_data [private]
```

**Units:** –

copy of the data we previously reflected

Definition at line 88 of file SineConditional.hh.

Referenced by `initialize()`, and `should_send()`.

#### 7.55.5.3 `sim_data`

```
SineData* TrickHLAModel::SineConditional::sim_data [private]
```

**Units:** –

pointer to the data to reflect in this cycle

Definition at line 87 of file SineConditional.hh.

Referenced by `initialize()`, and `should_send()`.

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

- [SineConditional.hh](#)
- [SineConditional.cpp](#)

## 7.56 TrickHLAModel::SineData Class Reference

```
#include <SineData.hh>
```

### Public Member Functions

- [SineData \(\)](#)

*Default constructor for the [TrickHLAModel SineData](#) class.*
- [SineData \(double phi, double omega, double mag\)](#)

*Initialization constructor for the [TrickHLAModel SineData](#) class.*

- virtual `~SineData ()`  
*Destructor for the `TrickHLAModel SineData` class.*
- int `set_time` (double t)  
*Set the current time for the sine wave.*
- double `get_time` () const  
*Get the current time value for the sine wave function.*
- void `set_value` (double val)  
*Set the current value of the sine wave function.*
- double `get_value` () const  
*Get the current value of the sine wave function.*
- void `set_derivative` (double deriv)  
*Set the value of the time derivative of the sine wave function.*
- double `get_derivative` () const  
*Get the value of the time derivative of the sine wave function.*
- void `set_phase` (double phi)  
*Set the phase value of the sine wave function.*
- double `get_phase` () const  
*Get the phase value of the sine wave function.*
- void `set_frequency` (double omega)  
*Set the frequency value of the sine wave function.*
- double `get_frequency` () const  
*Get the frequency value of the sine wave function.*
- void `set_amplitude` (double mag)  
*Set the amplitude value of the sine wave function.*
- double `get_amplitude` () const  
*Get the amplitude value of the sine wave function.*
- void `set_tolerance` (double epsilon)  
*Set the tolerance value of the sine wave function.*
- double `get_tolerance` () const  
*Get the tolerance value of the sine wave function.*
- const char \* `get_name` () const  
*Get the name of the sine wave object.*
- void `set_name` (const char \*n)  
*Set the name of the sine wave object.*
- void `copy_data` (SineData \*orig)  
*Utility function to copy data from source to this object.*
- void `compute_value` ()  
*Computes the value of the test data.*
- void `compute_value` (double t)  
*Computes the value of the test data.*
- void `compute_derivative` ()  
*Computes the derivative of the test data.*
- void `compute_derivative` (double t)  
*Computes the derivative of the test data.*
- void `adjust_phase` ()  
*Computes the current phase offset from the current value and derivative.*
- void `adjust_phase` (double t)  
*Computes the phase offset for the provided model time.*
- int `integration` ()  
*Sine wave integration routine.*

## Data Fields

- double `time`

**Units:** *s*  
*Current time for test data model.*
- double `value`

**Units:** *–*  
*Current output value for test data model.*
- double `dvdt`

**Units:** *–*  
*Current time derivative of the test data.*
- double `phase`

**Units:** *radian*  
*Phase offset.*
- double `freq`

**Units:** *radian/s*  
*Frequency of sine wave.*
- double `amp`

**Units:** *–*  
*Amplitude of sine wave.*
- double `tol`

**Units:** *–*  
*Tolerance for phase adjustment.*
- char \* `name`

**Units:** *–*  
*Name of the data.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLAModel__SineData ()`

### 7.56.1 Detailed Description

Definition at line 44 of file SineData.hh.

### 7.56.2 Constructor & Destructor Documentation

#### 7.56.2.1 SineData() [1/2]

`SineData::SineData ( )`

Default constructor for the `TrickHLAModel SineData` class.

**Trick Job Class:** *initialization*

Definition at line 48 of file SineData.cpp.

References `amp`, `compute_derivative()`, `compute_value()`, `dvdt`, `freq`, `phase`, `time`, `tol`, and `value`.

### 7.56.2.2 SineData() [2/2]

```
SineData::SineData (
    double phi,
    double omega,
    double mag )
```

Initialization constructor for the [TrickHLAModel SineData](#) class.

#### Parameters

<i>phi</i>	Sine wave phase {rad}.
<i>omega</i>	Sine wave frequency {rad/s}.
<i>mag</i>	Sine wave magnitude.

#### Trick Job Class: *initialization*

Definition at line 73 of file SineData.cpp.

References amp, compute\_derivative(), compute\_value(), freq, phase, time, and tol.

### 7.56.2.3 ~SineData()

```
SineData::~SineData ( ) [virtual]
Destructor for the TrickHLAModel SineData class.
```

#### Trick Job Class: *shutdown*

Definition at line 101 of file SineData.cpp.

References name.

## 7.56.3 Member Function Documentation

### 7.56.3.1 adjust\_phase() [1/2]

```
void SineData::adjust_phase ( )
```

Computes the current phase offset from the current value and derivative.

Definition at line 173 of file SineData.cpp.

References amp, dvdt, freq, phase, time, tol, and value.

Referenced by [adjust\\_phase\(\)](#), and [TrickHLAModel::SineLagCompensation::receive\\_lag\\_compensation\(\)](#).

### 7.56.3.2 adjust\_phase() [2/2]

```
void SineData::adjust_phase (
    double t )
```

Computes the phase offset for the provided model time.

#### Parameters

<i>t</i>	Current model time {s}.
----------	-------------------------

Definition at line 206 of file SineData.cpp.

References [adjust\\_phase\(\)](#), and [time](#).

### 7.56.3.3 compute\_derivative() [1/2]

```
void SineData::compute_derivative ( )
```

Computes the derivative of the test data.

**Trick Job Class:** *derivative*

Definition at line 151 of file SineData.cpp.

References amp, dvdt, freq, phase, and time.

Referenced by compute\_derivative(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLA←Model::SineLagCompensation::send\_lag\_compensation(), and SineData().

### 7.56.3.4 compute\_derivative() [2/2]

```
void SineData::compute_derivative (
    double t )
```

Computes the derivative of the test data.

#### Parameters

<i>t</i>	Current integration time.
----------	---------------------------

**Trick Job Class:** *derivative*

Definition at line 161 of file SineData.cpp.

References compute\_derivative(), and time.

### 7.56.3.5 compute\_value() [1/2]

```
void SineData::compute_value ( )
```

Computes the value of the test data.

**Trick Job Class:** *scheduled*

Definition at line 127 of file SineData.cpp.

References amp, freq, phase, time, and value.

Referenced by compute\_value(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLA←Model::SineLagCompensation::send\_lag\_compensation(), and SineData().

### 7.56.3.6 compute\_value() [2/2]

```
void SineData::compute_value (
    double t )
```

Computes the value of the test data.

#### Parameters

<i>t</i>	Current model time.
----------	---------------------

**Trick Job Class:** *scheduled*

Definition at line 137 of file SineData.cpp.

References compute\_value(), and time.

### 7.56.3.7 copy\_data()

```
void SineData::copy_data (
```

```
SineData * orig )
```

Utility function to copy data from source to this object.

#### Parameters

<code>orig</code>	Sine wave data object to copy into.
-------------------	-------------------------------------

#### Trick Job Class: *scheduled*

Definition at line 116 of file SineData.cpp.

### 7.56.3.8 `get_amplitude()`

```
double TrickHLAModel::SineData::get_amplitude ( ) const [inline]
```

Get the amplitude value of the sine wave function.

#### Returns

The amplitude value of the sine wave function.

Definition at line 135 of file SineData.hh.

References amp.

Referenced by TrickHLAModel::SinePacking::pack(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLAModel::SineConditional::should\_send(), and TrickHLAModel::SinePacking::unpack().

### 7.56.3.9 `get_derivative()`

```
double TrickHLAModel::SineData::get_derivative ( ) const [inline]
```

Get the value of the time derivative of the sine wave function.

#### Returns

The current value of the time derivative of the sine wave function.

Definition at line 111 of file SineData.hh.

References dvdt.

Referenced by TrickHLAModel::SinePacking::pack(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLAModel::SineConditional::should\_send(), and TrickHLAModel::SinePacking::unpack().

### 7.56.3.10 `get_frequency()`

```
double TrickHLAModel::SineData::get_frequency ( ) const [inline]
```

Get the frequency value of the sine wave function.

#### Returns

The frequency value of the sine wave function.

Definition at line 127 of file SineData.hh.

References freq.

Referenced by TrickHLAModel::SinePacking::pack(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLAModel::SineConditional::should\_send(), and TrickHLAModel::SinePacking::unpack().

### 7.56.3.11 get\_name()

```
const char* TrickHLAModel::SineData::get_name ( ) const [inline]  
Get the name of the sine wave object.
```

#### Returns

A constant pointer to the name of the sine wave object.

Definition at line 147 of file SineData.hh.

References name.

Referenced by TrickHLAModel::SinePacking::pack(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLAModel::SineConditional::should\_send(), and TrickHLAModel::SinePacking::unpack().

### 7.56.3.12 get\_phase()

```
double TrickHLAModel::SineData::get_phase ( ) const [inline]  
Get the phase value of the sine wave function.
```

#### Returns

The phase value of the sine wave function.

Definition at line 119 of file SineData.hh.

References phase.

Referenced by TrickHLAModel::SinePacking::pack(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLAModel::SineConditional::should\_send(), and TrickHLAModel::SinePacking::unpack().

### 7.56.3.13 get\_time()

```
double TrickHLAModel::SineData::get_time ( ) const [inline]  
Get the current time value for the sine wave function.
```

#### Returns

The current time value for the sine wave function.

Definition at line 95 of file SineData.hh.

References time.

Referenced by TrickHLAModel::SinePacking::pack(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLAModel::SineLagCompensation::send\_lag\_compensation(), TrickHLAModel::SineConditional::should\_send(), and TrickHLAModel::SinePacking::unpack().

### 7.56.3.14 get\_tolerance()

```
double TrickHLAModel::SineData::get_tolerance ( ) const [inline]  
Get the tolerance value of the sine wave function.
```

#### Returns

The tolerance value of the sine wave function.

Definition at line 143 of file SineData.hh.

References tol.

Referenced by TrickHLAModel::SinePacking::pack(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLAModel::SineConditional::should\_send(), and TrickHLAModel::SinePacking::unpack().

### 7.56.3.15 get\_value()

```
double TrickHLAModel::SineData::get_value ( ) const [inline]  
Get the current value of the sine wave function.
```

#### Returns

Current value of the sine wave function.

Definition at line 103 of file SineData.hh.

References value.

Referenced by TrickHLAModel::SinePacking::pack(), TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), TrickHLAModel::SineConditional::should\_send(), and TrickHLAModel::SinePacking::unpack().

### 7.56.3.16 integration()

```
int SineData::integration ( )  
Sine wave integration routine.
```

#### Returns

Intermediate step ID.

**Trick Job Class:** *integration*

Definition at line 220 of file SineData.cpp.

References dvdt, and value.

### 7.56.3.17 set\_amplitude()

```
void TrickHLAModel::SineData::set_amplitude (   
    double mag ) [inline]  
Set the amplitude value of the sine wave function.
```

#### Parameters

<i>mag</i>	The amplitude value of the sine wave function.
------------	--

Definition at line 131 of file SineData.hh.

Referenced by TrickHLAModel::SineLagCompensation::receive\_lag\_compensation().

### 7.56.3.18 set\_derivative()

```
void TrickHLAModel::SineData::set_derivative (   
    double deriv ) [inline]  
Set the value of the time derivative of the sine wave function.
```

#### Parameters

<i>deriv</i>	The time derivative value of the sine wave function.
--------------	--

Definition at line 107 of file SineData.hh.

References deriv.

Referenced by TrickHLAModel::SineLagCompensation::receive\_lag\_compensation().

### 7.56.3.19 set\_frequency()

```
void TrickHLAModel::SineData::set_frequency (
    double omega ) [inline]
```

Set the frequency value of the sine wave function.

#### Parameters

<i>omega</i>	The frequency value of the sine wave function.
--------------	--

Definition at line 123 of file SineData.hh.

Referenced by TrickHLAModel::SineLagCompensation::receive\_lag\_compensation().

### 7.56.3.20 set\_name()

```
void TrickHLAModel::SineData::set_name (
    const char * n ) [inline]
```

Set the name of the sine wave object.

#### Parameters

<i>n</i>	The name of the sine wave object.
----------	-----------------------------------

Definition at line 151 of file SineData.hh.

Referenced by TrickHLAModel::SineLagCompensation::receive\_lag\_compensation().

### 7.56.3.21 set\_phase()

```
void TrickHLAModel::SineData::set_phase (
    double phi ) [inline]
```

Set the phase value of the sine wave function.

#### Parameters

<i>phi</i>	The phase value of the sine wave function.
------------	--

Definition at line 115 of file SineData.hh.

Referenced by TrickHLAModel::SineLagCompensation::receive\_lag\_compensation(), and TrickHLAModel::SinePacking::unpack().

### 7.56.3.22 set\_time()

```
int TrickHLAModel::SineData::set_time (
    double t ) [inline]
```

Set the current time for the sine wave.

#### Parameters

<i>t</i>	Time value for sine wave function.
----------	------------------------------------

Definition at line 87 of file SineData.hh.

### 7.56.3.23 set\_tolerance()

```
void TrickHLAModel::SineData::set_tolerance ( double epsilon ) [inline]
```

Set the tolerance value of the sine wave function.

#### Parameters

<i>epsilon</i>	The tolerance value of the sine wave function.
----------------	--

Definition at line 139 of file SineData.hh.

Referenced by TrickHLAModel::SineLagCompensation::receive\_lag\_compensation().

### 7.56.3.24 set\_value()

```
void TrickHLAModel::SineData::set_value ( double val ) [inline]
```

Set the current value of the sine wave function.

#### Parameters

<i>val</i>	Sine wave value.
------------	------------------

Definition at line 99 of file SineData.hh.

Referenced by TrickHLAModel::SineLagCompensation::receive\_lag\_compensation().

## 7.56.4 Friends And Related Function Documentation

### 7.56.4.1 init\_attrTrickHLAModel\_\_SineData

```
void init_attrTrickHLAModel__SineData ( ) [friend]
```

### 7.56.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 51 of file SineData.hh.

## 7.56.5 Field Documentation

### 7.56.5.1 amp

```
double TrickHLAModel::SineData::amp
```

#### Units: -

Amplitude of sine wave.

Definition at line 65 of file SineData.hh.

Referenced by adjust\_phase(), compute\_derivative(), compute\_value(), get\_amplitude(), and SineData().

### 7.56.5.2 dvdt

```
double TrickHLAModel::SineData::dvdt
```

**Units:** –

Current time derivative of the test data.

Definition at line 60 of file SineData.hh.

Referenced by `adjust_phase()`, `compute_derivative()`, `get_derivative()`, `integration()`, and `SineData()`.

### 7.56.5.3 freq

```
double TrickHLAModel::SineData::freq
```

**Units:** *radian/s*

Frequency of sine wave.

Definition at line 64 of file SineData.hh.

Referenced by `adjust_phase()`, `compute_derivative()`, `compute_value()`, `get_frequency()`, and `SineData()`.

### 7.56.5.4 name

```
char* TrickHLAModel::SineData::name
```

**Units:** –

Name of the data.

Definition at line 68 of file SineData.hh.

Referenced by `get_name()`, and `~SineData()`.

### 7.56.5.5 phase

```
double TrickHLAModel::SineData::phase
```

**Units:** *radian*

Phase offset.

Definition at line 63 of file SineData.hh.

Referenced by `adjust_phase()`, `compute_derivative()`, `compute_value()`, `get_phase()`, and `SineData()`.

### 7.56.5.6 time

```
double TrickHLAModel::SineData::time
```

**Units:** *s*

Current time for test data model.

Definition at line 58 of file SineData.hh.

Referenced by `adjust_phase()`, `compute_derivative()`, `compute_value()`, `get_time()`, and `SineData()`.

### 7.56.5.7 tol

```
double TrickHLAModel::SineData::tol
```

**Units:** –

Tolerance for phase adjustment.

Definition at line 66 of file SineData.hh.

Referenced by `adjust_phase()`, `get_tolerance()`, and `SineData()`.

## 7.56.5.8 value

```
double TrickHLAModel::SineData::value
```

**Units:** –

Current output value for test data model.

Definition at line 59 of file SineData.hh.

Referenced by `adjust`, `phase()`, `compute`, `value()`, `get`, `value()`, `integration()`, and `SineData()`.

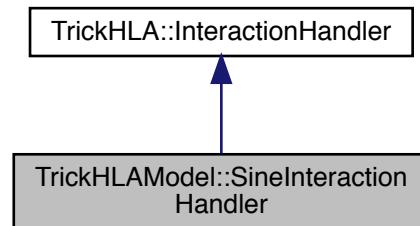
The documentation for this class was generated from the following files:

- SineData.hh
  - SineData.cpp

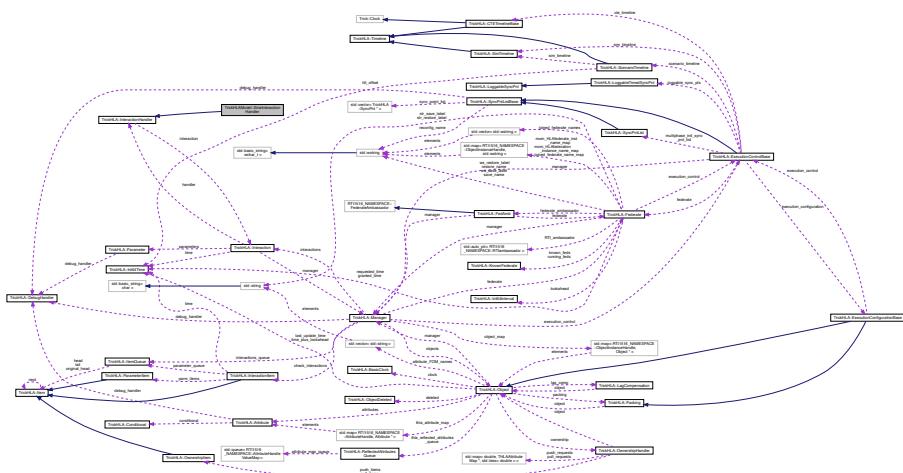
## 7.57 TrickHLAModel::SineInteractionHandler Class Reference

```
#include <SineInteractionHandler.hh>
```

## Inheritance diagram for TrickHLAModel::SineInteractionHandler:



## Collaboration diagram for TrickHLAModel::SineInteractionHandler:



## Public Member Functions

- `SineInteractionHandler()`

*Default constructor for the [TrickHLAModel SineInteractionHandler](#) class.*

- `virtual ~SineInteractionHandler ()`

*Destructor for the [TrickHLAModel SineInteractionHandler](#) class.*

- `void send_sine_interaction (double send_time)`

*Send the HLA interaction using either [Timestamp Order](#) or [Receive Order](#) which is determined at compile time.*

- `virtual void receive_interaction (RTI1516_USERDATA const &the_user_supplied_tag)`

*Receive the HLA interaction.*

## Data Fields

- `char * name`

**Units:** –

*Example of a unique name to identify the interaction handler.*

- `char * message`

**Units:** –

*Example of a static array of strings.*

## Protected Attributes

- `double time`

**Units:** *s*

*Example of floating-point data.*

- `int year`

**Units:** –

*Example of integer data.*

- `int send_cnt`

**Units:** –

*The number of times an interaction is sent.*

- `int receive_cnt`

**Units:** –

*The number of times an interaction was received.*

## Private Member Functions

- `SineInteractionHandler (const SineInteractionHandler &rhs)`

*Copy constructor for [SineInteractionHandler](#) class.*

- `SineInteractionHandler & operator= (const SineInteractionHandler &rhs)`

*Assignment operator for [SineInteractionHandler](#) class.*

## Friends

- `class InputProcessor`

- `void init_attrTrickHLAModel__SineInteractionHandler ()`

### 7.57.1 Detailed Description

Definition at line 45 of file `SineInteractionHandler.hh`.

### 7.57.2 Constructor & Destructor Documentation

**7.57.2.1 SineInteractionHandler() [1/2]**

```
SineInteractionHandler::SineInteractionHandler ( )
Default constructor for the TrickHLAModel SineInteractionHandler class.
Trick Job Class: initialization
Definition at line 59 of file SineInteractionHandler.cpp.
```

**7.57.2.2 ~SineInteractionHandler()**

```
SineInteractionHandler::~SineInteractionHandler ( ) [virtual]
Destructor for the TrickHLAModel SineInteractionHandler class.
Trick Job Class: shutdown
Definition at line 73 of file SineInteractionHandler.cpp.
```

**7.57.2.3 SineInteractionHandler() [2/2]**

```
TrickHLAModel::SineInteractionHandler::SineInteractionHandler (
    const SineInteractionHandler & rhs ) [private]
Copy constructor for SineInteractionHandler class.
This constructor is private to prevent inadvertent copies.
```

**7.57.3 Member Function Documentation****7.57.3.1 operator=()**

```
SineInteractionHandler& TrickHLAModel::SineInteractionHandler::operator= (
    const SineInteractionHandler & rhs ) [private]
Assignment operator for SineInteractionHandler class.
This assignment operator is private to prevent inadvertent copies.
```

**7.57.3.2 receive\_interaction()**

```
void SineInteractionHandler::receive_interaction (
    RTI1516\_USERDATA const & the_user_supplied_tag ) [virtual]
Receive the HLA interaction.
```

**Parameters**

<i>the_user_supplied_tag</i>	User tag.
------------------------------	-----------

Reimplemented from [TrickHLA::InteractionHandler](#).

Definition at line 164 of file SineInteractionHandler.cpp.

References [TrickHLA::DEBUG\\_LEVEL\\_1\\_TRACE](#), [TrickHLA::DEBUG\\_SOURCE\\_INTERACTION](#), [TrickHLA::InteractionHandler::get\\_scenario\\_time\(\)](#), [message](#), [name](#), [receive\\_cnt](#), [TrickHLA::InteractionHandler::should\\_print\(\)](#), [time](#), and [year](#).

**7.57.3.3 send\_sine\_interaction()**

```
void SineInteractionHandler::send_sine_interaction (
    double send_time )
```

Send the HLA interaction using either Timestamp Order or Receive Order which is determined at compile time.

#### Parameters

<i>send_time</i>	HLA time to send the interaction {s}.
------------------	---------------------------------------

Definition at line 78 of file SineInteractionHandler.cpp.

References `TrickHLA::DEBUG_LEVEL_1_TRACE`, `TrickHLA::DEBUG_SOURCE_INTERACTION`, `TrickHLA::InteractionHandler::get_fed_looking_ahead()`, `TrickHLA::InteractionHandler::get_granted_fed_time()`, `TrickHLA::Int64Interval::getDoubleTime()`, `TrickHLA::Int64Time::getDoubleTime()`, `message`, `name`, `RTI1516_USERDATA`, `send_cnt`, `TrickHLA::InteractionHandler::should_print()`, `time`, and `year`.

## 7.57.4 Friends And Related Function Documentation

### 7.57.4.1 `init_attrTrickHLAModel__SineInteractionHandler`

```
void init_attrTrickHLAModel__SineInteractionHandler ( ) [friend]
```

### 7.57.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 52 of file SineInteractionHandler.hh.

## 7.57.5 Field Documentation

### 7.57.5.1 `message`

```
char* TrickHLAModel::SineInteractionHandler::message
```

#### Units: –

Example of a static array of strings.

Definition at line 59 of file SineInteractionHandler.hh.

Referenced by `receive_interaction()`, and `send_sine_interaction()`.

### 7.57.5.2 `name`

```
char* TrickHLAModel::SineInteractionHandler::name
```

#### Units: –

Example of a unique name to identify the interaction handler.

Definition at line 58 of file SineInteractionHandler.hh.

Referenced by `receive_interaction()`, and `send_sine_interaction()`.

### 7.57.5.3 `receive_cnt`

```
int TrickHLAModel::SineInteractionHandler::receive_cnt [protected]
```

#### Units: –

The number of times an interaction was received.

Definition at line 85 of file SineInteractionHandler.hh.

Referenced by `receive_interaction()`.

#### 7.57.5.4 send\_cnt

```
int TrickHLAModel::SineInteractionHandler::send_cnt [protected]
```

**Units:** -

The number of times an interaction is sent.

Definition at line 84 of file SineInteractionHandler.hh.

Referenced by send\_sine\_interaction().

#### 7.57.5.5 time

```
double TrickHLAModel::SineInteractionHandler::time [protected]
```

**Units:** s

Example of floating-point data.

Definition at line 81 of file SineInteractionHandler.hh.

Referenced by receive\_interaction(), and send\_sine\_interaction().

#### 7.57.5.6 year

```
int TrickHLAModel::SineInteractionHandler::year [protected]
```

**Units:** -

Example of integer data.

Definition at line 82 of file SineInteractionHandler.hh.

Referenced by receive\_interaction(), and send\_sine\_interaction().

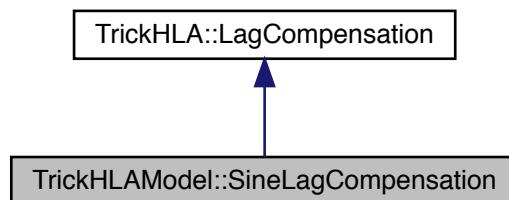
The documentation for this class was generated from the following files:

- [SineInteractionHandler.hh](#)
- [SineInteractionHandler.cpp](#)

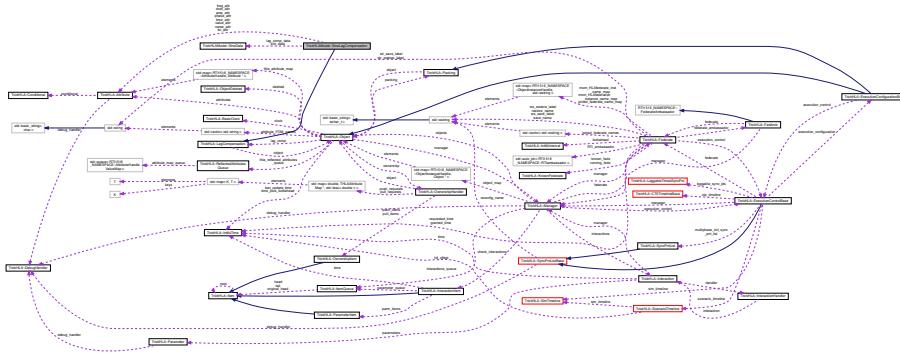
## 7.58 TrickHLAModel::SineLagCompensation Class Reference

```
#include <SineLagCompensation.hh>
```

Inheritance diagram for TrickHLAModel::SineLagCompensation:



Collaboration diagram for `TrickHLAModel::SineLagCompensation`:



## Public Member Functions

- `SineLagCompensation ()`  
*Default constructor for the `TrickHLAModel SineLagCompensation` class.*
- `virtual ~SineLagCompensation ()`  
*Destructor for the `TrickHLAModel SineLagCompensation` class.*
- `void initialize (SineData *sim_data, SineData *lag_comp_data)`  
*Initialize the LagCompensation object.*
- `virtual void initialize_callback (TrickHLA::Object *obj)`  
*Initialization callback as part of the `TrickHLA::LagCompensation` functions.*
- `virtual void send_lag_compensation ()`  
*Send side lag-compensation where we propagate the sine wave state head by dt to predict the value at the next data cycle.*
- `virtual void receive_lag_compensation ()`  
*Receive side lag-compensation where we propagate the sine wave state ahead by dt to predict the value at the next data cycle.*

## Private Member Functions

- `SineLagCompensation (const SineLagCompensation &rhs)`  
*Copy constructor for `SineLagCompensation` class.*
- `SineLagCompensation & operator= (const SineLagCompensation &rhs)`  
*Assignment operator for `SineLagCompensation` class.*

## Private Attributes

- `SineData * sim_data`  
**Units:** –  
*Simulation data.*
- `SineData * lag_comp_data`  
**Units:** –  
*Lag compensation data.*
- `TrickHLA::Attribute * time_attr`  
**Data I/O:** \*\*  
*Reference to the "Time" `TrickHLA::Attribute`.*

- `TrickHLA::Attribute * value_attr`  
`Data I/O: **`  
`Reference to the "Value" TrickHLA::Attribute.`
- `TrickHLA::Attribute * dvdt_attr`  
`Data I/O: **`  
`Reference to the "dvdt" TrickHLA::Attribute.`
- `TrickHLA::Attribute * phase_attr`  
`Data I/O: **`  
`Reference to the "Phase" TrickHLA::Attribute.`
- `TrickHLA::Attribute * freq_attr`  
`Data I/O: **`  
`Reference to the "Frequency" TrickHLA::Attribute.`
- `TrickHLA::Attribute * amp_attr`  
`Data I/O: **`  
`Reference to the "Amplitude" TrickHLA::Attribute.`
- `TrickHLA::Attribute * tol_attr`  
`Data I/O: **`  
`Reference to the "Tolerance" TrickHLA::Attribute.`
- `TrickHLA::Attribute * name_attr`  
`Data I/O: **`  
`Reference to the "Name" TrickHLA::Attribute.`

## Friends

- class `InputProcessor`
- void `init_attrTrickHLAModel__SineLagCompensation ()`

## Additional Inherited Members

### 7.58.1 Detailed Description

Definition at line 50 of file SineLagCompensation.hh.

### 7.58.2 Constructor & Destructor Documentation

#### 7.58.2.1 `SineLagCompensation()` [1/2]

`SineLagCompensation::SineLagCompensation ( )`  
Default constructor for the [TrickHLAModel SineLagCompensation](#) class.  
**Trick Job Class:** *initialization*  
Definition at line 51 of file SineLagCompensation.cpp.

#### 7.58.2.2 `~SineLagCompensation()`

`SineLagCompensation::~SineLagCompensation ( ) [virtual]`  
Destructor for the [TrickHLAModel SineLagCompensation](#) class.  
**Trick Job Class:** *shutdown*  
Definition at line 69 of file SineLagCompensation.cpp.

### 7.58.2.3 SineLagCompensation() [2/2]

```
TrickHLAModel::SineLagCompensation::SineLagCompensation (
    const SineLagCompensation & rhs ) [private]
```

Copy constructor for [SineLagCompensation](#) class.

This constructor is private to prevent inadvertent copies.

## 7.58.3 Member Function Documentation

### 7.58.3.1 initialize()

```
void SineLagCompensation::initialize (
    SineData * sim_data,
    SineData * lag_comp_data )
```

Initialize the LagCompensation object.

#### Parameters

<i>sim_data</i>	The sine wave data object.
<i>lag_comp_data</i>	The sine wave lag compensation data.

#### Trick Job Class: *initialization*

Definition at line 77 of file SineLagCompensation.cpp.

References *lag\_comp\_data*, and *sim\_data*.

### 7.58.3.2 initialize\_callback()

```
void SineLagCompensation::initialize_callback (
    TrickHLA::Object * obj ) [virtual]
```

Initialization callback as part of the [TrickHLA::LagCompensation](#) functions.

#### Parameters

<i>obj</i>	TrickHLA Object associated with this LagCompensation class.
------------	---

Use the initialize callback function as a way to setup [TrickHLA::Attribute](#) references which are used to determine ownership or if data for an attribute was received. **Trick Job Class: *initialization***

Reimplemented from [TrickHLA::LagCompensation](#).

Definition at line 92 of file SineLagCompensation.cpp.

References *amp\_attr*, *dvdt\_attr*, *freq\_attr*, [TrickHLA::LagCompensation::get\\_attribute\\_and\\_validate\(\)](#), *name\_attr*, *phase\_attr*, *time\_attr*, *tol\_attr*, and *value\_attr*.

### 7.58.3.3 operator=( )

```
SineLagCompensation& TrickHLAModel::SineLagCompensation::operator= (
    const SineLagCompensation & rhs ) [private]
```

Assignment operator for [SineLagCompensation](#) class.

This assignment operator is private to prevent inadvertent copies.

#### 7.58.3.4 receive\_lag\_compensation()

```
void SineLagCompensation::receive_lag_compensation ( ) [virtual]
```

Receive side lag-compensation where we propagate the sine wave state ahead by dt to predict the value at the next data cycle.

Reimplemented from [TrickHLA::LagCompensation](#).

Definition at line 136 of file SineLagCompensation.cpp.

References `TrickHLAModel::SineData::adjust_phase()`, `amp_attr`, `TrickHLAModel::SineData::compute_derivative()`, `TrickHLAModel::SineData::compute_value()`, `TrickHLA::DEBUG_LEVEL_6_TRACE`, `TrickHLA::DEBUG_SOURCE_LAG_COMPENSATION`, `dvdt_attr`, `freq_attr`, `TrickHLAModel::SineData::get_amplitude()`, `TrickHLAModel::SineData::get_derivative()`, `TrickHLAModel::SineData::get_frequency()`, `TrickHLAModel::SineData::get_name()`, `TrickHLAModel::SineData::get_phase()`, `TrickHLA::LagCompensation::get_scenario_time()`, `TrickHLAModel::SineData::get_time()`, `TrickHLAModel::SineData::get_tolerance()`, `TrickHLAModel::SineData::get_value()`, `TrickHLA::Attribute::is_received()`, `lag_comp_data`, `name_attr`, `phase_attr`, `TrickHLAModel::SineData::set_amplitude()`, `TrickHLAModel::SineData::set_derivative()`, `TrickHLAModel::SineData::set_frequency()`, `TrickHLAModel::SineData::set_name()`, `TrickHLAModel::SineData::set_phase()`, `TrickHLAModel::SineData::set_tolerance()`, `TrickHLAModel::SineData::set_value()`, `TrickHLA::LagCompensation::should_print()`, `sim_data`, `tol_attr`, and `value_attr`.

#### 7.58.3.5 send\_lag\_compensation()

```
void SineLagCompensation::send_lag_compensation ( ) [virtual]
```

Send side lag-compensation where we propagate the sine wave state head by dt to predict the value at the next data cycle.

Reimplemented from [TrickHLA::LagCompensation](#).

Definition at line 112 of file SineLagCompensation.cpp.

References `TrickHLAModel::SineData::compute_derivative()`, `TrickHLAModel::SineData::compute_value()`, `TrickHLA::DEBUG_LEVEL_6_TRACE`, `TrickHLA::DEBUG_SOURCE_LAG_COMPENSATION`, `TrickHLA::LagCompensation::get_fed_lookahead()`, `TrickHLA::LagCompensation::get_scenario_time()`, `TrickHLAModel::SineData::get_time()`, `TrickHLA::Int64Interval::getDoubleTime()`, `lag_comp_data`, `TrickHLA::LagCompensation::should_print()`, and `sim_data`.

### 7.58.4 Friends And Related Function Documentation

#### 7.58.4.1 init\_attrTrickHLAModel\_\_SineLagCompensation

```
void init_attrTrickHLAModel__SineLagCompensation ( ) [friend]
```

#### 7.58.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 57 of file SineLagCompensation.hh.

### 7.58.5 Field Documentation

#### 7.58.5.1 amp\_attr

```
TrickHLA::Attribute* TrickHLAModel::SineLagCompensation::amp_attr [private]
```

**Data I/O: \*\***

Reference to the "Amplitude" [TrickHLA::Attribute](#).

Definition at line 101 of file SineLagCompensation.hh.  
Referenced by initialize\_callback(), and receive\_lag\_compensation().

#### 7.58.5.2 dvdt\_attr

`TrickHLA::Attribute* TrickHLAModel::SineLagCompensation::dvdt_attr [private]`

**Data I/O:** \*\*

Reference to the "dvdt" [TrickHLA::Attribute](#).

Definition at line 98 of file SineLagCompensation.hh.

Referenced by initialize\_callback(), and receive\_lag\_compensation().

#### 7.58.5.3 freq\_attr

`TrickHLA::Attribute* TrickHLAModel::SineLagCompensation::freq_attr [private]`

**Data I/O:** \*\*

Reference to the "Frequency" [TrickHLA::Attribute](#).

Definition at line 100 of file SineLagCompensation.hh.

Referenced by initialize\_callback(), and receive\_lag\_compensation().

#### 7.58.5.4 lag\_comp\_data

`SineData* TrickHLAModel::SineLagCompensation::lag_comp_data [private]`

**Units:** –

Lag compensation data.

Definition at line 94 of file SineLagCompensation.hh.

Referenced by initialize(), receive\_lag\_compensation(), and send\_lag\_compensation().

#### 7.58.5.5 name\_attr

`TrickHLA::Attribute* TrickHLAModel::SineLagCompensation::name_attr [private]`

**Data I/O:** \*\*

Reference to the "Name" [TrickHLA::Attribute](#).

Definition at line 103 of file SineLagCompensation.hh.

Referenced by initialize\_callback(), and receive\_lag\_compensation().

#### 7.58.5.6 phase\_attr

`TrickHLA::Attribute* TrickHLAModel::SineLagCompensation::phase_attr [private]`

**Data I/O:** \*\*

Reference to the "Phase" [TrickHLA::Attribute](#).

Definition at line 99 of file SineLagCompensation.hh.

Referenced by initialize\_callback(), and receive\_lag\_compensation().

#### 7.58.5.7 sim\_data

`SineData* TrickHLAModel::SineLagCompensation::sim_data [private]`

**Units:** –

Simulation data.

Definition at line 93 of file SineLagCompensation.hh.

Referenced by initialize(), receive\_lag\_compensation(), and send\_lag\_compensation().

#### 7.58.5.8 time\_attr

`TrickHLA::Attribute* TrickHLAModel::SineLagCompensation::time_attr [private]`

**Data I/O: \*\***

Reference to the "Time" [TrickHLA::Attribute](#).

Definition at line 96 of file SineLagCompensation.hh.

Referenced by initialize\_callback().

#### 7.58.5.9 tol\_attr

`TrickHLA::Attribute* TrickHLAModel::SineLagCompensation::tol_attr [private]`

**Data I/O: \*\***

Reference to the "Tolerance" [TrickHLA::Attribute](#).

Definition at line 102 of file SineLagCompensation.hh.

Referenced by initialize\_callback(), and receive\_lag\_compensation().

#### 7.58.5.10 value\_attr

`TrickHLA::Attribute* TrickHLAModel::SineLagCompensation::value_attr [private]`

**Data I/O: \*\***

Reference to the "Value" [TrickHLA::Attribute](#).

Definition at line 97 of file SineLagCompensation.hh.

Referenced by initialize\_callback(), and receive\_lag\_compensation().

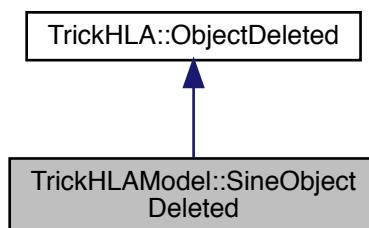
The documentation for this class was generated from the following files:

- [SineLagCompensation.hh](#)
- [SineLagCompensation.cpp](#)

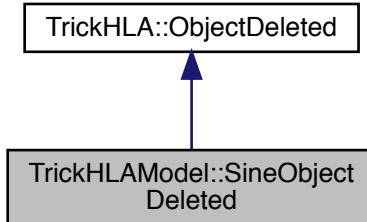
## 7.59 TrickHLAModel::SineObjectDeleted Class Reference

`#include <SineObjectDeleted.hh>`

Inheritance diagram for TrickHLAModel::SineObjectDeleted:



Collaboration diagram for TrickHLAModel::SineObjectDeleted:



## Public Member Functions

- `SineObjectDeleted ()`  
`Default constructor for the TrickHLAModel SineObjectDeleted class.`
- `virtual ~SineObjectDeleted ()`  
`Destructor for the TrickHLAModel SineObjectDeleted class.`
- `void deleted (TrickHLA::Object *obj)`  
`Callback routine implementation to report that this object has been deleted from the RTI.`

## Private Member Functions

- `SineObjectDeleted (const SineObjectDeleted &rhs)`  
`Copy constructor for SineObjectDeleted class.`
- `SineObjectDeleted & operator= (const SineObjectDeleted &rhs)`  
`Assignment operator for SineObjectDeleted class.`

## Friends

- `class InputProcessor`
- `void init_attrTrickHLAModel__SineObjectDeleted ()`

### 7.59.1 Detailed Description

Definition at line 47 of file SineObjectDeleted.hh.

### 7.59.2 Constructor & Destructor Documentation

#### 7.59.2.1 SineObjectDeleted() [1/2]

`SineObjectDeleted::SineObjectDeleted ( )`  
`Default constructor for the TrickHLAModel SineObjectDeleted class.`

**Trick Job Class:** *initialization*

Definition at line 49 of file SineObjectDeleted.cpp.

**7.59.2.2 ~SineObjectDeleted()**

`SineObjectDeleted::~SineObjectDeleted ( ) [virtual]`  
 Destructor for the [TrickHLAModel SineObjectDeleted](#) class.

**Trick Job Class:** *shutdown*

Definition at line 58 of file SineObjectDeleted.cpp.

**7.59.2.3 SineObjectDeleted() [2/2]**

`TrickHLAModel::SineObjectDeleted::SineObjectDeleted (`  
`const SineObjectDeleted & rhs ) [private]`

Copy constructor for [SineObjectDeleted](#) class.

This constructor is private to prevent inadvertent copies.

**7.59.3 Member Function Documentation****7.59.3.1 deleted()**

`void SineObjectDeleted::deleted (`  
`TrickHLA::Object * obj ) [virtual]`

Callback routine implementation to report that this object has been deleted from the RTI.

**Parameters**

<code>obj</code>	Object which was deleted.
------------------	---------------------------

Reimplemented from [TrickHLA::ObjectDeleted](#).

Definition at line 63 of file SineObjectDeleted.cpp.

References [TrickHLA::Object::get\\_name\(\)](#).

**7.59.3.2 operator=()**

`SineObjectDeleted& TrickHLAModel::SineObjectDeleted::operator= (`  
`const SineObjectDeleted & rhs ) [private]`

Assignment operator for [SineObjectDeleted](#) class.

This assignment operator is private to prevent inadvertent copies.

**7.59.4 Friends And Related Function Documentation****7.59.4.1 init\_attrTrickHLAModel\_\_SineObjectDeleted**

`void init_attrTrickHLAModel__SineObjectDeleted ( ) [friend]`

**7.59.4.2 InputProcessor**

`friend class InputProcessor [friend]`

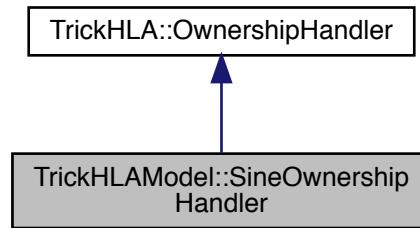
Definition at line 54 of file SineObjectDeleted.hh.

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

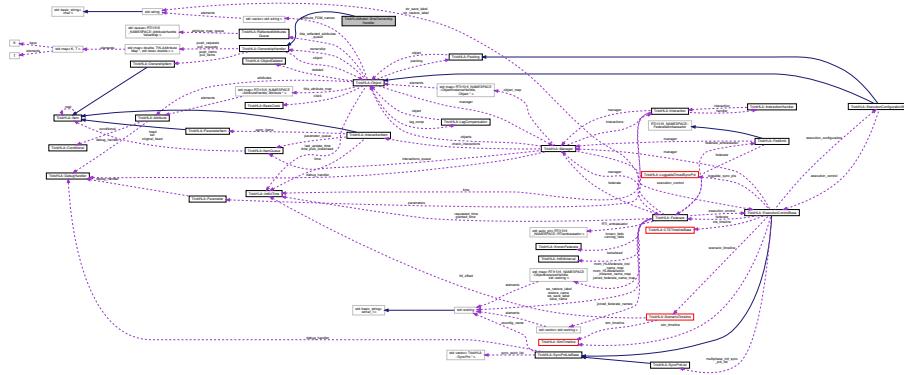
- [SineObjectDeleted.hh](#)
- [SineObjectDeleted.cpp](#)

## 7.60 TrickHLAModel::SineOwnershipHandler Class Reference

```
#include <SineOwnershipHandler.hh>
Inheritance diagram for TrickHLAModel::SineOwnershipHandler:
```



Collaboration diagram for TrickHLAModel::SineOwnershipHandler:



### Public Member Functions

- [SineOwnershipHandler \(\)](#)  
*Default constructor for the `TrickHLAModel SineOwnershipHandler` class.*
- [virtual ~SineOwnershipHandler \(\)](#)  
*Destructor for the `TrickHLAModel SineOwnershipHandler` class.*
- [virtual void initialize\\_callback \(TrickHLA::Object \\*obj\)](#)  
*Initialization callback as part of the `TrickHLA::OwnershipHandler` functions.*

### Private Member Functions

- [SineOwnershipHandler \(const SineOwnershipHandler &rhs\)](#)

*Copy constructor for [SineOwnershipHandler](#) class.*

- [SineOwnershipHandler & operator= \(const SineOwnershipHandler &rhs\)](#)

*Assignment operator for [SineOwnershipHandler](#) class.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrTrickHLAModel\\_\\_SineOwnershipHandler \(\)](#)

## Additional Inherited Members

### 7.60.1 Detailed Description

Definition at line 46 of file [SineOwnershipHandler.hh](#).

### 7.60.2 Constructor & Destructor Documentation

#### 7.60.2.1 [SineOwnershipHandler\(\) \[1/2\]](#)

`SineOwnershipHandler::SineOwnershipHandler ()`

Default constructor for the [TrickHLAModel SineOwnershipHandler](#) class.

**Trick Job Class:** *initialization*

Definition at line 46 of file [SineOwnershipHandler.cpp](#).

#### 7.60.2.2 [~SineOwnershipHandler\(\)](#)

`SineOwnershipHandler::~SineOwnershipHandler () [virtual]`

Destructor for the [TrickHLAModel SineOwnershipHandler](#) class.

**Trick Job Class:** *shutdown*

Definition at line 53 of file [SineOwnershipHandler.cpp](#).

#### 7.60.2.3 [SineOwnershipHandler\(\) \[2/2\]](#)

`TrickHLAModel::SineOwnershipHandler::SineOwnershipHandler (`  
`const SineOwnershipHandler & rhs ) [private]`

Copy constructor for [SineOwnershipHandler](#) class.

This constructor is private to prevent inadvertent copies.

### 7.60.3 Member Function Documentation

#### 7.60.3.1 [initialize\\_callback\(\)](#)

```
void SineOwnershipHandler::initialize_callback (
```

`TrickHLA::Object * obj ) [virtual]`

Initialization callback as part of the [TrickHLA::OwnershipHandler](#) functions.

#### Parameters

<code>obj</code>	Object associated with this OwnershipHandler class.
------------------	---

From the `TrickHLA::OwnershipHandler` class. We override this function so that we can initialize ownership transfer of some attributes at a specific time.

**Trick Job Class:** *initialization*

Reimplemented from `TrickHLA::OwnershipHandler`.

Definition at line 65 of file `SineOwnershipHandler.cpp`.

#### 7.60.3.2 `operator=()`

```
SineOwnershipHandler& TrickHLAModel::SineOwnershipHandler::operator= (
    const SineOwnershipHandler & rhs ) [private]
```

Assignment operator for `SineOwnershipHandler` class.

This assignment operator is private to prevent inadvertent copies.

### 7.60.4 Friends And Related Function Documentation

#### 7.60.4.1 `init_attrTrickHLAModel__SineOwnershipHandler`

```
void init_attrTrickHLAModel__SineOwnershipHandler ( ) [friend]
```

#### 7.60.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 53 of file `SineOwnershipHandler.hh`.

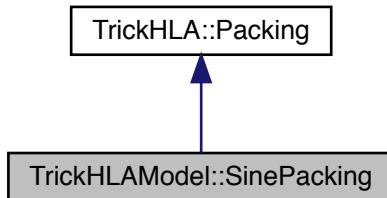
The documentation for this class was generated from the following files:

- `SineOwnershipHandler.hh`
- `SineOwnershipHandler.cpp`

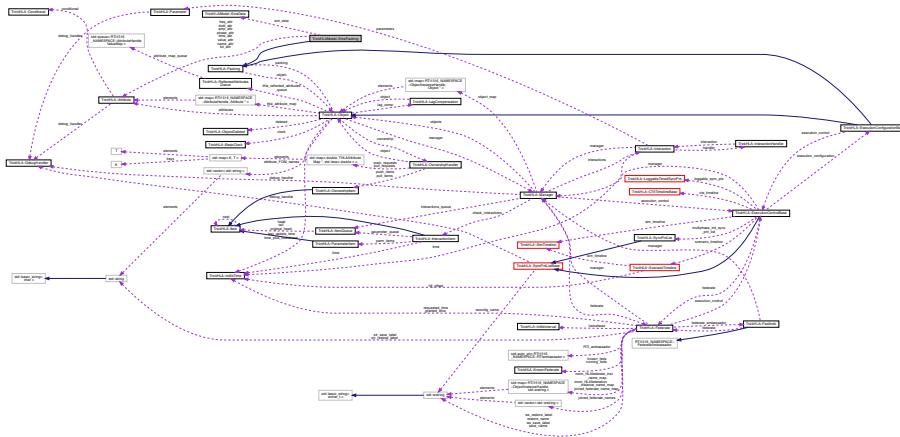
## 7.61 TrickHLAModel::SinePacking Class Reference

```
#include <SinePacking.hh>
```

Inheritance diagram for `TrickHLAModel::SinePacking`:



Collaboration diagram for TrickHLAModel::SinePacking:



## Public Member Functions

- `SinePacking ()`  
*Default constructor for the [TrickHLAModel SinePacking](#) class.*
- `virtual ~SinePacking ()`  
*Destructor for the [TrickHLAModel SinePacking](#) class.*
- `void initialize (SineData *sim_data)`  
*Initialize the packing object.*
- `virtual void initialize_callback (TrickHLA::Object *obj)`  
*Initialization callback as part of the [TrickHLA::Packing](#) functions.*
- `virtual void pack ()`  
*Called to pack the data before the data is sent to the RTI.*
- `virtual void unpack ()`  
*Called to unpack the data after data is received from the RTI.*

## Data Fields

- `SineData * sim_data`

**Units:** –  
*Simulation data.*
- `double phase_deg`

**Units:** degree  
*Phase offset in degrees.*
- `int pack_count`

**Units:** count  
*The number of times the pack routine has been called.*
- `bool initialized`

**Units:** –  
*Flag to indicate this class has been initialized.*
- `int buff_size`

**Units:** –  
*Size of the byte buffer.*
- `unsigned char * buff`

- Units:** –  
Byte buffer.
- `TrickHLA::Attribute * time_attr`

**Data I/O:** \*\*  
Reference to the "Time" `TrickHLA::Attribute`.
  - `TrickHLA::Attribute * value_attr`

**Data I/O:** \*\*  
Reference to the "Value" `TrickHLA::Attribute`.
  - `TrickHLA::Attribute * dvdt_attr`

**Data I/O:** \*\*  
Reference to the "dvdt" `TrickHLA::Attribute`.
  - `TrickHLA::Attribute * phase_attr`

**Data I/O:** \*\*  
Reference to the "Phase" `TrickHLA::Attribute`.
  - `TrickHLA::Attribute * freq_attr`

**Data I/O:** \*\*  
Reference to the "Frequency" `TrickHLA::Attribute`.
  - `TrickHLA::Attribute * amp_attr`

**Data I/O:** \*\*  
Reference to the "Amplitude" `TrickHLA::Attribute`.
  - `TrickHLA::Attribute * tol_attr`

**Data I/O:** \*\*  
Reference to the "Tolerance" `TrickHLA::Attribute`.
  - `TrickHLA::Attribute * name_attr`

**Data I/O:** \*\*  
Reference to the "Name" `TrickHLA::Attribute`.

## Private Member Functions

- `SinePacking (const SinePacking &rhs)`  
Copy constructor for `SinePacking` class.
- `SinePacking & operator= (const SinePacking &rhs)`  
Assignment operator for `SinePacking` class.

## Friends

- class `InputProcessor`
- void `init_attrTrickHLAModel__SinePacking ()`

## Additional Inherited Members

### 7.61.1 Detailed Description

Definition at line 50 of file `SinePacking.hh`.

### 7.61.2 Constructor & Destructor Documentation

### 7.61.2.1 SinePacking() [1/2]

`SinePacking::SinePacking ( )`  
 Default constructor for the [TrickHLAModel SinePacking](#) class.  
**Trick Job Class:** *initialization*  
 Definition at line 56 of file SinePacking.cpp.

### 7.61.2.2 ~SinePacking()

`SinePacking::~SinePacking ( ) [virtual]`  
 Destructor for the [TrickHLAModel SinePacking](#) class.  
**Trick Job Class:** *shutdown*  
 Definition at line 78 of file SinePacking.cpp.  
 References `buff`, and `buff_size`.

### 7.61.2.3 SinePacking() [2/2]

`TrickHLAModel::SinePacking::SinePacking (`  
`const SinePacking & rhs ) [private]`  
 Copy constructor for [SinePacking](#) class.  
 This constructor is private to prevent inadvertent copies.

## 7.61.3 Member Function Documentation

### 7.61.3.1 initialize()

`void SinePacking::initialize (`  
`SineData * sim_data )`  
 Initialize the packing object.

#### Parameters

<code>sim_data</code>	The sine wave data object for packing and unpacking.
-----------------------	--

**Trick Job Class:** *initialization*

Definition at line 93 of file SinePacking.cpp.  
 References `initialized`, and `sim_data`.

### 7.61.3.2 initialize\_callback()

`void SinePacking::initialize_callback (`  
`TrickHLA::Object * obj ) [virtual]`  
 Initialization callback as part of the [TrickHLA::Packing](#) functions.

#### Parameters

<code>obj</code>	Object associated with this packing class.
------------------	--

From the [TrickHLA::Packing](#) class. We override this function so that we can initialize references to the [TrickHLA::Attribute](#)'s that are used in the unpack function to handle attribute ownership and different attribute data

rates. Use the initialize callback function as a way to setup TrickHLA-Attribute references which are used to determine ownership or if data for an attribute was received.

**Trick Job Class:** *initialization*

Reimplemented from [TrickHLA::Packing](#).

Definition at line 111 of file SinePacking.cpp.

References amp\_attr, dvdt\_attr, freq\_attr, TrickHLA::Packing::get\_attribute\_and\_validate(), name\_attr, phase\_attr, time\_attr, tol\_attr, and value\_attr.

#### 7.61.3.3 operator=( )

```
SinePacking& TrickHLAModel::SinePacking::operator= (
    const SinePacking & rhs ) [private]
```

Assignment operator for [SinePacking](#) class.

This assignment operator is private to prevent inadvertent copies.

#### 7.61.3.4 pack()

```
void SinePacking::pack ( ) [virtual]
```

Called to pack the data before the data is sent to the RTI.

Implements [TrickHLA::Packing](#).

Definition at line 131 of file SinePacking.cpp.

References amp\_attr, buff, buff\_size, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_LEVEL\_6\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, dvdt\_attr, freq\_attr, TrickHLAModel::SineData::get\_amplitude(), TrickHLA::Packing::get\_attribute(), TrickHLA::Attribute::get\_attribute\_size(), TrickHLAModel::SineData::get\_derivative(), TrickHLAModel::SineData::get\_frequency(), TrickHLAModel::SineData::get\_name(), TrickHLA::Object::get\_name\_string(), TrickHLAModel::SineData::get\_phase(), TrickHLA::Attribute::get\_sim\_variable\_address(), TrickHLAModel::SineData::get\_time(), TrickHLAModel::SineData::get\_tolerance(), TrickHLAModel::SineData::get\_value(), initialized, TrickHLA::Attribute::is\_locally\_owned(), TrickHLA::Attribute::is\_publish(), name\_attr, TrickHLA::Packing::object, pack\_count, phase\_attr, phase\_deg, TrickHLA::Attribute::print\_buffer(), TrickHLA::Packing::should\_print(), sim\_data, time\_attr, tol\_attr, and value\_attr.

#### 7.61.3.5 unpack()

```
void SinePacking::unpack ( ) [virtual]
```

Called to unpack the data after data is received from the RTI.

Implements [TrickHLA::Packing](#).

Definition at line 241 of file SinePacking.cpp.

References amp\_attr, buff, buff\_size, TrickHLA::DEBUG\_LEVEL\_1\_TRACE, TrickHLA::DEBUG\_LEVEL\_6\_TRACE, TrickHLA::DEBUG\_SOURCE\_PACKING, dvdt\_attr, freq\_attr, TrickHLAModel::SineData::get\_amplitude(), TrickHLA::Packing::get\_attribute(), TrickHLAModel::SineData::get\_derivative(), TrickHLAModel::SineData::get\_frequency(), TrickHLAModel::SineData::get\_name(), TrickHLA::Object::get\_name\_string(), TrickHLAModel::SineData::get\_phase(), TrickHLA::Attribute::get\_sim\_variable\_address(), TrickHLAModel::SineData::get\_time(), TrickHLAModel::SineData::get\_tolerance(), TrickHLAModel::SineData::get\_value(), initialized, TrickHLA::Attribute::is\_received(), name\_attr, TrickHLA::Packing::object, phase\_attr, phase\_deg, TrickHLA::Attribute::print\_buffer(), TrickHLAModel::SineData::set\_phase(), TrickHLA::Packing::should\_print(), sim\_data, time\_attr, tol\_attr, and value\_attr.

### 7.61.4 Friends And Related Function Documentation

#### 7.61.4.1 init\_attrTrickHLAModel\_\_SinePacking

```
void init_attrTrickHLAModel__SinePacking ( ) [friend]
```

#### 7.61.4.2 InputProcessor

friend class InputProcessor [friend]  
Definition at line 57 of file SinePacking.hh.

### 7.61.5 Field Documentation

#### 7.61.5.1 amp\_attr

`TrickHLA::Attribute*` `TrickHLAModel::SinePacking::amp_attr`  
**Data I/O:** \*\*  
Reference to the "Amplitude" `TrickHLA::Attribute`.  
Definition at line 81 of file SinePacking.hh.  
Referenced by `initialize_callback()`, `pack()`, and `unpack()`.

#### 7.61.5.2 buff

`unsigned char*` `TrickHLAModel::SinePacking::buff`  
**Units:** –  
Byte buffer.  
Definition at line 74 of file SinePacking.hh.  
Referenced by `pack()`, `unpack()`, and `~SinePacking()`.

#### 7.61.5.3 buff\_size

`int` `TrickHLAModel::SinePacking::buff_size`  
**Units:** –  
Size of the byte buffer.  
Definition at line 72 of file SinePacking.hh.  
Referenced by `pack()`, `unpack()`, and `~SinePacking()`.

#### 7.61.5.4 dvdt\_attr

`TrickHLA::Attribute*` `TrickHLAModel::SinePacking::dvdt_attr`  
**Data I/O:** \*\*  
Reference to the "dvdt" `TrickHLA::Attribute`.  
Definition at line 78 of file SinePacking.hh.  
Referenced by `initialize_callback()`, `pack()`, and `unpack()`.

#### 7.61.5.5 freq\_attr

`TrickHLA::Attribute*` `TrickHLAModel::SinePacking::freq_attr`  
**Data I/O:** \*\*  
Reference to the "Frequency" `TrickHLA::Attribute`.  
Definition at line 80 of file SinePacking.hh.  
Referenced by `initialize_callback()`, `pack()`, and `unpack()`.

### 7.61.5.6 initialized

```
bool TrickHLAModel::SinePacking::initialized
```

**Units:** –

Flag to indicate this class has been initialized.

Definition at line 70 of file SinePacking.hh.

Referenced by initialize(), pack(), and unpack().

### 7.61.5.7 name\_attr

```
TrickHLA::Attribute* TrickHLAModel::SinePacking::name_attr
```

**Data I/O:** \*\*

Reference to the "Name" [TrickHLA::Attribute](#).

Definition at line 83 of file SinePacking.hh.

Referenced by initialize\_callback(), pack(), and unpack().

### 7.61.5.8 pack\_count

```
int TrickHLAModel::SinePacking::pack_count
```

**Units:** count

The number of times the pack routine has been called.

Definition at line 68 of file SinePacking.hh.

Referenced by pack().

### 7.61.5.9 phase\_attr

```
TrickHLA::Attribute* TrickHLAModel::SinePacking::phase_attr
```

**Data I/O:** \*\*

Reference to the "Phase" [TrickHLA::Attribute](#).

Definition at line 79 of file SinePacking.hh.

Referenced by initialize\_callback(), pack(), and unpack().

### 7.61.5.10 phase\_deg

```
double TrickHLAModel::SinePacking::phase_deg
```

**Units:** degree

Phase offset in degrees.

Definition at line 66 of file SinePacking.hh.

Referenced by pack(), and unpack().

### 7.61.5.11 sim\_data

```
SineData* TrickHLAModel::SinePacking::sim_data
```

**Units:** –

Simulation data.

Definition at line 64 of file SinePacking.hh.

Referenced by initialize(), pack(), and unpack().

### 7.61.5.12 time\_attr

`TrickHLA::Attribute* TrickHLAModel::SinePacking::time_attr`

**Data I/O:** \*\*

Reference to the "Time" `TrickHLA::Attribute`.

Definition at line 76 of file `SinePacking.hh`.

Referenced by `initialize_callback()`, `pack()`, and `unpack()`.

### 7.61.5.13 tol\_attr

`TrickHLA::Attribute* TrickHLAModel::SinePacking::tol_attr`

**Data I/O:** \*\*

Reference to the "Tolerance" `TrickHLA::Attribute`.

Definition at line 82 of file `SinePacking.hh`.

Referenced by `initialize_callback()`, `pack()`, and `unpack()`.

### 7.61.5.14 value\_attr

`TrickHLA::Attribute* TrickHLAModel::SinePacking::value_attr`

**Data I/O:** \*\*

Reference to the "Value" `TrickHLA::Attribute`.

Definition at line 77 of file `SinePacking.hh`.

Referenced by `initialize_callback()`, `pack()`, and `unpack()`.

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

- `SinePacking.hh`
- `SinePacking.cpp`

## 7.62 SpaceTimeCoordinateData Struct Reference

```
#include <SpaceTimeCoordinateData.h>
```

### Data Fields

- double `pos` [3]  
`trick_units{m}` Position in parent frame.
- double `vel` [3]  
`trick_units{m/s}` Velocity wrt. parent frame.
- double `quat_scalar`  
`trick_units{-}` Attitude quaternion scalar.
- double `quat_vector` [3]  
`trick_units{-}` Attitude quaternion vector.
- double `ang_vel` [3]  
`trick_units{rad/s}` Angular velocity vector.
- double `time`  
`trick_units{s}` Truncated Julian date in TT time scale.

### 7.62.1 Detailed Description

Definition at line 37 of file `SpaceTimeCoordinateData.h`.

## 7.62.2 Field Documentation

### 7.62.2.1 `ang_vel`

double SpaceTimeCoordinateData::ang\_vel[3]

trick\_units{rad/s} Angular velocity vector.

Definition at line 43 of file SpaceTimeCoordinateData.h.

Referenced by SpaceFOM::RefFrameBase::pack(), SpaceFOM::SpaceTimeCoordinateEncoder::SpaceTimeCoordinateEncoder(), and SpaceFOM::RefFrameBase::unpack().

### 7.62.2.2 `pos`

double SpaceTimeCoordinateData::pos[3]

trick\_units{m} Position in parent frame.

Definition at line 39 of file SpaceTimeCoordinateData.h.

Referenced by SpaceFOM::RefFrameBase::pack(), SpaceFOM::SpaceTimeCoordinateEncoder::SpaceTimeCoordinateEncoder(), and SpaceFOM::RefFrameBase::unpack().

### 7.62.2.3 `quat_scalar`

double SpaceTimeCoordinateData::quat\_scalar

trick\_units{—} Attitude quaternion scalar.

Definition at line 41 of file SpaceTimeCoordinateData.h.

Referenced by SpaceFOM::RefFrameBase::pack(), and SpaceFOM::RefFrameBase::unpack().

### 7.62.2.4 `quat_vector`

double SpaceTimeCoordinateData::quat\_vector[3]

trick\_units{—} Attitude quaternion vector.

Definition at line 42 of file SpaceTimeCoordinateData.h.

Referenced by SpaceFOM::RefFrameBase::pack(), SpaceFOM::SpaceTimeCoordinateEncoder::SpaceTimeCoordinateEncoder(), and SpaceFOM::RefFrameBase::unpack().

### 7.62.2.5 `time`

double SpaceTimeCoordinateData::time

trick\_units{s} Truncated Julian date in TT time scale.

Definition at line 44 of file SpaceTimeCoordinateData.h.

Referenced by SpaceFOM::PhysicalEntityBase::get\_time(), SpaceFOM::RefFrameBase::pack(), and SpaceFOM::RefFrameBase::unpack().

### 7.62.2.6 `vel`

double SpaceTimeCoordinateData::vel[3]

trick\_units{m/s} Velocity wrt. parent frame.

Definition at line 40 of file SpaceTimeCoordinateData.h.

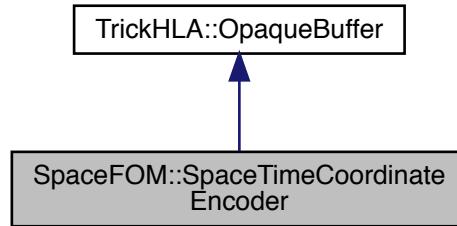
Referenced by SpaceFOM::RefFrameBase::pack(), SpaceFOM::SpaceTimeCoordinateEncoder::SpaceTimeCoordinateEncoder(), and SpaceFOM::RefFrameBase::unpack().

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

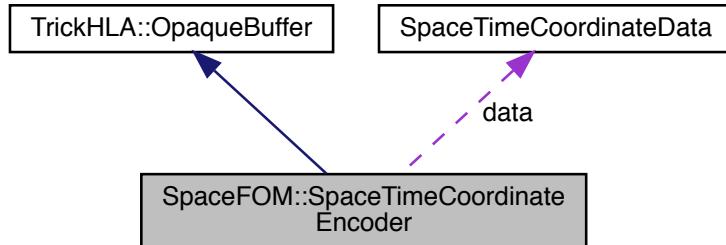
- [SpaceTimeCoordinateData.h](#)

## 7.63 SpaceFOM::SpaceTimeCoordinateEncoder Class Reference

```
#include <SpaceTimeCoordinateEncoder.hh>
Inheritance diagram for SpaceFOM::SpaceTimeCoordinateEncoder:
```



Collaboration diagram for SpaceFOM::SpaceTimeCoordinateEncoder:



### Public Member Functions

- [SpaceTimeCoordinateEncoder \(\)](#)  
*Default constructor for the SpaceFOM SpaceTimeCoordinateEncoder class.*
- [void encode \(\)](#)  
*Encode the spacetime coordinate data for sending out.*
- [void decode \(\)](#)  
*Decode the incoming spacetime coordinate data.*
- [SpaceTimeCoordinateData & get\\_data \(\)](#)  
*Get the spacetime coordinate data.*

## Protected Attributes

- `SpaceTimeCoordinateData data`  
`trick_units{--}` Reference frame transmission data.
- `rti1516e::HLAfloat64LE position` [3]  
`Data I/O: **`  
`HLAfloat64LE position array`
- `rti1516e::HLAfixedArray position_encoder`  
`Data I/O: **`  
`Position encoder`
- `rti1516e::HLAfloat64LE velocity` [3]  
`Data I/O: **`  
`HLAfloat64LE velocity array`
- `rti1516e::HLAfixedArray velocity_encoder`  
`Data I/O: **`  
`Velocity encoder`
- `rti1516e::HLAfixedRecord trans_state_encoder`  
`Data I/O: **`  
`Translational state encoder`
- `rti1516e::HLAfloat64LE quat_scalar_encoder`  
`Data I/O: **`  
`Quaternion scalar encoder`
- `rti1516e::HLAfloat64LE quat_vector` [3]  
`Data I/O: **`  
`HLAfloat64LE quaternion vector`
- `rti1516e::HLAfixedArray quat_vector_encoder`  
`Data I/O: **`  
`Quaternion vector encoder`
- `rti1516e::HLAfixedRecord quat_encoder`  
`Data I/O: **`  
`Attitude quaternion encoder`
- `rti1516e::HLAfloat64LE angular_velocity` [3]  
`Data I/O: **`  
`HLAfloat64LE angular velocity array.`
- `rti1516e::HLAfixedArray ang_vel_encoder`  
`Data I/O: **`  
`Angular velocity encoder`
- `rti1516e::HLAfixedRecord rot_state_encoder`  
`Data I/O: **`  
`Rotational state encoder`
- `rti1516e::HLAfloat64LE time_encoder`  
`Data I/O: **`  
`Time encoder`
- `rti1516e::HLAfixedRecord encoder`  
`Data I/O: **`  
`Space/Time coordinate encoder`

## Private Member Functions

- `SpaceTimeCoordinateEncoder (const SpaceTimeCoordinateEncoder &)`  
`Copy constructor for SpaceTimeCoordinateEncoder class.`
- `SpaceTimeCoordinateEncoder & operator= (const SpaceTimeCoordinateEncoder &)`  
`Assignment operator for SpaceTimeCoordinateEncoder class.`

## Friends

- class [InputProcessor](#)
- void [init\\_attrSpaceFOM\\_\\_SpaceTimeCoordinateEncoder \(\)](#)

## Additional Inherited Members

### 7.63.1 Detailed Description

Definition at line 50 of file SpaceTimeCoordinateEncoder.hh.

### 7.63.2 Constructor & Destructor Documentation

#### 7.63.2.1 SpaceTimeCoordinateEncoder() [1/2]

`SpaceTimeCoordinateEncoder::SpaceTimeCoordinateEncoder ()`  
Default constructor for the [SpaceFOM SpaceTimeCoordinateEncoder](#) class.

**Trick Job Class:** *initialization*

Definition at line 47 of file SpaceTimeCoordinateEncoder.cpp.

References `SpaceTimeCoordinateData::ang_vel`, `ang_vel_encoder`, `angular_velocity`, `data`, `encoder`, `TrickHLA::OpaqueBuffer::ensure_buffer_capacity()`, `SpaceTimeCoordinateData::pos`, `position`, `position_encoder`, `quat_encoder`, `quat_scalar_encoder`, `SpaceTimeCoordinateData::quat_vector`, `quat_vector`, `quat_vector_encoder`, `rot_state_encoder`, `TrickHLA::OpaqueBuffer::set_byte_alignment()`, `time_encoder`, `trans_state_encoder`, `SpaceTimeCoordinateData::vel`, `velocity`, and `velocity_encoder`.

#### 7.63.2.2 SpaceTimeCoordinateEncoder() [2/2]

`SpaceFOM::SpaceTimeCoordinateEncoder::SpaceTimeCoordinateEncoder (`  
    `const SpaceTimeCoordinateEncoder & )` [private]

Copy constructor for [SpaceTimeCoordinateEncoder](#) class.

This constructor is private to prevent inadvertent copies.

### 7.63.3 Member Function Documentation

#### 7.63.3.1 decode()

`void SpaceTimeCoordinateEncoder::decode ()`

Decode the incoming spacetime coordinate data.

**Trick Job Class:** *scheduled*

Definition at line 153 of file SpaceTimeCoordinateEncoder.cpp.

References `TrickHLA::OpaqueBuffer::buffer`, `TrickHLA::OpaqueBuffer::capacity`, and `encoder`.

Referenced by `SpaceFOM::PhysicalEntityBase::unpack()`, and `SpaceFOM::RefFrameBase::unpack()`.

#### 7.63.3.2 encode()

`void SpaceTimeCoordinateEncoder::encode ()`

Encode the spacetime coordinate data for sending out.

**Trick Job Class:** *scheduled*

Definition at line 128 of file SpaceTimeCoordinateEncoder.cpp.

References TrickHLA::OpaqueBuffer::buffer, encoder, and TrickHLA::OpaqueBuffer::get\_capacity().  
 Referenced by SpaceFOM::PhysicalEntityBase::pack(), and SpaceFOM::RefFrameBase::pack().

#### 7.63.3.3 `get_data()`

`SpaceTimeCoordinateData& SpaceFOM::SpaceTimeCoordinateEncoder::get_data ()` [inline]  
 Get the spacetim coordinate data.

##### Returns

A reference to the `SpaceTimeCoordinateData`.

Definition at line 73 of file `SpaceTimeCoordinateEncoder.hh`.

References data.

#### 7.63.3.4 `operator=()`

`SpaceTimeCoordinateEncoder& SpaceFOM::SpaceTimeCoordinateEncoder::operator= (`  
`const SpaceTimeCoordinateEncoder & )` [private]

Assignment operator for `SpaceTimeCoordinateEncoder` class.

This assignment operator is private to prevent inadvertent copies.

### 7.63.4 Friends And Related Function Documentation

#### 7.63.4.1 `init_attrSpaceFOM__SpaceTimeCoordinateEncoder`

`void init_attrSpaceFOM__SpaceTimeCoordinateEncoder ()` [friend]

#### 7.63.4.2 `InputProcessor`

`friend class InputProcessor` [friend]

Definition at line 57 of file `SpaceTimeCoordinateEncoder.hh`.

### 7.63.5 Field Documentation

#### 7.63.5.1 `ang_vel_encoder`

`rti1516e::HLAfixedArray SpaceFOM::SpaceTimeCoordinateEncoder::ang_vel_encoder` [protected]

##### **Data I/O: \*\***

Angular velocity encoder

Definition at line 90 of file `SpaceTimeCoordinateEncoder.hh`.

Referenced by `SpaceTimeCoordinateEncoder()`.

#### 7.63.5.2 `angular_velocity`

`rti1516e::HLAfloat64LE SpaceFOM::SpaceTimeCoordinateEncoder::angular_velocity[3]` [protected]

##### **Data I/O: \*\***

HLAfloat64LE angular velocity array.

Definition at line 89 of file SpaceTimeCoordinateEncoder.hh.  
Referenced by SpaceTimeCoordinateEncoder().

#### 7.63.5.3 data

```
SpaceTimeCoordinateData SpaceFOM::SpaceTimeCoordinateEncoder::data [protected]
trick_units{-->} Reference frame transmission data.
Definition at line 76 of file SpaceTimeCoordinateEncoder.hh.
Referenced by get_data(), and SpaceTimeCoordinateEncoder().
```

#### 7.63.5.4 encoder

```
rti1516e::HLAfixedRecord SpaceFOM::SpaceTimeCoordinateEncoder::encoder [protected]
Data I/O: **
Space/Time coordinate encoder
Definition at line 95 of file SpaceTimeCoordinateEncoder.hh.
Referenced by decode(), encode(), and SpaceTimeCoordinateEncoder().
```

#### 7.63.5.5 position

```
rti1516e::HLAfloat64LE SpaceFOM::SpaceTimeCoordinateEncoder::position[3] [protected]
Data I/O: **
HLAfloat64LE position array
Definition at line 79 of file SpaceTimeCoordinateEncoder.hh.
Referenced by SpaceTimeCoordinateEncoder().
```

#### 7.63.5.6 position\_encoder

```
rti1516e::HLAfixedArray SpaceFOM::SpaceTimeCoordinateEncoder::position_encoder [protected]
Data I/O: **
Position encoder
Definition at line 80 of file SpaceTimeCoordinateEncoder.hh.
Referenced by SpaceTimeCoordinateEncoder().
```

#### 7.63.5.7 quat\_encoder

```
rti1516e::HLAfixedRecord SpaceFOM::SpaceTimeCoordinateEncoder::quat_encoder [protected]
Data I/O: **
Attitude quaternion encoder
Definition at line 88 of file SpaceTimeCoordinateEncoder.hh.
Referenced by SpaceTimeCoordinateEncoder().
```

#### 7.63.5.8 quat\_scalar\_encoder

```
rti1516e::HLAfloat64LE SpaceFOM::SpaceTimeCoordinateEncoder::quat_scalar_encoder [protected]
Data I/O: **
Quaternion scalar encoder
Definition at line 85 of file SpaceTimeCoordinateEncoder.hh.
Referenced by SpaceTimeCoordinateEncoder().
```

### 7.63.5.9 quat\_vector

```
rti1516e::HLAfloat64LE SpaceFOM::SpaceTimeCoordinateEncoder::quat_vector[3] [protected]
```

**Data I/O:** \*\*

HLAfloat64LE quaternion vector

Definition at line 86 of file SpaceTimeCoordinateEncoder.hh.

Referenced by SpaceTimeCoordinateEncoder().

### 7.63.5.10 quat\_vector\_encoder

```
rti1516e::HLAfixedArray SpaceFOM::SpaceTimeCoordinateEncoder::quat_vector_encoder [protected]
```

**Data I/O:** \*\*

Quaternion vector encoder

Definition at line 87 of file SpaceTimeCoordinateEncoder.hh.

Referenced by SpaceTimeCoordinateEncoder().

### 7.63.5.11 rot\_state\_encoder

```
rti1516e::HLAfixedRecord SpaceFOM::SpaceTimeCoordinateEncoder::rot_state_encoder [protected]
```

**Data I/O:** \*\*

Rotational state encoder

Definition at line 91 of file SpaceTimeCoordinateEncoder.hh.

Referenced by SpaceTimeCoordinateEncoder().

### 7.63.5.12 time\_encoder

```
rti1516e::HLAfloat64LE SpaceFOM::SpaceTimeCoordinateEncoder::time_encoder [protected]
```

**Data I/O:** \*\*

Time encoder

Definition at line 93 of file SpaceTimeCoordinateEncoder.hh.

Referenced by SpaceTimeCoordinateEncoder().

### 7.63.5.13 trans\_state\_encoder

```
rti1516e::HLAfixedRecord SpaceFOM::SpaceTimeCoordinateEncoder::trans_state_encoder [protected]
```

**Data I/O:** \*\*

Translational state encoder

Definition at line 83 of file SpaceTimeCoordinateEncoder.hh.

Referenced by SpaceTimeCoordinateEncoder().

### 7.63.5.14 velocity

```
rti1516e::HLAfloat64LE SpaceFOM::SpaceTimeCoordinateEncoder::velocity[3] [protected]
```

**Data I/O:** \*\*

HLAfloat64LE velocity array

Definition at line 81 of file SpaceTimeCoordinateEncoder.hh.

Referenced by SpaceTimeCoordinateEncoder().

### 7.63.5.15 velocity\_encoder

`rti1516e::HLAfixedArray SpaceFOM::SpaceTimeCoordinateEncoder::velocity_encoder [protected]`

**Data I/O: \*\***

Velocity encoder

Definition at line 82 of file SpaceTimeCoordinateEncoder.hh.

Referenced by SpaceTimeCoordinateEncoder().

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

- [SpaceTimeCoordinateEncoder.hh](#)
- [SpaceTimeCoordinateEncoder.cpp](#)

## 7.64 TrickHLA::StringUtilities Class Reference

#include <StringUtilities.hh>

### Public Member Functions

- [StringUtilities \(\)](#)  
*Default constructor for the `TrickHLA StringUtilities` class.*
- [virtual ~StringUtilities \(\)](#)  
*Destructor for the `TrickHLA StringUtilities` class.*

### Static Public Member Functions

- [static void to\\_wstring \(std::wstring &output, const char \\*input\)](#)  
*C (char \*) string to C++ wide string conversion routine.*
- [static void to\\_wstring \(std::wstring &output, std::string const &input\)](#)  
*C++ string to C++ wide string conversion routine.*
- [static void to\\_string \(std::string &output, std::wstring const &input\)](#)  
*C++ wide string to C++ string conversion routine.*
- [static char \\* ip\\_strdup\\_wstring \(std::wstring const &input\)](#)  
*C++ wide string to C (char \*) string conversion routine with the string being placed into Trick memory space.*
- [static void to\\_printable\\_string \(std::string &output, RTI1516\\_USERDATA const &data\)](#)  
*HLA RTI User Data to printable C++ string conversion routine.*
- [static void to\\_string \(std::string &output, RTI1516\\_USERDATA const &data\)](#)  
*HLA RTI User Data to C++ string conversion routine.*
- [static void to\\_string \(std::string &output, RTI1516\\_NAMESPACE::FederateHandle handle\)](#)  
*Convert a federate handle to a C string representation.*
- [static void to\\_string \(std::string &output, RTI1516\\_NAMESPACE::InteractionClassHandle handle\)](#)  
*Convert an interaction class handle to a C string representation.*
- [static void to\\_string \(std::string &output, RTI1516\\_NAMESPACE::ParameterHandle handle\)](#)  
*Convert an interaction class handle to a C string representation.*
- [static void to\\_string \(std::string &output, RTI1516\\_NAMESPACE::ObjectInstanceHandle handle\)](#)  
*Convert an object instance handle to a C string representation.*
- [static void to\\_string \(std::string &output, RTI1516\\_NAMESPACE::ObjectClassHandle handle\)](#)  
*Convert an object class handle to a C string representation.*
- [static void to\\_string \(std::string &output, RTI1516\\_NAMESPACE::AttributeHandle handle\)](#)  
*Convert an attribute handle to a C string representation.*
- [static void trim\\_whitespace \(std::string &s\)](#)

- static void `trim_whitespace` (std::wstring &s)
 

*Trim any leading or trailing whitespace from the string.*
- static void `tokenize` (std::string const &str, std::vector< std::string > &tokens, std::string const &delimiters=",")
 

*Tokenize a given string for the specified delimiter characters.*
- static void `tokenize` (std::string const &str, std::vector< std::wstring > &tokens, std::string const &delimiters=",")
 

*Tokenize a given wstring for the specified delimiter characters.*

## Private Member Functions

- `StringUtilities` (const `StringUtilities` &rhs)
 

*Copy constructor for `StringUtilities` class.*
- `StringUtilities` & `operator=` (const `StringUtilities` &rhs)
 

*Assignment operator for `StringUtilities` class.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA__StringUtilities` ()

### 7.64.1 Detailed Description

Definition at line 64 of file `StringUtilities.hh`.

### 7.64.2 Constructor & Destructor Documentation

#### 7.64.2.1 `StringUtilities()` [1/2]

`TrickHLA::StringUtilities::StringUtilities ()` [inline]  
 Default constructor for the `TrickHLA StringUtilities` class.  
 Definition at line 81 of file `StringUtilities.hh`.

#### 7.64.2.2 `~StringUtilities()`

`virtual TrickHLA::StringUtilities::~StringUtilities ()` [inline], [virtual]  
 Destructor for the `TrickHLA StringUtilities` class.  
 Definition at line 84 of file `StringUtilities.hh`.

#### 7.64.2.3 `StringUtilities()` [2/2]

`TrickHLA::StringUtilities::StringUtilities (`  
`const StringUtilities & rhs )` [private]  
 Copy constructor for `StringUtilities` class.  
 This constructor is private to prevent inadvertent copies.

### 7.64.3 Member Function Documentation

### 7.64.3.1 ip\_strdup\_wstring()

```
static char* TrickHLA::StringUtilities::ip_strdup_wstring (
    std::wstring const & input ) [inline], [static]
```

C++ wide string to C (char \*) string conversion routine with the string being placed into Trick memory space. Make sure to use ip\_free() to free the memory otherwise you could end up with a memory leak.

#### Returns

C string.

#### Parameters

<i>input</i>	The input wide string.
--------------	------------------------

Definition at line 124 of file StringUtilities.hh.

Referenced by TrickHLA::Federate::add\_a\_single\_entry\_into\_running\_feds(), TrickHLA::TimedSyncPnt::convert(), TrickHLA::SyncPnt::convert(), TrickHLA::Federate::determine\_federate\_MOM\_object\_instance\_names(), TrickHLA::Object::register\_object\_with\_RTI(), TrickHLA::Federate::remove\_MOM\_HLAfederate\_instance\_id(), TrickHLA::Federate::set\_MOM\_HLAfederate\_instance\_attributes(), and TrickHLA::Federate::update\_running\_feds().

### 7.64.3.2 operator=( )

```
StringUtilities& TrickHLA::StringUtilities::operator= (
    const StringUtilities & rhs ) [private]
```

Assignment operator for [StringUtilities](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.64.3.3 to\_printable\_string()

```
static void TrickHLA::StringUtilities::to_printable_string (
    std::string & output,
    RTI1516_USERDATA const & data ) [inline], [static]
```

HLA RTI User Data to printable C++ string conversion routine.

#### Parameters

<i>output</i>	The output C++ string with only printable characters.
<i>data</i>	User supplied tag

Definition at line 136 of file StringUtilities.hh.

### 7.64.3.4 to\_string() [1/8]

```
static void TrickHLA::StringUtilities::to_string (
    std::string & output,
    RTI1516_NAMESPACE::AttributeHandle handle ) [inline], [static]
```

Convert an attribute handle to a C string representation.

#### Parameters

<i>output</i>	The output C++ string.
<i>handle</i>	<a href="#">Attribute</a> handle.

Definition at line 211 of file StringUtilities.hh.

References to\_string().

#### 7.64.3.5 to\_string() [2/8]

```
static void TrickHLA::StringUtilities::to_string (
    std::string & output,
    RTI1516_NAMESPACE::FederateHandle handle ) [inline], [static]
```

Convert a federate handle to a C string representation.

##### Parameters

<i>output</i>	The output C++ string.
<i>handle</i>	Federate handle.

Definition at line 161 of file StringUtilities.hh.

References to\_string().

#### 7.64.3.6 to\_string() [3/8]

```
static void TrickHLA::StringUtilities::to_string (
    std::string & output,
    RTI1516_NAMESPACE::InteractionClassHandle handle ) [inline], [static]
```

Convert an interaction class handle to a C string representation.

##### Parameters

<i>output</i>	The output C++ string.
<i>handle</i>	Class handle.

Definition at line 171 of file StringUtilities.hh.

References to\_string().

#### 7.64.3.7 to\_string() [4/8]

```
static void TrickHLA::StringUtilities::to_string (
    std::string & output,
    RTI1516_NAMESPACE::ObjectClassHandle handle ) [inline], [static]
```

Convert an object class handle to a C string representation.

##### Parameters

<i>output</i>	The output C++ string.
<i>handle</i>	Class handle.

Definition at line 201 of file StringUtilities.hh.

References to\_string().

**7.64.3.8 to\_string() [5/8]**

```
static void TrickHLA::StringUtilities::to_string (
    std::string & output,
    RTI1516_NAMESPACE::ObjectInstanceHandle handle ) [inline], [static]
```

Convert an object instance handle to a C string representation.

**Parameters**

<i>output</i>	The output C++ string.
<i>handle</i>	Instance handle.

Definition at line 191 of file StringUtilities.hh.

References to\_string().

**7.64.3.9 to\_string() [6/8]**

```
static void TrickHLA::StringUtilities::to_string (
    std::string & output,
    RTI1516_NAMESPACE::ParameterHandle handle ) [inline], [static]
```

Convert an interaction class handle to a C string representation.

**Parameters**

<i>output</i>	The output C++ string.
<i>handle</i>	Parameter Handle.

Definition at line 181 of file StringUtilities.hh.

References to\_string().

**7.64.3.10 to\_string() [7/8]**

```
static void TrickHLA::StringUtilities::to_string (
    std::string & output,
    RTI1516_USERDATA const & data ) [inline], [static]
```

HLA RTI User Data to C++ string conversion routine.

**Parameters**

<i>output</i>	The output C++ string.
<i>data</i>	User supplied tag

Definition at line 151 of file StringUtilities.hh.

**7.64.3.11 to\_string() [8/8]**

```
static void TrickHLA::StringUtilities::to_string (
    std::string & output,
    std::wstring const & input ) [inline], [static]
```

C++ wide string to C++ string conversion routine.

## Parameters

<i>output</i>	The output C++ string.
<i>input</i>	The input wide string.

Definition at line 111 of file StringUtilities.hh.

Referenced by TrickHLA::SyncPntListBase::achieve\_and\_wait\_for\_synchronization(), TrickHLA::Federate::achieve\_and\_wait\_for\_synchronization(), TrickHLA::Federate::achieve\_synchronization\_point(), TrickHLA::Federate::add\_MOM\_HLAfederation\_instance\_id(), TrickHLA::ExecutionControlBase::clear\_multiphase\_init\_sync\_points(), TrickHLA::FedAmb::connectionLost(), TrickHLA::Federate::create\_federation(), TrickHLA::Federate::create\_RTI\_ambassador\_and\_connect(), TrickHLA::Federate::destroy(), TrickHLA::Federate::determine\_federate\_MOM\_object\_instance\_names(), TrickHLA::Manager::discover\_object\_instance(), TrickHLA::FedAmb::discoverObjectInstance(), TrickHLA::Federate::enable\_async\_delivery(), TrickHLA::Interaction::extract\_data(), TrickHLA::Object::extract\_data(), TrickHLA::FedAmb::federationSynchronized(), TrickHLA::Federate::inform\_RTI\_of\_restore\_completion(), TrickHLA::Federate::initiate\_restore\_announce(), TrickHLA::FedAmb::initiateFederateRestore(), TrickHLA::Federate::join\_federation(), TrickHLA::Object::mark\_all\_attributes\_as\_nonlocal(), TrickHLA::ExecutionControlBase::mark\_object\_as\_deleted\_from\_federation(), TrickHLA::Manager::mark\_object\_as\_deleted\_from\_federation(), TrickHLA::FedAmb::multipleObjectInstanceNameReservationFailed(), TrickHLA::FedAmb::multipleObjectInstanceNameReservationSucceeded(), TrickHLA::Object::negotiated\_attribute\_ownership\_divestiture(), TrickHLA::FedAmb::objectInstanceNameReservationFailed(), TrickHLA::FedAmb::objectInstanceNameReservationSucceeded(), TrickHLA::Federate::perform\_checkpoint(), TrickHLA::Federate::perform\_restore(), TrickHLA::Federate::perform\_time\_advance\_request(), TrickHLA::Federate::post\_checkpoint(), TrickHLA::Federate::print\_requested\_federation\_restore\_status(), TrickHLA::Federate::print\_version(), TrickHLA::Object::process\_deleted\_object(), TrickHLA::Interaction::process\_interaction(), TrickHLA::Interaction::publish\_interaction(), TrickHLA::Object::publish\_object\_attributes(), TrickHLA::Object::pull\_ownership(), TrickHLA::Object::pull\_ownership\_upon\_rejoin(), TrickHLA::Object::push\_ownership(), TrickHLA::Federate::rebuild\_federate\_handles(), SpaceFOM::MTRInteractionHandler::receive\_interaction(), TrickHLA::Manager::receive\_interaction(), TrickHLA::FedAmb::reflectAttributeValues(), TrickHLA::Federate::register\_generic\_sync\_point(), TrickHLA::Object::register\_object\_with\_RTI(), TrickHLA::Object::release\_ownership(), TrickHLA::Interaction::remove(), TrickHLA::Object::remove(), TrickHLA::Federate::remove\_MOM\_HLAfederate\_instance\_id(), TrickHLA::Object::remove\_object\_instance(), TrickHLA::FedAmb::removeObjectInstance(), TrickHLA::Object::request\_attribute\_value\_update(), TrickHLA::Federate::request\_federation\_restore\_status(), TrickHLA::Federate::request\_federation\_save(), TrickHLA::Federate::request\_federation\_save\_status(), TrickHLA::FedAmb::requestAttributeOwnershipAssumption(), TrickHLA::FedAmb::requestDivestitureConfirmation(), TrickHLA::Object::reserve\_object\_name\_with\_RTI(), TrickHLA::Federate::resign(), TrickHLA::Federate::resign\_so\_we\_can\_rejoin(), TrickHLA::Interaction::send(), TrickHLA::Object::send\_cyclic\_data(), TrickHLA::Object::send\_init\_data(), SpaceFOM::MTRInteractionHandler::send\_interaction(), TrickHLA::Object::send\_requested\_data(), TrickHLA::Federate::set\_all\_federate\_MOM\_instance\_handles\_by\_name(), TrickHLA::Object::set\_instance\_handle\_and\_name(), TrickHLA::Federate::set\_MOM\_HLAfederate\_instance\_attributes(), TrickHLA::Manager::set\_object\_instance\_handles\_by\_name(), TrickHLA::Federate::setup\_checkpoint(), TrickHLA::Manager::setup\_interaction\_RTI\_handles(), TrickHLA::Manager::setup\_object\_RTI\_handles(), TrickHLA::Interaction::setup\_preferred\_order\_with\_RTI(), TrickHLA::Object::setup\_preferred\_order\_with\_RTI(), TrickHLA::Federate::setup\_time\_constrained(), TrickHLA::Federate::setup\_time\_regulation(), TrickHLA::Federate::shutdown\_time\_constrained(), TrickHLA::Federate::shutdown\_time\_regulating(), TrickHLA::Interaction::subscribe\_to\_interaction(), TrickHLA::Object::subscribe\_to\_object\_attributes(), to\_string(), TrickHLA::Object::unpublish\_all\_object\_attributes(), TrickHLA::Interaction::unpublish\_interaction(), TrickHLA::Object::unsubscribe\_all\_object\_attributes(), TrickHLA::Interaction::unsubscribe\_from\_interaction(), TrickHLA::Federate::wait\_for\_required\_federates\_to\_join(), TrickHLA::ExecutionControlBase::wait\_for\_sync\_point\_announce(), TrickHLA::ExecutionConfigurationBase::wait\_on\_registration(), and TrickHLA::Manager::wait\_on\_registration\_of\_required\_objects().

### 7.64.3.12 to\_wstring() [1/2]

```
static void TrickHLA::StringUtilities::to_wstring (
    std::wstring & output,
```

```
const char * input ) [inline], [static]
```

C (char \*) string to C++ wide string conversion routine.

#### Parameters

<i>output</i>	The output wide string.
<i>input</i>	The input C string.

Definition at line 90 of file StringUtilities.hh.

Referenced by TrickHLA::Federate::add\_MOM\_HLAfederation\_instance\_id(), TrickHLA::ExecutionControlBase::add\_multiphase\_init\_sync\_points(), TrickHLA::Federate::create\_federation(), TrickHLA::Federate::create\_RTI(), TrickHLA::Federate::ambassador\_and\_connect(), TrickHLA::Federate::destroy(), TrickHLA::Federate::destroy\_orphaned\_federation(), TrickHLA::Federate::determine\_federate\_MOM\_object\_instance\_names(), TrickHLA::Manager::get\_trickhla\_object(), TrickHLA::ExecutionControlBase::get\_trickhla\_object(), TrickHLA::ExecutionControlBase::get\_unregistered\_object(), TrickHLA::Manager::get\_unregistered\_object(), TrickHLA::Federate::initiate\_restore\_announce(), TrickHLA::Federate::initiate\_save\_announce(), TrickHLA::Federate::is\_a\_required\_startup\_federate(), TrickHLA::Federate::is\_joined\_federate(), TrickHLA::Federate::is\_required\_federate(), TrickHLA::Federate::join\_federation(), TrickHLA::Manager::object\_instance\_name\_reservation\_failed(), TrickHLA::ExecutionControlBase::object\_instance\_name\_reservation\_succeeded(), TrickHLA::Manager::receive\_init\_data(), TrickHLA::Object::register\_object\_with\_RTI(), TrickHLA::Manager::request\_data\_update(), TrickHLA::Object::reserve\_object\_name\_with\_RTI(), TrickHLA::Manager::send\_init\_data(), TrickHLA::Federate::set\_all\_federate\_MOM\_instance\_handles\_by\_name(), TrickHLA::Federate::set\_checkpoint\_file\_name(), TrickHLA::Manager::set\_object\_instance\_handles\_by\_name(), TrickHLA::Federate::setup\_checkpoint(), TrickHLA::Manager::setup\_interaction\_RTI\_handles(), TrickHLA::Manager::setup\_object\_RTI\_handles(), and TrickHLA::Manager::wait\_for\_init\_sync\_point().

#### 7.64.3.13 to\_wstring() [2/2]

```
static void TrickHLA::StringUtilities::to_wstring (
    std::wstring & output,
    std::string const & input ) [inline], [static]
```

C++ string to C++ wide string conversion routine.

#### Parameters

<i>output</i>	The output wide string.
<i>input</i>	The input C++ string.

Definition at line 101 of file StringUtilities.hh.

#### 7.64.3.14 tokenize() [1/2]

```
static void TrickHLA::StringUtilities::tokenize (
    std::string const & str,
    std::vector< std::string > & tokens,
    std::string const & delimiters = "," ) [inline], [static]
```

Tokenize a given string for the specified delimiter characters.

#### Parameters

<i>str</i>	The input string.
<i>tokens</i>	Tokens of the string.
<i>delimiters</i>	Delimiter characters for tokenizing the string.

Definition at line 287 of file StringUtilities.hh.

References trim\_whitespace().

Referenced by TrickHLA::ExecutionControlBase::add\_multiphase\_init\_sync\_points(), TrickHLA::Federate::create\_federation(), and TrickHLA::Federate::join\_federation().

#### 7.64.3.15 tokenize() [2/2]

```
static void TrickHLA::StringUtilities::tokenize (
    std::string const & str,
    std::vector< std::wstring > & tokens,
    std::string const & delimiters = "," ) [inline], [static]
```

Tokenize a given wstring for the specified delimiter characters.

##### Parameters

<i>str</i>	The input string.
<i>tokens</i>	Tokens of the string.
<i>delimiters</i>	Delimiter characters for tokenizing the string.

Definition at line 323 of file StringUtilities.hh.

References trim\_whitespace().

#### 7.64.3.16 trim\_whitespace() [1/2]

```
static void TrickHLA::StringUtilities::trim_whitespace (
    std::string & s ) [inline], [static]
```

Trim any leading or trailing whitespace from the string.

##### Parameters

<i>s</i>	The string to trim.
----------	---------------------

Definition at line 220 of file StringUtilities.hh.

References WHITESPACE\_CHARS.

Referenced by tokenize().

#### 7.64.3.17 trim\_whitespace() [2/2]

```
static void TrickHLA::StringUtilities::trim_whitespace (
    std::wstring & s ) [inline], [static]
```

Trim any leading or trailing whitespace from the wstring.

##### Parameters

<i>s</i>	The wstring to trim.
----------	----------------------

Definition at line 251 of file StringUtilities.hh.

## 7.64.4 Friends And Related Function Documentation

### 7.64.4.1 `init_attrTrickHLA__StringUtilities`

```
void init_attrTrickHLA__StringUtilities ( ) [friend]
```

### 7.64.4.2 `InputProcessor`

friend class InputProcessor [friend]

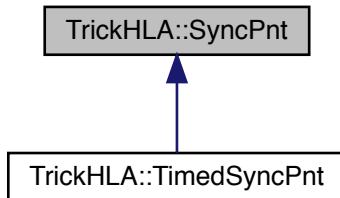
Definition at line 71 of file `StringUtilities.hh`.

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

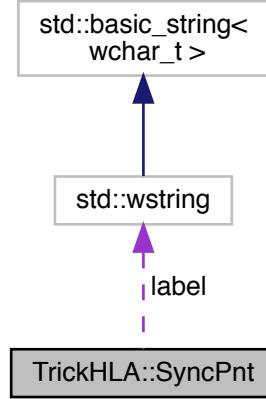
- [StringUtilities.hh](#)

## 7.65 TrickHLA::SyncPnt Class Reference

```
#include <SyncPnt.hh>
Inheritance diagram for TrickHLA::SyncPnt:
```



Collaboration diagram for TrickHLA::SyncPnt:



## Public Member Functions

- [SyncPnt \(\)](#)  
*Default constructor for the `TrickHLA SyncPnt` class.*
- [SyncPnt \(std::wstring const &l\)](#)  
*Initialization constructor.*
- [virtual ~SyncPnt \(\)](#)  
*Destructor for the `TrickHLA SyncPnt` class.*
- [virtual void register\\_sync\\_point \(RTI1516\\_NAMESPACE::RTIambassador &RTI\\_amb\)](#)  
*Register the synchronization point with the RTI.*
- [virtual void register\\_sync\\_point \(RTI1516\\_NAMESPACE::RTIambassador &RTI\\_amb, RTI1516\\_NAMESPACE::FederateHandleSet const &federate\\_handle\\_set\)](#)  
*Register the synchronization point with the RTI.*
- [bool wait\\_for\\_announce \(Federate \\*federate\)](#)  
*Wait for the announcement of the synchronization point.*
- [virtual void achieve\\_sync\\_point \(RTI1516\\_NAMESPACE::RTIambassador &RTI\\_amb\) throw \( RTI1516\\_NAMESPACE::SynchronizationPointLabelNotAnnounced, RTI1516\\_NAMESPACE::FederateNotExecutionMember, RTI1516\\_NAMESPACE::SaveInProgress, RTI1516\\_NAMESPACE::RestoreInProgress, RTI1516\\_NAMESPACE::NotConnected, RTI1516\\_NAMESPACE::RTIinternalError \)](#)  
*Notify the RTI that the synchronization point has been achieved.*
- [bool wait\\_for\\_synchronization \(Federate \\*federate\)](#)  
*Wait for this synchronization point to be synchronized.*
- [virtual bool is\\_valid \(\)](#)  
*Check if the synchronization point has been created and exists in at least one valid state.*
- [virtual bool exists \(\)](#)  
*Check if the synchronization point exists.*
- [virtual bool is\\_registered \(\)](#)  
*Check if the synchronization point is registered.*

- virtual bool `is_announced ()`  
*Check if the synchronization point is announced.*
- virtual bool `is_achieved ()`  
*Check if the synchronization point is achieved.*
- virtual bool `is_synchronized ()`  
*Check if the synchronization point is synchronized.*
- virtual bool `is_error ()`  
*Check if the synchronization point has a bad state.*
- virtual const std::wstring & `get_label () const`  
*Get the synchronization point label.*
- virtual const SyncPntStateEnum `get_state () const`  
*Get the synchronization point state.*
- virtual void `set_label (std::wstring const & l)`  
*Set the synchronization point label.*
- virtual void `set_state (SyncPntStateEnum s)`  
*Set the current state of the synchronization point.*
- virtual void `set_wait_sleep (unsigned int t)`  
*Set the sleep time for the wait loops.*
- virtual void `set_wait_timeout (unsigned int t)`  
*Set the timeout interval for the wait loops.*
- virtual std::wstring `to_string ()`  
*Create a C++ wide string with the synchronization point label and current state.*
- virtual void `convert (LoggableSyncPnt & log_sync_pnt)`  
*Convert the synchronization point into and loggable synchronization point.*

## Protected Attributes

- std::wstring `label`  
**Data I/O:** \*\*  
*Sync-point name.*
- SyncPntStateEnum `state`  
**Units:** –  
*Sync-point state.*
- unsigned int `wait_sleep`  
**Units:** us  
*Wait loop sleep times.*
- unsigned int `wait_timeout`  
**Units:** us  
*Wait loop timeout.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA_SyncPnt ()`

### 7.65.1 Detailed Description

Definition at line 55 of file SyncPnt.hh.

## 7.65.2 Constructor & Destructor Documentation

### 7.65.2.1 SyncPnt() [1/2]

```
SyncPnt::SyncPnt ( )
```

Default constructor for the [TrickHLA SyncPnt](#) class.  
**Trick Job Class:** *initialization*  
 Definition at line 49 of file SyncPnt.cpp.

### 7.65.2.2 SyncPnt() [2/2]

```
SyncPnt::SyncPnt (
    std::wstring const & l ) [explicit]
```

Initialization constructor.

#### Parameters

<i>l</i>	Synchronization point label.
----------	------------------------------

**Trick Job Class:** *initialization*  
 Definition at line 61 of file SyncPnt.cpp.

### 7.65.2.3 ~SyncPnt()

```
SyncPnt::~SyncPnt ( ) [virtual]
```

Destructor for the [TrickHLA SyncPnt](#) class.  
**Trick Job Class:** *shutdown*  
 Definition at line 73 of file SyncPnt.cpp.

## 7.65.3 Member Function Documentation

### 7.65.3.1 achieve\_sync\_point()

```
void SyncPnt::achieve_sync_point (
    RTI1516_NAMESPACE::RTIambassador & RTI_amb ) throw ( RTI1516_NAMESPACE::SynchronizationPointLabelNotAnnounced, RTI1516_NAMESPACE::FederateNotExecutionMember, RTI1516_NAMESPACE::SaveInProgress, RTI1516_NAMESPACE::RestoreInProgress, RTI1516_NAMESPACE::NotConnected, RTI1516_NAMESPACE::RTIinternalError ) [virtual]
```

Notify the RTI that the synchronization point has been achieved.

#### Parameters

<i>RTI_amb</i>	The HLA RTI Ambassador.
----------------	-------------------------

Definition at line 202 of file SyncPnt.cpp.

References [TrickHLA::SYNC\\_PNT\\_STATE\\_ACHIEVED](#), [TRICKHLA\\_RESTORE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_SAVE\\_FPU\\_CONTROL\\_WORD](#), and [TRICKHLA\\_VALIDATE\\_FPU\\_CONTROL\\_WORD](#).

Referenced by [TrickHLA::SyncPntListBase::achieve\\_and\\_wait\\_for\\_synchronization\(\)](#), [DSES::ExecutionControl::freeze\\_mode\\_transition\(\)](#), [DIS::ExecutionControl::freeze\\_mode\\_transition\(\)](#), [IMSim::ExecutionControl::freeze\\_mode\\_transition\(\)](#).

mode\_transition(), SpaceFOM::ExecutionControl::freeze\_mode\_transition(), DSES::ExecutionControl::run\_mode\_transition(), DIS::ExecutionControl::run\_mode\_transition(), IMSim::ExecutionControl::run\_mode\_transition(), and SpaceFOM::ExecutionControl::run\_mode\_transition().

### 7.65.3.2 convert()

```
void SyncPnt::convert (
    LoggableSyncPnt & log_sync_pnt ) [virtual]
```

Convert the synchronization point into and loggable synchronization point.

#### Parameters

<i>log_sync_pnt</i>	Reference to a loggable synchronization point.
---------------------	--

Reimplemented in [TrickHLA::TimedSyncPnt](#).

Definition at line 359 of file SyncPnt.cpp.

References TrickHLA::StringUtilities::ip\_strdup\_wstring(), TrickHLA::LoggableSyncPnt::label, label, TrickHLA::LoggableSyncPnt::state, and state.

### 7.65.3.3 exists()

```
bool SyncPnt::exists ( ) [virtual]
```

Check if the synchronization point exists.

#### Returns

True if the synchronization point exists.

Definition at line 289 of file SyncPnt.cpp.

References state, and TrickHLA::SYNC\_PNT\_STATE\_EXISTS.

Referenced by DSES::ExecutionControl::achieve\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::achieve\_all\_multiphase\_init\_sync\_pnts(), TrickHLA::TimedSyncPntList::achieve\_all\_sync\_pnts(), DSES::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), wait\_for\_announce(), and TrickHLA::ExecutionControlBase::wait\_for\_sync\_point\_announce().

### 7.65.3.4 get\_label()

```
virtual const std::wstring& TrickHLA::SyncPnt::get_label ( ) const [inline], [virtual]
```

Get the synchronization point label.

#### Returns

The synchronization point label.

Definition at line 144 of file SyncPnt.hh.

References label.

Referenced by TrickHLA::SyncPntListBase::achieve\_and\_wait\_for\_synchronization(), TrickHLA::SyncPntListBase::achieve\_sync\_pnt(), IMSim::PausePointList::clear\_sync\_pnt(), DIS::PausePointList::clear\_sync\_pnt(), TrickHLA::SyncPntListBase::clear\_sync\_pnt(), TrickHLA::SyncPntListBase::contains(), TrickHLA::SyncPntListBase::get\_sync\_pnt(), TrickHLA::SyncPntListBase::get\_sync\_pnt\_state(), TrickHLA::SyncPntListBase::mark\_announced(), TrickHLA::SyncPntListBase::mark\_registered(), TrickHLA::SyncPntListBase::mark\_synchronized(), TrickHLA::SyncPntListBase::register\_sync\_pnt(), and TrickHLA::ExecutionControlBase::wait\_for\_sync\_point\_announce().

### 7.65.3.5 `get_state()`

```
virtual const SyncPntStateEnum TrickHLA::SyncPnt::get_state ( ) const [inline], [virtual]
Get the synchronization point state.
```

#### Returns

The current state for this synchronization point.

Definition at line 148 of file SyncPnt.hh.

References state.

Referenced by TrickHLA::TimedSyncPntList::check\_sync\_pnts(), IMSim::PausePointList::clear\_sync\_pnt(), DIS::PausePointList::clear\_sync\_pnt(), and TrickHLA::SyncPntListBase::get\_sync\_pnt\_state().

### 7.65.3.6 `is_achieved()`

```
bool SyncPnt::is_achieved ( ) [virtual]
```

Check if the synchronization point is achieved.

#### Returns

True if the synchronization point is achieved.

Definition at line 304 of file SyncPnt.cpp.

References state, and TrickHLA::SYNC\_PNT\_STATE\_ACHIEVED.

Referenced by DSES::ExecutionControl::achieve\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::achieve\_all\_multiphase\_init\_sync\_pnts(), TrickHLA::TimedSyncPntList::achieve\_all\_sync\_pnts(), TrickHLA::SyncPntListBase::achieve\_all\_sync\_pnts(), TrickHLA::SyncPntListBase::achieve\_and\_wait\_for\_synchronization(), TrickHLA::SyncPntListBase::achieve\_sync\_pnt(), TrickHLA::SyncPntListBase::clear\_sync\_pnt(), DSES::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), and TrickHLA::SyncPntListBase::wait\_for\_list\_synchronization().

### 7.65.3.7 `is_announced()`

```
bool SyncPnt::is_announced ( ) [virtual]
```

Check if the synchronization point is announced.

#### Returns

True if the synchronization point is announced.

Definition at line 299 of file SyncPnt.cpp.

References state, and TrickHLA::SYNC\_PNT\_STATE\_ANNOUNCED.

Referenced by TrickHLA::SyncPntListBase::achieve\_and\_wait\_for\_synchronization(), SpaceFOM::ExecutionControl::role\_determination\_process(), wait\_for\_announce(), and TrickHLA::ExecutionControlBase::wait\_for\_sync\_point\_announce().

### 7.65.3.8 `is_error()`

```
bool SyncPnt::is_error ( ) [virtual]
```

Check if the synchronization point has a bad state.

**Returns**

True if the synchronization point has a bad state.

Definition at line 314 of file SyncPnt.cpp.

References state, TrickHLA::SYNC\_PNT\_STATE\_ACHIEVED, TrickHLA::SYNC\_PNT\_STATE\_ANNOUNCED, TrickHLA::SYNC\_PNT\_STATE\_EXISTS, TrickHLA::SYNC\_PNT\_STATE\_REGISTERED, and TrickHLA::SYNC\_PNT\_STATE\_SYNCHRONIZED.

### 7.65.3.9 **is\_registered()**

```
bool SyncPnt::is_registered () [virtual]
```

Check if the synchronization point is registered.

**Returns**

True if the synchronization point is registered.

Definition at line 294 of file SyncPnt.cpp.

References state, and TrickHLA::SYNC\_PNT\_STATE\_REGISTERED.

Referenced by TrickHLA::SyncPntListBase::register\_all\_sync\_pnts(), TrickHLA::SyncPntListBase::register\_sync\_pnt(), and wait\_for\_announce().

### 7.65.3.10 **is\_synchronized()**

```
bool SyncPnt::is_synchronized () [virtual]
```

Check if the synchronization point is synchronized.

**Returns**

True if the synchronization point is synchronized.

Definition at line 309 of file SyncPnt.cpp.

References state, and TrickHLA::SYNC\_PNT\_STATE\_SYNCHRONIZED.

Referenced by TrickHLA::SyncPntListBase::achieve\_and\_wait\_for\_synchronization(), and wait\_for\_synchronization().

### 7.65.3.11 **is\_valid()**

```
bool SyncPnt::is_valid () [virtual]
```

Check if the synchronization point has been created and exists in at least one valid state.

**Returns**

True if the synchronization point has a valid state.

Definition at line 280 of file SyncPnt.cpp.

References state, TrickHLA::SYNC\_PNT\_STATE\_ACHIEVED, TrickHLA::SYNC\_PNT\_STATE\_ANNOUNCED, TrickHLA::SYNC\_PNT\_STATE\_EXISTS, TrickHLA::SYNC\_PNT\_STATE\_REGISTERED, and TrickHLA::SYNC\_PNT\_STATE\_SYNCHRONIZED.

Referenced by TrickHLA::SyncPntListBase::achieve\_all\_sync\_pnts(), and TrickHLA::SyncPntListBase::wait\_for\_list\_synchronization().

### 7.65.3.12 **register\_sync\_point() [1/2]**

```
void SyncPnt::register_sync_point (
    RTI1516_NAMESPACE::RTIambassador & RTI_amb ) [virtual]
```

Register the synchronization point with the RTI.

## Parameters

<i>RTI_amb</i>	HLA RTI Ambassador.
----------------	---------------------

Definition at line 78 of file SyncPnt.cpp.

References label, RTI1516\_EXCEPTION, RTI1516\_USERDATA, set\_state(), TrickHLA::SYNC\_PNT\_STATE\_REGISTERED, THLA\_NEWLINE, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by TrickHLA::SyncPntListBase::register\_all\_sync\_pnts(), and TrickHLA::SyncPntListBase::register\_sync\_pnt().

### 7.65.3.13 register\_sync\_point() [2/2]

```
void SyncPnt::register_sync_point (
    RTI1516_NAMESPACE::RTIambassador & RTI_amb,
    RTI1516_NAMESPACE::FederateHandleSet const & federate_handle_set ) [virtual]
```

Register the synchronization point with the RTI.

## Parameters

<i>RTI_amb</i>	The HLA RTI Ambassador.
<i>federate_handle_set</i>	HLA Federation handle set.

Definition at line 113 of file SyncPnt.cpp.

References label, RTI1516\_EXCEPTION, RTI1516\_USERDATA, set\_state(), TrickHLA::SYNC\_PNT\_STATE\_REGISTERED, THLA\_NEWLINE, TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

### 7.65.3.14 set\_label()

```
virtual void TrickHLA::SyncPnt::set_label (
    std::wstring const & l ) [inline], [virtual]
```

Set the synchronization point label.

## Parameters

<i>l</i>	The synchronization point label.
----------	----------------------------------

Definition at line 152 of file SyncPnt.hh.

References label.

### 7.65.3.15 set\_state()

```
virtual void TrickHLA::SyncPnt::set_state (
    SyncPntStateEnum s ) [inline], [virtual]
```

Set the current state of the synchronization point.

## Parameters

<i>s</i>	Current synchronization point state.
----------	--------------------------------------

Definition at line 156 of file SyncPnt.hh.

References state.

Referenced by TrickHLA::SyncPntListBase::achieve\_sync\_pnt(), TrickHLA::SyncPntListBase::mark\_announced(), TrickHLA::SyncPntListBase::mark\_registered(), TrickHLA::SyncPntListBase::mark\_synchronized(), register\_↔ sync\_point(), wait\_for\_announce(), TrickHLA::SyncPntListBase::wait\_for\_list\_synchronization(), and wait\_for\_↔ synchronization().

#### 7.65.3.16 set\_wait\_sleep()

```
virtual void TrickHLA::SyncPnt::set_wait_sleep (
    unsigned int t ) [inline], [virtual]
```

Set the sleep time for the wait loops.

Parameters

<i>t</i>	Sleep time in microseconds.
----------	-----------------------------

Definition at line 160 of file SyncPnt.hh.

References wait\_sleep.

#### 7.65.3.17 set\_wait\_timeout()

```
virtual void TrickHLA::SyncPnt::set_wait_timeout (
    unsigned int t ) [inline], [virtual]
```

Set the timeout interval for the wait loops.

Parameters

<i>t</i>	Timeout interval in microseconds.
----------	-----------------------------------

Definition at line 164 of file SyncPnt.hh.

References wait\_timeout.

#### 7.65.3.18 to\_string()

```
std::wstring SyncPnt::to_string ( ) [virtual]
```

Create a C++ wide string with the synchronization point label and current state.

Returns

A string with the synchronization point label and current state.

Reimplemented in [TrickHLA::TimedSyncPnt](#).

Definition at line 323 of file SyncPnt.cpp.

References label, state, TrickHLA::SYNC\_PNT\_STATE\_ACHIEVED, TrickHLA::SYNC\_PNT\_STATE\_ANNOUNCED, TrickHLA::SYNC\_PNT\_STATE\_ERROR, TrickHLA::SYNC\_PNT\_STATE\_EXISTS, TrickHLA::SYNC\_PNT\_STATE\_↔ REGISTERED, and TrickHLA::SYNC\_PNT\_STATE\_SYNCHRONIZED.

Referenced by DIS::PausePointList::to\_string(), IMSim::PausePointList::to\_string(), TrickHLA::SyncPntListBase::to\_↔ string(), wait\_for\_announce(), and TrickHLA::SyncPntListBase::wait\_for\_announcement().

### 7.65.3.19 `wait_for_announce()`

```
bool SyncPnt::wait_for_announce (
    Federate * federate )
```

Wait for the announcement of the synchronization point.

#### Parameters

<code>federate</code>	The <a href="#">TrickHLA::Federate</a> instance.
-----------------------	--

Definition at line 150 of file SyncPnt.cpp.

References `TrickHLA::Federate::check_for_shutdown_with_termination()`, `exists()`, `is_announced()`, `TrickHLA::Federate::is_execution_member()`, `is_registered()`, `set_state()`, `TrickHLA::SYNC_PNT_STATE_ANNOUNCED`, `THLA_ENDL`, `to_string()`, and `wait_timeout`.

Referenced by `DSES::ExecutionControl::freeze_mode_transition()`, `DIS::ExecutionControl::freeze_mode_transition()`, `IMSim::ExecutionControl::freeze_mode_transition()`, `SpaceFOM::ExecutionControl::freeze_mode_transition()`, `DSES::ExecutionControl::run_mode_transition()`, `DIS::ExecutionControl::run_mode_transition()`, `IMSim::ExecutionControl::run_mode_transition()`, `SpaceFOM::ExecutionControl::run_mode_transition()`, `TrickHLA::SyncPntListBase::wait_for_all_announcements()`, and `TrickHLA::SyncPntListBase::wait_for_announcement()`.

### 7.65.3.20 `wait_for_synchronization()`

```
bool SyncPnt::wait_for_synchronization (
    Federate * federate )
```

Wait for this synchronization point to be synchronized.

#### Parameters

<code>federate</code>	The <a href="#">TrickHLA::Federate</a> instance.
-----------------------	--

Definition at line 240 of file SyncPnt.cpp.

References `TrickHLA::Federate::check_for_shutdown_with_termination()`, `TrickHLA::Federate::is_execution_member()`, `is_synchronized()`, `set_state()`, `TrickHLA::SYNC_PNT_STATE_EXISTS`, `THLA_ENDL`, `wait_sleep`, and `wait_timeout`.

Referenced by `TrickHLA::SyncPntListBase::achieve_and_wait_for_synchronization()`, `DSES::ExecutionControl::freeze_mode_transition()`, `DIS::ExecutionControl::freeze_mode_transition()`, `IMSim::ExecutionControl::freeze_mode_transition()`, `SpaceFOM::ExecutionControl::freeze_mode_transition()`, `DSES::ExecutionControl::run_mode_transition()`, `DIS::ExecutionControl::run_mode_transition()`, `IMSim::ExecutionControl::run_mode_transition()`, and `SpaceFOM::ExecutionControl::run_mode_transition()`.

## 7.65.4 Friends And Related Function Documentation

### 7.65.4.1 `init_attrTrickHLA__SyncPnt`

```
void init_attrTrickHLA__SyncPnt ( ) [friend]
```

### 7.65.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 62 of file SyncPnt.hh.

## 7.65.5 Field Documentation

### 7.65.5.1 label

std::wstring TrickHLA::SyncPnt::label [protected]

**Data I/O:** \*\*

Sync-point name.

Definition at line 178 of file SyncPnt.hh.

Referenced by DSES::ExecutionControl::achieve\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::achieve\_all\_multiphase\_init\_sync\_pnts(), TrickHLA::TimedSyncPnt::convert(), convert(), get\_label(), register\_sync\_point(), set\_label(), TrickHLA::TimedSyncPnt::to\_string(), to\_string(), DSES::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), and IMSim::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts().

### 7.65.5.2 state

SyncPntStateEnum TrickHLA::SyncPnt::state [protected]

**Units:** –

Sync-point state.

Definition at line 179 of file SyncPnt.hh.

Referenced by TrickHLA::TimedSyncPnt::convert(), convert(), exists(), get\_state(), is\_achieved(), is\_announced(), is\_error(), is\_registered(), is\_synchronized(), is\_valid(), set\_state(), TrickHLA::TimedSyncPnt::to\_string(), and to\_string().

### 7.65.5.3 wait\_sleep

unsigned int TrickHLA::SyncPnt::wait\_sleep [protected]

**Units:** us

Wait loop sleep times.

Definition at line 180 of file SyncPnt.hh.

Referenced by set\_wait\_sleep(), and wait\_for\_synchronization().

### 7.65.5.4 wait\_timeout

unsigned int TrickHLA::SyncPnt::wait\_timeout [protected]

**Units:** us

Wait loop timeout.

Definition at line 181 of file SyncPnt.hh.

Referenced by set\_wait\_timeout(), wait\_for\_announce(), and wait\_for\_synchronization().

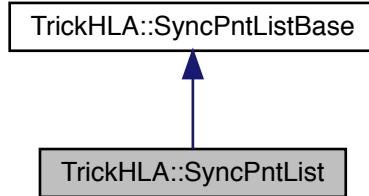
The documentation for this class was generated from the following files:

- [SyncPnt.hh](#)
- [SyncPnt.cpp](#)

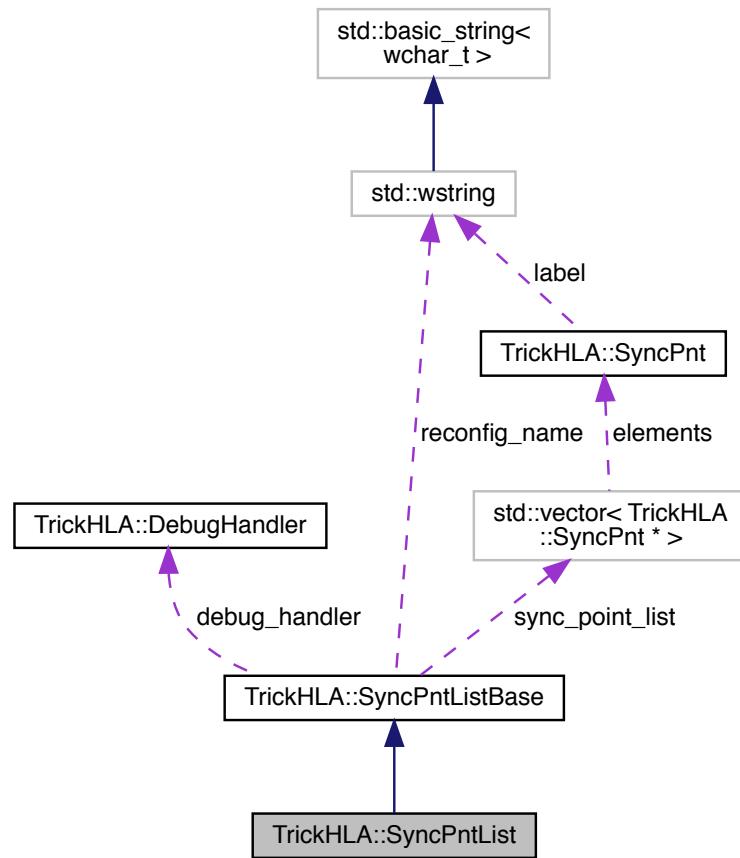
## 7.66 TrickHLA::SyncPntList Class Reference

#include <SyncPntList.hh>

Inheritance diagram for TrickHLA::SyncPntList:



Collaboration diagram for TrickHLA::SyncPntList:



## Public Member Functions

- [SyncPntList \(\)](#)  
*Default constructor for the `TrickHLA SyncPntList` class.*
- [virtual ~SyncPntList \(\)](#)  
*Pure virtual destructor for the `TrickHLA SyncPntList` class.*

## Private Member Functions

- [SyncPntList \(const SyncPntList &rhs\)](#)  
*Copy constructor for `SyncPntList` class.*
- [SyncPntList & operator= \(const SyncPntList &rhs\)](#)  
*Assignment operator for `SyncPntList` class.*

## Friends

- [class InputProcessor](#)
- [void init\\_attrTrickHLA\\_\\_SyncPntList \(\)](#)

## Additional Inherited Members

### 7.66.1 Detailed Description

Definition at line 58 of file SyncPntList.hh.

### 7.66.2 Constructor & Destructor Documentation

#### 7.66.2.1 SyncPntList() [1/2]

`TrickHLA::SyncPntList::SyncPntList ( ) [inline]`  
Default constructor for the `TrickHLA SyncPntList` class.  
Definition at line 75 of file SyncPntList.hh.

#### 7.66.2.2 ~SyncPntList()

`virtual TrickHLA::SyncPntList::~SyncPntList ( ) [inline], [virtual]`  
Pure virtual destructor for the `TrickHLA SyncPntList` class.  
Definition at line 78 of file SyncPntList.hh.

#### 7.66.2.3 SyncPntList() [2/2]

`TrickHLA::SyncPntList::SyncPntList (`  
    `const SyncPntList & rhs ) [private]`  
Copy constructor for `SyncPntList` class.  
This constructor is private to prevent inadvertent copies.

### 7.66.3 Member Function Documentation

### 7.66.3.1 operator=( )

```
SyncPntList& TrickHLA::SyncPntList::operator= (
    const SyncPntList & rhs ) [private]
```

Assignment operator for `SyncPntList` class.

This assignment operator is private to prevent inadvertent copies.

## 7.66.4 Friends And Related Function Documentation

### 7.66.4.1 init\_attrTrickHLA\_\_SyncPntList

```
void init_attrTrickHLA__SyncPntList ( ) [friend]
```

### 7.66.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 65 of file `SyncPntList.hh`.

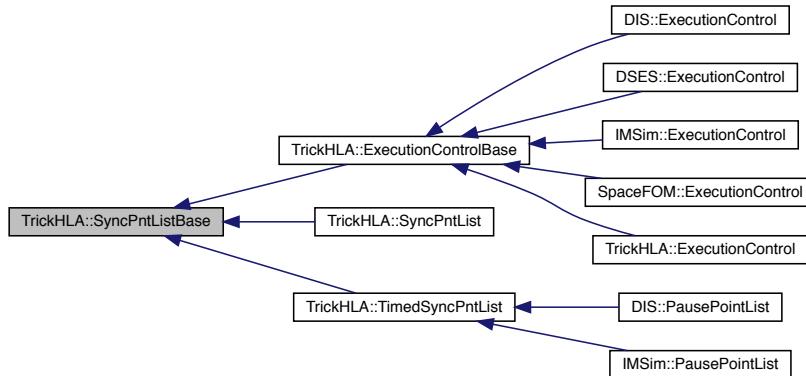
The documentation for this class was generated from the following file:

- [SyncPntList.hh](#)

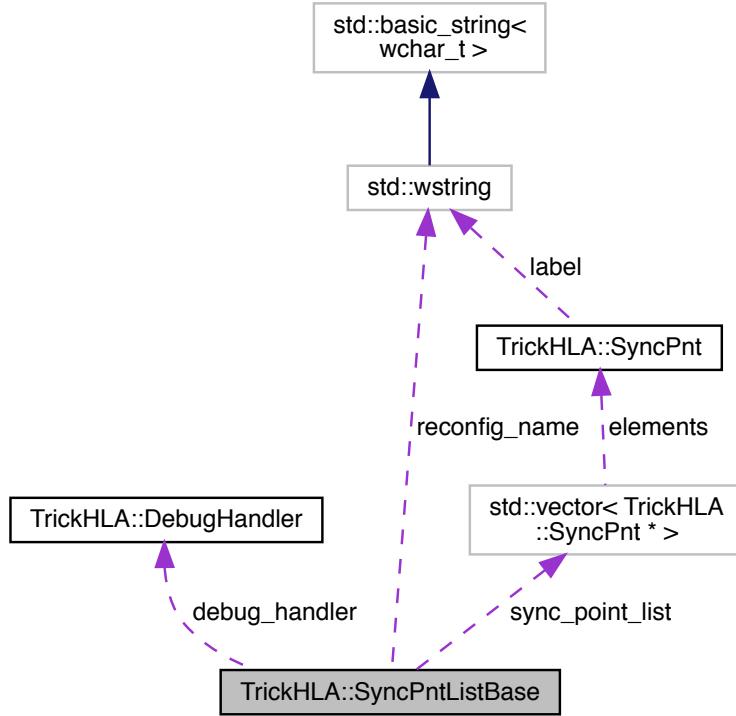
## 7.67 TrickHLA::SyncPntListBase Class Reference

```
#include <SyncPntListBase.hh>
```

Inheritance diagram for `TrickHLA::SyncPntListBase`:



Collaboration diagram for TrickHLA::SyncPntListBase:



## Public Member Functions

- **SyncPntListBase ()**  
*Default constructor for the [TrickHLA SyncPntListBase](#) class.*
- **virtual ~SyncPntListBase ()=0**  
*Pure virtual destructor for the [TrickHLA SyncPntListBase](#) class.*
- **virtual void add\_sync\_pnt (std::wstring const &label)**  
*Add the given synchronization point label to the list.*
- **virtual SyncPnt \* get\_sync\_pnt (std::wstring const &label)**  
*Get the pointer to the synchronization point associated with the given label.*
- **virtual SyncPnt \* register\_sync\_pnt (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador, std::wstring const &label)**  
*Register a synchronization point associated with the given label.*
- **virtual SyncPnt \* register\_sync\_pnt (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador, RTI1516\_NAMESPACE::FederateHandleSet const &federate\_handle\_set, std::wstring const &label)**  
*Register a synchronization point associated with the given label.*
- **virtual void register\_all\_sync\_pnts (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador)**  
*Register all the synchronization point in this list.*
- **virtual void register\_all\_sync\_pnts (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador, RTI1516\_NAMESPACE::FederateHandleSet const &federate\_handle\_set)**  
*Register all the synchronization point in this list.*

Register all the sync-points with the RTI for the given set of federate handles.

- virtual void `sync_point_registration_succeeded` (std::wstring const &label)

Callback from `TrickHLA::FedAmb` through `TrickHLA::Federate` for when registration of a synchronization point success. and is one of the sync-points created.

- virtual void `sync_point_registration_failed` (std::wstring const &label, bool not\_unique)

Callback from `TrickHLA::FedAmb` through `TrickHLA::Federate` for when registration of a synchronization point fails. and is one of the sync-points created.

- virtual void `wait_for_announcement` (Federate \*fed\_ptr, std::wstring const &label)

Wait for the sync-point to be announced by the RTI.

- virtual void `wait_for_all_announcements` (Federate \*fed\_ptr)

Wait for all the sync-points to be registered with the RTI.

- virtual void `announce_sync_point` (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador, std::wstring const &label, RTI1516\_USERDATA const &user\_supplied\_tag)

The RTI has announced the existence of a synchronization point.

- virtual bool `achieve_all_sync_pnts` (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador) throw ( RTI1516\_NAMESPACE::SynchronizationPointLabelNotAnnounced, RTI1516\_NAMESPACE::FederateNotExecutionMember, RTI1516\_NAMESPACE::SaveInProgress, RTI1516\_NAMESPACE::RestoreInProgress, RTI1516\_NAMESPACE::NotConnected, RTI1516\_NAMESPACE::RTIinternalError )

Achieve all the synchronization points in the list.

- void `wait_for_list_synchronization` (Federate \*federate)

Wait for all the synchronization points in the list to be achieved.

- virtual bool `achieve_sync_pnt` (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador, std::wstring const &label)

Achieve the synchronization point associated with this label.

- virtual bool `achieve_sync_pnt` (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador, SyncPnt \*sync\_pnt)

Achieve the given synchronization point.

- virtual void `achieve_and_wait_for_synchronization` (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador, Federate \*federate, std::wstring const &label) throw ( RTI1516\_NAMESPACE::SynchronizationPointLabelNotAnnounced, RTI1516\_NAMESPACE::FederateNotExecutionMember, RTI1516\_NAMESPACE::SaveInProgress, RTI1516\_NAMESPACE::RestoreInProgress, RTI1516\_NAMESPACE::NotConnected, RTI1516\_NAMESPACE::RTIinternalError )

Achieve the specified sync-point and wait for the federation to be synchronized on it.

- virtual bool `contains` (std::wstring const &label)

Determine if the synchronization point is known to be in the list of known synchronization points.

- virtual bool `mark_registered` (std::wstring const &label)

Mark the given synchronization point as registered in the federation.

- virtual bool `mark_announced` (std::wstring const &label)

Mark the given synchronization point as existing in the federation.

- virtual bool `mark_synchronized` (std::wstring const &label)

Mark the given synchronization point as synchronized in the federation.

- virtual SyncPntStateEnum `get_sync_pnt_state` (std::wstring const &label)

Get the state of the given synchronization point label.

- virtual bool `is_sync_pnt_announced` (std::wstring const &label)

Check if specified synchronization point has been announced.

- virtual bool `clear_sync_pnt` (std::wstring const &label)

Clear the given synchronization point label.

- virtual void `reset` ()

Reset the list of synchronization points.

- virtual size\_t `get_size` () const

*Get the number of synchronization points in the list.*

- virtual const std::wstring [get\\_reconfig\\_name \(\)](#)
- virtual std::wstring [to\\_string \(\)](#)

*Returns a wide string representing the state of the synchronization points.*

- virtual void [convert\\_sync\\_pts \(LoggableSyncPnt \\*sync\\_points\)](#)

*Converts the vector of synchronization points to a checkpoint-able class.*

- virtual void [print\\_sync\\_pnts \(\)](#)

*Dumps synchronization point information to the screen.*

- virtual void [set\\_debug\\_level \(DebugHandler hndlr\)](#)

*Set the debug handler.*

## Protected Member Functions

- void [lock\\_read\\_only \(\)](#)

*Put the lock in read only state.*

- void [lock\\_read\\_write \(\)](#)

*Put the lock in read/write state.*

- void [unlock\\_read\\_only \(\)](#)

*Unlock the read only lock.*

- void [unlock\\_read\\_write \(\)](#)

*Unlock the read/write lock.*

## Protected Attributes

- int [read\\_locks](#)

**Units:** –

*Read lock count.*

- int [write\\_locks](#)

**Units:** –

*Write lock count.*

- std::vector< [SyncPnt \\* > sync\\_point\\_list](#)

**Data I/O:** \*\*

*Vector of synchronization points.*

- std::wstring [reconfig\\_name](#)

**Data I/O:** \*\*

*Wide string of the reconfiguration name.*

- [DebugHandler debug\\_handler](#)

**Units:** –

*Decides whether to print any debug messages.*

## Private Member Functions

- [SyncPntListBase \(const SyncPntListBase &rhs\)](#)

*Copy constructor for [SyncPntListBase](#) class.*

- [SyncPntListBase & operator= \(const SyncPntListBase &rhs\)](#)

*Assignment operator for [SyncPntListBase](#) class.*

## Friends

- class [InputProcessor](#)

- void [init\\_attrTrickHLA\\_SyncPntListBase \(\)](#)

### 7.67.1 Detailed Description

Definition at line 55 of file SyncPntListBase.hh.

### 7.67.2 Constructor & Destructor Documentation

#### 7.67.2.1 SyncPntListBase() [1/2]

```
SyncPntListBase::SyncPntListBase ( )
Default constructor for the TrickHLA SyncPntListBase class.
Trick Job Class: initialization
Definition at line 55 of file SyncPntListBase.cpp.
```

#### 7.67.2.2 ~SyncPntListBase()

```
SyncPntListBase::~SyncPntListBase ( ) [pure virtual]
Pure virtual destructor for the TrickHLA SyncPntListBase class.
This is a pure virtual destructor. Trick Job Class: shutdown
Definition at line 65 of file SyncPntListBase.cpp.
References reset\(\).
```

#### 7.67.2.3 SyncPntListBase() [2/2]

```
TrickHLA::SyncPntListBase::SyncPntListBase (
    const SyncPntListBase & rhs ) [private]
Copy constructor for SyncPntListBase class.
This constructor is private to prevent inadvertent copies.
```

### 7.67.3 Member Function Documentation

#### 7.67.3.1 achieve\_all\_sync\_pnts()

```
bool SyncPntListBase::achieve_all_sync_pnts (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador ) throw ( RTI1516_NAMESPACE::←
SynchronizationPointLabelNotAnnounced, RTI1516_NAMESPACE::FederateNotExecutionMember, RTI1516←
NAMESPACE::SaveInProgress, RTI1516_NAMESPACE::RestoreInProgress, RTI1516_NAMESPACE::NotConnected,
RTI1516_NAMESPACE::RTIinternalError) [virtual]
Achieve all the synchronization points in the list.
```

##### Returns

True is any synchronization point in the list was achieved.

##### Parameters

<code>rti_ambassador</code>	Reference to the HLA RTI Ambassador instance.
-----------------------------	---

##### **Trick Job Class:** *initialization*

Reimplemented in [TrickHLA::TimedSyncPntList](#).

Definition at line 532 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::is\_achieved(), and TrickHLA::SyncPnt::is\_valid().

### 7.67.3.2 achieve\_and\_wait\_for\_synchronization()

```
void SyncPntListBase::achieve_and_wait_for_synchronization (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    Federate * federate,
    std::wstring const & label ) throw ( RTI1516_NAMESPACE::SynchronizationPointLabel<-
NotAnnounced, RTI1516_NAMESPACE::FederateNotExecutionMember, RTI1516_NAMESPACE::SaveInProgress,
RTI1516_NAMESPACE::RestoreInProgress, RTI1516_NAMESPACE::NotConnected, RTI1516_NAMESPACE::RT<-
InternalError) [virtual]
```

Achieve the specified sync-point and wait for the federation to be synchronized on it.

#### Parameters

<i>rti_ambassador</i>	Reference to RTI Ambassador.
<i>federate</i>	Associated federate.
<i>label</i>	Synchronization point label.

Definition at line 441 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::achieve\_sync\_point(), TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, TrickHLA::SyncPnt::get\_label(), TrickHLA::SyncPnt::is\_achieved(), TrickHLA::SyncPnt::is\_announced(), TrickHLA::SyncPnt::is\_synchronized(), TrickHLA::StringUtilities::to\_string(), and TrickHLA::SyncPnt::wait\_for\_synchronization().

Referenced by TrickHLA::Federate::achieve\_and\_wait\_for\_synchronization(), SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), TrickHLA::Manager::wait\_for\_init\_sync\_point(), and SpaceFOM::ExecutionControl::wait\_on\_root\_frame\_discovered\_synchronization().

### 7.67.3.3 achieve\_sync\_pnt() [1/2]

```
bool SyncPntListBase::achieve_sync_pnt (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    std::wstring const & label ) [virtual]
```

Achieve the synchronization point associated with this label.

#### Returns

True if achieved, false otherwise.

#### Parameters

<i>rti_ambassador</i>	Reference to RTI Ambassador.
<i>label</i>	Synchronization point label.

Definition at line 336 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::get\_label(), TrickHLA::SyncPnt::is\_achieved(), lock\_read\_only(), sync\_point\_list, and unlock\_read\_only().

Referenced by DIS::ExecutionControl::announce\_sync\_point(), DSES::ExecutionControl::announce\_sync\_point(), IMSim::ExecutionControl::announce\_sync\_point(), SpaceFOM::ExecutionControl::announce\_sync\_point(), and announce\_sync\_point().

#### 7.67.3.4 achieve\_sync\_pnt() [2/2]

```
bool SyncPntListBase::achieve_sync_pnt (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    SyncPnt * sync_pnt ) [virtual]
```

Achieve the given synchronization point.

##### Returns

True if achieved, false otherwise.

##### Parameters

<i>rti_ambassador</i>	Reference to RTI Ambassador.
<i>sync_pnt</i>	Specified synchronization point.

Definition at line 361 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::get\_label(), TrickHLA::SyncPnt::set\_state(), TrickHLA::SYNC\_PNT\_STATE\_ACHIEVED, and unlock\_read\_only().

#### 7.67.3.5 add\_sync\_pnt()

```
void SyncPntListBase::add_sync_pnt (
    std::wstring const & label ) [virtual]
```

Add the given synchronization point label to the list.

##### Parameters

<i>label</i>	Synchronization point label.
--------------	------------------------------

Reimplemented in [TrickHLA::TimedSyncPntList](#).

Definition at line 70 of file SyncPntListBase.cpp.

References lock\_read\_write(), sync\_point\_list, and unlock\_read\_write().

Referenced by SpaceFOM::ExecutionControl::add\_initialization\_sync\_points(), DIS::ExecutionControl::add\_multiphase\_init\_sync\_points(), DSES::ExecutionControl::add\_multiphase\_init\_sync\_points(), IMSim::ExecutionControl::add\_multiphase\_init\_sync\_points(), TrickHLA::ExecutionControlBase::add\_multiphase\_init\_sync\_points(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

#### 7.67.3.6 announce\_sync\_point()

```
void SyncPntListBase::announce_sync_point (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    std::wstring const & label,
    RTI1516_USERDATA const & user_supplied_tag ) [virtual]
```

The RTI has announced the existence of a synchronization point.

##### Parameters

<i>rti_ambassador</i>	Reference to the HLA RTI Ambassador instance.
<i>label</i>	Sync-point label.
<i>user_supplied_tag</i>	Use supplied tag.

Reimplemented in [SpaceFOM::ExecutionControl](#), [IMSim::ExecutionControl](#), [DSES::ExecutionControl](#), [DIS::ExecutionControl](#), and [TrickHLA::ExecutionControl](#).

Definition at line 304 of file SyncPntListBase.cpp.

References `achieve_sync_pnt()`, `contains()`, `debug_handler`, `TrickHLA::DEBUG_LEVEL_2_TRACE`, `TrickHLA::DEBU←G_SOURCE_FEDERATE`, `mark_announced()`, `TrickHLA::DebugHandler::should_print()`, and `THLA_NEWLINE`.

Referenced by `TrickHLA::ExecutionControl::announce_sync_point()`, and `TrickHLA::Federate::announce_sync_point()`.

#### 7.67.3.7 `clear_sync_pnt()`

```
bool SyncPntListBase::clear_sync_pnt (
    std::wstring const & label ) [virtual]
```

Clear the given synchronization point label.

Returns

True if synchronization point is cleared.

Parameters

<code>label</code>	The synchronization point label.
--------------------	----------------------------------

Reimplemented in [DIS::PausePointList](#), and [IMSim::PausePointList](#).

Definition at line 595 of file SyncPntListBase.cpp.

References `TrickHLA::SyncPnt::get_label()`, `TrickHLA::SyncPnt::is_achieved()`, `lock_read_write()`, `sync_point_list`, and `unlock_read_write()`.

#### 7.67.3.8 `contains()`

```
bool SyncPntListBase::contains (
    std::wstring const & label ) [virtual]
```

Determine if the synchronization point is known to be in the list of known synchronization points.

Returns

True if the label is a known synchronization point.

Parameters

<code>label</code>	The synchronization point label.
--------------------	----------------------------------

**Trick Job Class:** *initialization*

Definition at line 636 of file SyncPntListBase.cpp.

References `TrickHLA::SyncPnt::get_label()`, `lock_read_only()`, `sync_point_list`, and `unlock_read_only()`.

Referenced by `DIS::ExecutionControl::announce_sync_point()`, `DSES::ExecutionControl::announce_sync_point()`, `IMSim::ExecutionControl::announce_sync_point()`, `SpaceFOM::ExecutionControl::announce_sync_point()`, `announce_sync_point()`, `IMSim::ExecutionControl::determine_if_late_joining_or_restoring_federate()`, `sync_point_registration_failed()`, and `TrickHLA::Manager::wait_for_init_sync_point()`.

#### 7.67.3.9 `convert_sync_pts()`

```
void SyncPntListBase::convert_sync_pts (
```

`LoggableSyncPnt * sync_points ) [virtual]`

Converts the vector of synchronization points to a checkpoint-able class.

#### Parameters

<code>sync_points</code>	Area to populate.
--------------------------	-------------------

Reimplemented in [TrickHLA::TimedSyncPntList](#).

Definition at line 757 of file SyncPntListBase.cpp.

References `lock_read_only()`, `sync_point_list`, and `unlock_read_only()`.

#### 7.67.3.10 `get_reconfig_name()`

`virtual const std::wstring TrickHLA::SyncPntListBase::get_reconfig_name ( ) [inline], [virtual]`

#### Returns

Definition at line 236 of file SyncPntListBase.hh.

References `reconfig_name`.

#### 7.67.3.11 `get_size()`

`virtual size_t TrickHLA::SyncPntListBase::get_size ( ) const [inline], [virtual]`

Get the number of synchronization points in the list.

#### Returns

The number of synchronization points in the list.

Definition at line 232 of file SyncPntListBase.hh.

References `sync_point_list`.

#### 7.67.3.12 `get_sync_pnt()`

`SyncPnt * SyncPntListBase::get_sync_pnt ( std::wstring const & label ) [virtual]`

Get the pointer to the synchronization point associated with the given label.

#### Returns

The synchronization point associated with the label.

Definition at line 79 of file SyncPntListBase.cpp.

References `TrickHLA::SyncPnt::get_label()`, `lock_read_only()`, `sync_point_list`, and `unlock_read_only()`.

Referenced by `DSES::ExecutionControl::freeze_mode_transition()`, `DIS::ExecutionControl::freeze_mode_transition()`, `IMSim::ExecutionControl::freeze_mode_transition()`, `SpaceFOM::ExecutionControl::freeze_mode_transition()`, `SpaceFOM::ExecutionControl::role_determination_process()`, `DSES::ExecutionControl::run_mode_transition()`, `DIS::ExecutionControl::run_mode_transition()`, `IMSim::ExecutionControl::run_mode_transition()`, `SpaceFOM::ExecutionControl::run_mode_transition()`, and `wait_for_announcement()`.

### 7.67.3.13 `get_sync_pnt_state()`

```
SyncPntStateEnum SyncPntListBase::get_sync_pnt_state (
    std::wstring const & label ) [virtual]
```

Get the state of the given synchronization point label.

#### Returns

Synchronization point state.

#### Parameters

<i>label</i>	The synchronization point label.
--------------	----------------------------------

Definition at line 557 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::get\_label(), TrickHLA::SyncPnt::get\_state(), lock\_read\_write(), TrickHLA::SYNC\_PNT\_STATE\_ERROR, sync\_point\_list, and unlock\_read\_write().

Referenced by is\_sync\_pnt\_announced().

### 7.67.3.14 `is_sync_pnt_announced()`

```
bool SyncPntListBase::is_sync_pnt_announced (
    std::wstring const & label ) [virtual]
```

Check if specified synchronization point has been announced.

#### Returns

True if announced.

#### Parameters

<i>label</i>	The synchronization point label.
--------------	----------------------------------

Definition at line 584 of file SyncPntListBase.cpp.

References get\_sync\_pnt\_state(), state, and TrickHLA::SYNC\_PNT\_STATE\_ANNOUNCED.

Referenced by SpaceFOM::ExecutionControl::check\_for\_shutdown().

### 7.67.3.15 `lock_read_only()`

```
void SyncPntListBase::lock_read_only ( ) [protected]
```

Put the lock in read only state.

Definition at line 710 of file SyncPntListBase.cpp.

References read\_locks, and write\_locks.

Referenced by achieve\_sync\_pnt(), DIS::PausePointList::check\_state(), IMSim::PausePointList::check\_state(), contains(), convert\_sync\_pts(), get\_sync\_pnt(), IMSim::PausePointList::print\_sync\_pnts(), DIS::PausePointList::print\_sync\_pnts(), print\_sync\_pnts(), register\_sync\_pnt(), DIS::PausePointList::to\_string(), IMSim::PausePointList::to\_string(), and to\_string().

### 7.67.3.16 `lock_read_write()`

```
void SyncPntListBase::lock_read_write ( ) [protected]
```

Put the lock in read/write state.

Definition at line 718 of file SyncPntListBase.cpp.

References read\_locks, and write\_locks.

Referenced by add\_sync\_pnt(), IMSim::PausePointList::clear\_sync\_pnt(), DIS::PausePointList::clear\_sync\_pnt(), clear\_sync\_pnt(), get\_sync\_pnt\_state(), and reset().

#### 7.67.3.17 mark\_announced()

```
bool SyncPntListBase::mark_announced (
    std::wstring const & label ) [virtual]
```

Mark the given synchronization point as existing in the federation.

Returns

True if synchronization point label is valid.

Parameters

<i>label</i>	The synchronization point label.
--------------	----------------------------------

**Trick Job Class:** *initialization*

Definition at line 675 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::get\_label(), TrickHLA::SyncPnt::set\_state(), TrickHLA::SYNC\_PNT\_STATE\_ANNOUCED, and sync\_point\_list.

Referenced by DIS::ExecutionControl::announce\_sync\_point(), DSES::ExecutionControl::announce\_sync\_point(), IMSim::ExecutionControl::announce\_sync\_point(), SpaceFOM::ExecutionControl::announce\_sync\_point(), and announce\_sync\_point().

#### 7.67.3.18 mark\_registered()

```
bool SyncPntListBase::mark_registered (
    std::wstring const & label ) [virtual]
```

Mark the given synchronization point as registered in the federation.

Returns

True if synchronization point label is valid.

Parameters

<i>label</i>	The synchronization point label.
--------------	----------------------------------

**Trick Job Class:** *initialization*

Definition at line 657 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::get\_label(), TrickHLA::SyncPnt::set\_state(), TrickHLA::SYNC\_PNT\_STATE\_REGISTERED, and sync\_point\_list.

Referenced by sync\_point\_registration\_failed(), and sync\_point\_registration\_succeeded().

#### 7.67.3.19 mark\_synchronized()

```
bool SyncPntListBase::mark_synchronized (
    std::wstring const & label ) [virtual]
```

Mark the given synchronization point as synchronized in the federation.

#### Returns

True if synchronization point label is valid.

#### Parameters

<i>label</i>	The synchronization point label.
--------------	----------------------------------

#### Trick Job Class: *initialization*

Reimplemented in [IMSim::ExecutionControl](#).

Definition at line 693 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::get\_label(), TrickHLA::SyncPnt::set\_state(), TrickHLA::SYNC\_PNT\_STATE\_SYNCHRONIZED, and sync\_point\_list.

Referenced by TrickHLA::Federate::federation\_synchronized().

#### 7.67.3.20 operator=()

```
SyncPntListBase& TrickHLA::SyncPntListBase::operator= (
    const SyncPntListBase & rhs ) [private]
```

Assignment operator for [SyncPntListBase](#) class.

This assignment operator is private to prevent inadvertent copies.

#### 7.67.3.21 print\_sync\_pnts()

```
void SyncPntListBase::print_sync_pnts ( ) [virtual]
```

Dumps synchronization point information to the screen.

Reimplemented in [TrickHLA::TimedSyncPntList](#), [DIS::PausePointList](#), and [IMSim::PausePointList](#).

Definition at line 771 of file SyncPntListBase.cpp.

References lock\_read\_only(), sync\_point\_list, and unlock\_read\_only().

Referenced by TrickHLA::Federate::achieve\_and\_wait\_for\_synchronization(), TrickHLA::ExecutionControlBase::clear\_multiphase\_init\_sync\_points(), and wait\_for\_all\_announcements().

#### 7.67.3.22 register\_all\_sync\_pnts() [1/2]

```
void SyncPntListBase::register_all_sync_pnts (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador ) [virtual]
```

Register all the synchronization point in this list.

#### Parameters

<i>rti_ambassador</i>	Reference to RTI Ambassador.
-----------------------	------------------------------

Definition at line 168 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::is\_registered(), TrickHLA::SyncPnt::register\_sync\_point(), sync\_point\_list, TRICKHLA\_A\_RESTORE\_FPU\_CONTROL\_WORD, TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, and TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD.

Referenced by [DIS::ExecutionControl::determine\\_federation\\_master\(\)](#), [DSES::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), [IMSim::ExecutionControl::pre\\_multi\\_phase\\_init\\_processes\(\)](#), [register\\_all\\_sync\\_pnts\(\)](#), and [SpaceFOM::ExecutionControl::role\\_determination\\_process\(\)](#).

### 7.67.3.23 register\_all\_sync\_pnts() [2/2]

```
void SyncPntListBase::register_all_sync_pnts (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    RTI1516_NAMESPACE::FederateHandleSet const & federate_handle_set ) [virtual]
```

Register all the sync-points with the RTI for the given set of federate handles.

#### Parameters

<i>rti_ambassador</i>	Reference to the HLA RTI Ambassador instance.
<i>federate_handle_set</i>	<a href="#">Federate</a> Handle set.

Definition at line 192 of file SyncPntListBase.cpp.

References [TrickHLA::SyncPnt::is\\_registered\(\)](#), [register\\_all\\_sync\\_pnts\(\)](#), [TrickHLA::SyncPnt::register\\_sync\\_point\(\)](#), [sync\\_point\\_list](#), [TRICKHLA\\_RESTORE\\_FPU\\_CONTROL\\_WORD](#), [TRICKHLA\\_SAVE\\_FPU\\_CONTROL\\_WORD](#), and [TRICKHLA\\_VALIDATE\\_FPU\\_CONTROL\\_WORD](#).

### 7.67.3.24 register\_sync\_pnt() [1/2]

```
SyncPnt * SyncPntListBase::register_sync_pnt (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    RTI1516_NAMESPACE::FederateHandleSet const & federate_handle_set,
    std::wstring const & label ) [virtual]
```

Register a synchronization point associated with the given label.

#### Returns

The newly created synchronization point.

#### Parameters

<i>rti_ambassador</i>	Reference to RTI Ambassador.
<i>label</i>	Synchronization point label.
<i>federate_handle_set</i>	<a href="#">Federate</a> handle set.

Definition at line 140 of file SyncPntListBase.cpp.

References [TrickHLA::SyncPnt::get\\_label\(\)](#), [TrickHLA::SyncPnt::is\\_registered\(\)](#), [lock\\_read\\_only\(\)](#), [TrickHLA::SyncPnt::register\\_sync\\_point\(\)](#), [sync\\_point\\_list](#), and [unlock\\_read\\_only\(\)](#).

### 7.67.3.25 register\_sync\_pnt() [2/2]

```
SyncPnt * SyncPntListBase::register_sync_pnt (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    std::wstring const & label ) [virtual]
```

Register a synchronization point associated with the given label.

#### Returns

The newly created synchronization point.

#### Parameters

<i>rti_ambassador</i>	Reference to RTI Ambassador.
-----------------------	------------------------------

## Parameters

<i>label</i>	Synchronization point label.
--------------	------------------------------

Definition at line 113 of file SyncPntListBase.cpp.

References TrickHLA::SyncPnt::get\_label(), TrickHLA::SyncPnt::is\_registered(), lock\_read\_only(), TrickHLA::SyncPnt::register\_sync\_point(), sync\_point\_list, and unlock\_read\_only().

Referenced by DSES::ExecutionControl::freeze\_mode\_announce(), DIS::ExecutionControl::freeze\_mode\_announce(), IMSim::ExecutionControl::freeze\_mode\_announce(), SpaceFOM::ExecutionControl::freeze\_mode\_announce(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::role\_determination\_process(), DSES::ExecutionControl::run\_mode\_transition(), DIS::ExecutionControl::run\_mode\_transition(), IMSim::ExecutionControl::run\_mode\_transition(), SpaceFOM::ExecutionControl::run\_mode\_transition(), DSES::ExecutionControl::shutdown\_mode\_transition(), DIS::ExecutionControl::shutdown\_mode\_transition(), IMSim::ExecutionControl::shutdown\_mode\_transition(), and SpaceFOM::ExecutionControl::shutdown\_mode\_transition().

### 7.67.3.26 reset()

void SyncPntListBase::reset ( ) [virtual]

Reset the list of synchronization points.

Definition at line 620 of file SyncPntListBase.cpp.

References lock\_read\_write(), sync\_point\_list, and unlock\_read\_write().

Referenced by IMSim::ExecutionControl::reinstate\_logged\_sync\_pts(), and ~SyncPntListBase().

### 7.67.3.27 set\_debug\_level()

virtual void TrickHLA::SyncPntListBase::set\_debug\_level ( DebugHandler *hdlr* ) [inline], [virtual]

Set the debug handler.

## Parameters

<i>hdlr</i>	TrickHLA::DebugHandler.
-------------	-------------------------

Definition at line 253 of file SyncPntListBase.hh.

References debug\_handler, and TrickHLA::DebugHandler::set().

Referenced by TrickHLA::ExecutionControlBase::initialize().

### 7.67.3.28 sync\_point\_registration\_failed()

void SyncPntListBase::sync\_point\_registration\_failed ( std::wstring const & *label*, bool *not\_unique* ) [virtual]

Callback from TrickHLA::FedAmb through TrickHLA::Federate for when registration of a synchronization point fails. and is one of the sync-points created.

## Parameters

<i>label</i>	Sync-point label.
<i>not_unique</i>	True if not unique label.

Reimplemented in [DIS::ExecutionControl](#).

Definition at line 231 of file SyncPntListBase.cpp.

References contains(), debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDE RATE, mark\_registered(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

Referenced by TrickHLA::Federate::sync\_point\_registration\_failed().

#### 7.67.3.29 sync\_point\_registration\_succeeded()

```
void SyncPntListBase::sync_point_registration_succeeded (
    std::wstring const & label ) [virtual]
```

Callback from [TrickHLA::FedAmb](#) through [TrickHLA::Federate](#) for when registration of a synchronization point success. and is one of the sync-points created.

Parameters

<i>label</i>	Sync-point label.
--------------	-------------------

Reimplemented in [DIS::ExecutionControl](#).

Definition at line 219 of file SyncPntListBase.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, mark\_registered(), TrickHLA::DebugHandler::should\_print(), and THLA\_NEWLINE.

Referenced by TrickHLA::Federate::sync\_point\_registration\_succeeded().

#### 7.67.3.30 to\_string()

```
wstring SyncPntListBase::to_string ( ) [virtual]
```

Returns a wide string representing the state of the synchronization points.

Returns

String summary of synchronization points.

Reimplemented in [DIS::PausePointList](#), and [IMSim::PausePointList](#).

Definition at line 736 of file SyncPntListBase.cpp.

References lock\_read\_only(), sync\_point\_list, TrickHLA::SyncPnt::to\_string(), and unlock\_read\_only().

#### 7.67.3.31 unlock\_read\_only()

```
void SyncPntListBase::unlock_read_only ( ) [protected]
```

Unlock the read only lock.

Definition at line 726 of file SyncPntListBase.cpp.

References read\_locks.

Referenced by achieve\_sync\_pnt(), [DIS::PausePointList::check\\_state\(\)](#), [IMSim::PausePointList::check\\_state\(\)](#), contains(), convert\_sync\_pts(), get\_sync\_pnt(), [IMSim::PausePointList::print\\_sync\\_pnts\(\)](#), [DIS::PausePointList::print\\_sync\\_pnts\(\)](#), [print\\_sync\\_pnts\(\)](#), register\_sync\_pnt(), [DIS::PausePointList::to\\_string\(\)](#), [IMSim::PausePointList::to\\_string\(\)](#), and [to\\_string\(\)](#).

#### 7.67.3.32 unlock\_read\_write()

```
void SyncPntListBase::unlock_read_write ( ) [protected]
```

Unlock the read/write lock.

Definition at line 731 of file SyncPntListBase.cpp.

References write\_locks.

Referenced by add\_sync\_pnt(), IMSim::PausePointList::clear\_sync\_pnt(), DIS::PausePointList::clear\_sync\_pnt(), clear\_sync\_pnt(), get\_sync\_pnt\_state(), and reset().

### 7.67.3.33 wait\_for\_all\_announcements()

```
void SyncPntListBase::wait_for_all_announcements (
    Federate * fed_ptr ) [virtual]
```

Wait for all the sync-points to be registered with the RTI.

#### Parameters

<i>fed_ptr</i>	Pointer to <a href="#">TrickHLA::Federate</a> instance.
----------------	---

Definition at line 277 of file SyncPntListBase.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, print<->\_sync\_pnts(), TrickHLA::DebugHandler::should\_print(), sync\_point\_list, THLA\_NEWLINE, and TrickHLA::SyncPnt<->:wait\_for\_announce().

Referenced by DIS::ExecutionControl::determine\_federation\_master(), DSES::ExecutionControl::pre\_multi\_phase<->init\_processes(), and IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes().

### 7.67.3.34 wait\_for\_announcement()

```
void SyncPntListBase::wait_for_announcement (
    Federate * fed_ptr,
    std::wstring const & label ) [virtual]
```

Wait for the sync-point to be announced by the RTI.

#### Parameters

<i>fed_ptr</i>	Pointer to <a href="#">TrickHLA::Federate</a> instance.
<i>label</i>	Sync-point label.

Definition at line 258 of file SyncPntListBase.cpp.

References debug\_handler, TrickHLA::DEBUG\_LEVEL\_2\_TRACE, TrickHLA::DEBUG\_SOURCE\_FEDERATE, get<->\_sync\_pnt(), TrickHLA::DebugHandler::should\_print(), THLA\_ENDL, TrickHLA::SyncPnt::to\_string(), and TrickHLA::SyncPnt::wait\_for\_announce().

Referenced by SpaceFOM::ExecutionControl::early\_joiner\_hla\_init\_process().

### 7.67.3.35 wait\_for\_list\_synchronization()

```
void SyncPntListBase::wait_for_list_synchronization (
    Federate * federate )
```

Wait for all the synchronization points in the list to be achieved.

#### Parameters

<i>federate</i>	<a href="#">Federate</a> associated with this list.
-----------------	---

Definition at line 390 of file SyncPntListBase.cpp.

References TrickHLA::Federate::check\_for\_shutdown\_with\_termination(), TrickHLA::SyncPnt::is\_achieved(), TrickHLA::SyncPnt::wait\_for\_all\_announcements(), and TrickHLA::SyncPnt::wait\_for\_announcement().

A::Federate::is\_execution\_member(), TrickHLA::SyncPnt::is\_valid(), TrickHLA::SyncPnt::set\_state(), TrickHLA::SYNC←\_PNT\_STATE\_EXISTS, sync\_point\_list, and THLA\_ENDL.

Referenced by TrickHLA::ExecutionControlBase::wait\_for\_all\_multiphase\_init\_sync\_pnts().

## 7.67.4 Friends And Related Function Documentation

### 7.67.4.1 init\_attrTrickHLA\_\_SyncPntListBase

```
void init_attrTrickHLA__SyncPntListBase ( ) [friend]
```

### 7.67.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 62 of file SyncPntListBase.hh.

## 7.67.5 Field Documentation

### 7.67.5.1 debug\_handler

```
DebugHandler TrickHLA::SyncPntListBase::debug_handler [protected]
```

#### Units: –

Decides whether to print any debug messages.

Definition at line 278 of file SyncPntListBase.hh.

Referenced by DIS::ExecutionControl::announce\_sync\_point(), DSES::ExecutionControl::announce\_sync\_point(), IMSim::ExecutionControl::announce\_sync\_point(), SpaceFOM::ExecutionControl::announce\_sync\_point(), announce←\_sync\_point(), TrickHLA::ExecutionControl::clear\_multiphase\_init\_sync\_points(), DIS::ExecutionControl::clear←\_multiphase\_init\_sync\_points(), TrickHLA::ExecutionControlBase::clear\_multiphase\_init\_sync\_points(), DSES::ExecutionControl::determine\_federation\_master(), DIS::ExecutionControl::determine\_federation\_master(), IMSim::ExecutionControl::determine\_if\_late\_joining\_or\_restoring\_federate(), DSES::ExecutionControl::initialize(), DIS::ExecutionControl::initialize(), IMSim::ExecutionControl::initialize(), SpaceFOM::ExecutionControl::initialize(), TrickHLA::ExecutionControlBase::mark\_object\_as\_deleted\_from\_federation(), TrickHLA::ExecutionControlBase::object←\_instance\_name\_reservation\_succeeded(), TrickHLA::ExecutionControl::pre\_multi\_phase\_init\_processes(), DSES::ExecutionControl::pre\_multi\_phase\_init\_processes(), DIS::ExecutionControl::pre\_multi\_phase\_init\_processes(), IMSim::ExecutionControl::pre\_multi\_phase\_init\_processes(), SpaceFOM::ExecutionControl::pre\_multi\_phase\_init←\_processes(), TrickHLA::ExecutionControlBase::receive\_execution\_configuration(), SpaceFOM::ExecutionControl::receive\_init\_root\_ref\_frame(), SpaceFOM::ExecutionControl::receive\_interaction(), IMSim::ExecutionControl::receive\_interaction(), SpaceFOM::ExecutionControl::receive\_root\_ref\_frame(), TrickHLA::ExecutionControlBase::register\_objects\_with\_RTI(), SpaceFOM::ExecutionControl::role\_determination\_process(), DSES::ExecutionControl::run\_mode\_transition(), DIS::ExecutionControl::run\_mode\_transition(), IMSim::ExecutionControl::run\_mode←\_transition(), SpaceFOM::ExecutionControl::run\_mode\_transition(), TrickHLA::ExecutionControlBase::send\_execution←\_configuration(), SpaceFOM::ExecutionControl::send\_init\_root\_ref\_frame(), SpaceFOM::ExecutionControl::send←\_root\_ref\_frame(), set\_debug\_level(), IMSim::ExecutionControl::setup\_interaction\_ref\_attributes(), SpaceFOM::ExecutionControl::setup\_interaction\_ref\_attributes(), IMSim::ExecutionControl::start\_federation\_save\_at\_scenario←\_time(), sync\_point\_registration\_failed(), sync\_point\_registration\_succeeded(), wait\_for\_all\_announcements(), wait←\_for\_announcement(), and SpaceFOM::ExecutionControl::wait\_on\_root\_frame\_discovered\_synchronization().

### 7.67.5.2 read\_locks

```
int TrickHLA::SyncPntListBase::read_locks [protected]
```

**Units:** –

Read lock count.

Definition at line 271 of file SyncPntListBase.hh.

Referenced by lock\_read\_only(), lock\_read\_write(), and unlock\_read\_only().

**7.67.5.3 reconfig\_name**

```
std::wstring TrickHLA::SyncPntListBase::reconfig_name [protected]
```

**Data I/O:** \*\*

Wide string of the reconfiguration name.

Definition at line 276 of file SyncPntListBase.hh.

Referenced by DIS::PausePointList::clear\_sync\_pnt(), IMSim::PausePointList::clear\_sync\_pnt(), and get\_reconfig\_name().

**7.67.5.4 sync\_point\_list**

```
std::vector< SyncPnt * > TrickHLA::SyncPntListBase::sync_point_list [protected]
```

**Data I/O:** \*\*

Vector of synchronization points.

Definition at line 274 of file SyncPntListBase.hh.

Referenced by achieve\_sync\_pnt(), add\_sync\_pnt(), IMSim::PausePointList::check\_state(), DIS::PausePointList::check\_state(), DIS::PausePointList::clear\_sync\_pnt(), IMSim::PausePointList::clear\_sync\_pnt(), clear\_sync\_pnt(), contains(), convert\_sync\_pts(), get\_size(), get\_sync\_pnt(), get\_sync\_pnt\_state(), mark\_announced(), mark\_registered(), mark\_synchronized(), IMSim::PausePointList::print\_sync\_pnts(), DIS::PausePointList::print\_sync\_pnts(), print\_sync\_pnts(), register\_all\_sync\_pnts(), register\_sync\_pnt(), reset(), DIS::PausePointList::to\_string(), IMSim::PausePointList::to\_string(), to\_string(), wait\_for\_all\_announcements(), DSES::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), IMSim::ExecutionControl::wait\_for\_all\_multiphase\_init\_sync\_pnts(), wait\_for\_list\_synchronization(), and TrickHLA::ExecutionControlBase::wait\_for\_sync\_point\_announce().

**7.67.5.5 write\_locks**

```
int TrickHLA::SyncPntListBase::write_locks [protected]
```

**Units:** –

Write lock count.

Definition at line 272 of file SyncPntListBase.hh.

Referenced by lock\_read\_only(), lock\_read\_write(), and unlock\_read\_write().

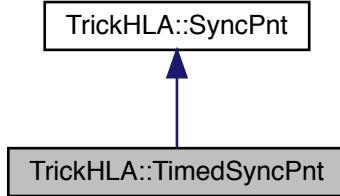
The documentation for this class was generated from the following files:

- [SyncPntListBase.hh](#)
- [SyncPntListBase.cpp](#)

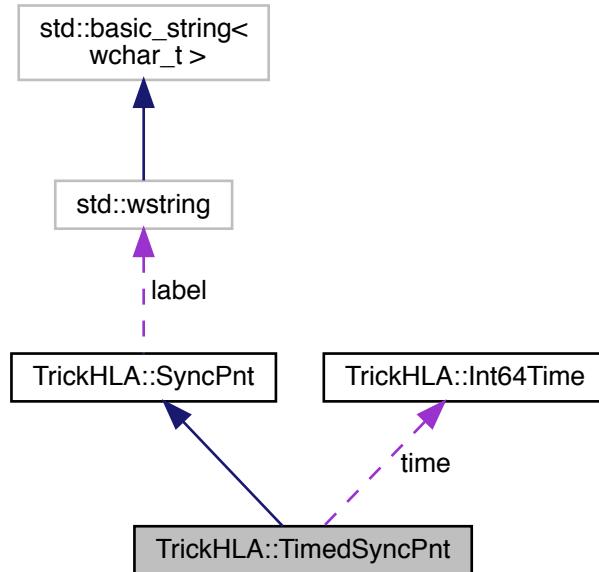
**7.68 TrickHLA::TimedSyncPnt Class Reference**

```
#include <TimedSyncPnt.hh>
```

Inheritance diagram for TrickHLA::TimedSyncPnt:



Collaboration diagram for TrickHLA::TimedSyncPnt:



## Public Member Functions

- [TimedSyncPnt \(\)](#)  
*Default constructor for the `TrickHLA TimedSyncPnt` class.*
- [TimedSyncPnt \(std::wstring const & l\)](#)  
*Initialization constructor.*
- [TimedSyncPnt \(const Int64Time & t, std::wstring const & l\)](#)  
*Initialization constructor.*

- virtual `~TimedSyncPnt ()`  
*Destructor for the `TrickHLA TimedSyncPnt` class.*
- virtual const `Int64Time & get_time () const`  
*Get the synchronization point action time.*
- virtual void `set_time (const Int64Time &t)`  
*Set the synchronization point action time.*
- virtual std::wstring `to_string ()`  
*Create a C++ wide string with the synchronization point label and current state.*
- virtual void `convert (LoggableSyncPnt &log_sync_pnt)`  
*Convert the synchronization point into and loggable synchronization point.*

## Protected Attributes

- `Int64Time time`  
**Units:** –  
*Synchronization point action time.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA_TimedSyncPnt ()`

### 7.68.1 Detailed Description

Definition at line 57 of file `TimedSyncPnt.hh`.

### 7.68.2 Constructor & Destructor Documentation

#### 7.68.2.1 `TimedSyncPnt()` [1/3]

`TimedSyncPnt::TimedSyncPnt ( )`

Default constructor for the `TrickHLA TimedSyncPnt` class.

**Trick Job Class:** *initialization*

Definition at line 49 of file `TimedSyncPnt.cpp`.

#### 7.68.2.2 `TimedSyncPnt()` [2/3]

`TimedSyncPnt::TimedSyncPnt ( std::wstring const & l ) [explicit]`

Initialization constructor.

#### Parameters

<code>l</code>	Synchronization point label.
----------------	------------------------------

**Trick Job Class:** *initialization*

Definition at line 58 of file `TimedSyncPnt.cpp`.

### 7.68.2.3 `TimedSyncPnt()` [3/3]

```
TimedSyncPnt::TimedSyncPnt (
    const Int64Time & t,
    std::wstring const & l )
```

Initialization constructor.

#### Parameters

<i>t</i>	Synchronization point action time.
<i>l</i>	Synchronization point label.

**Trick Job Class:** *initialization*

Definition at line 67 of file TimedSyncPnt.cpp.

### 7.68.2.4 `~TimedSyncPnt()`

```
TimedSyncPnt::~TimedSyncPnt ( ) [virtual]
```

Destructor for the [TrickHLA TimedSyncPnt](#) class.

**Trick Job Class:** *shutdown*

Definition at line 76 of file TimedSyncPnt.cpp.

## 7.68.3 Member Function Documentation

### 7.68.3.1 `convert()`

```
void TimedSyncPnt::convert (
    LoggableSyncPnt & log_sync_pnt ) [virtual]
```

Convert the synchronization point into and loggable synchronization point.

#### Parameters

<i>log_sync_pnt</i>	Reference to a loggable synchronization point.
---------------------	--

Reimplemented from [TrickHLA::SyncPnt](#).

Definition at line 113 of file TimedSyncPnt.cpp.

References `TrickHLA::Int64Time::getTimeInMicros()`, `TrickHLA::StringUtilities::ip_strdup_wstring()`, `TrickHLA::LoggableSyncPnt::label`, `TrickHLA::SyncPnt::label`, `TrickHLA::LoggableSyncPnt::state`, `TrickHLA::SyncPnt::state`, `TH_LA_ENDL`, `TrickHLA::LoggableTimedSyncPnt::time`, and `time`.

Referenced by `TrickHLA::TimedSyncPntList::convert_sync_pts()`.

### 7.68.3.2 `get_time()`

```
virtual const Int64Time& TrickHLA::TimedSyncPnt::get_time ( ) const [inline], [virtual]
```

Get the synchronization point action time.

#### Returns

Time for synchronization point action.

Definition at line 91 of file TimedSyncPnt.hh.

References `time`.

Referenced by TrickHLA::TimedSyncPntList::achieve\_all\_sync\_pnts(), TrickHLA::TimedSyncPntList::check\_sync\_pnts(), IMSim::PausePointList::print\_sync\_pnts(), DIS::PausePointList::print\_sync\_pnts(), and TrickHLA::TimedSyncPntList::print\_sync\_pnts().

### 7.68.3.3 set\_time()

```
virtual void TrickHLA::TimedSyncPnt::set_time (
    const Int64Time & t ) [inline], [virtual]
```

Set the synchronization point action time.

#### Parameters

<i>t</i>	The synchronization point action time.
----------	--

Definition at line 95 of file TimedSyncPnt.hh.

References time.

### 7.68.3.4 to\_string()

```
std::wstring TimedSyncPnt::to_string ( ) [virtual]
```

Create a C++ wide string with the synchronization point label and current state.

#### Returns

A string with the synchronization point label and current state.

Reimplemented from [TrickHLA::SyncPnt](#).

Definition at line 81 of file TimedSyncPnt.cpp.

References TrickHLA::SyncPnt::label, TrickHLA::SyncPnt::state, TrickHLA::SYNC\_PNT\_STATE\_ACHIEVED, TrickHLA::SYNC\_PNT\_STATE\_ANNOUNCED, TrickHLA::SYNC\_PNT\_STATE\_ERROR, TrickHLA::SYNC\_PNT\_STATE\_EXISTS, TrickHLA::SYNC\_PNT\_STATE\_REGISTERED, time, and TrickHLA::Int64Time::toString().

## 7.68.4 Friends And Related Function Documentation

### 7.68.4.1 init\_attrTrickHLA\_TimedSyncPnt

```
void init_attrTrickHLA_TimedSyncPnt ( ) [friend]
```

### 7.68.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 64 of file TimedSyncPnt.hh.

## 7.68.5 Field Documentation

### 7.68.5.1 time

```
Int64Time TrickHLA::TimedSyncPnt::time [protected]
```

**Units:** –

Synchronization point action time.

Definition at line 109 of file `TimedSyncPnt.hh`.

Referenced by `convert()`, `get_time()`, `set_time()`, and `to_string()`.

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

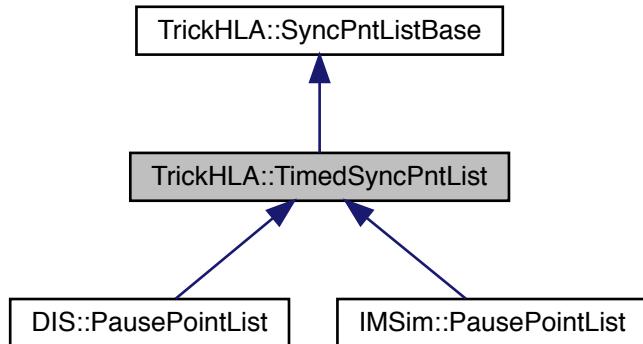
- [TimedSyncPnt.hh](#)

- [TimedSyncPnt.cpp](#)

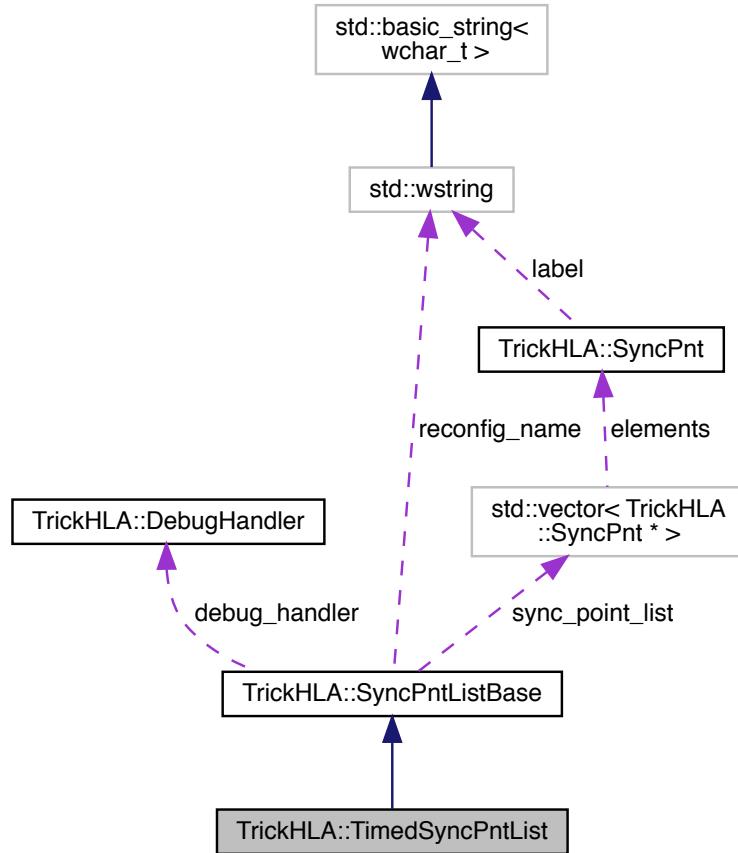
## 7.69 TrickHLA::TimedSyncPntList Class Reference

```
#include <TimedSyncPntList.hh>
```

Inheritance diagram for TrickHLA::TimedSyncPntList:



Collaboration diagram for TrickHLA::TimedSyncPntList:



## Public Member Functions

- **TimedSyncPntList ()**  
*Default constructor for the [TrickHLA TimedSyncPntList](#) class.*
- **virtual ~TimedSyncPntList ()**  
*Pure virtual destructor for the [TrickHLA TimedSyncPntList](#) class.*
- **virtual void add\_sync\_pnt (std::wstring const &label)**  
*Add the given synchronization point label to the list.*
- **virtual void add\_sync\_pnt (std::wstring const &label, const Int64Time &time)**  
*Add the given synchronization point label and action time to the list.*
- **virtual bool achieve\_all\_sync\_pnts (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador) throw ( R ← TI1516\_NAMESPACE::SynchronizationPointLabelNotAnnounced, R ← TI1516\_NAMESPACE::FederateNot ← ExecutionMember, R ← TI1516\_NAMESPACE::SaveInProgress, R ← TI1516\_NAMESPACE::RestoreInProgress, R ← TI1516\_NAMESPACE::NotConnected, R ← TI1516\_NAMESPACE::RTIinternalError )**  
*Achieve all the synchronization points in the list.*

- virtual bool `achieve_all_sync_pnts` (RTI1516\_NAMESPACE::RTIambassador &rti\_ambassador, const `Int64Time` &checkTime)  
*Acknowledge all the synchronization point less than or equal to the given time.*
- virtual bool `check_sync_pnts` (const `Int64Time` &checkTime)  
*Determine if we have any synchronization point that has a action time less than the given time.*
- virtual void `convert_sync_pts` (`LoggableSyncPnt` \*pts)  
*Converts the vector of synchronization points to a checkpoint-able class.*
- virtual void `print_sync_pnts` ()  
*Dumps synchronization point information to the screen.*

## Private Member Functions

- `TimedSyncPntList` (const `TimedSyncPntList` &rhs)  
*Copy constructor for `TimedSyncPntList` class.*
- `TimedSyncPntList` & `operator=` (const `TimedSyncPntList` &rhs)  
*Assignment operator for `TimedSyncPntList` class.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA_TimedSyncPntList` ()

## Additional Inherited Members

### 7.69.1 Detailed Description

Definition at line 59 of file `TimedSyncPntList.hh`.

### 7.69.2 Constructor & Destructor Documentation

#### 7.69.2.1 `TimedSyncPntList()` [1/2]

`TimedSyncPntList::TimedSyncPntList` ( )  
 Default constructor for the `TrickHLA TimedSyncPntList` class.

**Trick Job Class:** *initialization*

Definition at line 55 of file `TimedSyncPntList.cpp`.

#### 7.69.2.2 `~TimedSyncPntList()`

`virtual TrickHLA::TimedSyncPntList::~TimedSyncPntList` ( ) [inline], [virtual]  
 Pure virtual destructor for the `TrickHLA TimedSyncPntList` class.

Definition at line 79 of file `TimedSyncPntList.hh`.

#### 7.69.2.3 `TimedSyncPntList()` [2/2]

`TrickHLA::TimedSyncPntList::TimedSyncPntList` ( )  
 const `TimedSyncPntList` & `rhs` ) [private]

Copy constructor for `TimedSyncPntList` class.

This constructor is private to prevent inadvertent copies.

### 7.69.3 Member Function Documentation

#### 7.69.3.1 achieve\_all\_sync\_pnts() [1/2]

```
virtual bool TrickHLA::TimedSyncPntList::achieve_all_sync_pnts (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador ) throw ( RTI1516_NAMESPACE::←
SynchronizationPointLabelNotAnnounced, RTI1516_NAMESPACE::FederateNotExecutionMember, RTI1516←
NAMESPACE::SaveInProgress, RTI1516_NAMESPACE::RestoreInProgress, RTI1516_NAMESPACE::NotConnected,
RTI1516_NAMESPACE::RTIinternalError) [inline], [virtual]
```

Achieve all the synchronization points in the list.

##### Returns

True is any synchronization point in the list was achieved.

##### Parameters

<i>rti_ambassador</i>	Reference to the HLA RTI Ambassador instance.
-----------------------	---

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 93 of file TimedSyncPntList.hh.

Referenced by [IMSim::ExecutionControl::check\\_freeze\\_exit\(\)](#), [DIS::ExecutionControl::exit\\_freeze\(\)](#), [IMSim::ExecutionControl::exit\\_freeze\(\)](#), and [IMSim::ExecutionControl::is\\_save\\_initiated\(\)](#).

#### 7.69.3.2 achieve\_all\_sync\_pnts() [2/2]

```
bool TimedSyncPntList::achieve_all_sync_pnts (
    RTI1516_NAMESPACE::RTIambassador & rti_ambassador,
    const Int64Time & checkTime ) [virtual]
```

Acknowledge all the synchronization point less than or equal to the given time.

##### Returns

True if acknowledged, false otherwise.

##### Parameters

<i>rti_ambassador</i>	Reference to RTI Ambassador.
<i>checkTime</i>	Time to check.

Definition at line 77 of file TimedSyncPntList.cpp.

References [TrickHLA::SyncPnt::exists\(\)](#), [TrickHLA::TimedSyncPnt::get\\_time\(\)](#), and [TrickHLA::SyncPnt::is\\_achieved\(\)](#).

#### 7.69.3.3 add\_sync\_pnt() [1/2]

```
void TimedSyncPntList::add_sync_pnt (
    std::wstring const & label ) [virtual]
```

Add the given synchronization point label to the list.

## Parameters

<i>label</i>	Synchronization point label.
--------------	------------------------------

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 59 of file TimedSyncPntList.cpp.

Referenced by [DIS::ExecutionControl::add\\_pause\(\)](#), and [IMSim::ExecutionControl::add\\_pause\(\)](#).

#### 7.69.3.4 add\_sync\_pnt() [2/2]

```
void TimedSyncPntList::add_sync_pnt (
    std::wstring const & label,
    const Int64Time & time ) [virtual]
```

Add the given synchronization point label and action time to the list.

## Parameters

<i>label</i>	Synchronization point label.
<i>time</i>	Action time.

Definition at line 67 of file TimedSyncPntList.cpp.

#### 7.69.3.5 check\_sync\_pnts()

```
bool TimedSyncPntList::check_sync_pnts (
    const Int64Time & checkTime ) [virtual]
```

Determine if we have any synchronization point that has a action time less than the given time.

## Returns

True if sync-point is ready to be cleared.

## Parameters

<i>checkTime</i>	Time to check.
------------------	----------------

Definition at line 113 of file TimedSyncPntList.cpp.

References [TrickHLA::SyncPnt::get\\_state\(\)](#), [TrickHLA::TimedSyncPnt::get\\_time\(\)](#), and [TrickHLA::SYNC\\_PNT\\_STAT\\_E\\_EXISTS](#).

Referenced by [IMSim::ExecutionControl::check\\_pause\(\)](#), and [IMSim::ExecutionControl::exit\\_freeze\(\)](#).

#### 7.69.3.6 convert\_sync\_pts()

```
void TimedSyncPntList::convert_sync_pts (
    LoggableSyncPnt * pts ) [virtual]
```

Converts the vector of synchronization points to a checkpoint-able class.

## Parameters

<i>pts</i>	Area to populate.
------------	-------------------

Reimplemented from [TrickHLA::SyncPntListBase](#).

Definition at line 132 of file [TimedSyncPntList.cpp](#).

References [TrickHLA::TimedSyncPnt::convert\(\)](#), and [THLA\\_ENDL](#).

Referenced by [IMSim::ExecutionControl::convert\\_loggable\\_sync\\_pts\(\)](#).

#### 7.69.3.7 operator=()

```
TimedSyncPntList& TrickHLA::TimedSyncPntList::operator= (
    const TimedSyncPntList & rhs ) [private]
```

Assignment operator for [TimedSyncPntList](#) class.

This assignment operator is private to prevent inadvertent copies.

#### 7.69.3.8 print\_sync\_pnts()

```
void TimedSyncPntList::print_sync_pnts ( ) [virtual]
```

Dumps synchronization point information to the screen.

Reimplemented from [TrickHLA::SyncPntListBase](#).

Reimplemented in [DIS::PausePointList](#), and [IMSim::PausePointList](#).

Definition at line 164 of file [TimedSyncPntList.cpp](#).

References [TrickHLA::TimedSyncPnt::get\\_time\(\)](#), and [TrickHLA::Int64Time::getDoubleTime\(\)](#).

### 7.69.4 Friends And Related Function Documentation

#### 7.69.4.1 init\_attrTrickHLA\_\_TimedSyncPntList

```
void init_attrTrickHLA__TimedSyncPntList ( ) [friend]
```

#### 7.69.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 66 of file [TimedSyncPntList.hh](#).

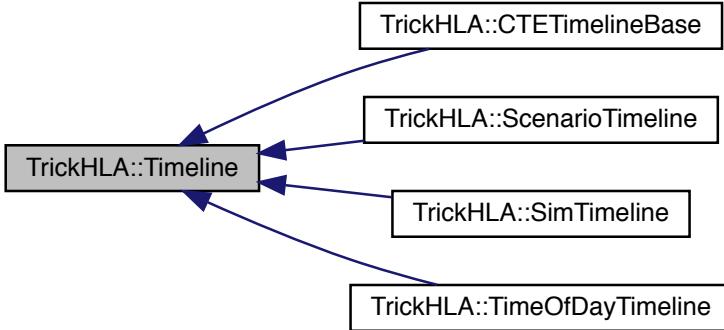
The documentation for this class was generated from the following files:

- [TimedSyncPntList.hh](#)
- [TimedSyncPntList.cpp](#)

## 7.70 TrickHLA::Timeline Class Reference

```
#include <Timeline.hh>
```

Inheritance diagram for TrickHLA::Timeline:



## Public Member Functions

- `Timeline (double t0=0.0)`  
*Initialization constructor for the `TrickHLA CTETimelineBase` class.*
- `virtual ~Timeline ()=0`  
*Pure virtual destructor for the `TrickHLA CTETimelineBase` class.*
- `virtual double get_time ()=0`  
*Get the current time for this timeline in seconds.*
- `virtual double get_elapsed_time ()`  
*Get the elapsed time for this timeline in seconds from epoch.*
- `virtual void set_epoch (double time)`  
*Set the epoch for this timeline in seconds.*
- `virtual double get_epoch ()`  
*Get the epoch for this timeline in seconds.*

## Protected Attributes

- `double epoch`  
**Units:** s  
*Epoch for the simulation.*

## Private Member Functions

- `Timeline (const Timeline &rhs)`  
*Copy constructor for `Timeline` class.*
- `Timeline & operator= (const Timeline &rhs)`  
*Assignment operator for `Timeline` class.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA_Timeline ()`

### 7.70.1 Detailed Description

Definition at line 39 of file Timeline.hh.

### 7.70.2 Constructor & Destructor Documentation

#### 7.70.2.1 Timeline() [1/2]

```
Timeline::Timeline (
    double t0 = 0.0 )
```

Initialization constructor for the [TrickHLA CTETimelineBase](#) class.

##### Parameters

<i>t0</i>	Epoch for the timeline.
-----------	-------------------------

##### Trick Job Class: initialization

Definition at line 40 of file Timeline.cpp.

#### 7.70.2.2 ~Timeline()

```
Timeline::~Timeline () [pure virtual]
```

Pure virtual destructor for the [TrickHLA CTETimelineBase](#) class.

##### Trick Job Class: shutdown

Definition at line 50 of file Timeline.cpp.

#### 7.70.2.3 Timeline() [2/2]

```
TrickHLA::Timeline::Timeline (
    const Timeline & rhs ) [private]
```

Copy constructor for [Timeline](#) class.

This constructor is private to prevent inadvertent copies.

### 7.70.3 Member Function Documentation

#### 7.70.3.1 get\_elapsed\_time()

```
virtual double TrickHLA::Timeline::get_elapsed_time () [inline], [virtual]
```

Get the elapsed time for this timeline in seconds from epoch.

##### Returns

Returns the elapsed time from epoch in seconds.

Definition at line 74 of file Timeline.hh.

References epoch, and [get\\_time\(\)](#).

### 7.70.3.2 `get_epoch()`

```
virtual double TrickHLA::Timeline::get_epoch ( ) [inline], [virtual]
Get the epoch for this timeline in seconds.
```

#### Returns

Returns the epoch for this timeline in seconds.

Definition at line 82 of file Timeline.hh.

References epoch.

Referenced by SpaceFOM::ExecutionControl::epoch\_and\_root\_frame\_discovery\_process(), SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process(), SpaceFOM::ExecutionControl::post\_multi\_phase\_init\_processes(), DSE::ExecutionControl::process\_execution\_control\_updates(), SpaceFOM::ExecutionControl::process\_execution\_control\_updates(), DIS::ExecutionControl::process\_execution\_control\_updates(), IMSim::ExecutionControl::process\_execution\_control\_updates(), DSES::ExecutionControl::process\_mode\_transition\_request(), SpaceFOM::ExecutionControl::process\_mode\_transition\_request(), DIS::ExecutionControl::process\_mode\_transition\_request(), and IMSim::ExecutionControl::process\_mode\_transition\_request().

### 7.70.3.3 `get_time()`

```
virtual double TrickHLA::Timeline::get_time ( ) [pure virtual]
Get the current time for this timeline in seconds.
```

#### Returns

Returns the current timeline time in seconds.

Implemented in [TrickHLA::ScenarioTimeline](#), [TrickHLA::CTETimelineBase](#), [TrickHLA::SimTimeline](#), and [TrickHLA::TimeOfDayTimeline](#). Referenced by `get_elapsed_time()`.

### 7.70.3.4 `operator=()`

```
Timeline& TrickHLA::Timeline::operator= (
    const Timeline & rhs ) [private]
```

Assignment operator for [Timeline](#) class.

This assignment operator is private to prevent inadvertent copies.

### 7.70.3.5 `set_epoch()`

```
virtual void TrickHLA::Timeline::set_epoch (
    double time ) [inline], [virtual]
```

Set the epoch for this timeline in seconds.

#### Parameters

<i>time</i>	New time value for epoch in seconds.
-------------	--------------------------------------

Definition at line 78 of file Timeline.hh.

References epoch.

Referenced by SpaceFOM::ExecutionControl::epoch\_and\_root\_frame\_discovery\_process(), and SpaceFOM::ExecutionControl::late\_joiner\_hla\_init\_process().

## 7.70.4 Friends And Related Function Documentation

### 7.70.4.1 `init_attrTrickHLA__Timeline`

```
void init_attrTrickHLA__Timeline ( ) [friend]
```

### 7.70.4.2 `InputProcessor`

```
friend class InputProcessor [friend]
```

Definition at line 46 of file Timeline.hh.

## 7.70.5 Field Documentation

### 7.70.5.1 `epoch`

```
double TrickHLA::Timeline::epoch [protected]
```

**Units:** s

Epoch for the simulation.

This is the value of the timeline when the execution starts up. This value is often zero but is note required to be zero.

Definition at line 85 of file Timeline.hh.

Referenced by TrickHLA::ScenarioTimeline::compute\_HLT(), TrickHLA::ScenarioTimeline::compute\_simulation\_time(), get\_elapsed\_time(), get\_epoch(), TrickHLA::ScenarioTimeline::get\_time(), set\_epoch(), TrickHLA::ScenarioTimeline::time\_from\_HLT(), and TrickHLA::ScenarioTimeline::time\_from\_simulation\_time().

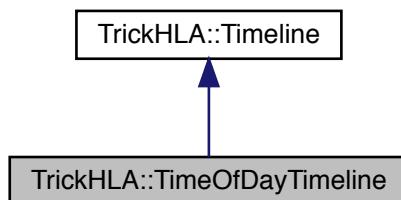
The documentation for this class was generated from the following files:

- [Timeline.hh](#)
- [Timeline.cpp](#)

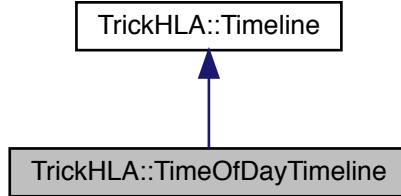
## 7.71 TrickHLA::TimeOfDayTimeline Class Reference

```
#include <TimeOfDayTimeline.hh>
```

Inheritance diagram for TrickHLA::TimeOfDayTimeline:



Collaboration diagram for TrickHLA::TimeOfDayTimeline:



## Public Member Functions

- `TimeOfDayTimeline ()`  
*Default constructor for the [TrickHLA TimeOfDayTimeline class](#).*
- `virtual ~TimeOfDayTimeline ()`  
*Destructor for the [TrickHLA TimeOfDayTimeline class](#).*
- `virtual double get_time ()`  
*Get the current time for this timeline in seconds.*

## Private Member Functions

- `TimeOfDayTimeline (const TimeOfDayTimeline &rhs)`  
*Copy constructor for [TimeOfDayTimeline class](#).*
- `TimeOfDayTimeline & operator= (const TimeOfDayTimeline &rhs)`  
*Assignment operator for [TimeOfDayTimeline class](#).*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA__TimeOfDayTimeline ()`

## Additional Inherited Members

### 7.71.1 Detailed Description

Definition at line 42 of file `TimeOfDayTimeline.hh`.

### 7.71.2 Constructor & Destructor Documentation

#### 7.71.2.1 `TimeOfDayTimeline()` [1/2]

`TimeOfDayTimeline::TimeOfDayTimeline ()`  
Default constructor for the [TrickHLA TimeOfDayTimeline class](#).  
**Trick Job Class:** *initialization*

Definition at line 44 of file TimeOfDayTimeline.cpp.

### 7.71.2.2 ~TimeOfDayTimeline()

TimeOfDayTimeline::~TimeOfDayTimeline ( ) [virtual]

Destructor for the [TrickHLA TimeOfDayTimeline](#) class.

**Trick Job Class:** *shutdown*

Definition at line 51 of file TimeOfDayTimeline.cpp.

### 7.71.2.3 TimeOfDayTimeline() [2/2]

TrickHLA::TimeOfDayTimeline::TimeOfDayTimeline (   
     const TimeOfDayTimeline & rhs ) [private]

Copy constructor for [TimeOfDayTimeline](#) class.

This constructor is private to prevent inadvertent copies.

## 7.71.3 Member Function Documentation

### 7.71.3.1 get\_time()

double TimeOfDayTimeline::get\_time ( ) [virtual]

Get the current time for this timeline in seconds.

**Returns**

Current Time-Of-Day time in seconds to represent realtime.

Implements [TrickHLA::Timeline](#).

Definition at line 55 of file TimeOfDayTimeline.cpp.

### 7.71.3.2 operator=()

TimeOfDayTimeline& TrickHLA::TimeOfDayTimeline::operator= (   
     const TimeOfDayTimeline & rhs ) [private]

Assignment operator for [TimeOfDayTimeline](#) class.

This assignment operator is private to prevent inadvertent copies.

## 7.71.4 Friends And Related Function Documentation

### 7.71.4.1 init\_attrTrickHLA\_\_TimeOfDayTimeline

void init\_attrTrickHLA\_\_TimeOfDayTimeline ( ) [friend]

### 7.71.4.2 InputProcessor

friend class InputProcessor [friend]

Definition at line 49 of file TimeOfDayTimeline.hh.

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

- [TimeOfDayTimeline.hh](#)
- [TimeOfDayTimeline.cpp](#)

## 7.72 TrickHLA::Utilities Class Reference

```
#include <Utilities.hh>
```

### Public Member Functions

- [Utilities \(\)](#)  
*Default constructor for the `TrickHLA Utilities` class.*
- [virtual ~Utilities \(\)](#)  
*Destructor for the `TrickHLA Utilities` class.*

### Static Public Member Functions

- [static const char get\\_endianness \(\)](#)  
*Get the endianness.*
- [static bool is\\_transmission\\_byteswap \(const EncodingEnum rti\\_encoding\)](#)  
*Determine if the RTI data needs a byteswap before transmission.*
- [static short byteswap\\_short \(short input\)](#)  
*Byteswap an short integer type.*
- [static unsigned short byteswap\\_unsigned\\_short \(unsigned short input\)](#)  
*Byteswap an unsigned short integer type.*
- [static int byteswap\\_int \(int input\)](#)  
*Byteswap an int integer type.*
- [static unsigned int byteswap\\_unsigned\\_int \(unsigned int input\)](#)  
*Byteswap an unsigned int integer type.*
- [static long byteswap\\_long \(long input\)](#)  
*Byteswap a long integer type.*
- [static unsigned long byteswap\\_unsigned\\_long \(unsigned long input\)](#)  
*Byteswap an unsigned long integer type.*
- [static long long byteswap\\_long\\_long \(long long input\)](#)  
*Byteswap a long long integer type.*
- [static unsigned long long byteswap\\_unsigned\\_long\\_long \(unsigned long long input\)](#)  
*Byteswap an unsigned long long integer type.*
- [static float byteswap\\_float \(float input\)](#)  
*Byteswap float floating-point type.*
- [static double byteswap\\_double \(double input\)](#)  
*Byteswap double floating-point type.*
- [static std::string get\\_version \(\)](#)  
*Return the current `TrickHLA` version string from the auto generated `Version.hh` header file.*
- [static std::string get\\_release\\_date \(\)](#)  
*Returns the `TrickHLA` release date from the auto generated `Version.hh` header file.*

## Private Member Functions

- `Utilities (const Utilities &rhs)`  
*Copy constructor for `Utilities` class.*
- `Utilities & operator= (const Utilities &rhs)`  
*Assignment operator for `Utilities` class.*

## Friends

- class `InputProcessor`
- void `init_attrTrickHLA__Utilities ()`

### 7.72.1 Detailed Description

Definition at line 122 of file Utilities.hh.

### 7.72.2 Constructor & Destructor Documentation

#### 7.72.2.1 Utilities() [1/2]

`TrickHLA::Utilities::Utilities ( ) [inline]`

Default constructor for the `TrickHLA Utilities` class.

Definition at line 139 of file Utilities.hh.

#### 7.72.2.2 ~Utilities()

`virtual TrickHLA::Utilities::~Utilities ( ) [inline], [virtual]`

Destructor for the `TrickHLA Utilities` class.

Definition at line 142 of file Utilities.hh.

#### 7.72.2.3 Utilities() [2/2]

`TrickHLA::Utilities::Utilities (`  
    `const Utilities & rhs ) [private]`

Copy constructor for `Utilities` class.

This constructor is private to prevent inadvertent copies.

### 7.72.3 Member Function Documentation

#### 7.72.3.1 byteswap\_double()

`double Utilities::byteswap_double (`  
    `double input ) [static]`

Byteswap double floating-point type.

##### Returns

Byteswap value.

## Parameters

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 336 of file Utilities.cpp.

Referenced by TrickHLA::OpaqueBuffer::byteswap\_buffer\_copy(), TrickHLA::Parameter::byteswap\_buffer\_copy(), TrickHLA::Attribute::byteswap\_buffer\_copy(), TrickHLA::Parameter::print\_buffer(), and TrickHLA::Attribute::print\_buffer().

### 7.72.3.2 byteswap\_float()

```
float Utilities::byteswap_float (
    float input ) [static]
```

Byteswap float floating-point type.

## Returns

Byteswap value.

## Parameters

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 301 of file Utilities.cpp.

Referenced by TrickHLA::Parameter::byteswap\_buffer\_copy(), and TrickHLA::Attribute::byteswap\_buffer\_copy().

### 7.72.3.3 byteswap\_int()

```
int Utilities::byteswap_int (
    int input ) [static]
```

Byteswap an int integer type.

## Returns

Byteswap value.

## Parameters

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 115 of file Utilities.cpp.

Referenced by TrickHLA::Parameter::byteswap\_buffer\_copy(), TrickHLA::Attribute::byteswap\_buffer\_copy(), TrickHLA::Federate::enable\_MOM\_auto\_provide\_setting(), TrickHLA::Federate::rebuild\_federate\_handles(), TrickHLA::Federate::set\_MOM\_HLAfederate\_instance\_attributes(), and TrickHLA::Federate::set\_MOM\_HLAfederation\_instance\_attributes().

### 7.72.3.4 byteswap\_long()

```
long Utilities::byteswap_long (
    long input ) [static]
```

Byteswap a long integer type.

**Returns**

Byteswap value.

**Parameters**

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 167 of file Utilities.cpp.

Referenced by TrickHLA::Parameter::byteswap\_buffer\_copy(), and TrickHLA::Attribute::byteswap\_buffer\_copy().

### 7.72.3.5 byteswap\_long\_long()

```
long long Utilities::byteswap_long_long (
    long long input ) [static]
```

Byteswap a long long integer type.

The long long type is defined in the C99 standard and is at least 64 bits.

**Returns**

Byteswap value.

**Parameters**

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 239 of file Utilities.cpp.

Referenced by TrickHLA::Parameter::byteswap\_buffer\_copy(), and TrickHLA::Attribute::byteswap\_buffer\_copy().

### 7.72.3.6 byteswap\_short()

```
short Utilities::byteswap_short (
    short input ) [static]
```

Byteswap an short integer type.

**Returns**

Byteswap value.

**Parameters**

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 55 of file Utilities.cpp.

Referenced by TrickHLA::Parameter::byteswap\_buffer\_copy(), and TrickHLA::Attribute::byteswap\_buffer\_copy().

### 7.72.3.7 byteswap\_unsigned\_int()

```
unsigned int Utilities::byteswap_unsigned_int (
    unsigned int input ) [static]
```

Byteswap an unsigned int integer type.

**Returns**

Byteswap value.

**Parameters**

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 141 of file Utilities.cpp.

Referenced by TrickHLA::OpaqueBuffer::byteswap\_buffer\_copy(), TrickHLA::Parameter::byteswap\_buffer\_copy(), and TrickHLA::Attribute::byteswap\_buffer\_copy().

**7.72.3.8 byteswap\_unsigned\_long()**

```
unsigned long Utilities::byteswap_unsigned_long (
    unsigned long input ) [static]
```

Byteswap an unsigned long integer type.

**Returns**

Byteswap value.

**Parameters**

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 203 of file Utilities.cpp.

Referenced by TrickHLA::Parameter::byteswap\_buffer\_copy(), and TrickHLA::Attribute::byteswap\_buffer\_copy().

**7.72.3.9 byteswap\_unsigned\_long\_long()**

```
unsigned long long Utilities::byteswap_unsigned_long_long (
    unsigned long long input ) [static]
```

Byteswap an unsigned long long integer type.

The unsigned long long type is defined in the C99 standard and is at least 64 bits.

**Returns**

Byteswap value.

**Parameters**

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 270 of file Utilities.cpp.

Referenced by TrickHLA::OpaqueBuffer::byteswap\_buffer\_copy(), TrickHLA::Parameter::byteswap\_buffer\_copy(), and TrickHLA::Attribute::byteswap\_buffer\_copy().

**7.72.3.10 byteswap\_unsigned\_short()**

```
unsigned short Utilities::byteswap_unsigned_short (
```

```
unsigned short byteswap( unsigned short input ) [static]
Byteswap an unsigned short integer type.
```

**Returns**

Byteswap value.

**Parameters**

<i>input</i>	The input value to byteswap.
--------------	------------------------------

Definition at line 85 of file Utilities.cpp.

Referenced by TrickHLA::OpaqueBuffer::byteswap\_buffer\_copy(), TrickHLA::Parameter::byteswap\_buffer\_copy(), and TrickHLA::Attribute::byteswap\_buffer\_copy().

**7.72.3.11 get\_endianness()**

```
static const char TrickHLA::Utilities::get_endianness( ) [inline], [static]
Get the endianness.
```

**Returns**

Either TRICK\_BIG\_ENDIAN (char)0x00 or TRICK\_LITTLE\_ENDIAN (char)0x01

Definition at line 147 of file Utilities.hh.

Referenced by TrickHLA::Attribute::Attribute(), TrickHLA::Parameter::decode\_opaque\_data\_from\_buffer(), TrickHLA::Attribute::decode\_opaque\_data\_from\_buffer(), TrickHLA::Parameter::decode\_string\_from\_buffer(), TrickHLA::Attribute::decode\_string\_from\_buffer(), TrickHLA::Parameter::encode\_opaque\_data\_to\_buffer(), TrickHLA::Attribute::encode\_opaque\_data\_to\_buffer(), TrickHLA::Parameter::encode\_string\_to\_buffer(), TrickHLA::Attribute::encode\_string\_to\_buffer(), and TrickHLA::Parameter::Parameter().

**7.72.3.12 get\_release\_date()**

```
string Utilities::get_release_date( ) [static]
Returns the TrickHLA release date from the auto generated Version.hh header file.
```

**Returns**

Byteswap value.

Definition at line 370 of file Utilities.cpp.

References TRICKHLA\_RELEASE\_DATE.

Referenced by TrickHLA::Federate::print\_version().

**7.72.3.13 get\_version()**

```
string Utilities::get_version( ) [static]
Return the current TrickHLA version string from the auto generated Version.hh header file.
```

**Returns**

Byteswap value.

Definition at line 365 of file Utilities.cpp.

References TRICKHLA\_VERSION.

Referenced by TrickHLA::Federate::print\_version().

### 7.72.3.14 is\_transmission\_byteswap()

```
bool Utilities::is_transmission_byteswap (
    const EncodingEnum rti_encoding ) [static]
```

Determine if the RTI data needs a byteswap before transmission.

#### Returns

True if byteswap is needed.

#### Parameters

<i>rti_encoding</i>	TrickHLA RTI encoding of the data.
---------------------	------------------------------------

Definition at line 46 of file Utilities.cpp.

References TrickHLA::ENCODING\_BIG\_ENDIAN, and TrickHLA::ENCODING\_LITTLE\_ENDIAN.

Referenced by TrickHLA::OpaqueBuffer::byteswap\_buffer\_copy(), TrickHLA::Parameter::complete\_initialization(), TrickHLA::Federate::enable\_MOM\_auto\_provide\_setting(), TrickHLA::Attribute::initialize(), TrickHLA::Federate::rebuild\_federate\_handles(), TrickHLA::Parameter::set\_encoding(), TrickHLA::Attribute::set\_encoding(), TrickHLA::Federate::set\_MOM\_HLAfederate\_instance\_attributes(), and TrickHLA::Federate::set\_MOM\_HLAfederation\_instance\_attributes().

### 7.72.3.15 operator=()

```
Utilities& TrickHLA::Utilities::operator= (
    const Utilities & rhs ) [private]
```

Assignment operator for **Utilities** class.

This assignment operator is private to prevent inadvertent copies.

## 7.72.4 Friends And Related Function Documentation

### 7.72.4.1 init\_attrTrickHLA\_\_Utilities

```
void init_attrTrickHLA__Utilities ( ) [friend]
```

### 7.72.4.2 InputProcessor

```
friend class InputProcessor [friend]
```

Definition at line 129 of file Utilities.hh.

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

- [Utilities.hh](#)
- [Utilities.cpp](#)

# Chapter 8

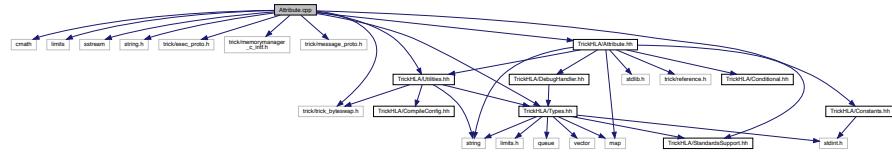
# File Documentation

## 8.1 Attribute.cpp File Reference

This class represents the HLA attributes of an object that is managed by Trick.

```
#include <cmath>
#include <limits>
#include <iostream>
#include <string.h>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "trick/trick_byteswap.h"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Constants.hh"
#include "TrickHLA/Types.hh"
#include "TrickHLA/Utilities.hh"
Include dependency graph for Attribute.cpp:
```

Include dependency graph for Attribute.cpp:



### 8.1.1 Detailed Description

This class represents the HLA attributes of an object that is managed by Trick.

### Assumptions and Limitations:

- Only primitive types and static arrays of primitive type are supported for now.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- [Conditional.cpp](#)
- [Utilities.cpp](#)
- [Attribute.cpp](#)

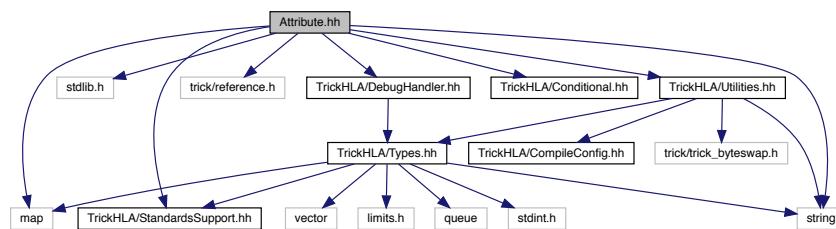
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	May 2006	–	Initial version.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

## Revision History

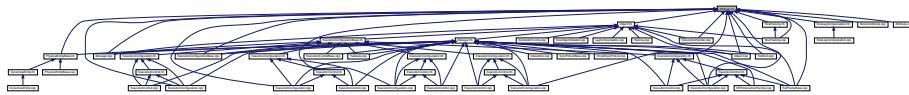
## 8.2 Attribute.hh File Reference

This class represents the HLA attributes of an object that is managed by Trick.

```
#include <map>
#include <stdlib.h>
#include <string>
#include "trick/reference.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Conditional.hh"
#include "TrickHLA/DebugHandler.hh"
#include "TrickHLA/Utilities.hh"
Include dependency graph for Attribute.hh:
```



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



## Data Structures

- class [TrickHLA::Attribute](#)

## Namespaces

- [TrickHLA](#)

## Typedefs

- `typedef std::map< RTI1516_NAMESPACE::AttributeHandle, Attribute * > TrickHLA::AttributeMap`

*Data I/O: \*\**

*Map of attributes.*

### 8.2.1 Detailed Description

This class represents the HLA attributes of an object that is managed by Trick.

This class represents the simulation timeline.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../../source/TrickHLA/Attribute.cpp`
- `../../source/TrickHLA/Utilities.cpp`
- `../../source/TrickHLA/Conditional.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

## Revision History

### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Python Module: *trick.TrickHLA*

#### Link Dependencies

- `../source/TrickHLA/Timeline.cpp`
- `../source/TrickHLA/SimTimeline.cpp`

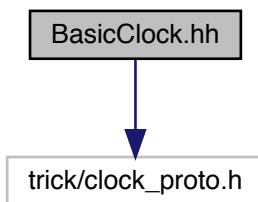
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	NExSyS	April 2016	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

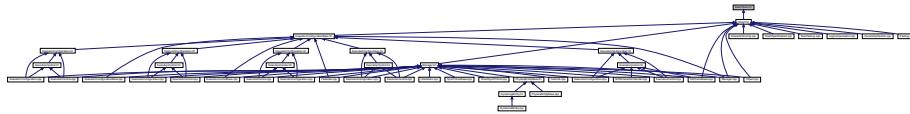
### 8.3 BasicClock.hh File Reference

This class provides a function to get the current GMT time by using either the CTE external clock if enabled or the system time-of-day.

```
#include "trick/clock_proto.h"
Include dependency graph for BasicClock.hh:
```



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



## Data Structures

- class [TrickHLA::BasicClock](#)

## Namespaces

- [TrickHLA](#)

### 8.3.1 Detailed Description

This class provides a function to get the current GMT time by using either the CTE external clock if enabled or the system time-of-day.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

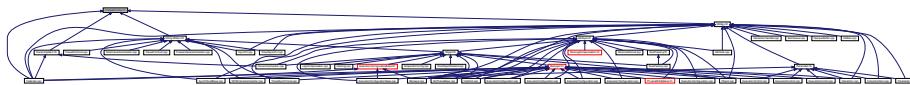
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L-3 Communications	<a href="#">IMSim</a>	March 2019	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.4 CompileConfig.hh File Reference

[TrickHLA](#) compile time configuration settings.

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



## Macros

- `#define NO_THLA_THREAD_WAIT_FOR_DATA`
- `#define THLA_THREAD_TIMED_WAIT_FOR_DATA`
- `#define THLA_10SEC_TIMEOUT_WHILE_WAITING_FOR_DATA`
- `#define THLA_QUEUE_REFLECTED_ATTRIBUTES`
- `#define THLA_INTRAFRAME_BLOCKING_READ_CONFIG`
- `#define MIN_TRICK_VER 17`
- `#define MIN_TRICK_MINOR 5`
- `#define MIN_TRICK_PATCH 0`
- `#define THLA_ENDL std::endl`
- `#define THLA_NEWLINE '\n'`
- `#define NO_TRICKHLA_ENABLE_FPU_CONTROL_WORD_VALIDATION`
- `#define NO_TRICKHLA_ENABLE_FOM_DUMP`

### 8.4.1 Detailed Description

[TrickHLA](#) compile time configuration settings.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	<a href="#">TrickHLA</a>	February 2009	–	Consolidated config settings.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

#### Revision History

### 8.4.2 Macro Definition Documentation

#### 8.4.2.1 MIN\_TRICK\_MINOR

`#define MIN_TRICK_MINOR 5`  
 Definition at line 104 of file `CompileConfig.hh`.

#### 8.4.2.2 MIN\_TRICK\_PATCH

```
#define MIN_TRICK_PATCH 0
Definition at line 105 of file CompileConfig.hh.
```

#### 8.4.2.3 MIN\_TRICK\_VER

```
#define MIN_TRICK_VER 17
Definition at line 103 of file CompileConfig.hh.
```

#### 8.4.2.4 NO\_THLA\_THREAD\_WAIT\_FOR\_DATA

```
#define NO_THLA_THREAD_WAIT_FOR_DATA
Definition at line 43 of file CompileConfig.hh.
```

#### 8.4.2.5 NO\_TRICKHLA\_ENABLE\_FOM\_DUMP

```
#define NO_TRICKHLA_ENABLE_FOM_DUMP
Definition at line 151 of file CompileConfig.hh.
```

#### 8.4.2.6 NO\_TRICKHLA\_ENABLE\_FPU\_CONTROL\_WORD\_VALIDATION

```
#define NO_TRICKHLA_ENABLE_FPU_CONTROL_WORD_VALIDATION
Definition at line 147 of file CompileConfig.hh.
```

#### 8.4.2.7 THLA\_10SEC\_TIMEOUT WHILE\_WAITING\_FOR\_DATA

```
#define THLA_10SEC_TIMEOUT WHILE_WAITING_FOR_DATA
Definition at line 53 of file CompileConfig.hh.
```

#### 8.4.2.8 THLA\_ENDL

```
#define THLA_ENDL std::endl
Definition at line 131 of file CompileConfig.hh.
```

#### 8.4.2.9 THLA\_INTRAFRAME\_BLOCKING\_READ\_CONFIG

```
#define THLA_INTRAFRAME_BLOCKING_READ_CONFIG
Definition at line 63 of file CompileConfig.hh.
```

#### 8.4.2.10 THLA\_NEWLINE

```
#define THLA_NEWLINE '\n'
Definition at line 132 of file CompileConfig.hh.
```

#### 8.4.2.11 THLA\_QUEUE\_REFLECTED\_ATTRIBUTES

```
#define THLA_QUEUE_REFLECTED_ATTRIBUTES
Definition at line 58 of file CompileConfig.hh.
```

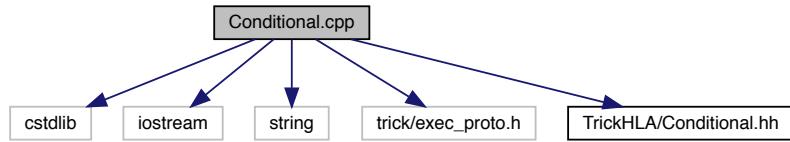
#### 8.4.2.12 THLA\_THREAD\_TIMED\_WAIT\_FOR\_DATA

```
#define THLA_THREAD_TIMED_WAIT_FOR_DATA
Definition at line 48 of file CompileConfig.hh.
```

## 8.5 Conditional.cpp File Reference

This class provides a user API for the handling of a CONDITIONAL attribute.

```
#include <cstdlib>
#include <iostream>
#include <string>
#include "trick/exec_proto.h"
#include "TrickHLA/Conditional.hh"
Include dependency graph for Conditional.cpp:
```



### 8.5.1 Detailed Description

This class provides a user API for the handling of a CONDITIONAL attribute.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Conditional.cpp](#)
- [Attribute.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3	DSES	Oct 2009	–	Initial version.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

## 8.6 Conditional.hh File Reference

This class provides a user API for the handling of a CONDITIONAL attribute.

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



## Data Structures

- class [TrickHLA::Conditional](#)

## Namespaces

- [TrickHLA](#)

### 8.6.1 Detailed Description

This class provides a user API for the handling of a CONDITIONAL attribute.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../../source/TrickHLA/Conditional.cpp`
- `../../source/TrickHLA/Attribute.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3 Titan Group	IMSim	Oct 2009	–	Initial version.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

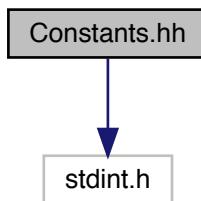
## Revision History

## 8.7 Constants.hh File Reference

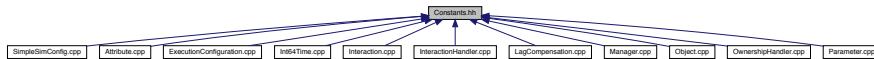
Define the global time constants used in [TrickHLA](#).

```
#include <stdint.h>
```

Include dependency graph for Constants.hh:



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



## Namespaces

- [TrickHLA](#)

## Variables

- const int64\_t [TrickHLA::MICROS\\_MULTIPLIER](#) = 1000000  
*Units:* –
- const int64\_t [TrickHLA::MAX\\_VALUE\\_IN\\_MICROS](#) = std::numeric\_limits< int64\_t >::max()  
*Units:* us
- const double [TrickHLA::MAX\\_LOGICAL\\_TIME\\_SECONDS](#) = (double)MAX\_VALUE\_IN\_MICROS / MICROS\_←  
 MULTIPLIER  
*Units:* s

### 8.7.1 Detailed Description

Define the global time constants used in [TrickHLA](#).

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module:** *trick.TrickHLA***Link Dependencies**

- ..../source/TrickHLA/Int64Interval.cpp

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

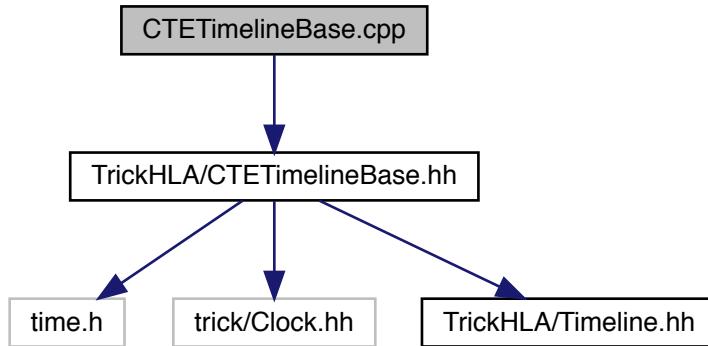
**Revision History**

## 8.8 CTETimelineBase.cpp File Reference

A base implementation of a SRFOM CTE timeline.

```
#include "TrickHLA/CTETimelineBase.hh"
```

Include dependency graph for CTETimelineBase.cpp:



### 8.8.1 Detailed Description

A base implementation of a SRFOM CTE timeline.

This is a baseline implementation based off of the system clock. It is intended to provide the definition of the CTE Timeline interface in the form of a base class. It also provides a working implementation based on the system clock (see **Assumptions and Limitations below**).

The expectation is that a hardware specific implementation will be developed and provided for the CTE hardware used in a specific federation execution.

#### Assumptions and Limitations:

- This implementation uses the system clock which is assumed to be synchronized using an external mechanism like NTP.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Timeline.cpp](#)
- [CTETimelineBase.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	June 2016	–	Initial version.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

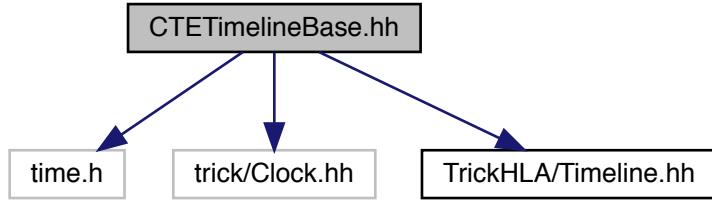
#### Revision History

## 8.9 CTETimelineBase.hh File Reference

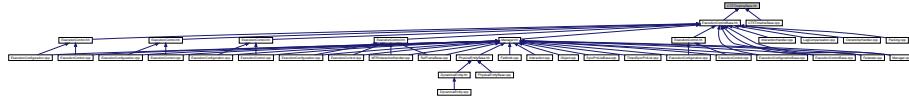
This class represents the CTE timeline.

```
#include <time.h>
#include "trick/Clock.hh"
#include "TrickHLA/Timeline.hh"
```

Include dependency graph for CTETimelineBase.hh:



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



## Data Structures

- class [TrickHLA::CTETimelineBase](#)

## Namespaces

- [TrickHLA](#)

### 8.9.1 Detailed Description

This class represents the CTE timeline.

**Assumptions and Limitations:**

- Instances of this class represent the timeline for the CTE associated with the problem.
- The time scale for this timeline is always Terrestrial Time (TT) which complies with the Space Reference FOM standard.
- Note that the epoch value for this CTE timeline represents the epoch or starting point of the CTE timeline. This will correspond to the starting time in the TT time standard represented in Truncated Julian Date format (TJD) expressed in seconds.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Python Module: `trick.TrickHLA`

### Link Dependencies

- `../source/TrickHLA/Timeline.cpp`
- `../source/TrickHLA/CTETimelineBase.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	January 2019	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

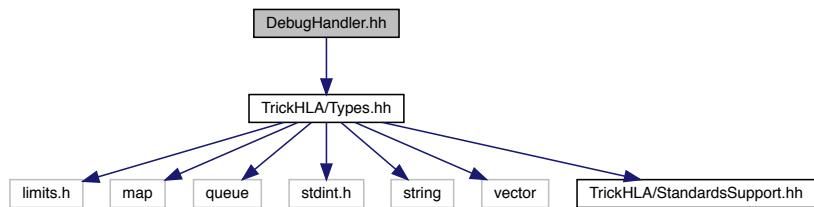
### Revision History

## 8.10 DebugHandler.hh File Reference

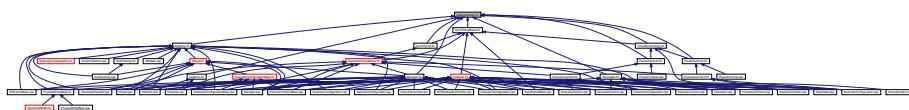
Multi-level debug reporter.

```
#include "TrickHLA/Types.hh"
```

Include dependency graph for DebugHandler.hh:



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



## Data Structures

- class `TrickHLA::DebugHandler`

## Namespaces

- [TrickHLA](#)

### 8.10.1 Detailed Description

Multi-level debug reporter.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: [trick.TrickHLA](#)

#### Link Dependencies

- [..../source/TrickHLA/Types.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3 Titan Group	<a href="#">IMSim</a>	Jan 2010	–	Initial version.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

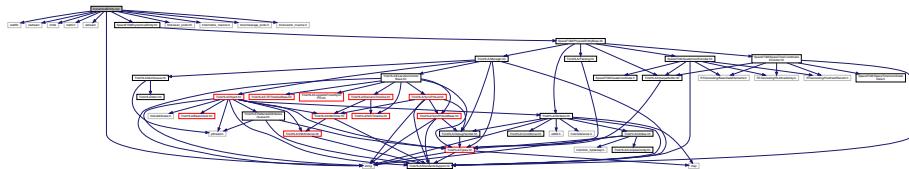
#### Revision History

## 8.11 DynamicalEntity.cpp File Reference

This class provides data packing for the [SpaceFOM](#) Reference Frames.

```
#include <cstdlib>
#include <iostream>
#include <limits>
#include <math.h>
#include <sstream>
#include <string>
#include "trick/exec_proto.hh"
#include "trick/matrix_macros.h"
#include "trick/message_proto.h"
#include "trick/vector_macros.h"
#include "SpaceFOM/DynamicalEntity.hh"
```

Include dependency graph for `DynamicalEntity.cpp`:



### 8.11.1 Detailed Description

This class provides data packing for the [SpaceFOM](#) Reference Frames.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [../TrickHLA/Packing.cpp](#)
- [DynamicalEntity.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	Sept 2006	–	Initial implementation.
Edwin Z. Crues	NASA ER7	SISO	Sept 2010	–	Smackdown implementation.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

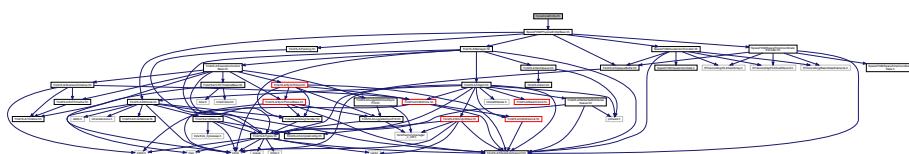
#### Revision History

## 8.12 DynamicalEntity.hh File Reference

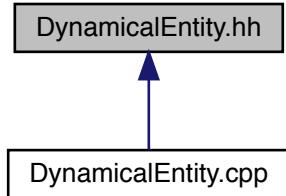
Definition of the [TrickHLA SpaceFOM](#) Dynamical entity type.

```
#include "SpaceFOM/PhysicalEntityBase.hh"
```

Include dependency graph for `DynamicalEntity.hh`:



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



## Data Structures

- class [SpaceFOM::DynamicalEntity](#)

## Namespaces

- [SpaceFOM](#)

### 8.12.1 Detailed Description

Definition of the [TrickHLA SpaceFOM](#) Dynamical entity type.

This is the base implementation for the Space Reference FOM ([SpaceFOM](#)) interface to the Reference Frame object. This needs to be available to the [SpaceFOM](#) initialization process for the root reference frame discovery step in the initialization process.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.SpaceFOM`

#### Link Dependencies

- [../../source/SpaceFOM/PhysicalEntityBase.cpp](#)
- [../../source/SpaceFOM/DynamicalEntity.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

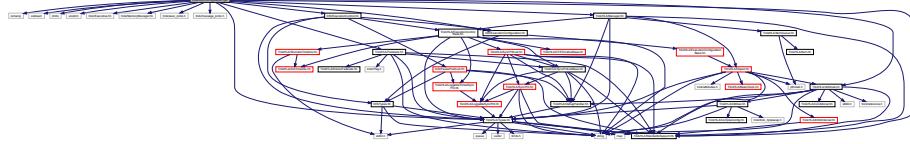
## Revision History

## 8.13 ExecutionConfiguration.cpp File Reference

Implementation of the [TrickHLA DIS](#) Execution Configuration Object (ExCO).

```
#include <iomanip>
#include <iostream>
#include <limits>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/MemoryManager.hh"
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Types.hh"
#include "DIS/ExecutionConfiguration.hh"
#include "DIS/ExecutionControl.hh"
```

Include dependency graph for DIS/ExecutionConfiguration.cpp:



### Variables

- ATTRIBUTES [attrDIS\\_\\_ExecutionConfiguration](#) []

#### 8.13.1 Detailed Description

Implementation of the [TrickHLA DIS](#) Execution Configuration Object (ExCO).

Assumptions and Limitations:

- One and only one ExecutionConfiguration object should exist in an federation execution.

### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Link Dependencies

- ../TrickHLA/Object.cpp
- ../TrickHLA/Packing.cpp
- ../TrickHLA/Federate.cpp
- ../TrickHLA/Manager.cpp
- ../TrickHLA/ExecutionConfigurationBase.cpp
- ExecutionConfiguration.cpp

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DIS	June 2007	–	Initial version.
Edwin Z. Crues	NASA ER7	TrickHLA	Jan 2019	–	DIS support and testing.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

### Revision History

## 8.13.2 Variable Documentation

### 8.13.2.1 attrDIS\_\_ExecutionConfiguration

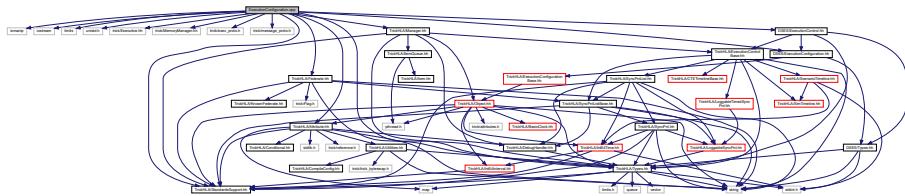
ATTRIBUTES attrDIS\_\_ExecutionConfiguration[]  
 Referenced by SpaceFOM::ExecutionConfiguration::setup\_ref\_attributes().

## 8.14 ExecutionConfiguration.cpp File Reference

Implementation of the [TrickHLA DSES](#) Execution Configuration Object (ExCO).

```
#include <iomanip>
#include <iostream>
#include <limits>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/MemoryManager.hh"
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Types.hh"
#include "DSES/ExecutionConfiguration.hh"
#include "DSES/ExecutionControl.hh"
```

Include dependency graph for DSES/ExecutionConfiguration.cpp:



## Variables

- ATTRIBUTES `attrDSES__ExecutionConfiguration []`

### 8.14.1 Detailed Description

Implementation of the [TrickHLA DSES](#) Execution Configuration Object (ExCO).

Assumptions and Limitations:

- One and only one ExecutionConfiguration object should exist in an federation execution.

## Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- ../TrickHLA/Object.cpp
- ../TrickHLA/Packing.cpp
- ../TrickHLA/Federate.cpp
- ../TrickHLA/Manager.cpp
- ../TrickHLA/ExecutionConfigurationBase.cpp
- ExecutionConfiguration.cpp

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	June 2007	–	Initial version.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	Jan 2019	–	<a href="#">DSES</a> support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

## Revision History

## 8.14.2 Variable Documentation

## 8.14.2.1 attrDSES\_\_ExecutionConfiguration

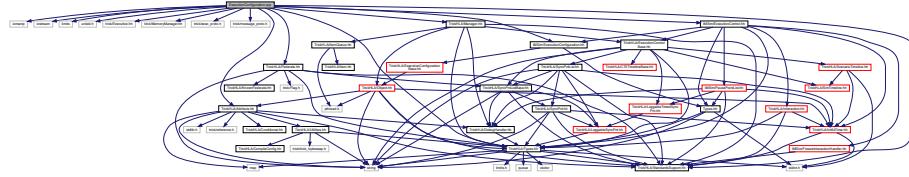
```
ATTRIBUTES attrDSES__ExecutionConfiguration[]
```

## 8.15 ExecutionConfiguration.cpp File Reference

Implementation of the [TrickHLA IMSim](#) Execution Configuration Object (ExCO).

```
#include <iomanip>
#include <iostream>
#include <limits>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/MemoryManager.hh"
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Types.hh"
#include "IMSim/ExecutionConfiguration.hh"
#include "IMSim/ExecutionControl.hh"
```

Include dependency graph for IMSim/ExecutionConfiguration.cpp:



## Variables

- ATTRIBUTES attrIMSim\_\_ExecutionConfiguration []

## 8.15.1 Detailed Description

Implementation of the [TrickHLA IMSim](#) Execution Configuration Object (ExCO).

## Assumptions and Limitations:

- One and only one ExecutionConfiguration object should exist in an federation execution.

## Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- ../TrickHLA/Object.cpp
- ../TrickHLA/Packing.cpp
- ../TrickHLA/Federate.cpp
- ../TrickHLA/Manager.cpp
- ../TrickHLA/ExecutionConfigurationBase.cpp
- ExecutionConfiguration.cpp

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	IMSim	June 2007	–	Initial version.
Edwin Z. Crues	NASA ER7	TrickHLA	Jan 2019	–	IMSim support and testing.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

### 8.15.2 Variable Documentation

#### 8.15.2.1 attrIMSim\_\_ExecutionConfiguration

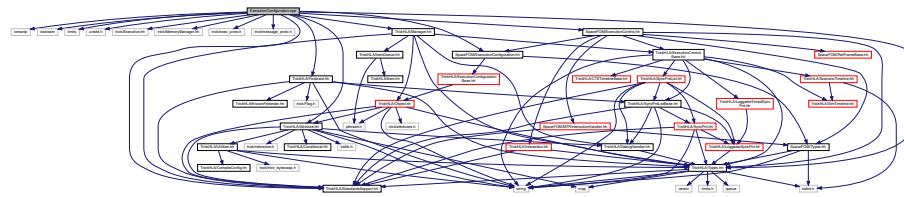
ATTRIBUTES attrIMSim\_\_ExecutionConfiguration[ ]

## 8.16 ExecutionConfiguration.cpp File Reference

Implementation of the [TrickHLA SpaceFOM](#) Execution Configuration Object (ExCO).

```
#include <iomanip>
#include <iostream>
#include <limits>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/MemoryManager.hh"
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
```

```
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Types.hh"
#include "SpaceFOM/ExecutionConfiguration.hh"
#include "SpaceFOM/ExecutionControl.hh"
Include dependency graph for SpaceFOM/ExecutionConfiguration.cpp:
```



## Variables

- ATTRIBUTES [attrSpaceFOM\\_\\_ExecutionConfiguration \[\]](#)

### 8.16.1 Detailed Description

Implementation of the [TrickHLA SpaceFOM](#) Execution Configuration Object (ExCO).

Assumptions and Limitations:

- One and only one ExecutionConfiguration object should exist in an federation execution.

## Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- [../TrickHLA/Object.cpp](#)
- [../TrickHLA/Packing.cpp](#)
- [../TrickHLA/Federate.cpp](#)
- [../TrickHLA/Manager.cpp](#)
- [../TrickHLA/ExecutionConfigurationBase.cpp](#)
- [ExecutionConfiguration.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	June 2007	–	Initial version.
Edwin Z. Crues	NASA ER7	TrickHLA	Jan 2019	–	SpaceFOM support and testing.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

### 8.16.2 Variable Documentation

#### 8.16.2.1 attrSpaceFOM\_\_ExecutionConfiguration

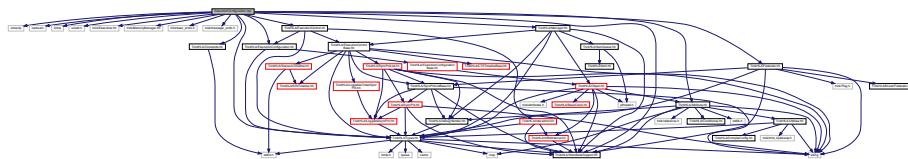
ATTRIBUTES attrSpaceFOM\_\_ExecutionConfiguration[]

## 8.17 ExecutionConfiguration.cpp File Reference

Implementation of the [TrickHLA](#) [TrickHLA](#) Execution Configuration Object (ExCO).

```
#include <iomanip>
#include <iostream>
#include <limits>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/MemoryManager.hh"
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Constants.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Types.hh"
#include "TrickHLA/ExecutionConfiguration.hh"
#include "TrickHLA/ExecutionControl.hh"
```

Include dependency graph for TrickHLA/ExecutionConfiguration.cpp:



## Variables

- ATTRIBUTES attrTrickHLA\_\_ExecutionConfiguration[]

### 8.17.1 Detailed Description

Implementation of the [TrickHLA](#) [TrickHLA](#) Execution Configuration Object (ExCO).

**Assumptions and Limitations:**

- One and only one ExecutionConfiguration object should exist in an federation execution.

**Copyright**

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Link Dependencies**

- [Object.cpp](#)
- [Packing.cpp](#)
- [Federate.cpp](#)
- [Manager.cpp](#)
- [ExecutionConfigurationBase.cpp](#)
- [ExecutionConfiguration.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	June 2007	–	Initial version.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	Jan 2019	–	<a href="#">TrickHLA</a> support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

**Revision History****8.17.2 Variable Documentation****8.17.2.1 attrTrickHLA\_\_ExecutionConfiguration**

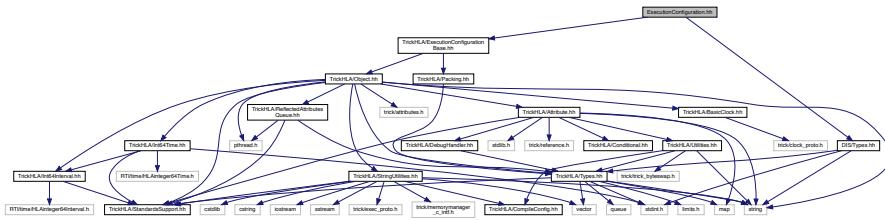
```
ATTRIBUTES attrTrickHLA__ExecutionConfiguration[ ]
```

**8.18 ExecutionConfiguration.hh File Reference**

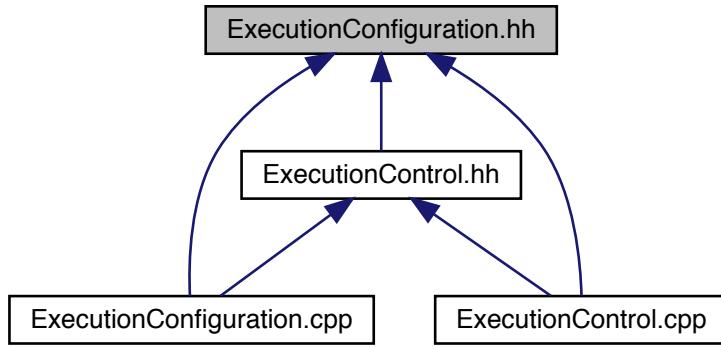
Definition of the [TrickHLA DIS](#) Execution Configuration Object (ExCO).

```
#include "TrickHLA/ExecutionConfigurationBase.hh"
#include "DIS/Types.hh"
```

Include dependency graph for DIS/ExecutionConfiguration.hh:



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



## Data Structures

- class [DIS::ExecutionConfiguration](#)

## Namespaces

- [DIS](#)

### 8.18.1 Detailed Description

Definition of the [TrickHLA DIS](#) Execution Configuration Object (ExCO).

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

Python Module: *trick.DIS*

## Link Dependencies

- *../source/TrickHLA/Object.cpp*
- *../source/TrickHLA/Packing.cpp*
- *../source/TrickHLA/ExecutionConfigurationBase.cpp*
- *../source/DIS/ExecutionConfiguration.cpp*

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	June 2016	–	SISO DIS Initialization.
Edwin Z. Crues	NASA ER7	TrickHLA	January 2019	–	DIS support and testing.
Edwin Z. Crues	NASA ER7	TrickHLA	June 2019	–	Version 3 rewrite.

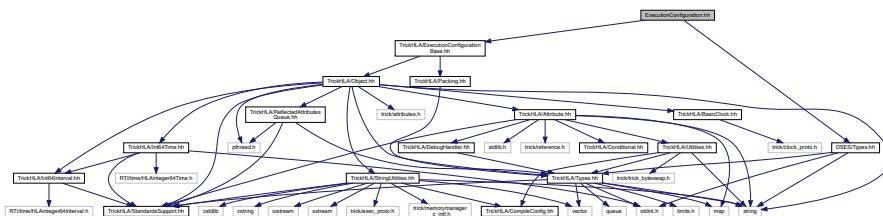
## Revision History

## 8.19 ExecutionConfiguration.hh File Reference

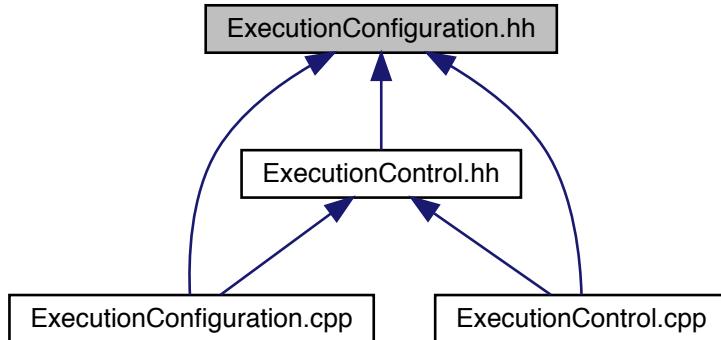
Definition of the [TrickHLA DSES](#) Execution Configuration Object (ExCO).

```
#include "TrickHLA/ExecutionConfigurationBase.hh"
#include "DSES/Types.hh"
```

Include dependency graph for DSES/ExecutionConfiguration.hh:



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



## Data Structures

- class [DSES::ExecutionConfiguration](#)

## Namespaces

- [DSES](#)

### 8.19.1 Detailed Description

Definition of the [TrickHLA DSES](#) Execution Configuration Object (ExCO).

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.DSES*

#### Link Dependencies

- [..../source/TrickHLA/Object.cpp](#)
- [..../source/TrickHLA/Packing.cpp](#)
- [..../source/TrickHLA/ExecutionConfigurationBase.cpp](#)
- [..../source/DSES/ExecutionConfiguration.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	June 2016	–	SISO <a href="#">DSES</a> Initialization.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	January 2019	–	<a href="#">DSES</a> support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

## Revision History

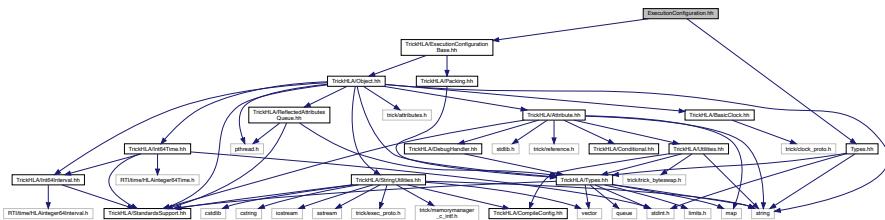
## 8.20 ExecutionConfiguration.hh File Reference

Definition of the [TrickHLA IMSim](#) Execution Configuration Object (ExCO).

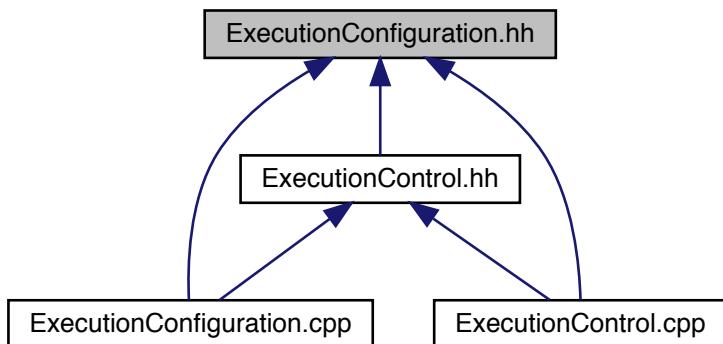
```
#include "TrickHLA/ExecutionConfigurationBase.hh"
```

```
#include "Types.hh"
```

Include dependency graph for IMSim/ExecutionConfiguration.hh:



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



## Data Structures

- class `IMSim::ExecutionConfiguration`

## Namespaces

- [IMSim](#)

### 8.20.1 Detailed Description

Definition of the [TrickHLA IMSim](#) Execution Configuration Object (ExCO).

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.IMSim*

#### Link Dependencies

- [..../source/TrickHLA/Object.cpp](#)
- [..../source/TrickHLA/Packing.cpp](#)
- [..../source/TrickHLA/ExecutionConfigurationBase.cpp](#)
- [..../source/IMSim/ExecutionConfiguration.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	June 2016	–	SISO <a href="#">IMSim</a> Initialization.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	January 2019	–	<a href="#">IMSim</a> support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

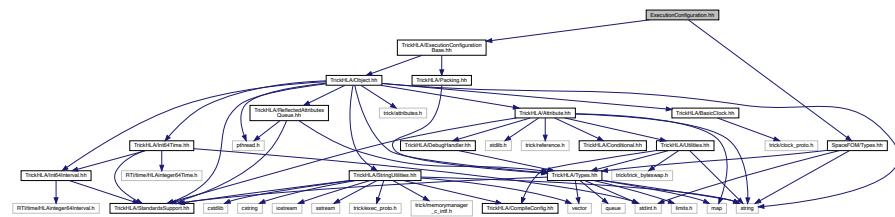
#### Revision History

## 8.21 ExecutionConfiguration.hh File Reference

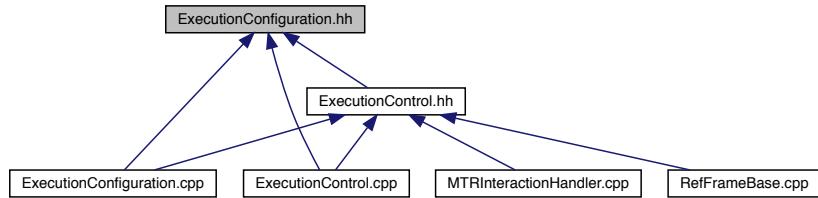
Definition of the [TrickHLA SpaceFOM](#) Execution Configuration Object (ExCO).

```
#include "TrickHLA/ExecutionConfigurationBase.hh"
#include "SpaceFOM/Types.hh"
```

Include dependency graph for SpaceFOM/ExecutionConfiguration.hh:



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



## Data Structures

- class [SpaceFOM::ExecutionConfiguration](#)

## Namespaces

- SpaceFOM

### 8.21.1 Detailed Description

Definition of the [TrickHLA SpaceFOM](#) Execution Configuration Object (ExCO).

## Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.SpaceFOM*

## Link Dependencies

- `../source/TrickHLA/Object.cpp`
  - `../source/TrickHLA/Packing.cpp`

- ..../source/TrickHLA/ExecutionConfigurationBase.cpp
- ..../source/SpaceFOM/ExecutionConfiguration.cpp

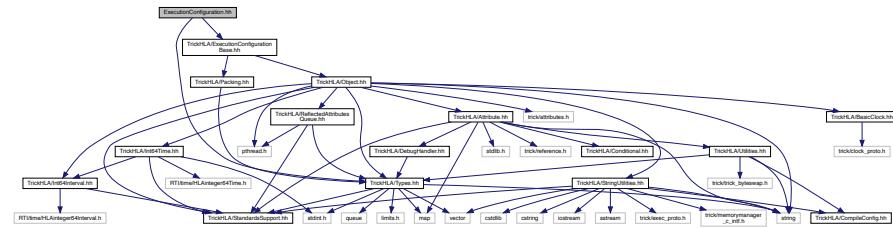
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	June 2016	–	SISO SpaceFOM Initialization.
Edwin Z. Crues	NASA ER7	TrickHLA	January 2019	–	SpaceFOM support and testing.
Edwin Z. Crues	NASA ER7	TrickHLA	June 2019	–	Version 3 rewrite.

#### Revision History

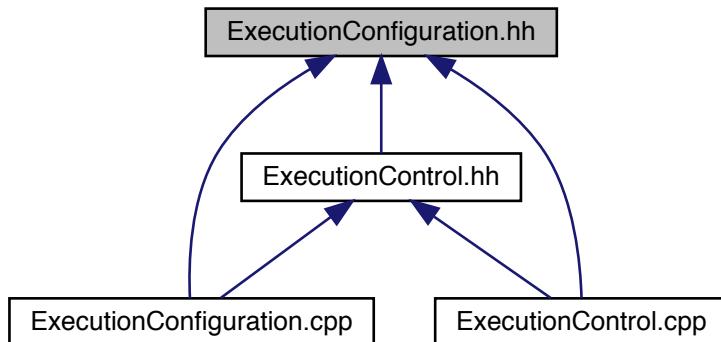
## 8.22 ExecutionConfiguration.hh File Reference

Definition of the [TrickHLA](#) simple Execution Configuration Object (ExCO).

```
#include "TrickHLA/ExecutionConfigurationBase.hh"
#include "TrickHLA/Types.hh"
Include dependency graph for TrickHLA/ExecutionConfiguration.hh:
```



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



## Data Structures

- class [TrickHLA::ExecutionConfiguration](#)

## Namespaces

- [TrickHLA](#)

### 8.22.1 Detailed Description

Definition of the [TrickHLA](#) simple Execution Configuration Object (ExCO).

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLA*

#### Link Dependencies

- [../source/TrickHLA/Object.cpp](#)
- [../source/TrickHLA/Packing.cpp](#)
- [../source/TrickHLA/ExecutionConfigurationBase.cpp](#)
- [../source/TrickHLA/ExecutionConfiguration.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	June 2016	—	SISO <a href="#">TrickHLA</a> Initialization.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	January 2019	—	<a href="#">TrickHLA</a> support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	—	Version 3 rewrite.

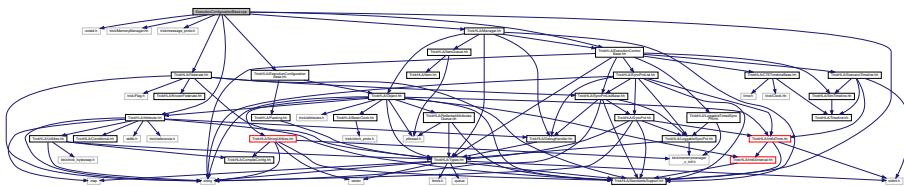
#### Revision History

## 8.23 ExecutionConfigurationBase.cpp File Reference

The abstract base class for the [TrickHLA](#) simulation execution configuration class.

```
#include <unistd.h>
#include "trick/MemoryManager.hh"
#include "trick/message_proto.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/ExecutionConfigurationBase.hh"
#include "TrickHLA/ExecutionControlBase.hh"
```

```
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
Include dependency graph for ExecutionConfigurationBase.cpp:
```



## Variables

- ATTRIBUTES [attrTrickHLA\\_\\_ExecutionConfigurationBase \[\]](#)

### 8.23.1 Detailed Description

The abstract base class for the [TrickHLA](#) simulation execution configuration class.

This class is used to provide the fundamentals for exchanging startup, initialization, and run time configuration information between participating federates in an HLA federation execution.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Assumptions and Limitations:

- One and only one ExecutionConfigurationBase object should exist in an federation execution.

#### Link Dependencies

- [Object.cpp](#)
- [Packing.cpp](#)
- [Federate.cpp](#)
- [Manager.cpp](#)
- [ExecutionConfigurationBase.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2020	–	Version 3 rewrite.

## Revision History

### 8.23.2 Variable Documentation

### 8.23.2.1 attrTrickHLA\_\_ExecutionConfigurationBase

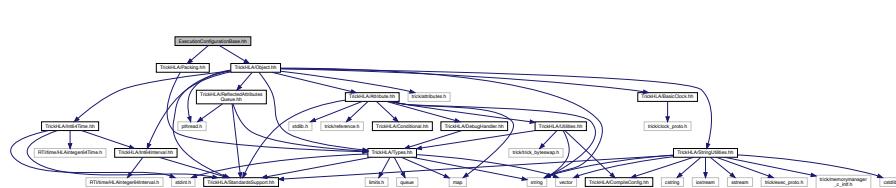
ATTRIBUTES attrTrickHLA\_\_ExecutionConfigurationBase[ ]

## 8.24 ExecutionConfigurationBase.hh File Reference

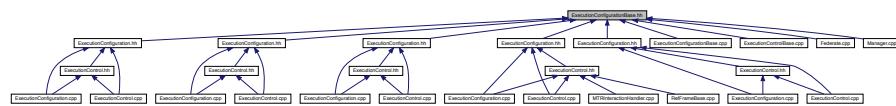
The abstract base class for the [TrickHLA](#) simulation execution configuration class.

```
#include "TrickHLA/Object.hh"
```

```
#include "TrickHLA/Packing.hh"
```



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



## Data Structures

- class `TrickHLA::ExecutionConfigurationBase`

## Namespaces

- TrickHLA

### 8.24.1 Detailed Description

The abstract base class for the [TrickHLA](#) simulation execution configuration class.

This class is used to provide the fundamentals for exchanging startup, initialization, and run time configuration information between participating federates in an HLA federation execution.

## Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Assumptions and Limitations:

- One and only one `ExecutionConfigurationBase` object should exist in an federation execution.

### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Object.cpp`
- `../source/TrickHLA/Packing.cpp`
- `../source/TrickHLA/ExecutionConfigurationBase.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2020	–	Version 3 rewrite.

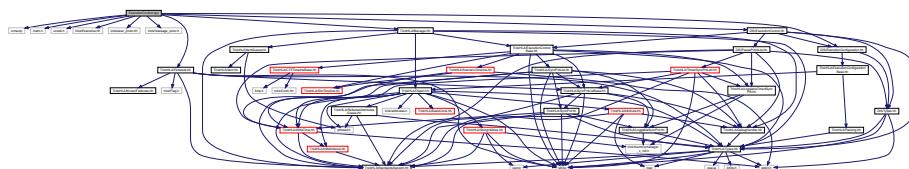
### Revision History

## 8.25 ExecutionControl.cpp File Reference

This class provides and abstract base class as the base implementation for managing mode transitions.

```
#include <iomanip>
#include <math.h>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/exec_proto.hh"
#include "trick/message_proto.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "DIS/ExecutionConfiguration.hh"
#include "DIS/ExecutionControl.hh"
```

Include dependency graph for `DIS/ExecutionControl.cpp`:



### Namespaces

- `DIS`

## Variables

- static const std::wstring **DIS::INITIALIZE\_SYNC\_POINT** = L"initialize"
- static const std::wstring **DIS::STARTUP\_SYNC\_POINT** = L"startup"
- static const std::wstring **DIS::STARTUP\_FREEZE\_SYNC\_POINT** = L"pause\_0.0"

### 8.25.1 Detailed Description

This class provides and abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- ../TrickHLA/SyncPntListBase.cpp
- ExecutionControl.cpp

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	Jan 2019	–	<b>DIS</b> support and testing.
Edwin Z. Crues	NASA ER7	TrickHLA	June 2019	–	Version 3 rewrite.

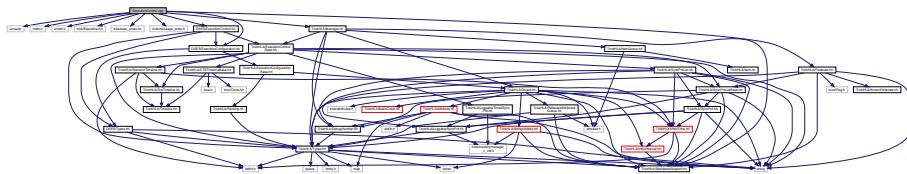
#### Revision History

## 8.26 ExecutionControl.cpp File Reference

This class provides and abstract base class as the base implementation for managing mode transitions.

```
#include <iomanip>
#include <math.h>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/exec_proto.hh"
#include "trick/message_proto.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "DSES/ExecutionConfiguration.hh"
#include "DSES/ExecutionControl.hh"
```

Include dependency graph for DSES/ExecutionControl.cpp:



## Namespaces

- [DSES](#)

## Variables

- static const std::wstring [DSES::SIM\\_CONFIG\\_SYNC\\_POINT](#) = L"sim\_config"
- static const std::wstring [DSES::INITIALIZE\\_SYNC\\_POINT](#) = L"initialize"
- static const std::wstring [DSES::STARTUP\\_SYNC\\_POINT](#) = L"startup"

### 8.26.1 Detailed Description

This class provides an abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [../TrickHLA/SyncPntListBase.cpp](#)
- [ExecutionControl.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	Jan 2019	–	DSES support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

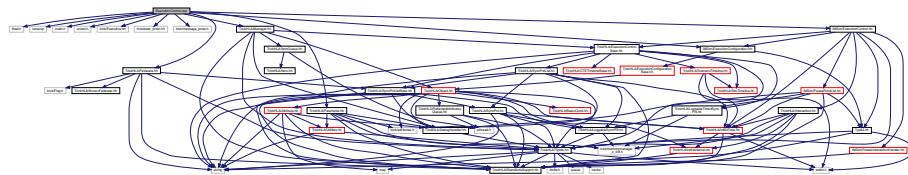
#### Revision History

## 8.27 ExecutionControl.cpp File Reference

This class provides and abstract base class as the base implementation for managing mode transitions.

```
#include <float.h>
#include <iomanip>
#include <math.h>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/exec_proto.hh"
#include "trick/message_proto.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Parameter.hh"
#include "IMSim/ExecutionConfiguration.hh"
#include "IMSim/ExecutionControl.hh"
```

Include dependency graph for IMSim/ExecutionControl.cpp:



### Namespaces

- [IMSim](#)

### Variables

- static const std::wstring [IMSim::SIM\\_CONFIG\\_SYNC\\_POINT](#) = L"sim\_config\_v2"
- static const std::wstring [IMSim::INITIALIZE\\_SYNC\\_POINT](#) = L"initialize\_v2"
- static const std::wstring [IMSim::INIT\\_COMPLETE\\_SYNC\\_POINT](#) = L"initialization\_complete\_v2"
- static const std::wstring [IMSim::STARTUP\\_SYNC\\_POINT](#) = L"startup\_v2"
- static const std::wstring [IMSim::FEDSAVE\\_SYNC\\_POINT](#) = L"FEDSAVE\_v2"
- static const std::wstring [IMSim::FEDRUN\\_SYNC\\_POINT](#) = L"FEDRUN\_v2"
- static const std::wstring [IMSim::STARTUP\\_FREEZE\\_SYNC\\_POINT](#) = L"pause\_0.0"
- ATTRIBUTES [attrTrickHLA\\_FreezeInteractionHandler](#) []

#### 8.27.1 Detailed Description

This class provides and abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- ../TrickHLA/SyncPntListBase.cpp
- ExecutionControl.cpp
- PausePointList.cpp

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	Jan 2019	–	IMSim support and testing.
Edwin Z. Crues	NASA ER7	TrickHLA	June 2019	–	Version 3 rewrite.

## Revision History

### 8.27.2 Variable Documentation

#### 8.27.2.1 attrTrickHLA\_\_FreezeInteractionHandler

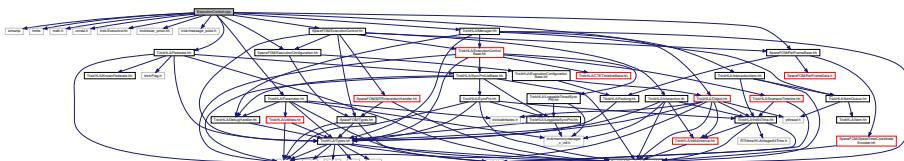
ATTRIBUTES attrTrickHLA\_\_FreezeInteractionHandler[]  
 Referenced by IMSim::ExecutionControl::setup\_interaction\_ref\_attributes().

## 8.28 ExecutionControl.cpp File Reference

This class provides an abstract base class as the base implementation for managing mode transitions.

```
#include <iomanip>
#include <limits>
#include <math.h>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/exec_proto.hh"
#include "trick/message_proto.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/InteractionItem.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Parameter.hh"
#include "SpaceFOM/ExecutionConfiguration.hh"
#include "SpaceFOM/ExecutionControl.hh"
#include "SpaceFOM/RefFrameBase.hh"
```

Include dependency graph for SpaceFOM/ExecutionControl.cpp:



## Namespaces

- [SpaceFOM](#)

## Variables

- static const std::wstring [SpaceFOM::INIT\\_STARTED\\_SYNC\\_POINT](#) = L"initialization\_started"
- static const std::wstring [SpaceFOM::INIT\\_COMPLETED\\_SYNC\\_POINT](#) = L"initialization\_completed"
- static const std::wstring [SpaceFOM::OBJECTS\\_DISCOVERED\\_SYNC\\_POINT](#) = L"objects\_discovered"
- static const std::wstring [SpaceFOM::ROOT\\_FRAME\\_DISCOVERED\\_SYNC\\_POINT](#) = L"root\_frame\_discovered"
- Trick::Clock \* [the\\_clock](#)
- ATTRIBUTES [attrSpaceFOM\\_\\_MTRInteractionHandler](#) []

### 8.28.1 Detailed Description

This class provides an abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [../TrickHLA/SyncPntListBase.cpp](#)
- [RefFrameBase.cpp](#)
- [ExecutionControl.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	Jan 2019	–	<a href="#">SpaceFOM</a> support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

## Revision History

## 8.28.2 Variable Documentation

## 8.28.2.1 attrSpaceFOM\_\_MTRInteractionHandler

ATTRIBUTES attrSpaceFOM\_\_MTRInteractionHandler[]  
 Referenced by SpaceFOM::ExecutionControl::setup\_interaction\_ref\_attributes().

## 8.28.2.2 the\_clock

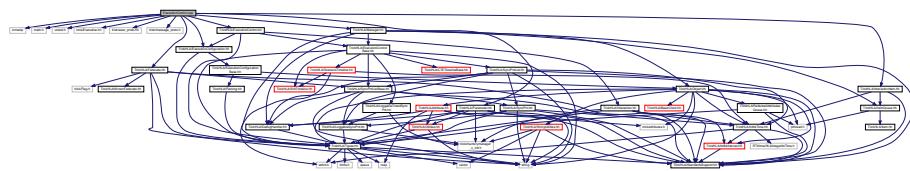
Trick::Clock\* the\_clock  
 Referenced by SpaceFOM::ExecutionControl::exit\_freeze().

## 8.29 ExecutionControl.cpp File Reference

This class provides an abstract base class as the base implementation for managing mode transitions.

```
#include <iomanip>
#include <math.h>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/exec_proto.hh"
#include "trick/message_proto.h"
#include "TrickHLA/ExecutionConfiguration.hh"
#include "TrickHLA/ExecutionControl.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/InteractionItem.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Parameter.hh"
```

Include dependency graph for TrickHLA/ExecutionControl.cpp:



## Namespaces

- [TrickHLA](#)

## Variables

- ATTRIBUTES [attrTrickHLA\\_\\_MTRInteractionHandler](#) []

### 8.29.1 Detailed Description

This class provides and abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [SyncPntListBase.cpp](#)
- [ExecutionControl.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	Jan 2019	–	<a href="#">TrickHLA</a> support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

#### Revision History

### 8.29.2 Variable Documentation

#### 8.29.2.1 attrTrickHLA\_\_MTRInteractionHandler

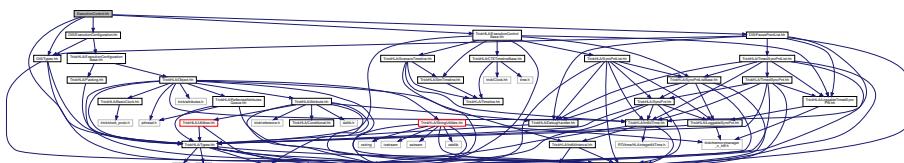
ATTRIBUTES attrTrickHLA\_\_MTRInteractionHandler[ ]

## 8.30 ExecutionControl.hh File Reference

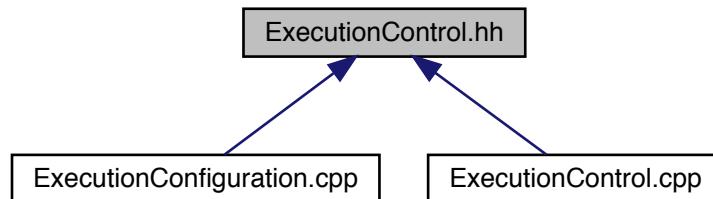
This class provides and abstract base class as the base implementation for managing mode transitions.

```
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Types.hh"
#include "DIS/ExecutionConfiguration.hh"
#include "DIS/PausePointList.hh"
#include "DIS/Types.hh"
```

Include dependency graph for DIS/ExecutionControl.hh:



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



## Data Structures

- class [DIS::ExecutionControl](#)

## Namespaces

- [DIS](#)

### 8.30.1 Detailed Description

This class provides an abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.DIS*

#### Link Dependencies

- [../source/DIS/ExecutionControl.cpp](#)

- [../../source/TrickHLA/ExecutionControlBase.cpp](#)
- [../../source/DIS/PausePointList.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2020	–	<a href="#">DIS</a> development.

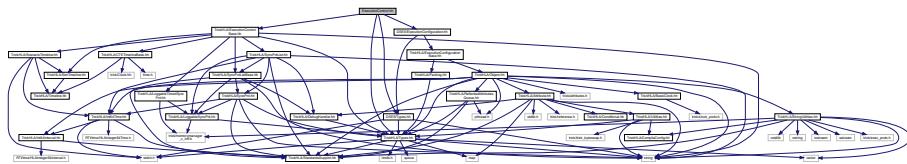
#### Revision History

## 8.31 ExecutionControl.hh File Reference

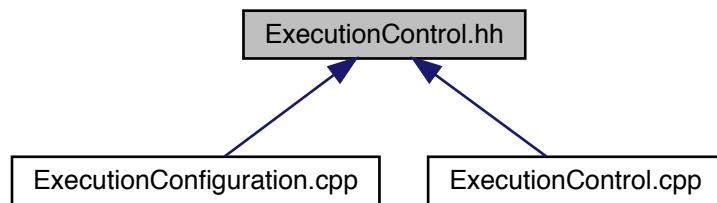
This class provides an abstract base class as the base implementation for managing mode transitions.

```
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Types.hh"
#include "DSES/ExecutionConfiguration.hh"
#include "DSES/Types.hh"
```

Include dependency graph for DSES/ExecutionControl.hh:



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



## Data Structures

- class [DSES::ExecutionControl](#)

## Namespaces

- [DSES](#)

### 8.31.1 Detailed Description

This class provides and abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.DSES*

#### Link Dependencies

- *../source/DIS/ExecutionControl.cpp*
- *../source/TrickHLA/ExecutionControlBase.cpp*

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	DSES development.

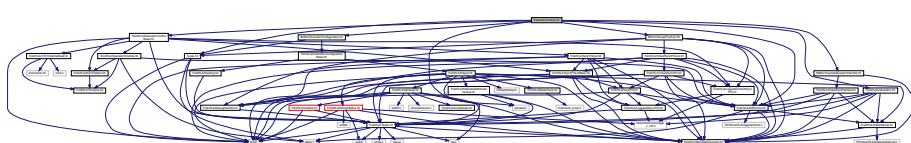
#### Revision History

## 8.32 ExecutionControl.hh File Reference

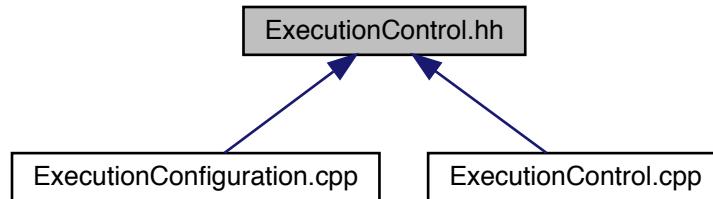
This class provides and abstract base class as the base implementation for managing mode transitions.

```
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/Interaction.hh"
#include "TrickHLA/Types.hh"
#include "IMSim/ExecutionConfiguration.hh"
#include "IMSim/FreezeInteractionHandler.hh"
#include "IMSim/PausePointList.hh"
#include "IMSim/Types.hh"
```

Include dependency graph for *IMSim/ExecutionControl.hh*:



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



## Data Structures

- class [IMSim::ExecutionControl](#)

## Namespaces

- [IMSim](#)

### 8.32.1 Detailed Description

This class provides an abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.IMSim*

#### Link Dependencies

- [../source/TrickHLA/ExecutionControlBase.cpp](#)
- [../source/IMSim/ExecutionControl.cpp](#)
- [../source/IMSim/FreezeInteractionHandler.cpp](#)
- [../source/IMSim/PausePointList.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	<a href="#">IMSim</a> development.

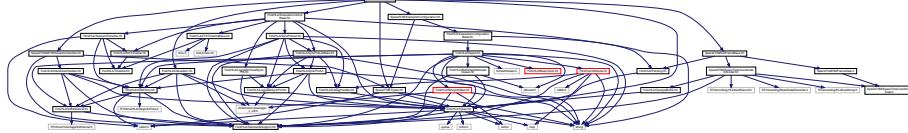
## Revision History

## 8.33 ExecutionControl.hh File Reference

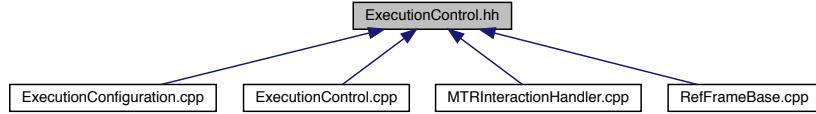
This class provides and abstract base class as the base implementation for managing mode transitions.

```
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Interaction.hh"
#include "TrickHLA/Types.hh"
#include "SpaceFOM/ExecutionConfiguration.hh"
#include "SpaceFOM/MTRInteractionHandler.hh"
#include "SpaceFOM/RefFrameBase.hh"
#include "SpaceFOM/Types.hh"
```

Include dependency graph for SpaceFOM/ExecutionControl.hh:



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



## Data Structures

- class [SpaceFOM::ExecutionControl](#)

## Namespaces

- [SpaceFOM](#)

### 8.33.1 Detailed Description

This class provides and abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

Python Module: *trick.SpaceFOM*

## Link Dependencies

- `../source/TrickHLA/ExecutionControlBase.cpp`
- `../source/SpaceFOM/ExecutionControl.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	<a href="#">SpaceFOM</a> development.

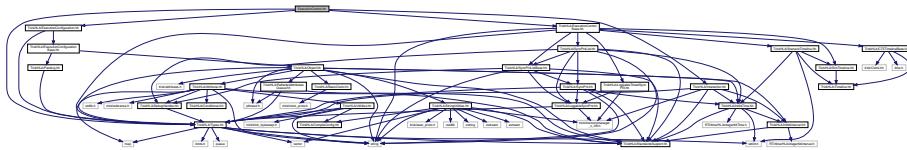
## Revision History

## 8.34 ExecutionControl.hh File Reference

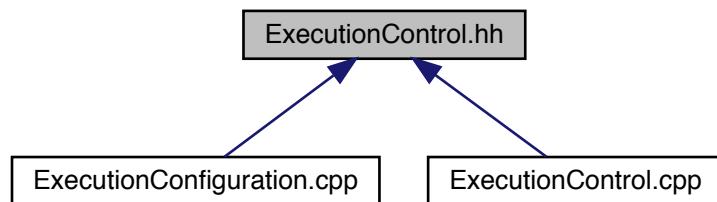
This class provides and abstract base class as the base implementation for [TrickHLA](#) simple execution control.

```
#include "TrickHLA/ExecutionConfiguration.hh"
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Interaction.hh"
#include "TrickHLA/Types.hh"
```

Include dependency graph for `TrickHLA/ExecutionControl.hh`:



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



## Data Structures

- class [TrickHLA::ExecutionControl](#)

## Namespaces

- [TrickHLA](#)

### 8.34.1 Detailed Description

This class provides and abstract base class as the base implementation for [TrickHLA](#) simple execution control.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLA*

#### Link Dependencies

- [../source/TrickHLA/ExecutionControl.cpp](#)
- [../source/TrickHLA/ExecutionControlBase.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2020	–	<a href="#">TrickHLA</a> development.

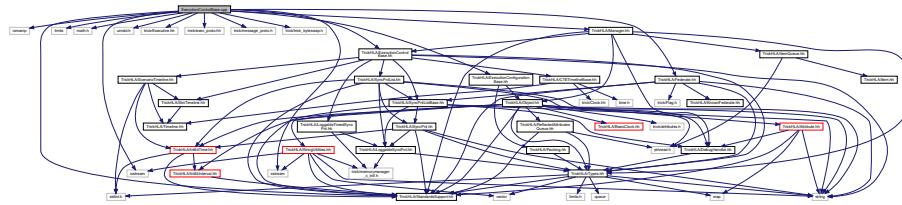
#### Revision History

## 8.35 ExecutionControlBase.cpp File Reference

This class provides and abstract base class as the base implementation for managing mode transitions.

```
#include <iomanip>
#include <iostream>
#include <limits>
#include <math.h>
#include <sstream>
#include <unistd.h>
#include "trick/Executive.hh"
#include "trick/exec_proto.hh"
#include "trick/message_proto.h"
#include "trick/trick_byteswap.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/ExecutionConfigurationBase.hh"
```

```
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
Include dependency graph for ExecutionControlBase.cpp:
```



## Namespaces

- [TrickHLA](#)

## Variables

- [SimTimeline](#) `TrickHLA::def_sim_timeline`
- [ScenarioTimeline](#) `TrickHLA::def_scenario_timeline` (`def_sim_timeline`)

### 8.35.1 Detailed Description

This class provides an abstract base class as the base implementation for managing mode transitions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [SyncPntListBase.cpp](#)
- [ExecutionControlBase.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	Jan 2019	–	<a href="#">TrickHLA</a> support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

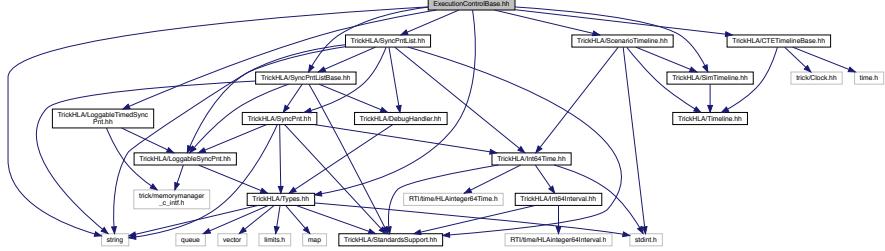
#### Revision History

## 8.36 ExecutionControlBase.hh File Reference

This class provides and abstract base class as the base implementation for execution control.

```
#include <string>
#include "TrickHLA/CTETimelineBase.hh"
#include "TrickHLA/LoggableTimedSyncPnt.hh"
#include "TrickHLA/ScenarioTimeline.hh"
#include "TrickHLA/SimTimeline.hh"
#include "TrickHLA/SyncPntList.hh"
#include "TrickHLA/SyncPntListBase.hh"
#include "TrickHLA/Types.hh"
```

Include dependency graph for ExecutionControlBase.hh:



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



### Data Structures

- class [TrickHLA::ExecutionControlBase](#)

### Namespaces

- [TrickHLA](#)

#### 8.36.1 Detailed Description

This class provides and abstract base class as the base implementation for execution control.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module:** *trick.TrickHLA*

#### Link Dependencies

- `../source/TrickHLA/ExecutionControlBase.cpp`
- `../source/TrickHLA/Types.cpp`
- `../source/TrickHLA/SyncPntListBase.cpp`
- `../source/TrickHLA/Int64Time.cpp`
- `../source/TrickHLA/Timeline.cpp`
- `../source/TrickHLA/SimTimeline.cpp`
- `../source/TrickHLA/ScenarioTimeline.cpp`
- `../source/TrickHLA/CTETimelineBase.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	Oct 2019	–	Initial version.

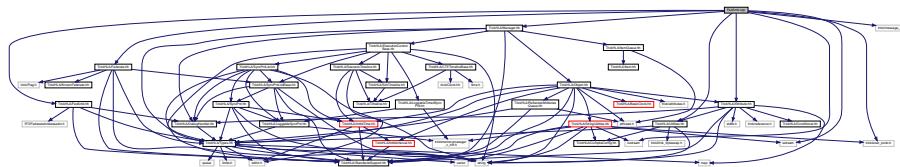
#### Revision History

## 8.37 FedAmb.cpp File Reference

Provides methods called by the RTI Ambassador for simulation object, interaction and time management.

```
#include <iostream>
#include <sstream>
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/FedAmb.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for FedAmb.cpp:



### 8.37.1 Detailed Description

Provides methods called by the RTI Ambassador for simulation object, interaction and time management. Methods of objects of this class are not intended to be called from the Trick S\_define level. However, this class is essentially a polymorphic callback class provided to the RTI Ambassador.

#### Assumptions and Limitations:

- Derived class of abstract FederateAmbassador class to implement methods so that RTI can call functions in the federate.
- Based on HelloWorld example code.
- None of the methods in this class are intended to be called from the Trick S\_define level. However, an instance of this class can be declared in the S\_define.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Federate.cpp](#)
- [Manager.cpp](#)
- [FedAmb.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
DMSO Programmer	DMSO	HLA	Mar 1998	–	HelloWorld Federate Ambassador.
Edwin Z. Crues	Titan Systems Corp.	<a href="#">DIS</a>	Feb 2002	–	HLA Ball Sim.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

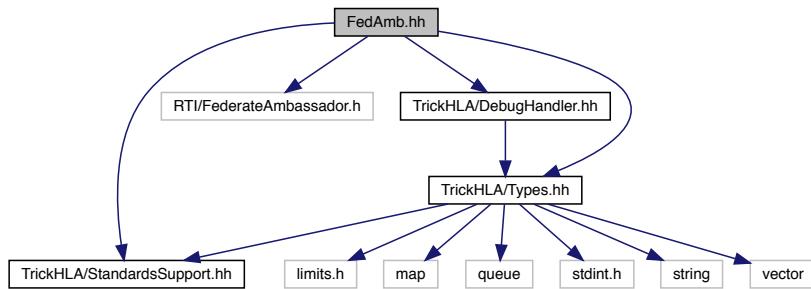
#### Revision History

## 8.38 FedAmb.hh File Reference

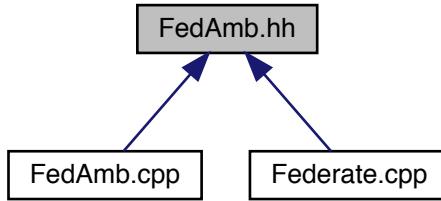
Provides methods called by the HLA RTI Ambassador.

```
#include "TrickHLA/StandardsSupport.hh"
#include "RTI/FederateAmbassador.h"
#include "TrickHLA/DebugHandler.hh"
#include "TrickHLA/Types.hh"
```

Include dependency graph for FedAmb.hh:



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



## Data Structures

- class [TrickHLA::FedAmb](#)

## Namespaces

- [TrickHLA](#)

### 8.38.1 Detailed Description

Provides methods called by the HLA RTI Ambassador.

This class is essentially a polymorphic callback class provided to the RTI Ambassador. It provides methods called by the RTI Ambassador for simulation object, interaction and time management.

The methods in this class fill out the required virtual methods of the `RTI1516_NAMESPACE::FederateAmbassador` abstract class to enable the `FedAmb` class to be instantiated. While this class is instantiable, for almost all practical applications, a simulation developer will want to overload the necessary attribute and interaction callback functions to make things work properly for their particular federation needs.

#### Assumptions and Limitations:

- Derived class of abstract FederateAmbassador class to implement methods so that RTI can call functions in the federate.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLA*

##### Link Dependencies

- ../source/TrickHLA/Federate.cpp
- ../source/TrickHLA/Manager.cpp
- ../source/TrickHLA/FedAmb.cpp

Author	Organization	Project	Date	Rev. ID	Description
DMSO Programmer	DMSO	HLA	Mar 1998	–	HelloWorld Federate Ambassador.
Edwin Z. Crues	Titan Systems Corp.	DIS	Feb 2002	–	HLA Ball Sim.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

#### Revision History

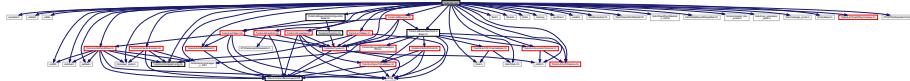
## 8.39 Federate.cpp File Reference

This class is the abstract base class for representing timelines.

```
#include <arpa/inet.h>
#include <cstddef>
#include <cstdio>
#include <cstdlib>
#include <float.h>
#include <fstream>
#include <iostream>
#include <limits>
#include <memory>
#include <sstream>
#include <stdint.h>
#include <sys/time.h>
#include <time.h>
```

```
#include <unistd.h>
#include "trick/Clock.hh"
#include "trick/Executive.hh"
#include "trick/exec_proto.h"
#include "trick/CheckPointRestart.hh"
#include "trick/CheckPointRestart_c_intf.hh"
#include "trick/DataRecordDispatcher.hh"
#include "trick/command_line_protos.h"
#include "trick/input_processor_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "trick/release.h"
#include "TrickHLA/CompileConfig.hh"
#include "TrickHLA/ExecutionConfigurationBase.hh"
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/FedAmb.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Int64Interval.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/OwnershipItem.hh"
#include "TrickHLA/ScenarioTimeline.hh"
#include "TrickHLA/SimTimeline.hh"
#include "TrickHLA/StringUtilities.hh"
#include "TrickHLA/TimeOfDayTimeline.hh"
#include "TrickHLA/Types.hh"
#include "TrickHLA/Utilities.hh"
#include <RTI/RTIambassadorFactory.h>
```

Include dependency graph for Federate.cpp:



## Macros

- `#define THLA_TAG_USE_USLEEP 1`

## Variables

- `Trick::CheckPointRestart * the_cpr`
- `pthread_mutex_t ip_mutex`

### 8.39.1 Detailed Description

This class is the abstract base class for representing timelines.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- [Int64Time.cpp](#)
- [Timeline.cpp](#)
- [SimTimeline.cpp](#)
- [ScenarioTimeline.cpp](#)
- [CTETimelineBase.cpp](#)
- [FedAmb.cpp](#)
- [Federate.cpp](#)
- [Manager.cpp](#)
- [ExecutionControlBase.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	Titan Systems Corp.	<a href="#">DIS</a>	Titan Systems Corp.	–	Initial investigation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	Jan 2019	–	SRFOM support & test.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

## Revision History

### 8.39.2 Macro Definition Documentation

#### 8.39.2.1 THLA\_TAG\_USE\_USLEEP

```
#define THLA_TAG_USE_USLEEP 1
```

### 8.39.3 Variable Documentation

#### 8.39.3.1 ip\_mutex

```
pthread_mutex_t ip_mutex
```

#### 8.39.3.2 the\_cpr

```
Trick::CheckPointRestart* the_cpr
```

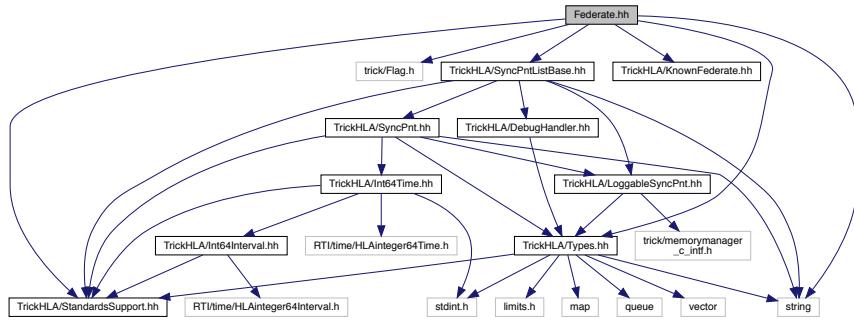
Referenced by TrickHLA::Federate::setup\_checkpoint().

## 8.40 Federate.hh File Reference

This class provides basic services for connecting a Trick based simulation in to a HLA based distributed simulation environment.

```
#include <string>
#include "trick/Flag.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/KnownFederate.hh"
#include "TrickHLA/SyncPntListBase.hh"
#include "TrickHLA/Types.hh"
```

Include dependency graph for Federate.hh:



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



### Data Structures

- class [TrickHLA::Federate](#)

### Namespaces

- [TrickHLA](#)

### Enumerations

- enum [TrickHLA::THLASaveRestoreProcEnum](#) {
 [TrickHLA::No\\_Restore](#) = 0, [TrickHLA::Restore\\_Request\\_Failed](#) = 1, [TrickHLA::Restore\\_Request\\_Succeeded](#) = 2, [TrickHLA::Initiate\\_Restore](#) = 3, [TrickHLA::Restore\\_In\\_Progress](#) = 4, [TrickHLA::Restore\\_Complete](#) = 5, [TrickHLA::Restore\\_Failed](#) = 6 }

#### 8.40.1 Detailed Description

This class provides basic services for connecting a Trick based simulation in to a HLA based distributed simulation environment.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.TrickHLA*

### Link Dependencies

- `../source/TrickHLA/Int64Time.cpp`
- `../source/TrickHLA/FedAmb.cpp`
- `../source/TrickHLA/Federate.cpp`
- `../source/TrickHLA/Manager.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	DSES	Sept 2005	–	DSES Test Sim.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	Jan 2019	–	SRFOM support & test.
Edwin Z. Crues	NASA ER7	TrickHLA	June 2019	–	Version 3 rewrite.

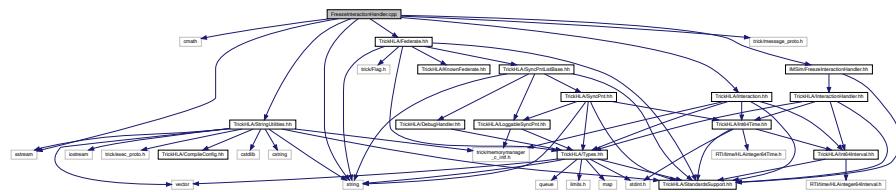
## Revision History

## 8.41 `FreezeInteractionHandler.cpp` File Reference

This class handles the HLA Freeze interactions.

```
#include <cmath>
#include <sstream>
#include <string>
#include "trick/message_proto.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Interaction.hh"
#include "TrickHLA/StringUtilities.hh"
#include "IMSim/FreezeInteractionHandler.hh"
```

Include dependency graph for FreezeInteractionHandler.cpp:



## Macros

- `#define THLA_FREEZE_INTERACTION_DEBUG 0`

### 8.41.1 Detailed Description

This class handles the HLA Freeze interactions.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- `../TrickHLA/Int64Time.cpp`
  - `../TrickHLA/Interaction.cpp`
  - `FreezeInteractionHandler.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3	DSES	July 2009	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

## 8.41.2 Macro Definition Documentation

#### 8.41.2.1 THLA\_FREEZE\_INTERACTION\_DEBUG

```
#define THLA_FREEZE_INTERACTION_DEBUG 0  
Definition at line 51 of file FreezeInteractionHandler.cpp.
```

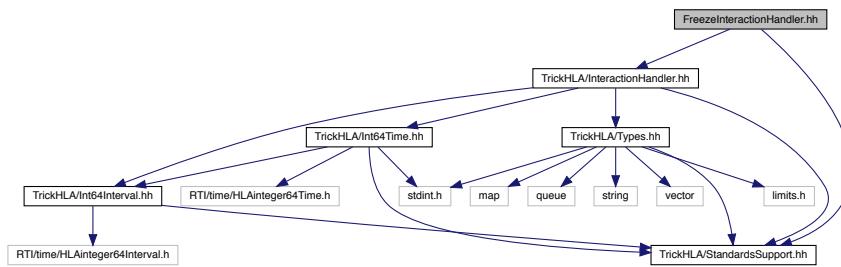
## 8.42 FreezelInteractionHandler.hh File Reference

This class is a specialized class for handling HLA Freeze interactions.

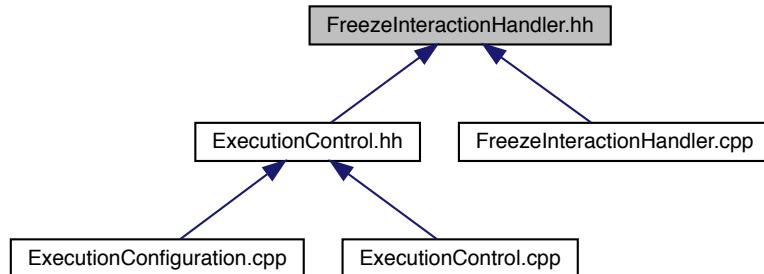
```
#include "TrickHLA/StandardsSupport.hh"
```

```
#include "TrickHLA/InteractionHandler.hh"
```

Include dependency graph for FreezeInteractionHandler.hh:



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



## Data Structures

- class [IMSim::FreezeInteractionHandler](#)

## Namespaces

- IMSim

### 8.42.1 Detailed Description

This class is a specialized class for handling HLA Freeze interactions.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module: *trick.IMSim*****Link Dependencies**

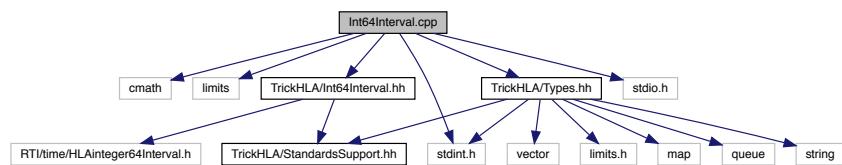
- `../source/TrickHLA/Int64Time.cpp`
- `../source/TrickHLA/InteractionHandler.cpp`
- `../source/IMSim/FreezeInteractionHandler.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3	DSES	July 2009	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

**Revision History****8.43 Int64Interval.cpp File Reference**

This class represents the HLA Interval time.

```
#include <cmath>
#include <limits>
#include <stdint.h>
#include <stdio.h>
#include "TrickHLA/Int64Interval.hh"
#include "TrickHLA/Types.hh"
Include dependency graph for Int64Interval.cpp:
```



## Namespaces

- [TrickHLA](#)

### 8.43.1 Detailed Description

This class represents the HLA Interval time.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Int64Time.cpp](#)
- [Int64Interval.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Robert G. Phillips	Titan Corp.	<a href="#">DIS</a>	October 2004	–	Initial implementation for ISS HTV Sim
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

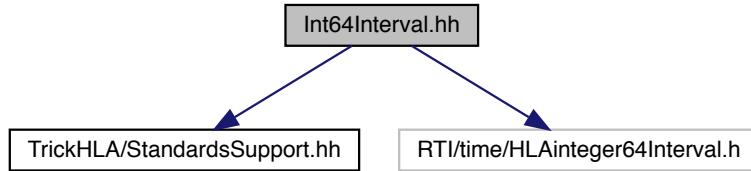
#### Revision History

## 8.44 Int64Interval.hh File Reference

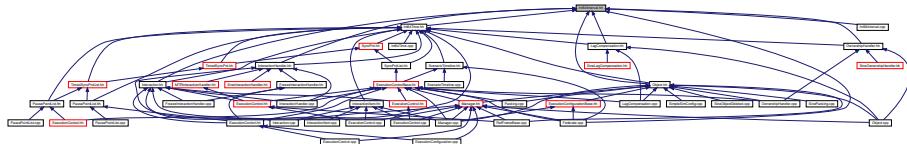
This class represents the HLA Interval time.

```
#include "TrickHLA/StandardsSupport.hh"
#include <RTI/time/HLAinteger64Interval.h>
```

Include dependency graph for Int64Interval.hh:



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



## Data Structures

- class [TrickHLA::Int64Interval](#)

## Namespaces

- [TrickHLA](#)

### 8.44.1 Detailed Description

This class represents the HLA Interval time.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLA*

#### Link Dependencies

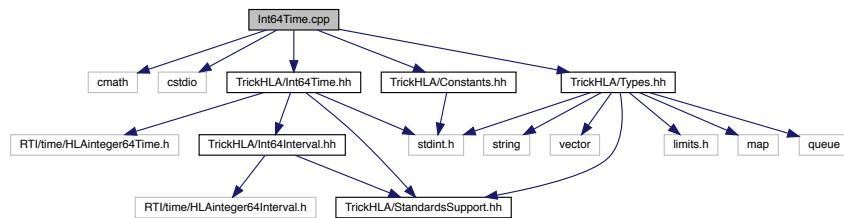
- [../source/TrickHLA/Int64Time.cpp](#)
- [../source/TrickHLA/Int64Interval.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Robert G. Phillips	Titan Corp.	DIS	October 2004	–	Initial implementation for ISS HTV Sim
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.45 Int64Time.cpp File Reference

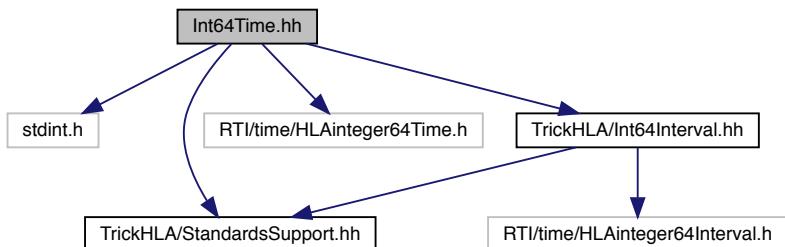
```
#include <cmath>
#include <cstdio>
#include "TrickHLA/Constants.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/Types.hh"
Include dependency graph for Int64Time.cpp:
```



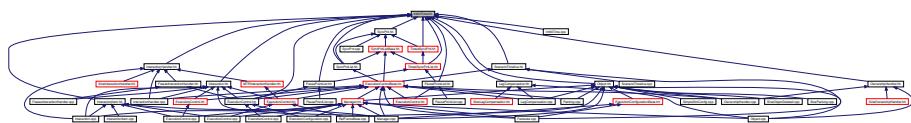
## 8.46 Int64Time.hh File Reference

This class represents the HLA time.

```
#include <stdint.h>
#include "TrickHLA/StandardsSupport.hh"
#include <RTI/time/HLAinteger64Time.h>
#include "TrickHLA/Int64Interval.hh"
Include dependency graph for Int64Time.hh:
```



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



## Data Structures

- class [TrickHLA::Int64Time](#)

## Namespaces

- [TrickHLA](#)

### 8.46.1 Detailed Description

This class represents the HLA time.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Int64Time.cpp`
- `../source/TrickHLA/Int64Interval.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Robert G. Phillips	Titan Corp.	<a href="#">DIS</a>	October 2004	–	Initial implementation for ISS HTV Sim
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.47 integrator.d File Reference

### Variables

- INTEGRATOR. `state` [NUM\_VARIABLES]
- INTEGRATOR. `deriv` [NUM\_STEP][NUM\_VARIABLES]
- INTEGRATOR. `state_ws` [NUM\_STEP][NUM\_VARIABLES]
- INTEGRATOR. `num_state` = NUM\_VARIABLES
- INTEGRATOR. `option` = Runge\_Kutta\_2
- INTEGRATOR. `init` = True
- INTEGRATOR. `first_step_deriv` = Yes

### 8.47.1 Variable Documentation

#### 8.47.1.1 deriv

INTEGRATOR. `deriv`[NUM\_STEP] [NUM\_VARIABLES]

Definition at line 67 of file integrator.d.

Referenced by TrickHLAModel::SineData::set\_derivative().

#### 8.47.1.2 first\_step\_deriv

INTEGRATOR. `first_step_deriv` = Yes

Definition at line 73 of file integrator.d.

#### 8.47.1.3 init

INTEGRATOR. `init` = True

Definition at line 72 of file integrator.d.

#### 8.47.1.4 num\_state

INTEGRATOR. `num_state` = NUM\_VARIABLES

Definition at line 70 of file integrator.d.

#### 8.47.1.5 option

INTEGRATOR. `option` = Runge\_Kutta\_2

Definition at line 71 of file integrator.d.

#### 8.47.1.6 state

INTEGRATOR. `state`[NUM\_VARIABLES]

Definition at line 66 of file integrator.d.

Referenced by TrickHLA::SyncPntListBase::is\_sync\_pnt\_announced(), DIS::pause\_pnt\_state\_enum\_to\_int16(), IM←Sim::pause\_pnt\_state\_enum\_to\_int16(), DIS::pause\_pnt\_state\_enum\_to\_string(), IMSim::pause\_pnt\_state\_enum←to\_string(), TrickHLA::sync\_pnt\_state\_enum\_to\_int16(), and TrickHLA::sync\_pnt\_state\_enum\_to\_string().

### 8.47.1.7 state\_ws

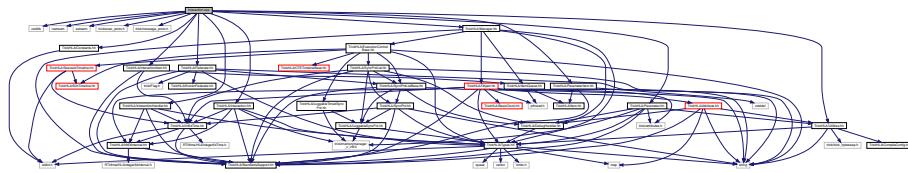
INTEGRATOR. state\_ws [NUM\_STEP] [NUM\_VARIABLES]  
 Definition at line 68 of file integrator.d.

## 8.48 Interaction.cpp File Reference

This class represents an HLA Interaction that is managed by Trick.

```
#include <cstdlib>
#include <iostream>
#include <sstream>
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "TrickHLA/Constants.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Interaction.hh"
#include "TrickHLA/InteractionHandler.hh"
#include "TrickHLA/InteractionItem.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Parameter.hh"
#include "TrickHLA/ParameterItem.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for Interaction.cpp:



### 8.48.1 Detailed Description

This class represents an HLA Interaction that is managed by Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Int64Interval.cpp](#)
- [Int64Time.cpp](#)
- [Parameter.cpp](#)

- [Manager.cpp](#)
- [Federate.cpp](#)
- [InteractionItem.cpp](#)
- [InteractionHandler.cpp](#)
- [Interaction.cpp](#)

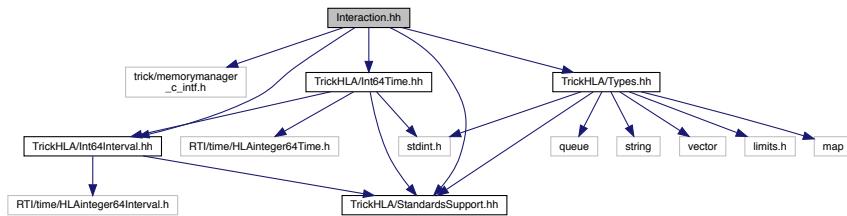
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	Aug 2006	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

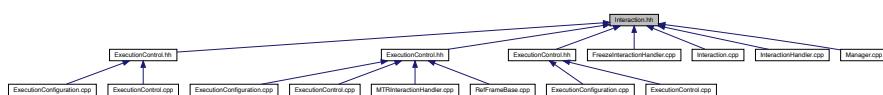
## 8.49 Interaction.hh File Reference

This class represents an HLA Interaction that is managed by Trick.

```
#include "trick/memorymanager_c_intf.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Int64Interval.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/Types.hh"
Include dependency graph for Interaction.hh:
```



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



## Data Structures

- class [TrickHLA::Interaction](#)

## Namespaces

- [TrickHLA](#)

### 8.49.1 Detailed Description

This class represents an HLA Interaction that is managed by Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLA*

#### Link Dependencies

- [../source/TrickHLA/Types.cpp](#)
- [../source/TrickHLA/Int64Interval.cpp](#)
- [../source/TrickHLA/Int64Time.cpp](#)
- [../source/TrickHLA/Parameter.cpp](#)
- [../source/TrickHLA/Manager.cpp](#)
- [../source/TrickHLA/Federate.cpp](#)
- [../source/TrickHLA/InteractionItem.cpp](#)
- [../source/TrickHLA/InteractionHandler.cpp](#)
- [../source/TrickHLA/Interaction.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	Aug 2006	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

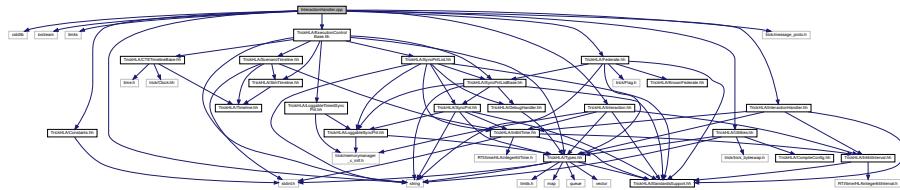
#### Revision History

## 8.50 InteractionHandler.cpp File Reference

This class is the abstract base class for handling HLA interactions.

```
#include <cstdlib>
#include <iostream>
```

```
#include <limits>
#include <string>
#include "trick/message_proto.h"
#include "TrickHLA/Constants.hh"
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Interaction.hh"
#include "TrickHLA/InteractionHandler.hh"
#include "TrickHLA/Utilities.hh"
Include dependency graph for InteractionHandler.cpp:
```



### 8.50.1 Detailed Description

This class is the abstract base class for handling HLA interactions.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- `Int64Interval.cpp`
  - `Int64Time.cpp`
  - `Interaction.cpp`
  - `InteractionHandler.cpp`
  - `ExecutionControlBase.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	April 2016	–	Initial version.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

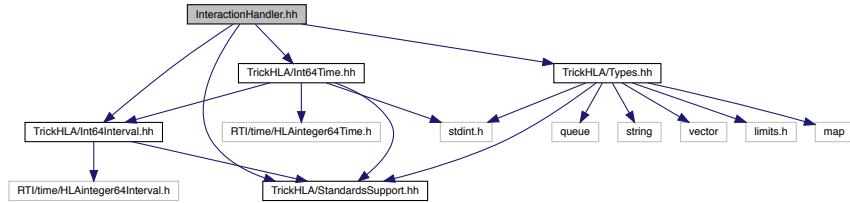
## Revision History

## 8.51 InteractionHandler.hh File Reference

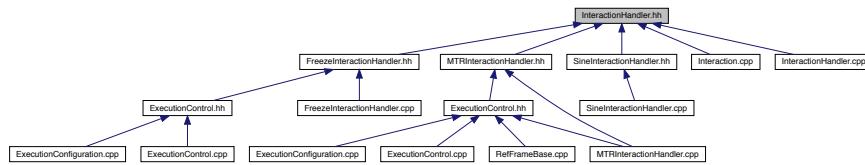
This class is the abstract base class for handling HLA interactions.

```
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Int64Interval.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/Types.hh"
```

Include dependency graph for InteractionHandler.hh:



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



## Data Structures

- class [TrickHLA::InteractionHandler](#)

## Namespaces

- [TrickHLA](#)

### 8.51.1 Detailed Description

This class is the abstract base class for handling HLA interactions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module:** *trick.TrickHLA***Link Dependencies**

- ..../source/TrickHLA/Int64Interval.cpp
- ..../source/TrickHLA/Int64Time.cpp
- ..../source/TrickHLA/Interaction.cpp
- ..../source/TrickHLA/InteractionHandler.cpp

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	Aug 2006	–	Initial Interaction Handler.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

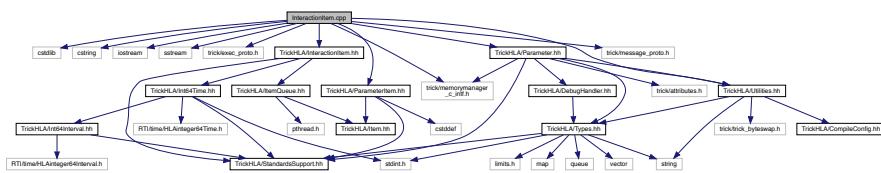
**Revision History**

## 8.52 InteractionItem.cpp File Reference

This class represents a queue for holding HLA Interactions of either Timestamp Order (TSO) or Receive Order (RO).

```
#include <cstdlib>
#include <cstring>
#include <iostream>
#include <sstream>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "TrickHLA/InteractionItem.hh"
#include "TrickHLA/Parameter.hh"
#include "TrickHLA/ParameterItem.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for `InteractionItem.cpp`:

**Variables**

- ATTRIBUTES `attrParameterItem` []

### 8.52.1 Detailed Description

This class represents a queue for holding HLA Interactions of either Timestamp Order (TSO) or Receive Order (RO).

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Link Dependencies**

- [Int64Time.cpp](#)
- [ItemQueue.cpp](#)
- [Parameter.cpp](#)
- [ParameterItem.cpp](#)
- [InteractionItem.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	May 2007	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

**Revision History****8.52.2 Variable Documentation****8.52.2.1 attrParameterItem**

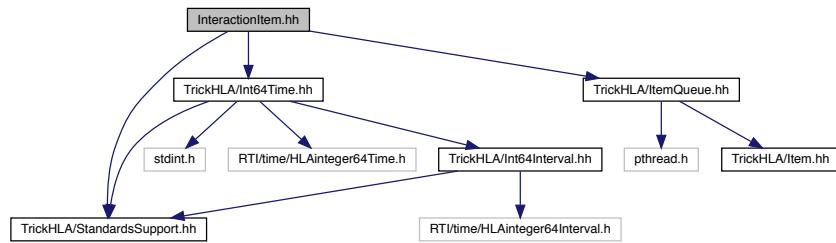
```
ATTRIBUTES attrParameterItem[ ]
```

**8.53 InteractionItem.hh File Reference**

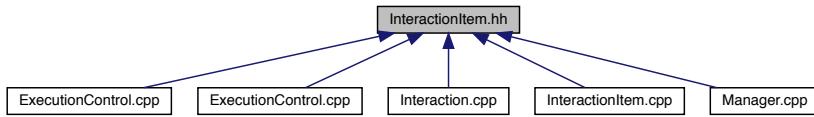
This class represents a queue for holding HLA Interactions of either Timestamp Order (TSO) or Receive Order (RO).

```
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/ItemQueue.hh"
```

Include dependency graph for `InteractionItem.hh`:



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



## Data Structures

- class [TrickHLA::InteractionItem](#)

## Namespaces

- [TrickHLA](#)

### 8.53.1 Detailed Description

This class represents a queue for holding HLA Interactions of either Timestamp Order (TSO) or Receive Order (RO).

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Int64Time.cpp`
- `../source/TrickHLA/ItemQueue.cpp`

- [..../source/TrickHLA/Parameter.cpp](#)
- [..../source/TrickHLA/ParameterItem.cpp](#)
- [..../source/TrickHLA/InteractionItem.cpp](#)

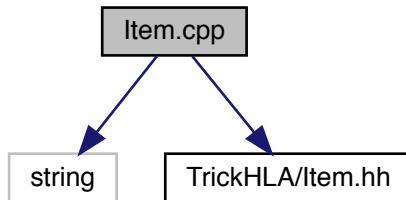
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	May 2007	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.54 Item.cpp File Reference

This class represents an Item in the item queue.

```
#include <string>
#include "TrickHLA/Item.hh"
Include dependency graph for Item.cpp:
```



### 8.54.1 Detailed Description

This class represents an Item in the item queue.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- [Item.cpp](#)

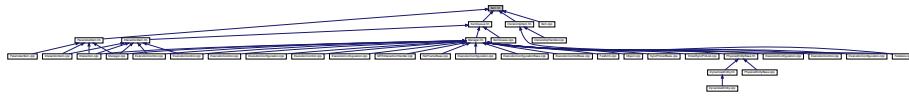
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	<a href="#">TrickHLA</a>	Feb 2009	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

## Revision History

## 8.55 Item.hh File Reference

This class represents a item to be held in an Item Queue.

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



## Data Structures

- class [TrickHLA::Item](#)

## Namespaces

- [TrickHLA](#)

### 8.55.1 Detailed Description

This class represents a item to be held in an Item Queue.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

## Link Dependencies

- [../source/TrickHLA/Item.cpp](#)

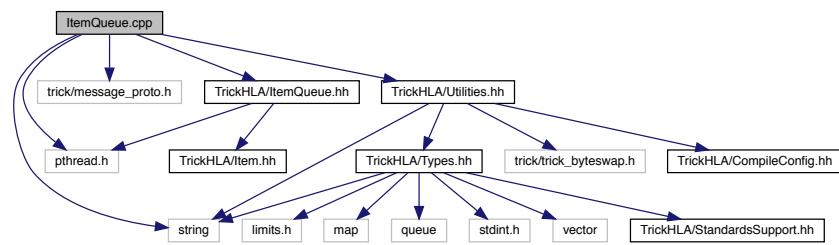
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	TrickHLA	Feb 2009	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

## 8.56 ItemQueue.cpp File Reference

This class represents a queue for holding Items.

```
#include <pthread.h>
#include <string>
#include "trick/message_proto.h"
#include "TrickHLA/ItemQueue.hh"
#include "TrickHLA/Utilities.hh"
Include dependency graph for ItemQueue.cpp:
```



### 8.56.1 Detailed Description

This class represents a queue for holding Items.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Item.cpp](#)
- [ItemQueue.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	IMSim	Feb 2009	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

### Revision History

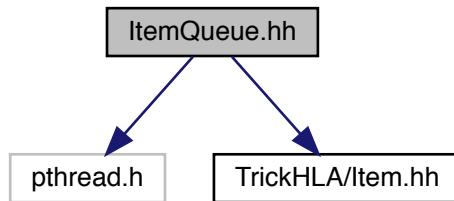
## 8.57 ItemQueue.hh File Reference

This class represents a queue for holding Items.

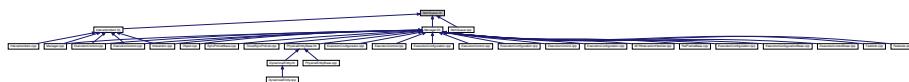
```
#include <pthread.h>
```

```
#include "TrickHLA/Item.hh"
```

Include dependency graph for ItemQueue.hh:



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



### Data Structures

- class [TrickHLA::ItemQueue](#)

### Namespaces

- [TrickHLA](#)

#### 8.57.1 Detailed Description

This class represents a queue for holding Items.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module:** *trick.TrickHLA***Link Dependencies**

- [../source/TrickHLA/ItemQueue.cpp](#)
- [../source/TrickHLA/Item.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	<a href="#">IMSim</a>	Feb 2009	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

**Revision History**

## 8.58 KnownFederate.hh File Reference

A class representing an HLA federate known to the Federation.

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

**Data Structures**

- class [TrickHLA::KnownFederate](#)

**Namespaces**

- [TrickHLA](#)

### 8.58.1 Detailed Description

A class representing an HLA federate known to the Federation.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.TrickHLA*

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	April 2006	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

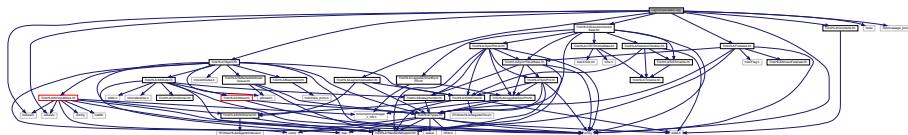
## Revision History

## 8.59 LagCompensation.cpp File Reference

This class is the abstract base class for Trick HLA lag compensation.

```
#include <iostream>
#include <limits>
#include <sstream>
#include <string>
#include "trick/message_proto.h"
#include "TrickHLA/Constants.hh"
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/LagCompensation.hh"
#include "TrickHLA/Object.hh"
```

Include dependency graph for LagCompensation.cpp:



### 8.59.1 Detailed Description

This class is the abstract base class for Trick HLA lag compensation.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Link Dependencies**

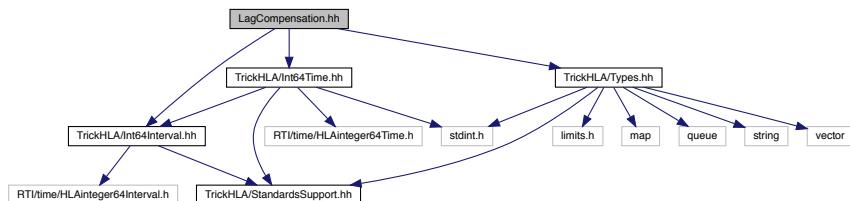
- [Types.cpp](#)
- [ExecutionControlBase.cpp](#)
- [Attribute.cpp](#)
- [Object.cpp](#)
- [Int64Interval.cpp](#)
- [Int64Time.cpp](#)
- [LagCompensation.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	June 2006	–	<a href="#">DSES</a> Initial Lag Compensation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

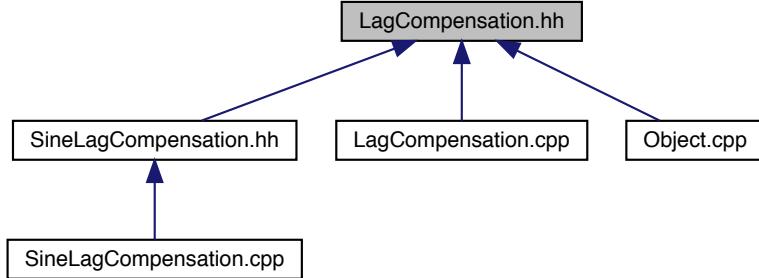
**Revision History****8.60 LagCompensation.hh File Reference**

This class is the abstract base class for [TrickHLA](#) lag compensation.

```
#include "TrickHLA/Int64Interval.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/Types.hh"
Include dependency graph for LagCompensation.hh:
```



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



## Data Structures

- class [TrickHLA::LagCompensation](#)

## Namespaces

- [TrickHLA](#)

### 8.60.1 Detailed Description

This class is the abstract base class for [TrickHLA](#) lag compensation.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Types.cpp`
- `../source/TrickHLA/Attribute.cpp`
- `../source/TrickHLA/Object.cpp`
- `../source/TrickHLA/Int64Interval.cpp`
- `../source/TrickHLA/Int64Time.cpp`
- `../source/TrickHLA/LagCompensation.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	June 2006	–	DSES Initial Lag Compensation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

#### Revision History

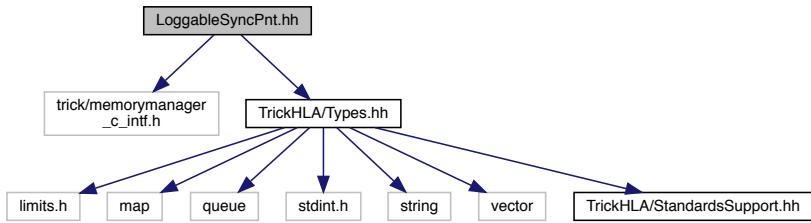
## 8.61 LoggableSyncPnt.hh File Reference

This class provides a mechanism for logging sync point data and retrieving it from the log file.

```
#include "trick/memorymanager_c_intf.h"
```

```
#include "TrickHLA/Types.hh"
```

Include dependency graph for LoggableSyncPnt.hh:



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



### Data Structures

- class [TrickHLA::LoggableSyncPnt](#)

### Namespaces

- [TrickHLA](#)

#### 8.61.1 Detailed Description

This class provides a mechanism for logging sync point data and retrieving it from the log file.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Python Module: `trick.TrickHLA`

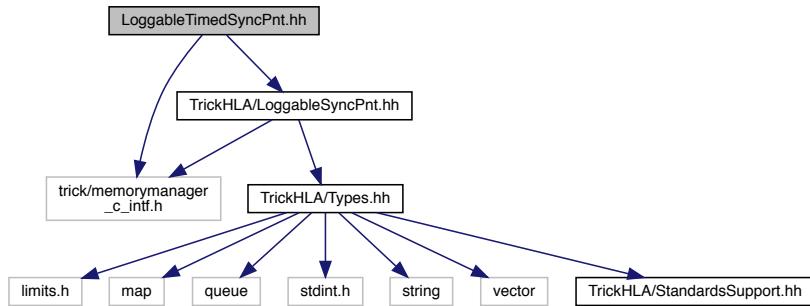
Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3	<a href="#">TrickHLA</a>	July 2009	–	Checkpoint / restore of <a href="#">TrickHLA</a> .
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

### Revision History

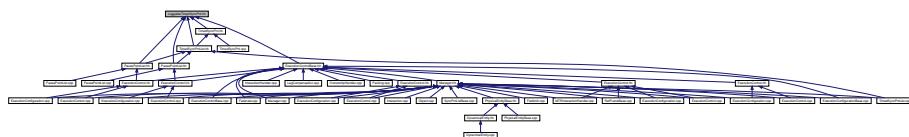
## 8.62 LoggableTimedSyncPnt.hh File Reference

This class provides a mechanism for logging timed synchronization point data and retrieving it from the log file.

```
#include "trick/memorymanager_c_intf.h"
#include "TrickHLA/LoggableSyncPnt.hh"
Include dependency graph for LoggableTimedSyncPnt.hh:
```



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



### Data Structures

- class [TrickHLA::LoggableTimedSyncPnt](#)

## Namespaces

- [TrickHLA](#)

### 8.62.1 Detailed Description

This class provides a mechanism for logging timed synchronization point data and retrieving it from the log file.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLA*

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3	<a href="#">TrickHLA</a>	July 2009	–	Checkpoint / restore of <a href="#">TrickHLA</a> .
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

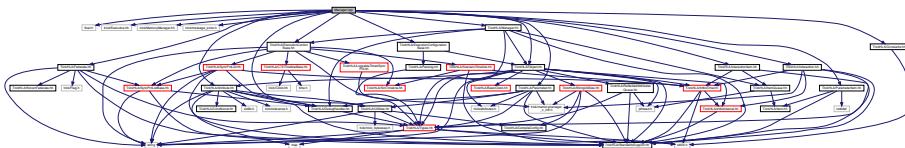
#### Revision History

## 8.63 Manager.cpp File Reference

This class manages the interface between a Trick simulation and HLA.

```
#include <float.h>
#include "trick/Executive.hh"
#include "trick/MemoryManager.hh"
#include "trick/message_proto.h"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Constants.hh"
#include "TrickHLA/ExecutionConfigurationBase.hh"
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Interaction.hh"
#include "TrickHLA/InteractionItem.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Object.hh"
#include "TrickHLA/Parameter.hh"
#include "TrickHLA/ParameterItem.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for Manager.cpp:



## Variables

- ATTRIBUTES `attrTrickHLA__InteractionItem []`
- ATTRIBUTES `attrTrickHLA__Interaction []`
- ATTRIBUTES `attrSRFOM__MTRInteractionHandler []`
- ATTRIBUTES `attrTrickHLA__Parameter []`

### 8.63.1 Detailed Description

This class manages the interface between a Trick simulation and HLA.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- `Types.cpp`
- `Object.cpp`
- `Manager.cpp`
- `ExecutionControlBase.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	May 2006	–	DSES Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	Jan 2019	–	SRFOM support and testing.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

#### Revision History

### 8.63.2 Variable Documentation

#### 8.63.2.1 attrSRFOM\_\_MTRInteractionHandler

```
ATTRIBUTES attrSRFOM__MTRInteractionHandler[ ]
```

#### 8.63.2.2 attrTrickHLA\_\_Interaction

```
ATTRIBUTES attrTrickHLA__Interaction[]
```

#### 8.63.2.3 attrTrickHLA\_\_InteractionItem

```
ATTRIBUTES attrTrickHLA__InteractionItem[ ]
```

#### 8.63.2.4 attrTrickHLA\_\_Parameter

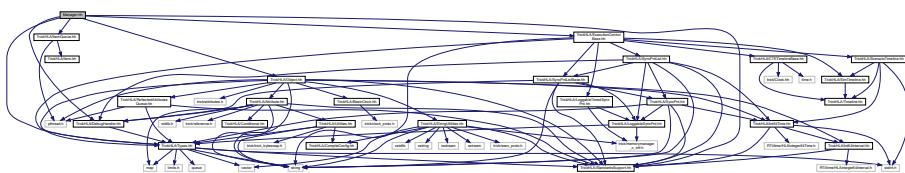
```
ATTRIBUTES attrTrickHLA__Parameter[ ]
```

## 8.64 Manager.hh File Reference

This class manages the interface between a Trick simulation and HLA.

```
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/DebugHandler.hh"
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/ItemQueue.hh"
#include "TrickHLA/Object.hh"
#include "TrickHLA/Types.hh"
```

Include dependency graph for Manager.hh:



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



## Data Structures

- class [TrickHLA::Manager](#)

## Namespaces

- [TrickHLA](#)

## Enumerations

- enum [TrickHLA::ManagerTypeOfInteractionEnum](#) { [TrickHLA::TRICKHLA\\_MANAGER\\_USER\\_DEFINED\\_INTERACTION](#) = 0, [TrickHLA::TRICKHLA\\_MANAGER\\_BUILTIN\\_FREEZE\\_INTERACTION](#) = 1, [TrickHLA::TRICKHLA\\_MANAGER\\_BUILTIN\\_MTR](#) = 2 }

### 8.64.1 Detailed Description

This class manages the interface between a Trick simulation and HLA.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: [trick.TrickHLA](#)

#### Link Dependencies

- [../source/TrickHLA/Types.cpp](#)
- [../source/TrickHLA/Object.cpp](#)
- [../source/TrickHLA/Manager.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	May 2006	—	<a href="#">DSES</a> Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	—	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	Jan 2019	—	SRFOM support and testing.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	—	Version 3 rewrite.

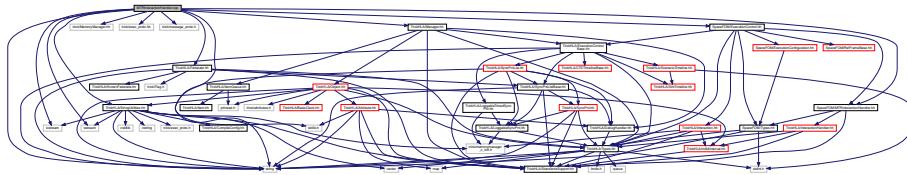
#### Revision History

## 8.65 MTRInteractionHandler.cpp File Reference

This class handles the HLA interactions for Space Reference FOM ([SpaceFOM](#)) Mode Transition Request (MTR) interaction.

```
#include <iostream>
#include <sstream>
#include <stdlib.h>
#include <string>
```

```
#include "trick/MemoryManager.hh"
#include "trick/exec_proto.hh"
#include "trick/message_proto.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/StringUtilities.hh"
#include "SpaceFOM/ExecutionControl.hh"
#include "SpaceFOM/MTRInteractionHandler.hh"
Include dependency graph for MTRInteractionHandler.cpp:
```



### 8.65.1 Detailed Description

This class handles the HLA interactions for Space Reference FOM ([SpaceFOM](#)) Mode Transition Request (MTR) interaction.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- ..../TrickHLA/InteractionHandler.cpp
- [MTRInteractionHandler.cpp](#)
- ExecutionControl.cpp

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	Jan 2019	–	Initial implementation.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

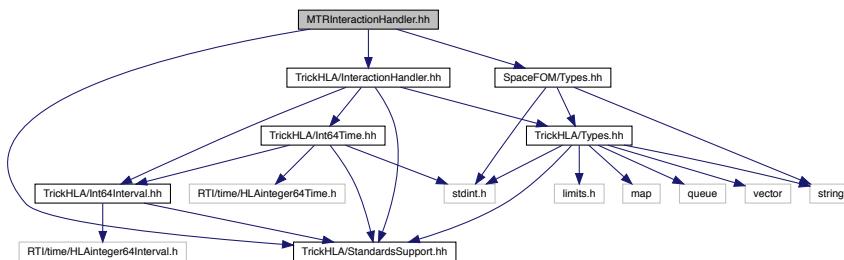
#### Revision History

## 8.66 MTRInteractionHandler.hh File Reference

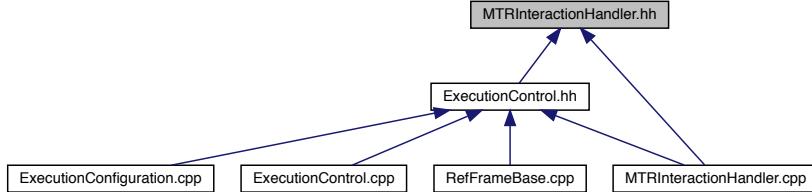
This is the base implementation for the Space Reference FOM ([SpaceFOM](#)) Mode Transition Request (MTR) interaction handler.

```
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/InteractionHandler.hh"
#include "SpaceFOM/Types.hh"
```

Include dependency graph for MTRInteractionHandler.hh:



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



### Data Structures

- class [SpaceFOM::MTRInteractionHandler](#)

### Namespaces

- [TrickHLA](#)
- [SpaceFOM](#)

#### 8.66.1 Detailed Description

This is the base implementation for the Space Reference FOM ([SpaceFOM](#)) Mode Transition Request (MTR) interaction handler.

##### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Python Module: *trick.SpaceFOM*

#### Link Dependencies

- *../source/TrickHLA/InteractionHandler.cpp*
- *../source/SpaceFOM/MTRInteractionHandler.cpp*

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	Jan 2019	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

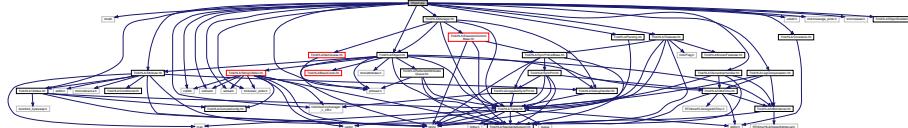
#### Revision History

## 8.67 Object.cpp File Reference

This class represents an HLA object that is managed by Trick.

```
#include <cmath>
#include <cstdlib>
#include <iostream>
#include <pthread.h>
#include <sstream>
#include <unistd.h>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "trick/release.h"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Constants.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/LagCompensation.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Object.hh"
#include "TrickHLA/ObjectDeleted.hh"
#include "TrickHLA/OwnershipHandler.hh"
#include "TrickHLA/Packing.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for Object.cpp:



## Macros

- `#define THLA_OBJ_DEBUG 0`
- `#define THLA_OBJ_DEBUG_SEND 0`
- `#define THLA_OBJ_DEBUG_RECEIVE 0`
- `#define THLA_OBJ_OWNERSHIP_DEBUG 0`
- `#define THLA_OBJ_DEBUG_VALID_OBJECT_RECEIVE 0`

## Functions

- `void * grant_push_pthread_function (void *arg)`  
*The function that runs in the push P-thread that handles the push request grant.*
- `void * ownership_divestiture_pthread_function (void *arg)`  
*The function that runs in the ownership divestiture P-thread that handles ownership transfer.*

### 8.67.1 Detailed Description

This class represents an HLA object that is managed by Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Object.cpp](#)
- [Attribute.cpp](#)
- [Manager.cpp](#)
- [OwnershipHandler.cpp](#)
- [Int64Time.cpp](#)
- [ReflectedAttributesQueue.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	May 2006	–	DSES Created Object
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.67.2 Macro Definition Documentation

### 8.67.2.1 THLA\_OBJ\_DEBUG

```
#define THLA_OBJ_DEBUG 0
```

Definition at line 70 of file Object.cpp.

### 8.67.2.2 THLA\_OBJ\_DEBUG\_RECEIVE

```
#define THLA_OBJ_DEBUG_RECEIVE 0
```

Definition at line 72 of file Object.cpp.

### 8.67.2.3 THLA\_OBJ\_DEBUG\_SEND

```
#define THLA_OBJ_DEBUG_SEND 0
```

Definition at line 71 of file Object.cpp.

### 8.67.2.4 THLA\_OBJ\_DEBUG\_VALID\_OBJECT\_RECEIVE

```
#define THLA_OBJ_DEBUG_VALID_OBJECT_RECEIVE 0
```

Definition at line 79 of file Object.cpp.

### 8.67.2.5 THLA\_OBJ\_OWNERSHIP\_DEBUG

```
#define THLA_OBJ_OWNERSHIP_DEBUG 0
```

Definition at line 73 of file Object.cpp.

## 8.67.3 Function Documentation

### 8.67.3.1 grant\_push\_pthread\_function()

```
void* grant_push_pthread_function (
    void * arg )
```

The function that runs in the push P-thread that handles the push request grant.

This function is local to this file and is NOT part of the class.

#### Returns

Void pointer and is always NULL.

#### Parameters

<i>arg</i>	Arguments list. <b>Trick Job Class:</b> <i>scheduled</i>
------------	--

Definition at line 3420 of file Object.cpp.

References TrickHLA::Object::grant\_push\_request().

Referenced by TrickHLA::Object::grant\_push\_request\_pthread().

### 8.67.3.2 ownership\_divestiture\_pthread\_function()

```
void* ownership_divestiture_pthread_function (
    void * arg )
```

The function that runs in the ownership divestiture P-thread that handles ownership transfer.  
This function is local to this file and is NOT part of the class.

#### Returns

Void pointer and is always NULL.

#### Parameters

<i>arg</i>	Arguments list. <b>Trick Job Class:</b> <i>scheduled</i>
------------	--

Definition at line 3535 of file Object.cpp.

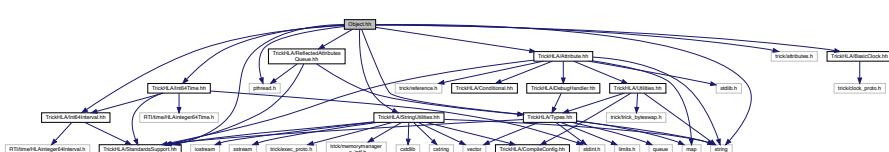
References TrickHLA::DivestThreadArgs::handle\_set, TrickHLA::Object::negotiated\_attribute\_ownership\_divestiture(), THLA\_NEWLINE, and TrickHLA::DivestThreadArgs::trick\_hla\_obj.

Referenced by TrickHLA::Object::push\_ownership().

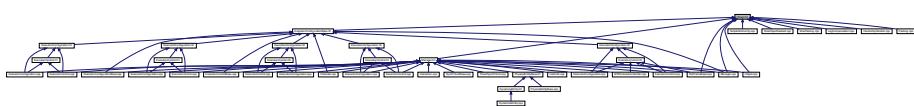
## 8.68 Object.hh File Reference

This class represents an HLA object that is managed by Trick.

```
#include <pthread.h>
#include <string>
#include "trick/attributes.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/BasicClock.hh"
#include "TrickHLA/Int64Interval.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/ReflectedAttributesQueue.hh"
#include "TrickHLA/StringUtilities.hh"
#include "TrickHLA/Types.hh"
Include dependency graph for Object.hh:
```



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



## Data Structures

- class [TrickHLA::Object](#)
- struct [TrickHLA::DivestThreadArgs](#)

## Namespaces

- [TrickHLA](#)

## Macros

- `#define _TRICKHLA_OBJECT_HH_`
- `#define THLA_USE_ATTRIBUTE_MAP_CLASS_INSTANCE`

## Typedefs

- `typedef std::map< RTI1516_NAMESPACE::ObjectHandle, Object * > TrickHLA::ObjectInstanceMap`  
*trick\_io{\*\*} Map of [TrickHLA](#) objects.*

### 8.68.1 Detailed Description

This class represents an HLA object that is managed by Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Object.cpp`
- `../source/TrickHLA/Attribute.cpp`
- `../source/TrickHLA/Manager.cpp`
- `../source/TrickHLA/OwnershipHandler.cpp`
- `../source/TrickHLA/Int64Time.cpp`
- `../source/TrickHLA/ReflectedAttributesQueue.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	May 2006	–	<a href="#">DSES</a> Created Object
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

## 8.68.2 Macro Definition Documentation

### 8.68.2.1 `_TRICKHLA_OBJECT_HH_`

```
#define _TRICKHLA_OBJECT_HH_
```

Definition at line 46 of file Object.hh.

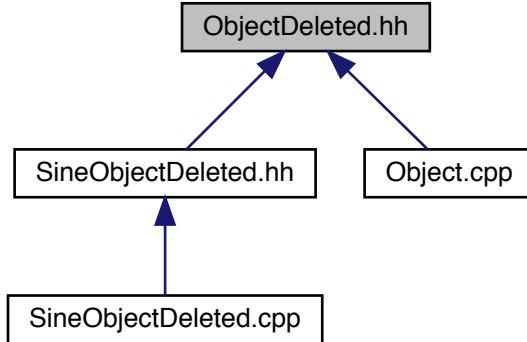
### 8.68.2.2 `THLA_USE_ATTRIBUTE_MAP_CLASS_INSTANCE`

```
#define THLA_USE_ATTRIBUTE_MAP_CLASS_INSTANCE
```

Definition at line 50 of file Object.hh.

## 8.69 ObjectDeleted.hh File Reference

This class is the abstract base class for a callback of identification of deleted objects from the RTI.  
This graph shows which files directly or indirectly include this file:



## Data Structures

- class [TrickHLA::ObjectDeleted](#)

## Namespaces

- [TrickHLA](#)

## 8.69.1 Detailed Description

This class is the abstract base class for a callback of identification of deleted objects from the RTI.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module: *trick.TrickHLA*****Link Dependencies**

- ..../source/TrickHLA/Object.cpp

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3-Communications	DSES	June 2008	–	IMSim: report <a href="#">TrickHLA::Object</a> as deleted via a callback.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

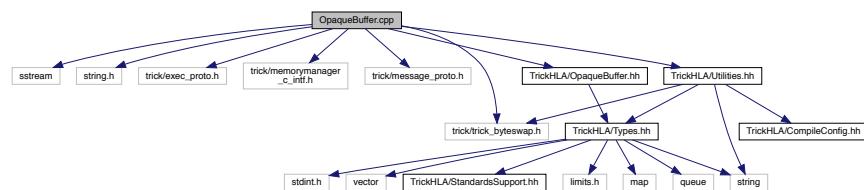
**Revision History**

## 8.70 OpaqueBuffer.cpp File Reference

This class provides a generic opaque buffer that is in the Trick managed memory space.

```
#include <iostream>
#include <string.h>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "trick/trick_byteswap.h"
#include "TrickHLA/OpaqueBuffer.hh"
#include "TrickHLA/Utilities.hh"

Include dependency graph for OpaqueBuffer.cpp:
```



### 8.70.1 Detailed Description

This class provides a generic opaque buffer that is in the Trick managed memory space. This is useful for sending a fixed record of data that includes byte padding to ensure a byte alignment.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [OpaqueBuffer.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	July 2009	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	June 2019	–	Version 3 rewrite.

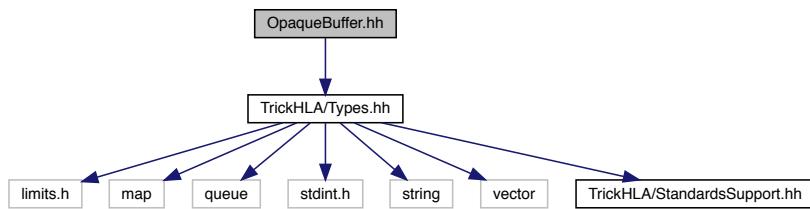
#### Revision History

## 8.71 OpaqueBuffer.hh File Reference

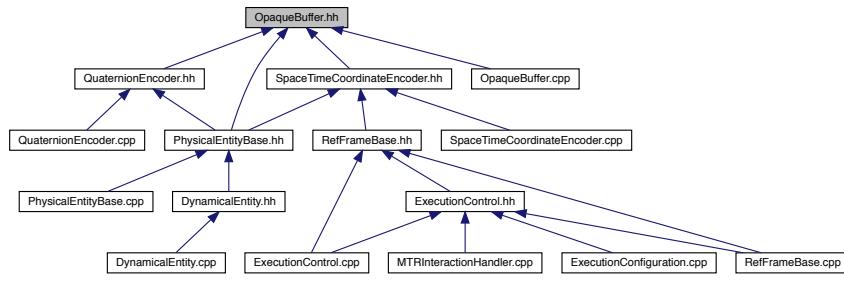
This class provides a generic opaque buffer that is in the Trick managed memory space. This is useful for sending a fixed record of data that includes byte padding to ensure a byte alignment.

#include "TrickHLA/Types.hh"

Include dependency graph for OpaqueBuffer.hh:



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



## Data Structures

- class [TrickHLA::OpaqueBuffer](#)

## Namespaces

- [TrickHLA](#)

### 8.71.1 Detailed Description

This class provides a generic opaque buffer that is in the Trick managed memory space. This is useful for sending a fixed record of data that includes byte padding to ensure a byte alignment.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/OpaqueBuffer.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	July 2009	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

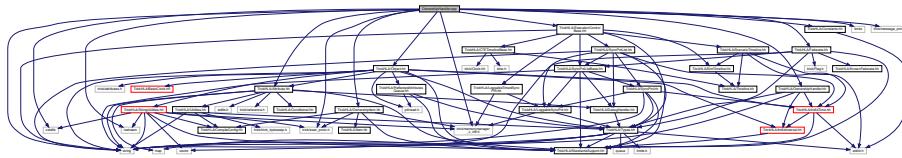
#### Revision History

## 8.72 OwnershipHandler.cpp File Reference

This class represents ownership transfer of HLA attributes for a specific object.

```
#include <cstdlib>
#include <iostream>
#include <limits>
#include <string>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Constants.hh"
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Object.hh"
#include "TrickHLA/OwnershipHandler.hh"
#include "TrickHLA/OwnershipItem.hh"
```

Include dependency graph for OwnershipHandler.cpp:



### 8.72.1 Detailed Description

This class represents ownership transfer of HLA attributes for a specific object.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Types.cpp](#)
- [Int64Time.cpp](#)
- [Int64Interval.cpp](#)
- [Attribute.cpp](#)
- [Object.cpp](#)
- [ExecutionControlBase.cpp](#)
- [OwnershipHandler.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	December 2006	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

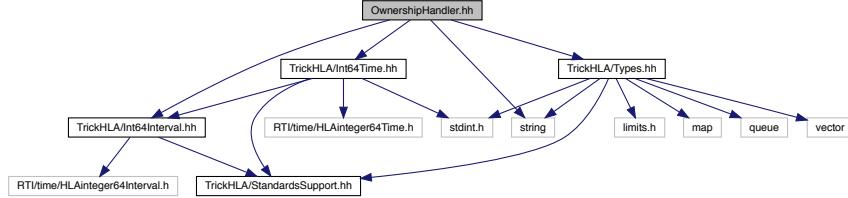
#### Revision History

## 8.73 OwnershipHandler.hh File Reference

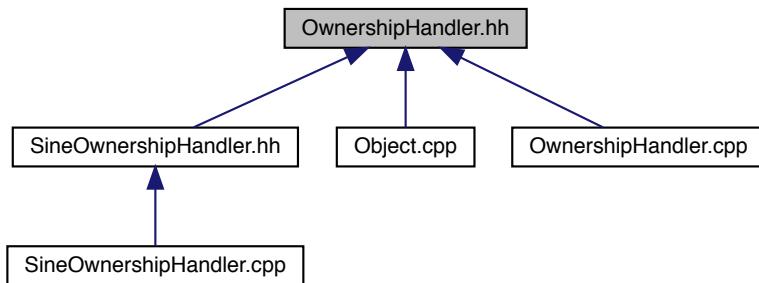
This class represents ownership transfer of HLA attributes for a specific object.

```
#include <string>
#include "TrickHLA/Int64Interval.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/Types.hh"
```

Include dependency graph for OwnershipHandler.hh:



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



## Data Structures

- class [TrickHLA::OwnershipHandler](#)

## Namespaces

- [TrickHLA](#)

## Typedefs

- `typedef std::map< std::string, Attribute * > TrickHLA::THLAAttributeMap`
- `typedef std::map< double, THLAAttributeMap *, std::less< double > > TrickHLA::AttributeOwnershipMap`

### 8.73.1 Detailed Description

This class represents ownership transfer of HLA attributes for a specific object.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Types.cpp`
- `../source/TrickHLA/Int64Time.cpp`
- `../source/TrickHLA/Int64Interval.cpp`
- `../source/TrickHLA/Attribute.cpp`
- `../source/TrickHLA/Object.cpp`
- `../source/TrickHLA/OwnershipHandler.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	December 2006	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

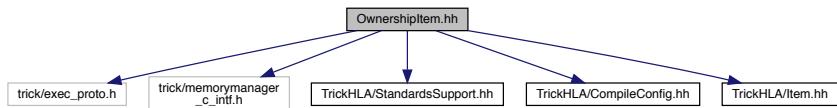
#### Revision History

## 8.74 OwnershipItem.hh File Reference

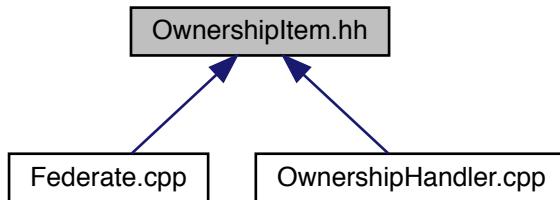
This class represents a queue item for holding ownership transfers of an attribute.

```
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/CompileConfig.hh"
#include "TrickHLA/Item.hh"
```

Include dependency graph for OwnershipItem.hh:



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



## Data Structures

- class [TrickHLA::OwnershipItem](#)

## Namespaces

- [TrickHLA](#)

### 8.74.1 Detailed Description

This class represents a queue item for holding ownership transfers of an attribute.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Item.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3	TS21	July 2009	–	Checkpoint / restart of <a href="#">TrickHLA</a> .
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

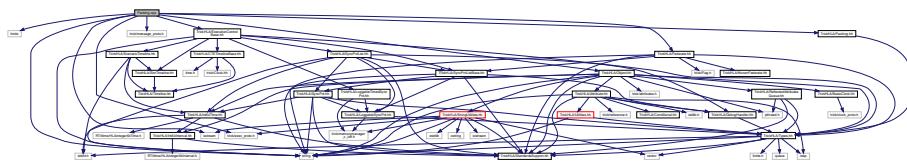
#### Revision History

## 8.75 Packing.cpp File Reference

This class is the abstract base class for Trick HLA Packing class.

```
#include <limits>
#include <sstream>
#include <string>
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "TrickHLA/ExecutionControlBase.hh"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Object.hh"
#include "TrickHLA/Packing.hh"
```

Include dependency graph for Packing.cpp:



### 8.75.1 Detailed Description

This class is the abstract base class for Trick HLA Packing class.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Attribute.cpp](#)
- [Object.cpp](#)
- [ExecutionControlBase.cpp](#)
- [Packing.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	Sept 2006	–	Initial version.
Dan Dexter	NASA/ER7	IMSim	Sept 2009	–	Updated Packing API.
Dan Dexter	NASA/ER7	IMSim	Oct 2009	–	Added get attribute function.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

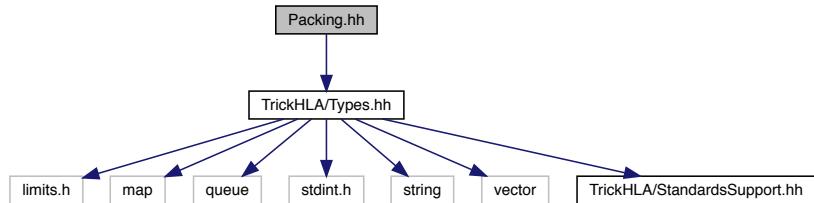
## Revision History

## 8.76 Packing.hh File Reference

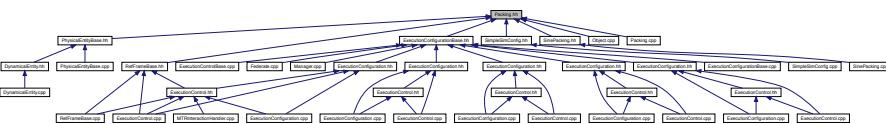
## Definition of the TrickHLA Packing class.

```
#include "TrickHLA/Types.hh"
```

Include dependency graph for Packing.hh:



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



## Data Structures

- class `TrickHLA::Packing`

## Namespaces

- TrickHLA

### 8.76.1 Detailed Description

## Definition of the TrickHLA Packing class.

This class is the abstract base class for packing before data is sent to the RTI and unpacking just after data is received from the RTI.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.TrickHLA*

### Link Dependencies

- `../source/TrickHLA/Attribute.cpp`
- `../source/TrickHLA/Object.cpp`
- `../source/TrickHLA/Packing.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	Sept 2006	–	Initial version.
Dan Dexter	NASA/ER7	IMSim	Sept 2009	–	Updated Packing API.
Dan Dexter	NASA/ER7	IMSim	Oct 2009	–	Added get attribute function.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

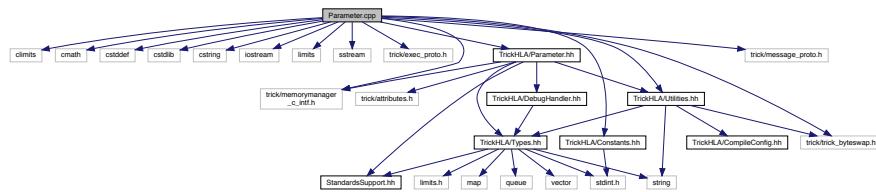
## Revision History

## 8.77 Parameter.cpp File Reference

This class represents the HLA parameters of an interaction that is managed by Trick.

```
#include <climits>
#include <cmath>
#include <cstddef>
#include <cstdlib>
#include <cstring>
#include <iostream>
#include <limits>
#include <sstream>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "trick/trick_bytesswap.h"
#include "TrickHLA/Constants.hh"
#include "TrickHLA/Parameter.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for Parameter.cpp:



### 8.77.1 Detailed Description

This class represents the HLA parameters of an interaction that is managed by Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Utilities.cpp](#)
- [Parameter.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	Aug 2006	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

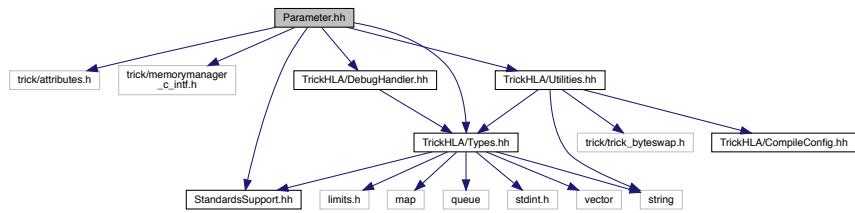
#### Revision History

## 8.78 Parameter.hh File Reference

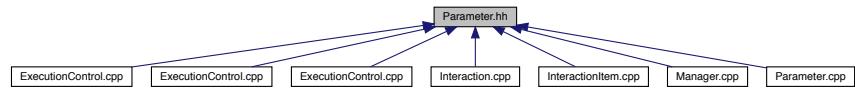
This class represents the HLA parameters of an interaction that is managed by Trick.

```
#include "trick/attributes.h"
#include "trick/memorymanager_c_intf.h"
#include "StandardSupport.hh"
#include "TrickHLA/DebugHandler.hh"
#include "TrickHLA/Types.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for Parameter.hh:



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



## Data Structures

- class [TrickHLA::Parameter](#)

## Namespaces

- [TrickHLA](#)

### 8.78.1 Detailed Description

This class represents the HLA parameters of an interaction that is managed by Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Utilities.cpp`
- `../source/TrickHLA/Parameter.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	Aug 2006	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

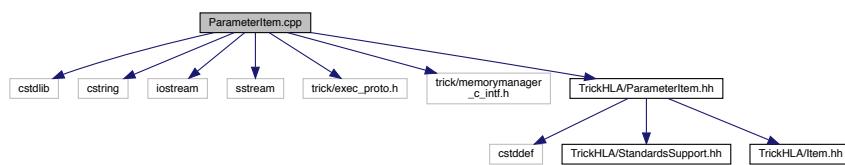
#### Revision History

## 8.79 ParameterItem.cpp File Reference

This class represents a queue for holding HLA parameters.

```
#include <cstdlib>
#include <cstring>
#include <iostream>
#include <sstream>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "TrickHLA/ParameterItem.hh"
```

Include dependency graph for ParameterItem.cpp:



### 8.79.1 Detailed Description

This class represents a queue for holding HLA parameters.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [ParameterItem.cpp](#)
- [Item.cpp](#)

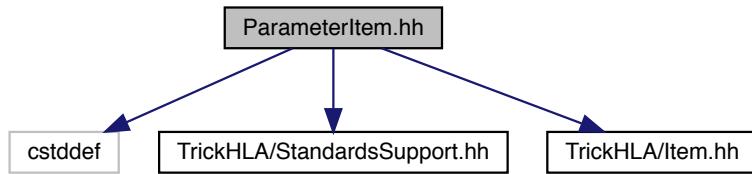
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	TrickHLA	Feb 2009	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

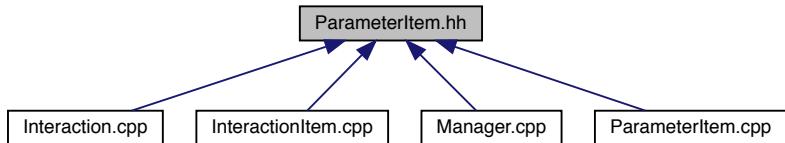
## 8.80 ParameterItem.hh File Reference

This class represents a queue for holding HLA parameters.

```
#include <cstddef>
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Item.hh"
Include dependency graph for ParameterItem.hh:
```



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



## Data Structures

- class [TrickHLA::ParameterItem](#)

## Namespaces

- [TrickHLA](#)

### 8.80.1 Detailed Description

This class represents a queue for holding HLA parameters.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.TrickHLA*

## Link Dependencies

- `./source/TrickHLA/ParameterItem.cpp`
  - `./source/TrickHLA/Item.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	<a href="#">TrickHLA</a>	Feb 2009	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

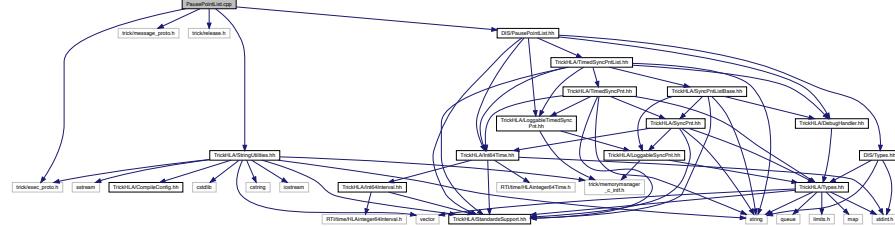
## Revision History

## 8.81 PausePointList.cpp File Reference

Represents an HLA Synchronization Point in Trick.

```
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "trick/release.h"
#include "TrickHLA/StringUtilities.hh"
#include "DIS/PausePointList.hh"
Include dependency graph for DIS/PausePointList.cpp:
```

Include dependency graph for DIS/PausePointList.cpp.



### 8.81.1 Detailed Description

Represents an HLA Synchronization Point in Trick.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.DIS*

## Link Dependencies

- Int64Time.cpp
  - PausePointList.cpp

Author	Organization	Project	Date	Rev. ID	Description
Robert G. Phillips	Titan Systems Corp.	DIS	Oct 2004	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

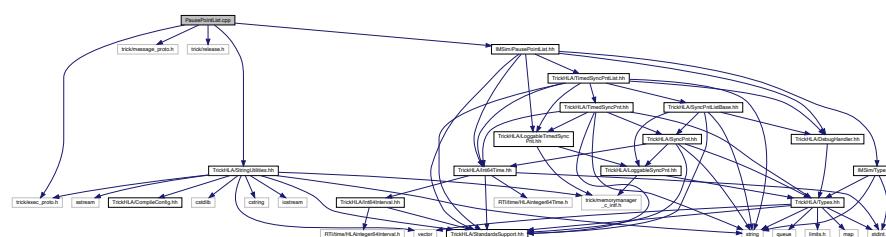
## Revision History

## 8.82 PausePointList.cpp File Reference

Represents an HLA Synchronization Point in Trick.

```
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "trick/release.h"
#include "TrickHLA/StringUtilities.hh"
#include "IMSim/PausePointList.hh"
Include dependency graph for IMSim/PausePointList.cpp:
```

Include dependency graph for IMSim/PausePointList.cpp:



### 8.82.1 Detailed Description

Represents an HLA Synchronization Point in Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.IMSim*

#### Link Dependencies

- [Int64Time.cpp](#)
- [PausePointList.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Robert G. Phillips	Titan Systems Corp.	<a href="#">DIS</a>	Oct 2004	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

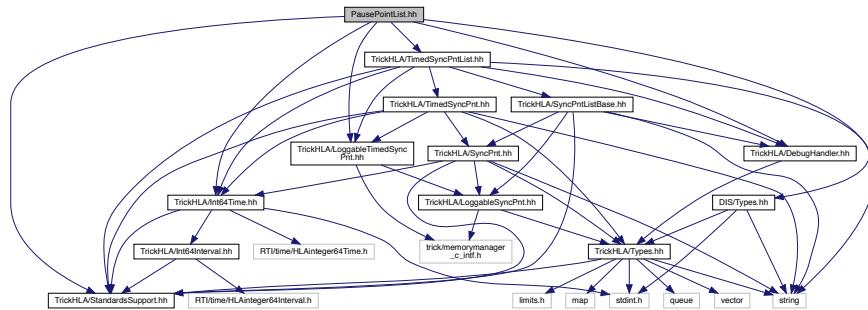
#### Revision History

## 8.83 PausePointList.hh File Reference

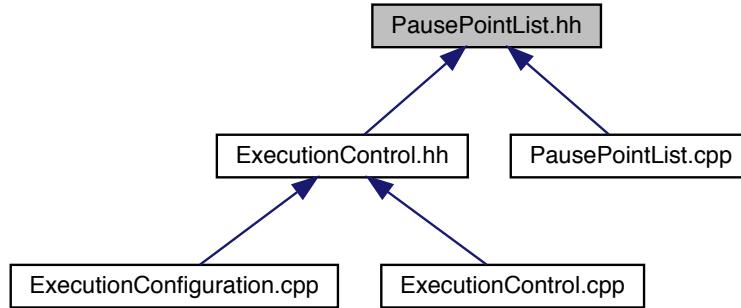
This class provides a mechanism for storing and managing HLA synchronization points for Trick.

```
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/DebugHandler.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/LoggableTimedSyncPnt.hh"
#include "TrickHLA/TimedSyncPntList.hh"
#include "DIS/Types.hh"
```

Include dependency graph for DIS/PausePointList.hh:



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



## Data Structures

- class [DIS::PausePointList](#)

## Namespaces

- [DIS](#)

### 8.83.1 Detailed Description

This class provides a mechanism for storing and managing HLA synchronization points for Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

Python Module: *trick.DIS*

## Link Dependencies

- ..../source/TrickHLA/Int64Time.cpp
- ..../source/DIS/PausePointList.cpp

Author	Organization	Project	Date	Rev. ID	Description
Robert G. Phillips	Titan Systems Corp.	DIS	Oct 2004	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

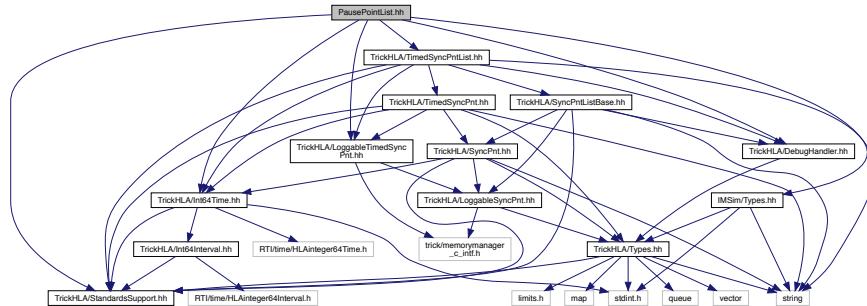
## Revision History

## 8.84 PausePointList.hh File Reference

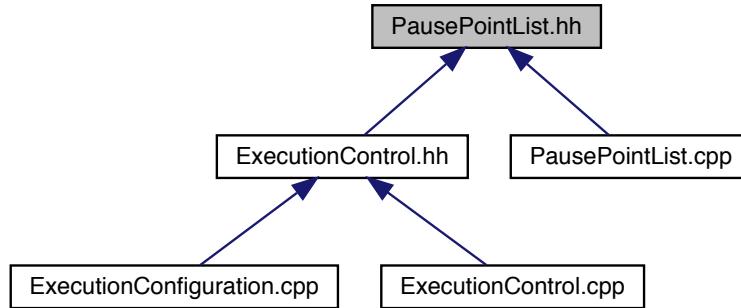
This class provides a mechanism for storing and managing HLA synchronization points for Trick.

```
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/DebugHandler.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/LoggableTimedSyncPnt.hh"
#include "TrickHLA/TimedSyncPntList.hh"
#include "IMSim/Types.hh"
```

Include dependency graph for IMSim/PausePointList.hh:



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



## Data Structures

- class [IMSim::PausePointList](#)

## Namespaces

- [IMSim](#)

### 8.84.1 Detailed Description

This class provides a mechanism for storing and managing HLA synchronization points for Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.IMSim`

#### Link Dependencies

- `../source/TrickHLA/Int64Time.cpp`
- `../source/IMSim/PausePointList.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Robert G. Phillips	Titan Systems Corp.	<a href="#">DIS</a>	Oct 2004	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

Generated by Doxygen

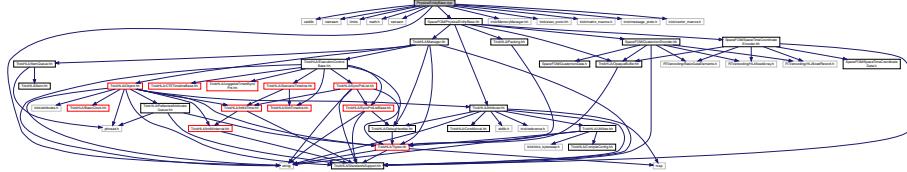
## Revision History

## 8.85 PhysicalEntityBase.cpp File Reference

This class provides data packing for the [SpaceFOM](#) Reference Frames.

```
#include <cstdlib>
#include <iostream>
#include <limits>
#include <math.h>
#include <sstream>
#include <string>
#include "trick/MemoryManager.hh"
#include "trick/exec_proto.hh"
#include "trick/matrix_macros.h"
#include "trick/message_proto.h"
#include "trick/vector_macros.h"
#include "SpaceFOM/PhysicalEntityBase.hh"
```

Include dependency graph for PhysicalEntityBase.cpp:



## Variables

- Trick::MemoryManager \* `trick_MM`

### 8.85.1 Detailed Description

This class provides data packing for the [SpaceFOM](#) Reference Frames.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- `../TrickHLA/Packing.cpp`
- `SpaceTimeCoordinateEncoder.cpp`

- [QuaternionEncoder.cpp](#)
- [PhysicalEntityBase.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	Sept 2006	–	Initial implementation.
Edwin Z. Crues	NASA ER7	SISO	Sept 2010	–	Smackdown implementation.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

### 8.85.2 Variable Documentation

#### 8.85.2.1 `trick_MM`

`Trick::MemoryManager* trick_MM`

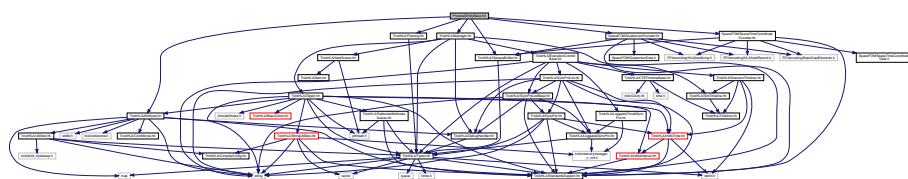
Referenced by `TrickHLA::Manager::clear_interactions()`, `TrickHLA::ExecutionConfiguration::configure_attributes()`, `DIS::ExecutionConfiguration::configure_attributes()`, `DSES::ExecutionConfiguration::configure_attributes()`, `IMSim::ExecutionConfiguration::configure_attributes()`, `SpaceFOM::ExecutionConfiguration::configure_attributes()`, `TrickHLA::ExecutionConfigurationBase::ExecutionConfigurationBase()`, `SpaceFOM::PhysicalEntityBase::initialize()`, `TrickHLA::Manager::restore_interactions()`, `SpaceFOM::PhysicalEntityBase::set_name()`, `SpaceFOM::MTRInteractionHandler::set_name()`, `SpaceFOM::PhysicalEntityBase::set_parent_ref_frame()`, `TrickHLA::ExecutionConfigurationBase::set_S_define_name()`, `SpaceFOM::PhysicalEntityBase::set_status()`, `SpaceFOM::PhysicalEntityBase::set_type()`, `TrickHLA::Manager::setup_checkpoint_interactions()`, `SpaceFOM::ExecutionConfiguration::setup_ref_attributes()`, `TrickHLA::ExecutionConfigurationBase::~ExecutionConfigurationBase()`, `SpaceFOM::MTRInteractionHandler::~MTRInteractionHandler()`, and `SpaceFOM::PhysicalEntityBase::~PhysicalEntityBase()`.

## 8.86 PhysicalEntityBase.hh File Reference

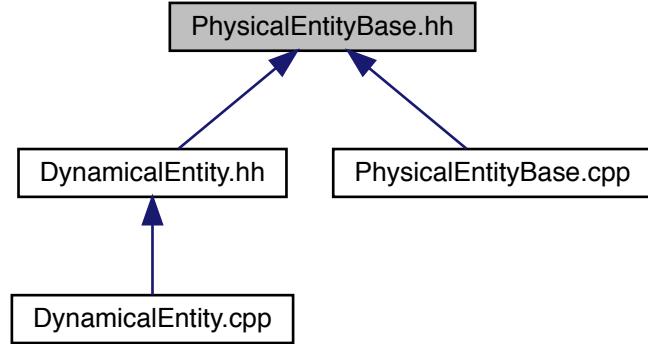
Definition of the [TrickHLA SpaceFOM](#) physical entity type.

```
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/OpaqueBuffer.hh"
#include "TrickHLA/Packing.hh"
#include "SpaceFOM/QuaternionEncoder.hh"
#include "SpaceFOM/SpaceTimeCoordinateEncoder.hh"
```

Include dependency graph for `PhysicalEntityBase.hh`:



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



## Data Structures

- class [SpaceFOM::PhysicalEntityBase](#)

## Namespaces

- [TrickHLA](#)
- [SpaceFOM](#)

### 8.86.1 Detailed Description

Definition of the [TrickHLA SpaceFOM](#) physical entity type.

This is the base implementation for the Space Reference FOM ([SpaceFOM](#)) interface to the Reference Frame object. This needs to be available to the [SpaceFOM](#) initialization process for the root reference frame discovery step in the initialization process.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.SpaceFOM*

#### Link Dependencies

- [../../source/TrickHLA/Packing.cpp](#)
- [../../source/SpaceFOM/PhysicalEntityBase.cpp](#)

- [..../source/SpaceFOM/SpaceTimeCoordinateEncoder.cpp](#)
- [..../source/SpaceFOM/QuaternionEncoder.cpp](#)

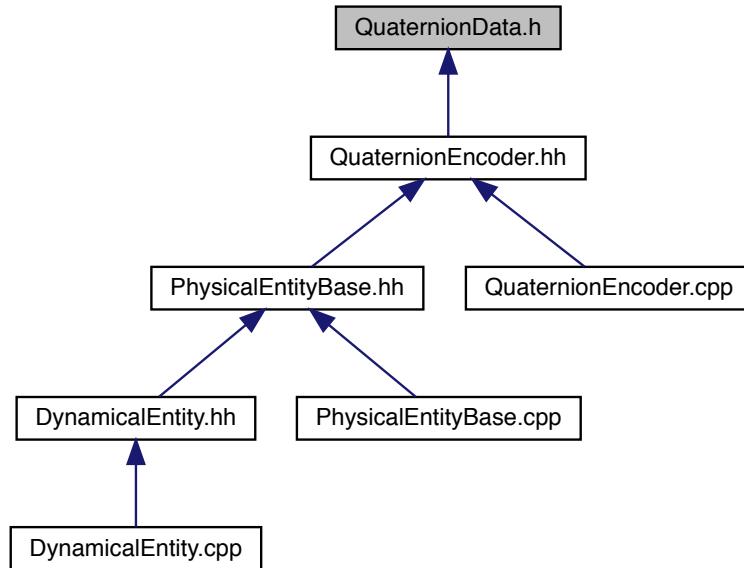
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.87 QuaternionData.h File Reference

A simple structure that contains the date fields required to encode and decode a SISO Space Reference FOM Quaternion data type.

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



## Data Structures

- struct [QuaternionData](#)

### 8.87.1 Detailed Description

A simple structure that contains the date fields required to encode and decode a SISO Space Reference FOM Quaternion data type.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

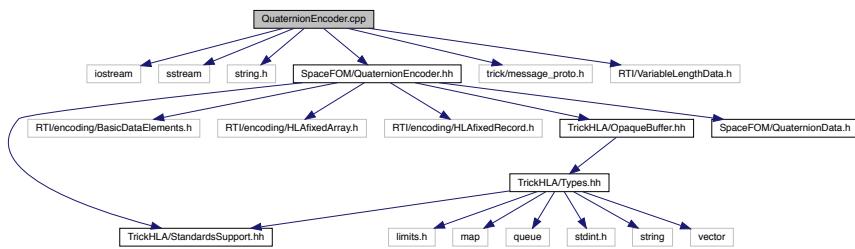
**Python Module: *trick.SpaceFOM***

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	NExSyS	July 2018	–	Initial version

## 8.88 QuaternionEncoder.cpp File Reference

This file contains the methods for the QuaternionEncoder class.

```
#include <iostream>
#include <sstream>
#include <string.h>
#include "SpaceFOM/QuaternionEncoder.hh"
#include "trick/message_proto.h"
#include "RTI/VariableLengthData.h"
Include dependency graph for QuaternionEncoder.cpp:
```



### 8.88.1 Detailed Description

This file contains the methods for the QuaternionEncoder class.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- QuaternionEncoder.cpp

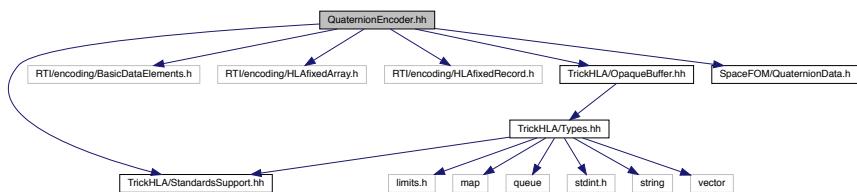
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	–	July 2018	NExSyS	Initial version

## 8.89 QuaternionEncoder.hh File Reference

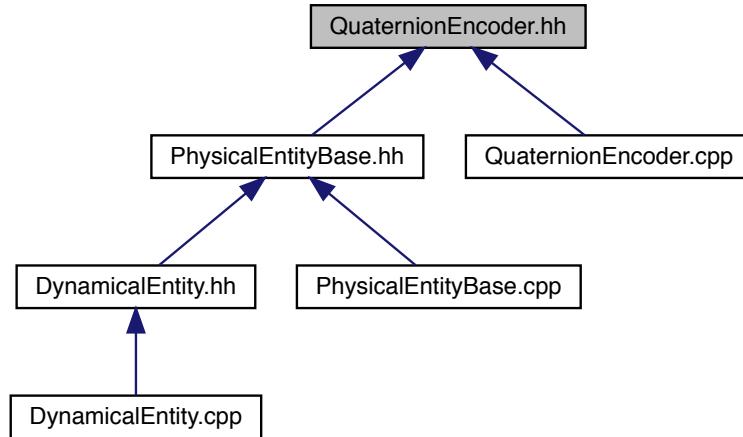
Definition of the [TrickHLA SpaceFOM](#) quaternion encoding utility.

```
#include "TrickHLA/StandardsSupport.hh"
#include "RTI/encoding/BasicDataElements.h"
#include "RTI/encoding/HLAfixedArray.h"
#include "RTI/encoding/HLAfixedRecord.h"
#include "TrickHLA/OpaqueBuffer.hh"
#include "SpaceFOM/QuaternionData.h"
```

Include dependency graph for QuaternionEncoder.hh:



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



## Data Structures

- class [SpaceFOM::QuaternionEncoder](#)

## Namespaces

- [SpaceFOM](#)

### 8.89.1 Detailed Description

Definition of the [TrickHLA SpaceFOM](#) quaternion encoding utility.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.SpaceFOM*

#### Link Dependencies

- [../../source/SpaceFOM/QuaternionEncoder.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

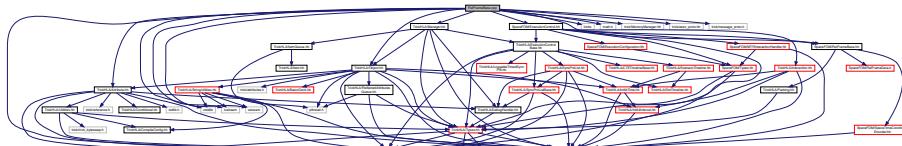
## 8.90 README.md File Reference

## 8.91 RefFrameBase.cpp File Reference

This class provides data packing for the [SpaceFOM](#) Reference Frames.

```
#include <cstdlib>
#include <iostream>
#include <limits>
#include <math.h>
#include <sstream>
#include <string>
#include "trick/MemoryManager.hh"
#include "trick/exec_proto.hh"
#include "trick/message_proto.h"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/Object.hh"
#include "SpaceFOM/ExecutionControl.hh"
#include "SpaceFOM/RefFrameBase.hh"
```

Include dependency graph for RefFrameBase.cpp:



## Macros

- `#define REF_FRAME_PACKING_DEBUG 0`
- `#define REF_FRAME_PACKING_EXTRA_DEBUG 0`

## Variables

- `Trick::MemoryManager * trick_MM`

### 8.91.1 Detailed Description

This class provides data packing for the [SpaceFOM](#) Reference Frames.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Link Dependencies**

- ../TrickHLA/Packing.cpp
- [SpaceTimeCoordinateEncoder.cpp](#)
- [RefFrameBase.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	Sept 2006	–	Initial implementation.
Edwin Z. Crues	NASA ER7	SISO	Sept 2010	–	Smackdown implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

**Revision History****8.91.2 Macro Definition Documentation****8.91.2.1 REF\_FRAME\_PACKING\_DEBUG**

```
#define REF_FRAME_PACKING_DEBUG 0
Definition at line 56 of file RefFrameBase.cpp.
```

**8.91.2.2 REF\_FRAME\_PACKING\_EXTRA\_DEBUG**

```
#define REF_FRAME_PACKING_EXTRA_DEBUG 0
Definition at line 57 of file RefFrameBase.cpp.
```

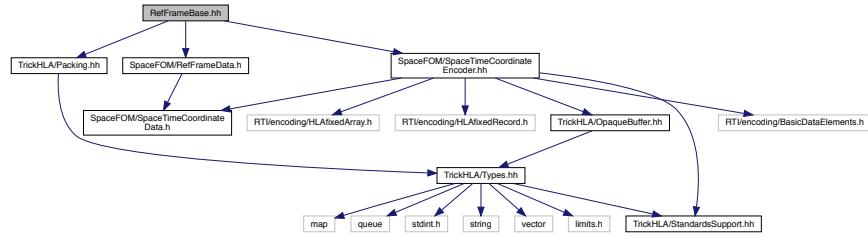
**8.91.3 Variable Documentation****8.91.3.1 trick\_MM**

```
Trick::MemoryManager* trick_MM
Referenced by SpaceFOM::RefFrameBase::default_data(), SpaceFOM::RefFrameBase::initialize(), SpaceFOM::RefFrameBase::set_name(), SpaceFOM::RefFrameBase::set_parent_name(), and SpaceFOM::RefFrameBase::~RefFrameBase().
```

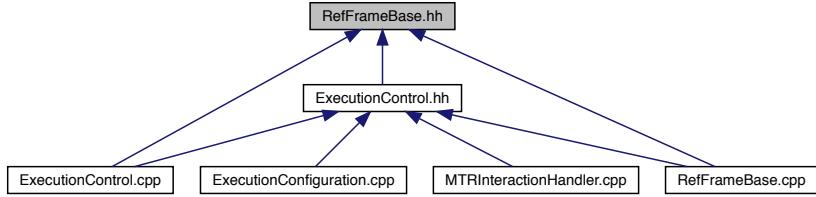
**8.92 RefFrameBase.hh File Reference**

This class provides data packing for the [SpaceFOM](#) Reference Frames.

```
#include "TrickHLA/Packing.hh"
#include "SpaceFOM/RefFrameData.h"
#include "SpaceFOM/SpaceTimeCoordinateEncoder.hh"
Include dependency graph for RefFrameBase.hh:
```



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



## Data Structures

- class [SpaceFOM::RefFrameBase](#)

## Namespaces

- [TrickHLA](#)
- [SpaceFOM](#)

### 8.92.1 Detailed Description

This class provides data packing for the [SpaceFOM](#) Reference Frames.

This is the base implementation for the Space Reference FOM ([SpaceFOM](#)) interface to the Reference Frame object. This needs to be available to the [SpaceFOM](#) initialization process for the root reference frame discovery step in the initialization process.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

**Python Module:** *trick.SpaceFOM*

#### Link Dependencies

- *../source/TrickHLA/Packing.cpp*
- *../source/SpaceFOM/SpaceTimeCoordinateEncoder.cpp*
- *../source/SpaceFOM/RefFrameBase.cpp*

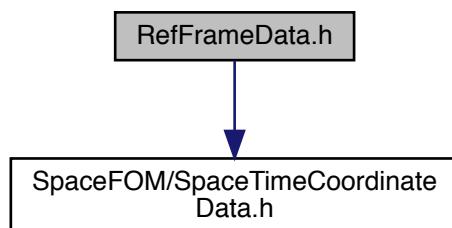
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

#### Revision History

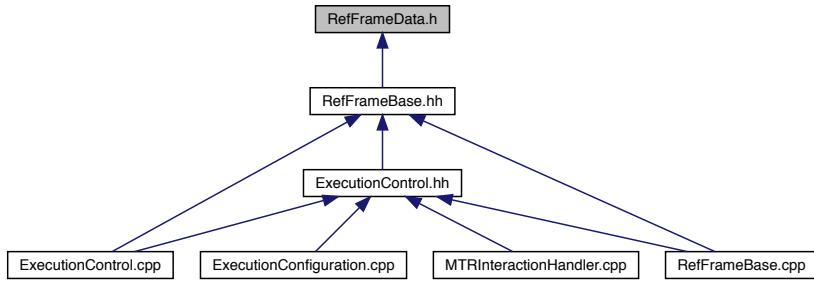
## 8.93 RefFrameData.h File Reference

A simple structure that contains the date fields required to encode and decode a SISO Space Reference FOM ReferenceFrame data type.

```
#include "SpaceFOM/SpaceTimeCoordinateData.h"  
Include dependency graph for RefFrameData.h:
```



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



## Data Structures

- struct [RefFrameData](#)

### 8.93.1 Detailed Description

A simple structure that contains the date fields required to encode and decode a SISO Space Reference FOM ReferenceFrame data type.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

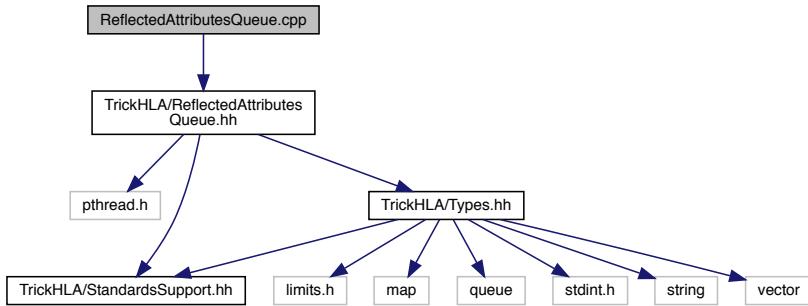
#### Python Module: *trick.SpaceFOM*

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	NExSyS	July 2018	–	Initial version

## 8.94 ReflectedAttributesQueue.cpp File Reference

This is a protected queue class to hold reflected attributes.

```
#include "TrickHLA/ReflectedAttributesQueue.hh"
Include dependency graph for ReflectedAttributesQueue.cpp:
```



### 8.94.1 Detailed Description

This is a protected queue class to hold reflected attributes.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [ReflectedAttributesQueue.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER6	TrickHLA	Feb 2019	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

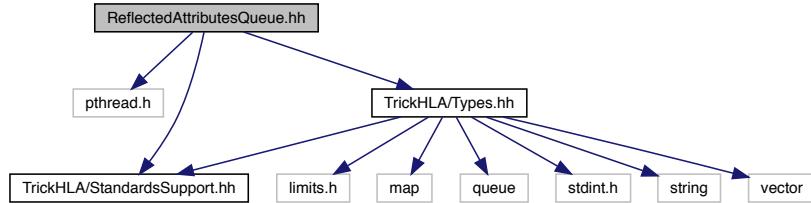
#### Revision History

## 8.95 ReflectedAttributesQueue.hh File Reference

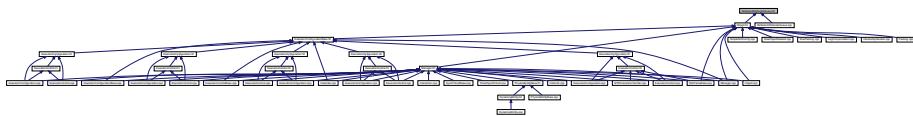
[TrickHLA](#) Queue of AttributeHandleValueMap.

```
#include <pthread.h>
#include "TrickHLA/StandardsSupport.hh"
```

```
#include "TrickHLA/Types.hh"
Include dependency graph for ReflectedAttributesQueue.hh:
```



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



## Data Structures

- class [TrickHLA::ReflectedAttributesQueue](#)

## Namespaces

- [TrickHLA](#)

### 8.95.1 Detailed Description

[TrickHLA](#) Queue of AttributeHandleValueMap.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/ReflectedAttributesQueue.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER6	TrickHLA	Feb 2019	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

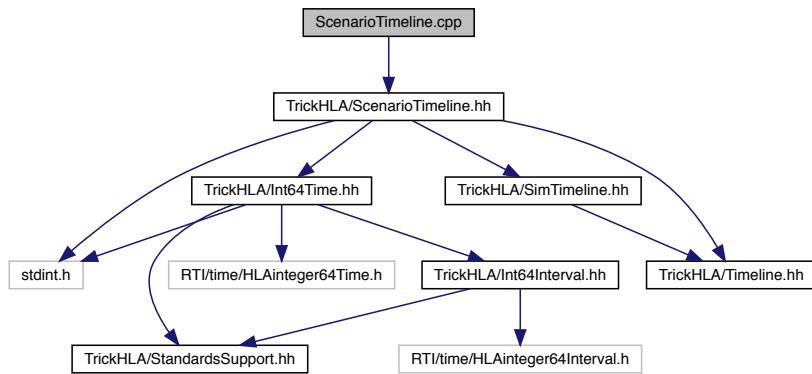
#### Revision History

## 8.96 ScenarioTimeline.cpp File Reference

This class represents the simulation timeline.

```
#include "TrickHLA/ScenarioTimeline.hh"
```

Include dependency graph for ScenarioTimeline.cpp:



### 8.96.1 Detailed Description

This class represents the simulation timeline.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Timeline.cpp](#)
- [ScenarioTimeline.cpp](#)

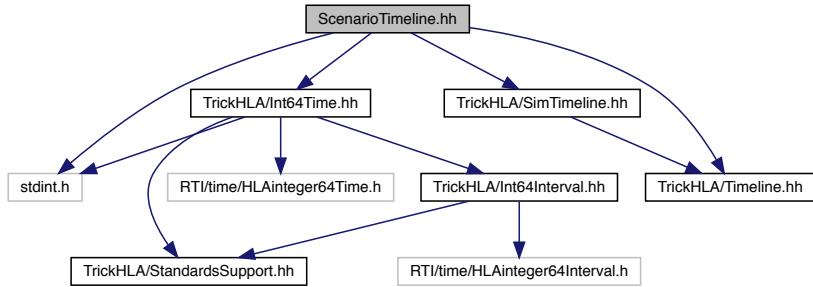
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	January 2019	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

### Revision History

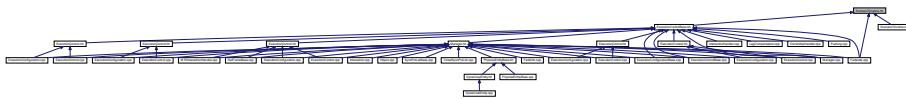
## 8.97 ScenarioTimeline.hh File Reference

This class represents the scenario timeline.

```
#include <stdint.h>
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/SimTimeline.hh"
#include "TrickHLA/Timeline.hh"
Include dependency graph for ScenarioTimeline.hh:
```



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



## Data Structures

- class [TrickHLA::ScenarioTimeline](#)

## Namespaces

- [TrickHLA](#)

### 8.97.1 Detailed Description

This class represents the scenario timeline.

**Assumptions and Limitations:**

- Instances of this class represent the timeline for the scenario associated with the problem.
- The time scale for this timeline is always Terrestrial Time (TT) which complies with the Space Reference FOM standard.
- Note that the epoch value for this CTE timeline represents the epoch or starting point of the CTE timeline. This will correspond to the starting time in the TT time standard represented in Truncated Julian Date format (TJD) expressed in seconds.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module: *trick.TrickHLA*****Link Dependencies**

- `../source/TrickHLA/Timeline.cpp`
- `../source/TrickHLA/ScenarioTimeline.cpp`

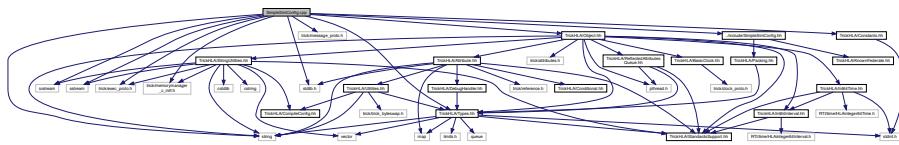
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	January 2019	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

**Revision History****8.98 SimpleSimConfig.cpp File Reference**

This class contains a basic simulation configuration.

```
#include <iostream>
#include <sstream>
#include <stdlib.h>
#include <string>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "TrickHLA/Object.hh"
#include "TrickHLA/Types.hh"
#include "TrickHLA/Constants.hh"
#include "../include/SimpleSimConfig.hh"
```

Include dependency graph for SimpleSimConfig.cpp:



### 8.98.1 Detailed Description

This class contains a basic simulation configuration.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLAModel*

#### Link Dependencies

- ..../source/TrickHLA/Packing.o
- simconfig/src/SimpleSimConfig.o

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	June 2007	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

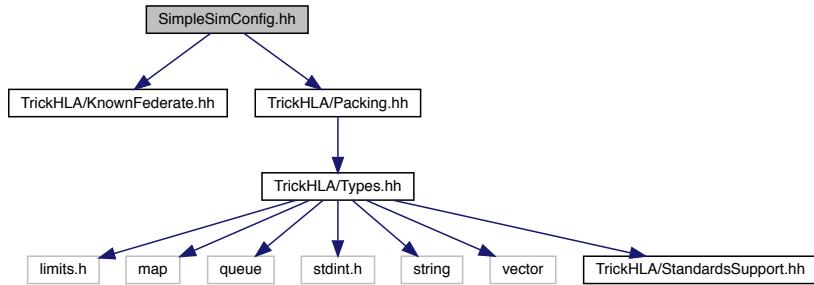
#### Revision History

## 8.99 SimpleSimConfig.hh File Reference

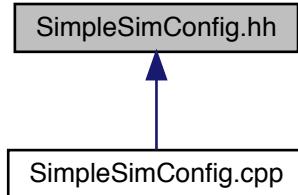
This class contains a basic simulation configuration.

```
#include "TrickHLA/KnownFederate.hh"
#include "TrickHLA/Packing.hh"
```

Include dependency graph for SimpleSimConfig.hh:



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



## Data Structures

- class [TrickHLAModel::SimpleSimConfig](#)

## Namespaces

- [TrickHLAModel](#)

### 8.99.1 Detailed Description

This class contains a basic simulation configuration.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Python Module: `trick.TrickHLAModel`

#### Link Dependencies

- `../source/TrickHLA/Packing.o`
- `simconfig/src/SimpleSimConfig.o`

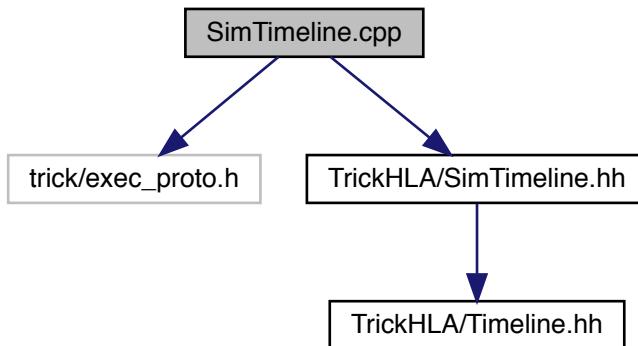
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	June 2007	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

#### Revision History

## 8.100 SimTimeline.cpp File Reference

This class represents the simulation timeline.

```
#include "trick/exec_proto.h"
#include "TrickHLA/SimTimeline.hh"
Include dependency graph for SimTimeline.cpp:
```



### 8.100.1 Detailed Description

This class represents the simulation timeline.

### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

### Link Dependencies

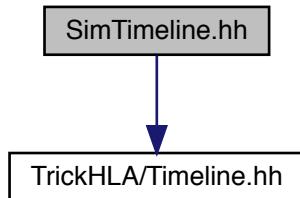
- [Timeline.cpp](#)
- [SimTimeline.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	NExSyS	April 2016	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

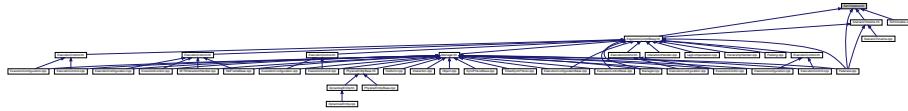
### Revision History

## 8.101 SimTimeline.hh File Reference

```
#include "TrickHLA/Timeline.hh"
Include dependency graph for SimTimeline.hh:
```



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



## Data Structures

- class [TrickHLA::SimTimeline](#)

## Namespaces

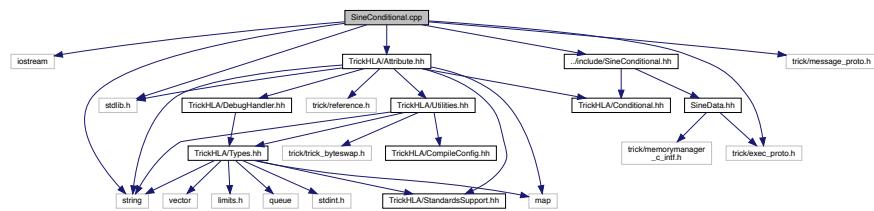
- [TrickHLA](#)

## 8.102 SineConditional.cpp File Reference

Subclass the base class to provide sine wave-specific CONDITIONAL attribute.

```
#include <iostream>
#include <stdlib.h>
#include <string>
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "TrickHLA/Attribute.hh"
#include "../include/SineConditional.hh"
```

Include dependency graph for SineConditional.cpp:



### 8.102.1 Detailed Description

Subclass the base class to provide sine wave-specific CONDITIONAL attribute.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Link Dependencies

- sine/src/SineConditional.o
- sine/src/SineData.o

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3	DSES	October 2009	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

### Revision History

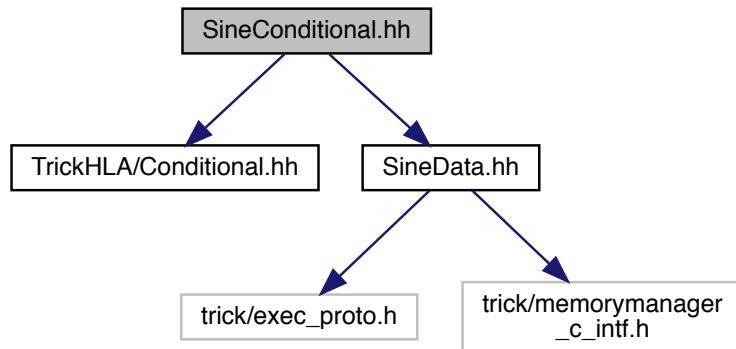
## 8.103 SineConditional.hh File Reference

Subclass the base class to provide sine wave-specific CONDITIONAL attribute.

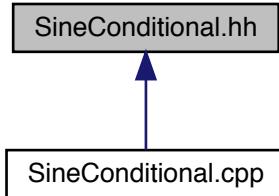
```
#include "TrickHLA/Conditional.hh"
```

```
#include "SineData.hh"
```

Include dependency graph for SineConditional.hh:



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



## Data Structures

- class [TrickHLAModel::SineConditional](#)

## Namespaces

- [TrickHLAModel](#)

### 8.103.1 Detailed Description

Subclass the base class to provide sine wave-specific CONDITIONAL attribute.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLAModel*

#### Link Dependencies

- sine/src/SineConditional.o
- sine/src/SineData.o
- ../../source/TrickHLA/Conditional.o
- ../../source/TrickHLA/Attribute.o

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3	<a href="#">TrickHLA</a>	Oct 2009	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2020	–	Version 3 rewrite.

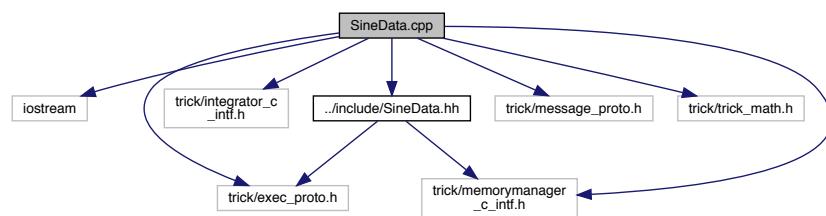
## Revision History

## 8.104 SineData.cpp File Reference

This class is the working class for the Sine HLA/RTI example problem.

```
#include <iostream>
#include "trick/exec_proto.h"
#include "trick/integrator_c_intf.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "trick/trick_math.h"
#include "../include/SineData.hh"

Include dependency graph for SineData.cpp:
```



### 8.104.1 Detailed Description

This class is the working class for the Sine HLA/RTI example problem.

## Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- sine/src/SineData.o
- sine/src/SinePacking.o

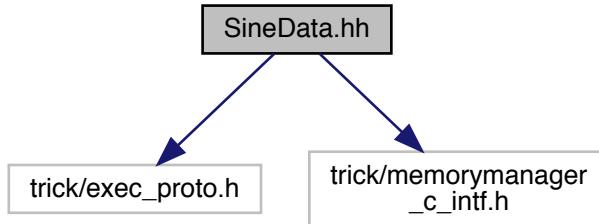
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	Titan-AEU	DSES	January 2003	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

## Revision History

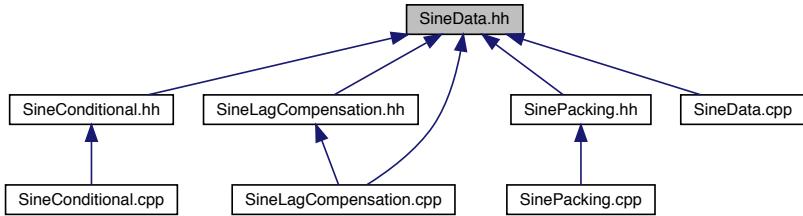
## 8.105 SineData.hh File Reference

This is a container class for general test data used in the general HLA test routines.

```
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
Include dependency graph for SineData.hh:
```



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



### Data Structures

- class [TrickHLAModel::SineData](#)

### Namespaces

- [TrickHLAModel](#)

#### 8.105.1 Detailed Description

This is a container class for general test data used in the general HLA test routines.

**Copyright**

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module: *trick.TrickHLAModel*****Link Dependencies**

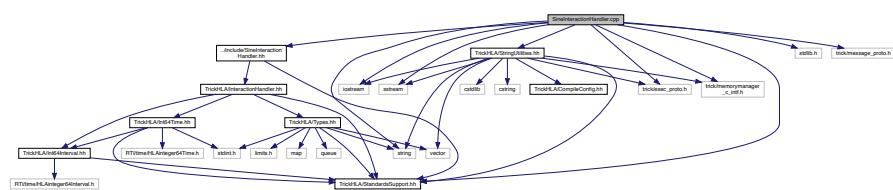
- sine/src/SineData.o
- sine/src/SinePacking.o

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	Titan-AEU	DSES	January 2003	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

**Revision History****8.106 SineInteractionHandler.cpp File Reference**

This class handles the HLA interactions for the sine wave simulation.

```
#include <iostream>
#include <sstream>
#include <stdlib.h>
#include <string>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "../include/SineInteractionHandler.hh"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/StringUtilities.hh"
Include dependency graph for SineInteractionHandler.cpp:
```



## Macros

- `#define SINE_SEND_INTERACTION_TSO 1`

### 8.106.1 Detailed Description

This class handles the HLA interactions for the sine wave simulation.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

**Python Module:** `trick.TrickHLAModel`

#### Link Dependencies

- `sine/src/SineInteractionHandler.o`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	Aug 2006	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

#### Revision History

### 8.106.2 Macro Definition Documentation

#### 8.106.2.1 SINE\_SEND\_INTERACTION\_TSO

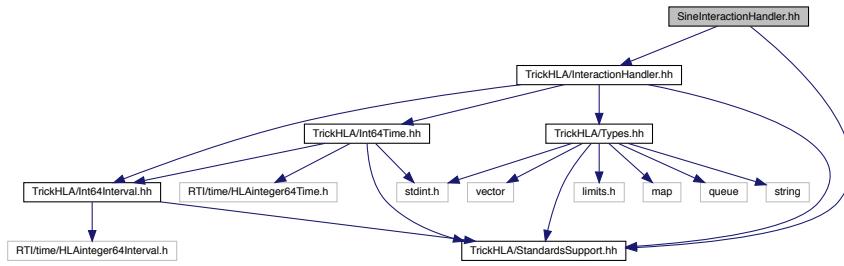
```
#define SINE_SEND_INTERACTION_TSO 1
Definition at line 53 of file SineInteractionHandler.cpp.
```

## 8.107 SineInteractionHandler.hh File Reference

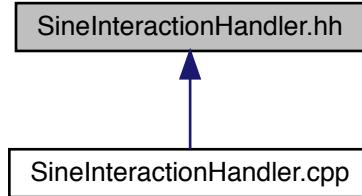
This class handles the HLA interactions for the sine wave simulation.

```
#include "TrickHLA/InteractionHandler.hh"
#include "TrickHLA/StandardsSupport.hh"
```

Include dependency graph for SineInteractionHandler.hh:



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



## Data Structures

- class [TrickHLAModel::SineInteractionHandler](#)

## Namespaces

- [TrickHLAModel](#)

## Macros

- `#define SINE_MSG_SIZE 2`

### 8.107.1 Detailed Description

This class handles the HLA interactions for the sine wave simulation.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.TrickHLAModel*

### Link Dependencies

- sine/src/SineInteractionHandler.o

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	Aug 2006	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

### Revision History

## 8.107.2 Macro Definition Documentation

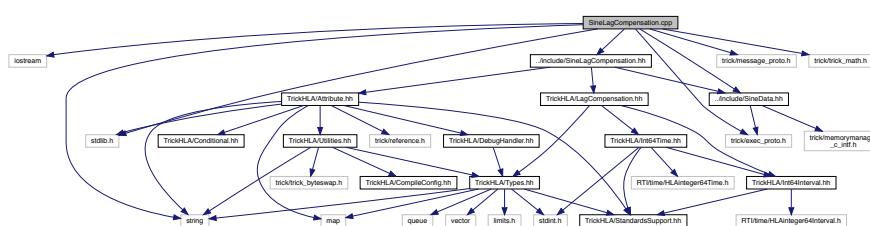
### 8.107.2.1 SINE\_MSG\_SIZE

```
#define SINE_MSG_SIZE 2
Definition at line 40 of file SineInteractionHandler.hh.
```

## 8.108 SineLagCompensation.cpp File Reference

This class provides lag compensation for the sine wave object.

```
#include <iostream>
#include <stdlib.h>
#include <string>
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "trick/trick_math.h"
#include "../include/SineData.hh"
#include "../include/SineLagCompensation.hh"
Include dependency graph for SineLagCompensation.cpp:
```



### 8.108.1 Detailed Description

This class provides lag compensation for the sine wave object.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- ..../source/TrickHLA/LagCompensation.o
- sine/src/SineLagCompensation.o

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	June 2006	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

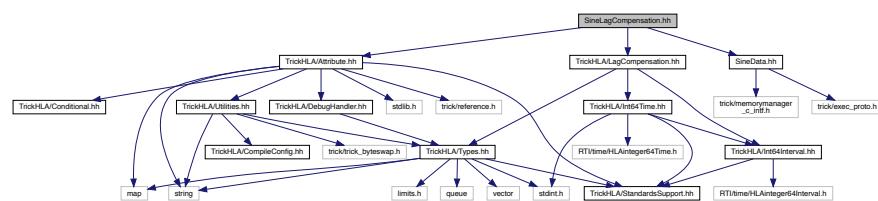
#### Revision History

## 8.109 SineLagCompensation.hh File Reference

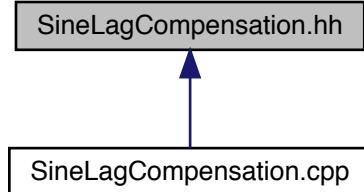
Send and receiving side lag compensation.

```
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/LagCompensation.hh"
#include "SineData.hh"

Include dependency graph for SineLagCompensation.hh:
```



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



## Data Structures

- class [TrickHLAModel::SineLagCompensation](#)

## Namespaces

- [TrickHLA](#)
- [TrickHLAModel](#)

### 8.109.1 Detailed Description

Send and receiving side lag compensation.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLAModel`

#### Link Dependencies

- `sine/src/SineLagCompensation.o`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	June 2006	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2020	–	Version 3 rewrite.

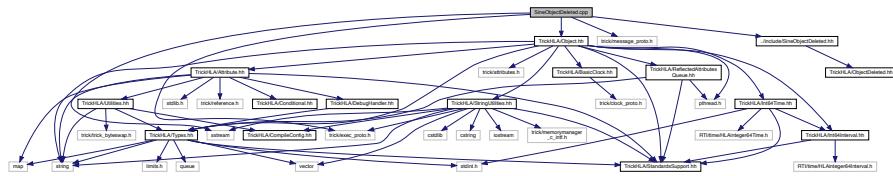
## Revision History

## 8.110 SineObjectDeleted.cpp File Reference

Callback class the user writes to do something once the object has been deleted from the RTI.

```
#include <sstream>
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "TrickHLA/Object.hh"
#include "../include/SineObjectDeleted.hh"
Include dependency graph for SineObjectDeleted.cpp:
```

Include dependency graph for SineObjectDeleted.cpp:



### 8.110.1 Detailed Description

Callback class the user writes to do something once the object has been deleted from the RTI.

## Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- sine/src/SineObjectDeleted.o

Author	Organization	Project	Date	Rev. ID	Description
Tony Varesic	L3-Communications	DSES	June 2008	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

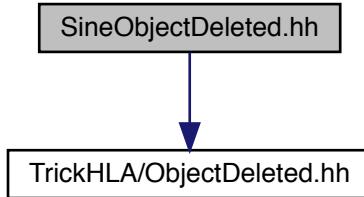
## Revision History

## 8.111 SineObjectDeleted.hh File Reference

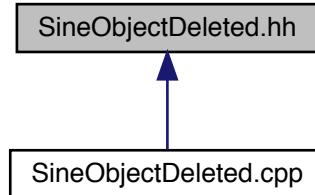
Callback class the user writes to do something once the object has been deleted from the RTI.

```
#include "TrickHLA/ObjectDeleted.hh"
```

Include dependency graph for SineObjectDeleted.hh:



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



### Data Structures

- class [TrickHLAModel::SineObjectDeleted](#)

### Namespaces

- [TrickHLA](#)
- [TrickHLAModel](#)

#### 8.111.1 Detailed Description

Callback class the user writes to do something once the object has been deleted from the RTI.

### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Python Module: *trick.TrickHLAModel*

#### Link Dependencies

- sine/src/SineObjectDeleted.o

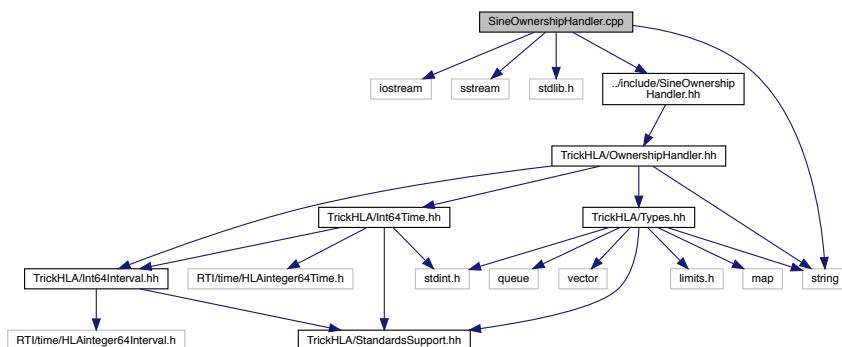
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	June 2008	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

### Revision History

## 8.112 SineOwnershipHandler.cpp File Reference

This class handles the HLA ownership transfer for the sine wave simulation.

```
#include <iostream>
#include <sstream>
#include <stdlib.h>
#include <string>
#include "../include/SineOwnershipHandler.hh"
Include dependency graph for SineOwnershipHandler.cpp:
```



### 8.112.1 Detailed Description

This class handles the HLA ownership transfer for the sine wave simulation.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- ..../source/TrickHLA/OwnershipHandler.o
- sine/src/SineOwnershipHandler.o

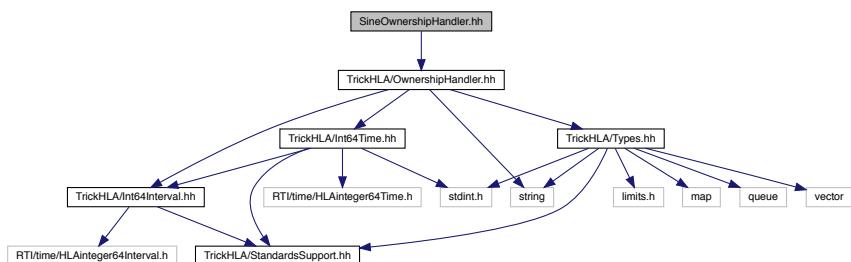
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	Sept 2006	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

#### Revision History

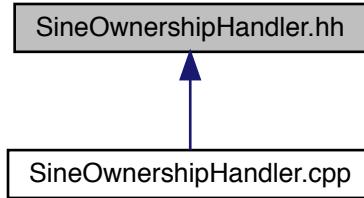
### 8.113 SineOwnershipHandler.hh File Reference

Ownership transfer for the HLA object attributes.

```
#include "TrickHLA/OwnershipHandler.hh"
Include dependency graph for SineOwnershipHandler.hh:
```



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



## Data Structures

- class [TrickHLAModel::SineOwnershipHandler](#)

## Namespaces

- [TrickHLA](#)
- [TrickHLAModel](#)

### 8.113.1 Detailed Description

Ownership transfer for the HLA object attributes.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLAModel*

#### Link Dependencies

- sine/src/SineOwnershipHandler.o

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	August 2006	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2020	–	Version 3 rewrite.

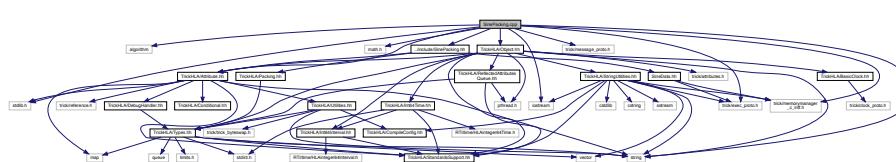
## Revision History

## 8.114 SinePacking.cpp File Reference

This class provides data packing for the sine wave data.

```
#include <algorithm>
#include <iostream>
#include <math.h>
#include <stdlib.h>
#include <string>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "trick/message_proto.h"
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Object.hh"
#include "../include/SinePacking.hh"
```

Include dependency graph for SinePacking.cpp:



### 8.114.1 Detailed Description

This class provides data packing for the sine wave data.

#### Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- ..../source/TrickHLA/Packing.o
- sine/src/SinePacking.o

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	DSES	Sept 2006	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

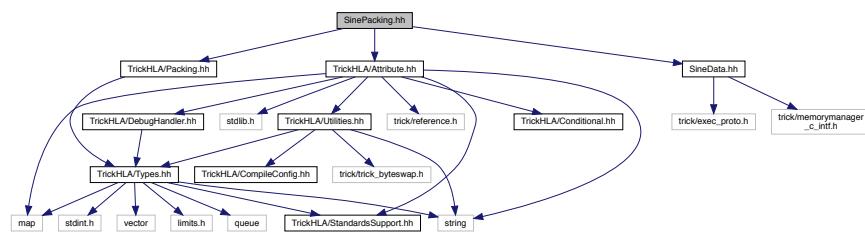
## Revision History

## 8.115 SinePacking.hh File Reference

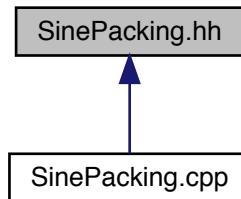
This class provides data packing.

```
#include "TrickHLA/Attribute.hh"
#include "TrickHLA/Packing.hh"
#include "SineData.hh"
```

Include dependency graph for SinePacking.hh:



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



### Data Structures

- class [TrickHLAModel::SinePacking](#)

### Namespaces

- [TrickHLA](#)
- [TrickHLAModel](#)

#### 8.115.1 Detailed Description

This class provides data packing.

## Copyright

Copyright 2020 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.TrickHLAModel*

### Link Dependencies

- sine/src/SinePacking.o

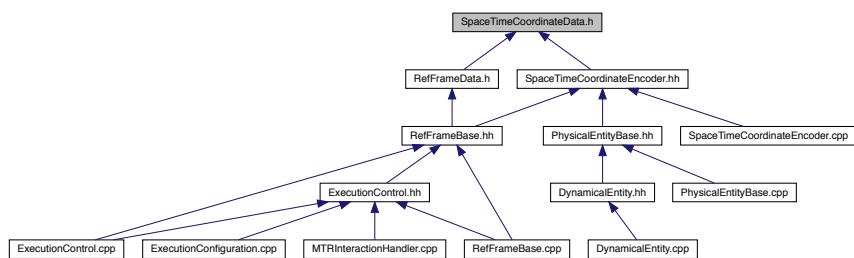
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	Sept 2009	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2020	–	Version 3 rewrite.

## Revision History

## 8.116 SpaceTimeCoordinateData.h File Reference

A simple structure that contains the date fields required to encode and decode a SISO Space Reference FOM Space/← Time Coordinate data type.

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



## Data Structures

- struct [SpaceTimeCoordinateData](#)

### 8.116.1 Detailed Description

A simple structure that contains the date fields required to encode and decode a SISO Space Reference FOM Space/← Time Coordinate data type.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

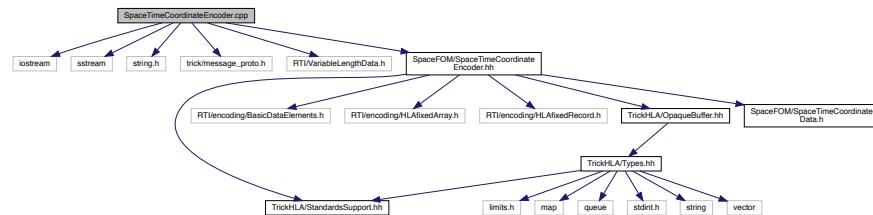
**Python Module:** *trick.SpaceFOM*

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	NExSyS	Jan 2019	-	<a href="#">SpaceFOM</a> support and testing.

## 8.117 SpaceTimeCoordinateEncoder.cpp File Reference

This file contains the methods for the SpaceTimeCoordinate encoder class.

```
#include <iostream>
#include <sstream>
#include <string.h>
#include "trick/message_proto.h"
#include "RTI/VariableLengthData.h"
#include "SpaceFOM/SpaceTimeCoordinateEncoder.hh"
Include dependency graph for SpaceTimeCoordinateEncoder.cpp:
```



### 8.117.1 Detailed Description

This file contains the methods for the SpaceTimeCoordinate encoder class.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- [SpaceTimeCoordinateEncoder.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	–	May 2016	NExSyS	Initial version

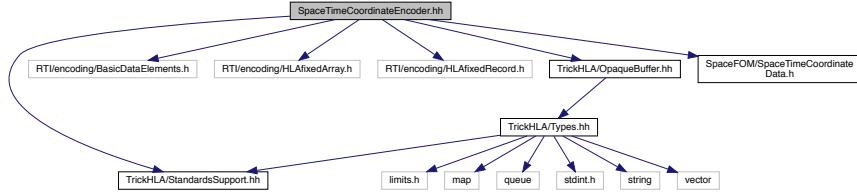
## Revision History

## 8.118 SpaceTimeCoordinateEncoder.hh File Reference

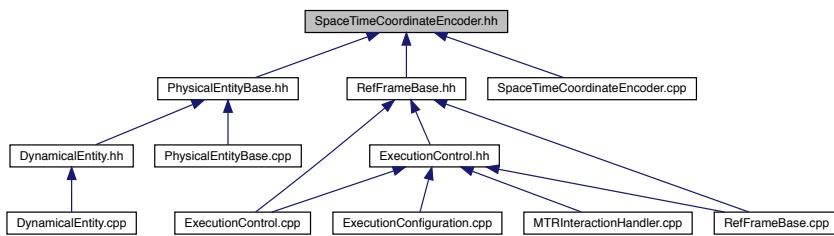
Definition of the [TrickHLA SpaceFOM](#) Space/Time coordinate encoder.

```
#include "TrickHLA/StandardsSupport.hh"
#include "RTI/encoding/BasicDataElements.h"
#include "RTI/encoding/HLAfixedArray.h"
#include "RTI/encoding/HLAfixedRecord.h"
#include "TrickHLA/OpaqueBuffer.hh"
#include "SpaceFOM/SpaceTimeCoordinateData.h"
```

Include dependency graph for SpaceTimeCoordinateEncoder.hh:



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



## Data Structures

- class [SpaceFOM::SpaceTimeCoordinateEncoder](#)

## Namespaces

- [SpaceFOM](#)

### 8.118.1 Detailed Description

Definition of the [TrickHLA SpaceFOM](#) Space/Time coordinate encoder.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.SpaceFOM*

#### Link Dependencies

- ../../source/SpaceFOM/SpaceTimeCoordinateEncoder.cpp

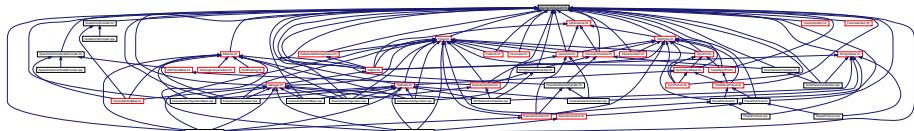
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.119 StandardsSupport.hh File Reference

This header file provides the [TrickHLA](#) support necessary to hide the differences between the different HLA Standards that implement the Runtime Infrastructure (RTI).

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



#### Macros

- #define IEEE\_1516\_2010
- #define RTI1516\_HEADER "RTI/RTI1516.h"
- #define RTI1516\_NAMESPACE rti1516e
- #define RTI1516\_USERDATA rti1516e::VariableLengthData
- #define RTI1516\_EXCEPTION rti1516e::Exception

### 8.119.1 Detailed Description

This header file provides the [TrickHLA](#) support necessary to hide the differences between the different HLA Standards that implement the Runtime Infrastructure (RTI).

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.TrickHLA*

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA/ER7	<a href="#">TrickHLA</a>	February 2009	–	HLA Standards Support
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

## Revision History

### 8.119.2 Macro Definition Documentation

#### 8.119.2.1 IEEE\_1516\_2010

```
#define IEEE_1516_2010
Definition at line 36 of file StandardsSupport.hh.
```

#### 8.119.2.2 RTI1516\_EXCEPTION

```
#define RTI1516_EXCEPTION rti1516e::Exception
Definition at line 55 of file StandardsSupport.hh.
```

#### 8.119.2.3 RTI1516\_HEADER

```
#define RTI1516_HEADER "RTI/RTI1516.h"
Definition at line 52 of file StandardsSupport.hh.
```

#### 8.119.2.4 RTI1516\_NAMESPACE

```
#define RTI1516_NAMESPACE rti1516e
Definition at line 53 of file StandardsSupport.hh.
```

#### 8.119.2.5 RTI1516\_USERDATA

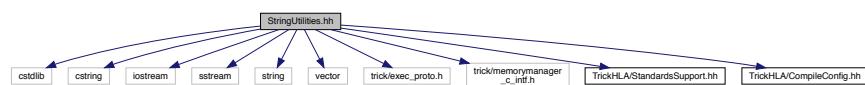
```
#define RTI1516_USERDATA rti1516e::VariableLengthData
Definition at line 54 of file StandardsSupport.hh.
```

## 8.120 StringUtilities.hh File Reference

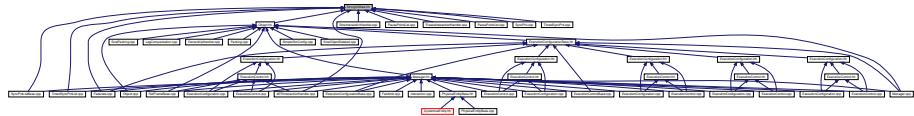
String utilities.

```
#include <cstdlib>
#include <cstring>
#include <iostream>
#include <sstream>
#include <string>
#include <vector>
#include "trick/exec_proto.h"
#include "trick/memorymanager_c_intf.h"
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/CompileConfig.hh"
```

Include dependency graph for StringUtilities.hh:



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



### Data Structures

- class [TrickHLA::StringUtilities](#)

### Namespaces

- [TrickHLA](#)

### Macros

- `#define WHITESPACE_CHARS " \t\r\n\f\v"`

#### 8.120.1 Detailed Description

String utilities.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.TrickHLA*

## Link Dependencies

- `../source/TrickHLA/Attribute.cpp`
  - `../source/TrickHLA/Utilities.cpp`
  - `../source/TrickHLA/Conditional.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">DSES</a>	June 2006	–	Initial implementation.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

## Revision History

## 8.120.2 Macro Definition Documentation

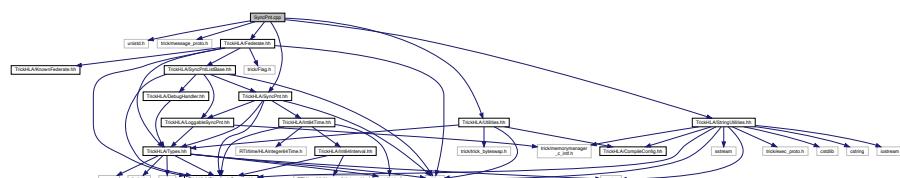
### 8.120.2.1 WHITESPACE\_CHARS

```
#define WHITESPACE_CHARS " \t\r\n\f\v"  
Definition at line 59 of file StringUtilities.hh.
```

## 8.121 SyncPnt.cpp File Reference

This class provides a sync-point implementation for storing and managing [TrickHLA](#) synchronization points.

```
#include <unistd.h>
#include "trick/message_proto.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/StringUtilities.hh"
#include "TrickHLA/SyncPnt.hh"
#include "TrickHLA/Utilities.hh"
Include dependency graph for SyncPnt.cpp:
```



### 8.121.1 Detailed Description

This class provides a sync-point implementation for storing and managing [TrickHLA](#) synchronization points.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [SyncPnt.cpp](#)
- [Int64Time.cpp](#)

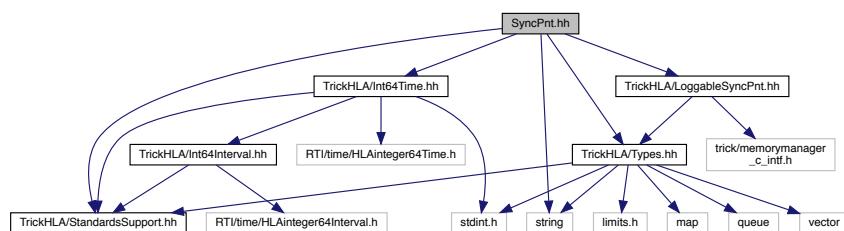
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA JSC ER7	<a href="#">TrickHLA</a>	Jan 2019	–	Create from old TrickHLASyncPntsBase class.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

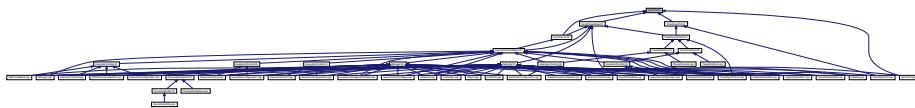
## 8.122 SyncPnt.hh File Reference

This class provides a sync-point implementation for storing and managing [TrickHLA](#) synchronization points.

```
#include <string>
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/LoggableSyncPnt.hh"
#include "TrickHLA/Types.hh"
Include dependency graph for SyncPnt.hh:
```



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



## Data Structures

- class [TrickHLA::SyncPnt](#)

## Namespaces

- [TrickHLA](#)

### 8.122.1 Detailed Description

This class provides a sync-point implementation for storing and managing [TrickHLA](#) synchronization points.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLA*

#### Link Dependencies

- [../source/TrickHLA/SyncPnt.cpp](#)
- [../source/TrickHLA/Int64Time.cpp](#)

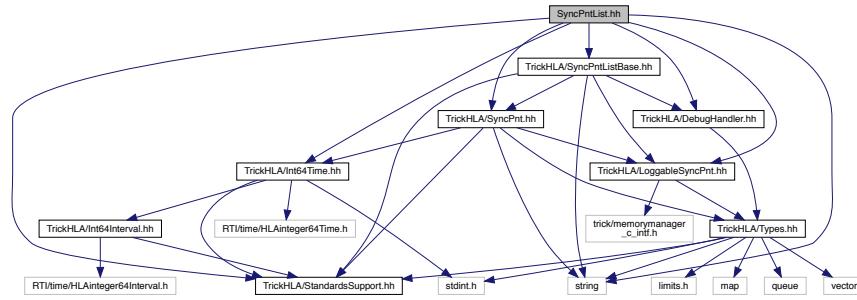
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA JSC ER7	<a href="#">TrickHLA</a>	Jan 2019	–	Create from old TrickHLASync← PtsBase class.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

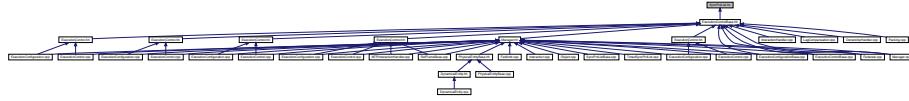
### 8.123 SyncPntList.hh File Reference

This class extends the [TrickHLA::SyncPntListBase](#) class and provides an instantiable implementation for storing and managing HLA synchronization points for [TrickHLA](#).

```
#include <string>
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/DebugHandler.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/LoggableSyncPnt.hh"
#include "TrickHLA/SyncPnt.hh"
#include "TrickHLA/SyncPntListBase.hh"
Include dependency graph for SyncPntList.hh:
```



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



## Data Structures

- class [TrickHLA::SyncPntList](#)

## Namespaces

- [TrickHLA](#)

### 8.123.1 Detailed Description

This class extends the [TrickHLA::SyncPntListBase](#) class and provides an instantiable implementation for storing and managing HLA synchronization points for [TrickHLA](#).

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module:** *trick.TrickHLA***Link Dependencies**

- ..../source/TrickHLA/Int64Time.cpp
- ..../source/TrickHLA/SyncPnt.cpp
- ..../source/TrickHLA/SyncPntListBase.cpp

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

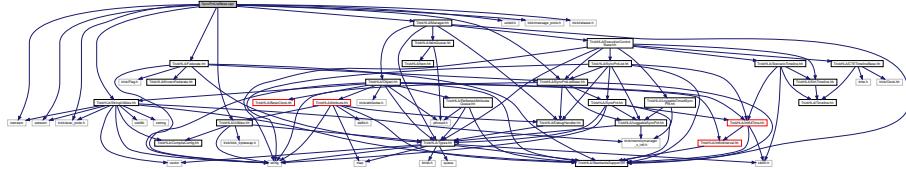
**Revision History**

## 8.124 SyncPntListBase.cpp File Reference

This class provides and abstract base class as the base implementation for storing and managing HLA synchronization points for Trick.

```
#include <iostream>
#include <sstream>
#include <unistd.h>
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "trick/release.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/StringUtilities.hh"
#include "TrickHLA/SyncPntListBase.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for SyncPntListBase.cpp:



### 8.124.1 Detailed Description

This class provides and abstract base class as the base implementation for storing and managing HLA synchronization points for Trick.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- [Int64Time.cpp](#)
- [SyncPnt.cpp](#)
- [SyncPntListBase.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

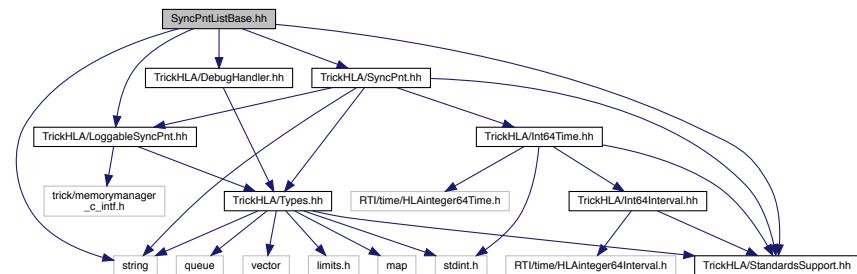
## Revision History

## 8.125 SyncPntListBase.hh File Reference

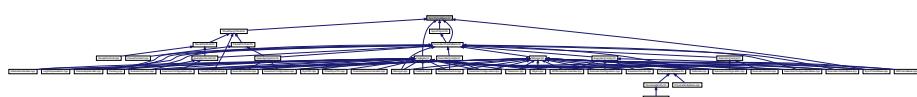
This class provides an abstract base class as the base implementation for storing and managing HLA synchronization points for Trick.

```
#include <string>
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/DebugHandler.hh"
#include "TrickHLA/LoggableSyncPnt.hh"
#include "TrickHLA/SyncPnt.hh"
```

Include dependency graph for SyncPntListBase.hh:



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



## Data Structures

- class [TrickHLA::SyncPntListBase](#)

## Namespaces

- [TrickHLA](#)

### 8.125.1 Detailed Description

This class provides an abstract base class as the base implementation for storing and managing HLA synchronization points for Trick.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLA*

#### Link Dependencies

- [../source/TrickHLA/Int64Time.cpp](#)
- [../source/TrickHLA/SyncPnt.cpp](#)
- [../source/TrickHLA/SyncPntListBase.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

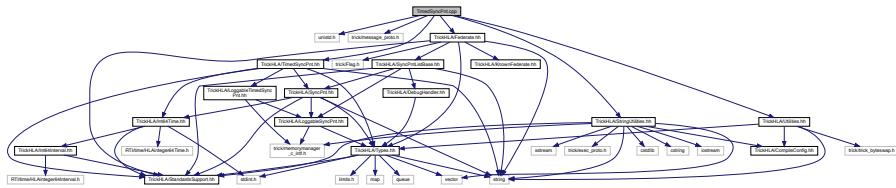
#### Revision History

### 8.126 TimedSyncPnt.cpp File Reference

This class provides a sync-point implementation for storing and managing [TrickHLA](#) synchronization points.

```
#include <unistd.h>
#include "trick/message_proto.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/StringUtilities.hh"
#include "TrickHLA/TimedSyncPnt.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for TimedSyncPnt.cpp:



### 8.126.1 Detailed Description

This class provides a sync-point implementation for storing and managing [TrickHLA](#) synchronization points.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [TimedSyncPnt.cpp](#)
- [Int64Time.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA JSC ER7	<a href="#">TrickHLA</a>	Jan 2019	–	Create from old TrickHLASync→ PtsBase class.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

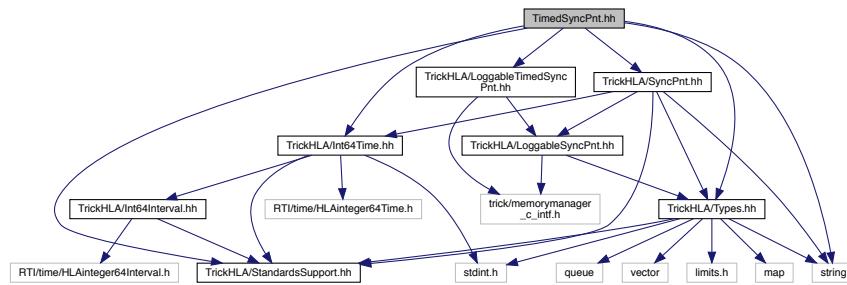
#### Revision History

### 8.127 TimedSyncPnt.hh File Reference

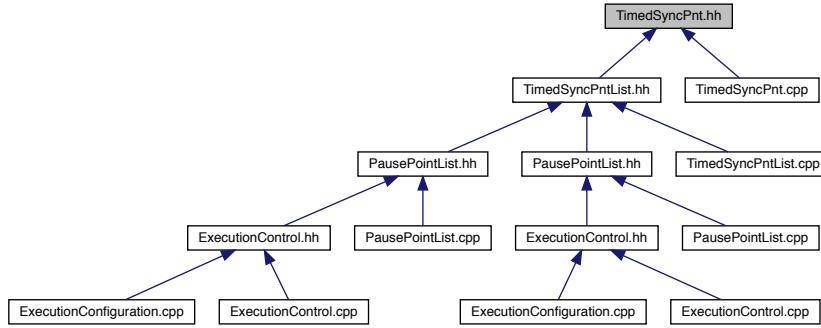
This class extends the basis [TrickHLA::SyncPnt](#) synchronization point implementation to add a time stamp.

```
#include <string>
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/LoggableTimedSyncPnt.hh"
#include "TrickHLA/SyncPnt.hh"
#include "TrickHLA/Types.hh"
```

Include dependency graph for TimedSyncPnt.hh:



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



## Data Structures

- class [TrickHLA::TimedSyncPnt](#)

## Namespaces

- [TrickHLA](#)

### 8.127.1 Detailed Description

This class extends the basis [TrickHLA::SyncPnt](#) synchronization point implementation to add a time stamp.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Python Module: *trick.TrickHLA*

#### Link Dependencies

- `../source/TrickHLA/SyncPnt.cpp`
- `../source/TrickHLA/TimedSyncPnt.cpp`
- `../source/TrickHLA/Int64Time.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA JSC ER7	<a href="#">TrickHLA</a>	Jan 2019	–	Create from old TrickHLASync PtsBase class.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

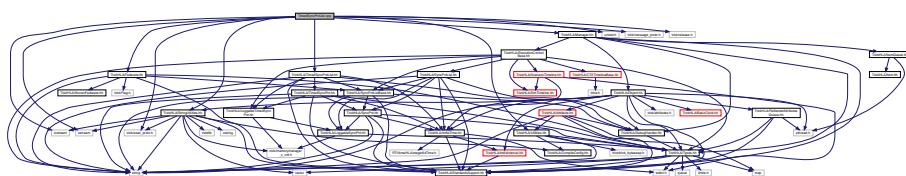
#### Revision History

## 8.128 TimedSyncPntList.cpp File Reference

This class provides and abstract base class as the base implementation for storing and managing HLA synchronization points for Trick.

```
#include <iostream>
#include <sstream>
#include <unistd.h>
#include "trick/exec_proto.h"
#include "trick/message_proto.h"
#include "trick/release.h"
#include "TrickHLA/Federate.hh"
#include "TrickHLA/Manager.hh"
#include "TrickHLA/StringUtilities.hh"
#include "TrickHLA/TimedSyncPntList.hh"
#include "TrickHLA/Utilities.hh"
```

Include dependency graph for TimedSyncPntList.cpp:



### 8.128.1 Detailed Description

This class provides and abstract base class as the base implementation for storing and managing HLA synchronization points for Trick.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- [Int64Time.cpp](#)
- [SyncPnt.cpp](#)
- [TimedSyncPntList.cpp](#)

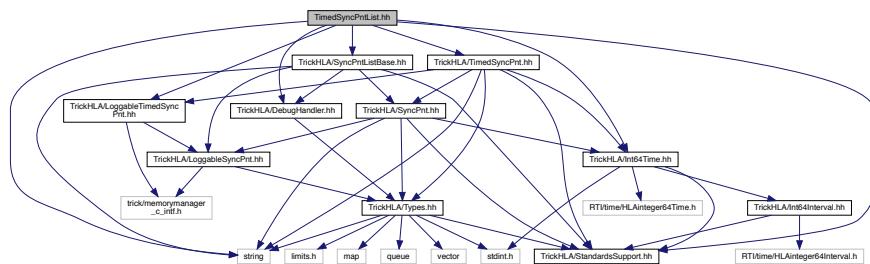
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

## Revision History

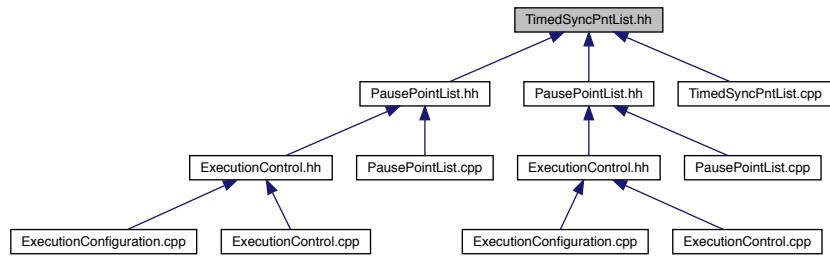
## 8.129 TimedSyncPntList.hh File Reference

This class extends the [TrickHLA::SyncPntListBase](#) class and provides an instantiable implementation for storing and managing HLA synchronization points for [TrickHLA](#).

```
#include <string>
#include "TrickHLA/StandardsSupport.hh"
#include "TrickHLA/DebugHandler.hh"
#include "TrickHLA/Int64Time.hh"
#include "TrickHLA/LoggableTimedSyncPnt.hh"
#include "TrickHLA/SyncPntListBase.hh"
#include "TrickHLA/TimedSyncPnt.hh"
Include dependency graph for TimedSyncPntList.hh:
```



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



## Data Structures

- class [TrickHLA::TimedSyncPntList](#)

## Namespaces

- [TrickHLA](#)

### 8.129.1 Detailed Description

This class extends the [TrickHLA::SyncPntListBase](#) class and provides an instantiable implementation for storing and managing HLA synchronization points for [TrickHLA](#).

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Int64Time.cpp`
- `../source/TrickHLA/SyncPnt.cpp`
- `../source/TrickHLA/SyncPntListBase.cpp`
- `../source/TrickHLA/TimedSyncPntList.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

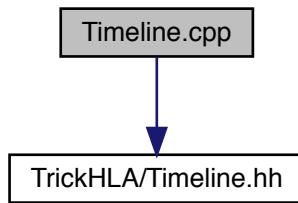
## Revision History

## 8.130 Timeline.cpp File Reference

This class represents the HLA time.

```
#include "TrickHLA/Timeline.hh"
```

Include dependency graph for Timeline.cpp:



### 8.130.1 Detailed Description

This class represents the HLA time.

This class is the abstract base class for representing timelines.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- [Int64Time.cpp](#)
- [Int64Interval.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Robert G. Phillips	Titan Corp.	<a href="#">DIS</a>	October 2004	–	Initial implementation for ISS HTV Sim
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

## Revision History

### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

### Link Dependencies

- [Timeline.cpp](#)

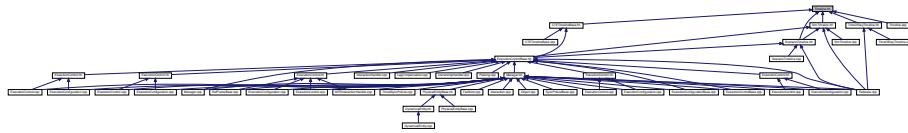
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	April 2016	–	Initial version.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

## Revision History

## 8.131 Timeline.hh File Reference

This class is the abstract base class for representing timelines.

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



### Data Structures

- class [TrickHLA::Timeline](#)

### Namespaces

- [TrickHLA](#)

#### 8.131.1 Detailed Description

This class is the abstract base class for representing timelines.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Python Module: *trick.TrickHLA*****Link Dependencies**

- ..../source/TrickHLA/Timeline.cpp

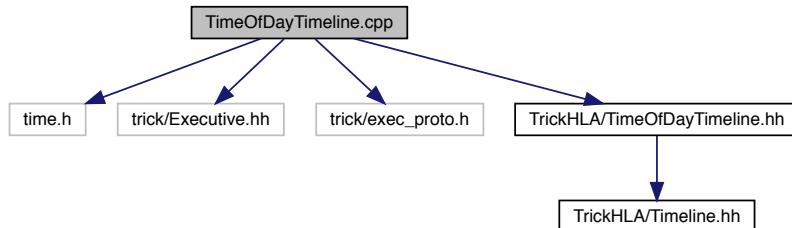
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	TrickHLA	April 2016	–	Initial implementation.
Dan Dexter	NASA ER7	TrickHLA	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

**Revision History**

## 8.132 TimeOfDayTimeline.cpp File Reference

This class represents the simulation timeline.

```
#include <time.h>
#include "trick/Executive.hh"
#include "trick/exec_proto.h"
#include "TrickHLA/TimeOfDayTimeline.hh"
Include dependency graph for TimeOfDayTimeline.cpp:
```



### 8.132.1 Detailed Description

This class represents the simulation timeline.

### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
Software, Robotics & Simulation Division  
NASA, Johnson Space Center  
2101 NASA Parkway, Houston, TX 77058

### Link Dependencies

- [Timeline.cpp](#)
- [TimeOfDayTimeline.cpp](#)

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	SpaceFOM	June 2016	–	Initial implementation.
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

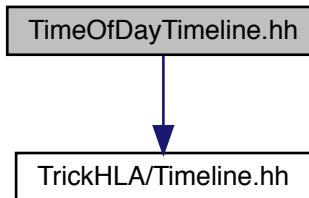
### Revision History

## 8.133 TimeOfDayTimeline.hh File Reference

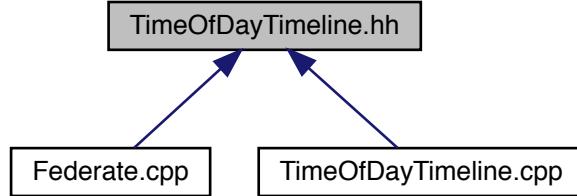
This class represents the time of day timeline.

```
#include "TrickHLA/Timeline.hh"
```

Include dependency graph for TimeOfDayTimeline.hh:



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



## Data Structures

- class [TrickHLA::TimeOfDayTimeline](#)

## Namespaces

- [TrickHLA](#)

### 8.133.1 Detailed Description

This class represents the time of day timeline.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Timeline.cpp`
- `../source/TrickHLA/TimeOfDayTimeline.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">SpaceFOM</a>	June 2016	–	Initial implementation.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

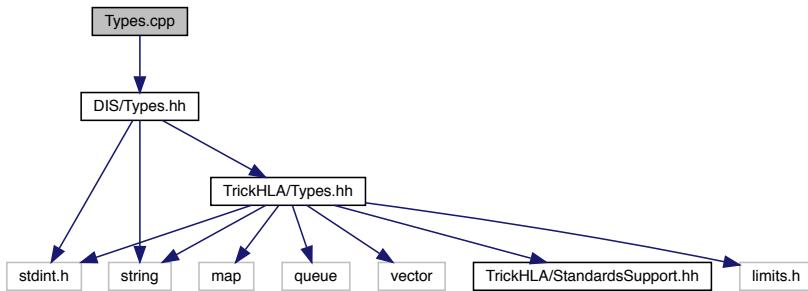
## Revision History

## 8.134 Types.cpp File Reference

Implementation of the [TrickHLA DIS](#) types utility functions.

```
#include "DIS/Types.hh"
```

Include dependency graph for DIS/Types.cpp:



### 8.134.1 Detailed Description

Implementation of the [TrickHLA DIS](#) types utility functions.

## Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Link Dependencies

- Types.cpp

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

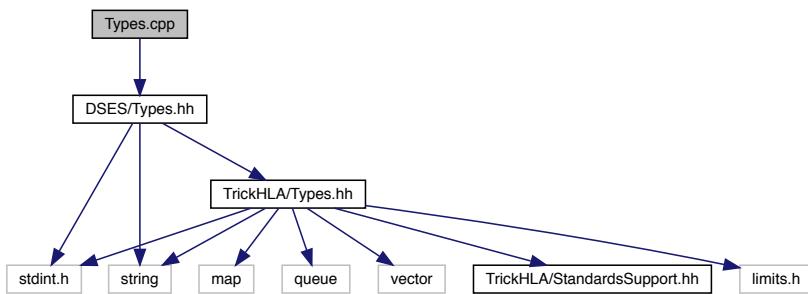
## Revision History

## 8.135 Types.cpp File Reference

Implementation of the [TrickHLA DSES](#) types utility functions.

```
#include "DSES/Types.hh"
```

Include dependency graph for DSES/Types.cpp:



### 8.135.1 Detailed Description

Implementation of the [TrickHLA DSES](#) types utility functions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- Types.cpp

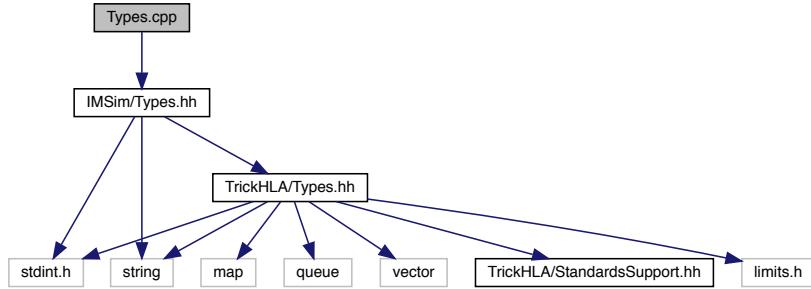
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.136 Types.cpp File Reference

Implementation of the [TrickHLA IMSim](#) types utility functions.

```
#include "IMSim/Types.hh"
Include dependency graph for IMSim/Types.cpp:
```



### 8.136.1 Detailed Description

Implementation of the [TrickHLA IMSim](#) types utility functions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- Types.cpp

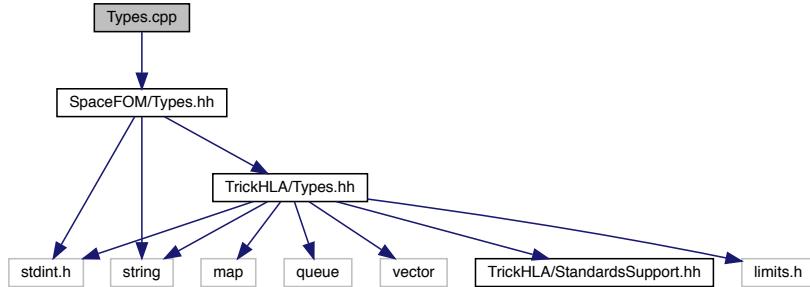
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.137 Types.cpp File Reference

Implementation of the [TrickHLA SpaceFOM](#) types utility functions.

```
#include "SpaceFOM/Types.hh"
Include dependency graph for SpaceFOM/Types.cpp:
```



### 8.137.1 Detailed Description

Implementation of the [TrickHLA SpaceFOM](#) types utility functions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- Types.cpp

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

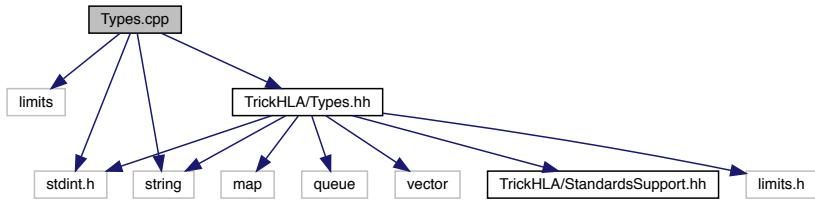
#### Revision History

## 8.138 Types.cpp File Reference

Implementation of the [TrickHLA](#) types utility functions.

```
#include <limits>
#include <stdint.h>
#include <string>
#include "TrickHLA/Types.hh"
```

Include dependency graph for TrickHLA/Types.cpp:



### 8.138.1 Detailed Description

Implementation of the [TrickHLA](#) types utility functions.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Link Dependencies

- Types.cpp

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

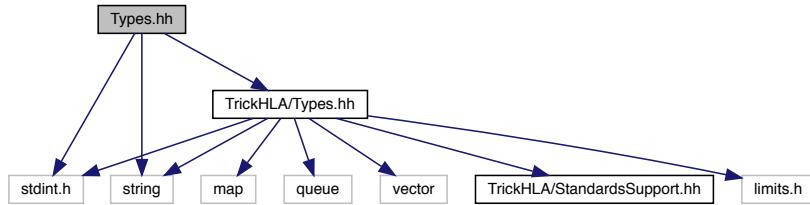
#### Revision History

## 8.139 Types.hh File Reference

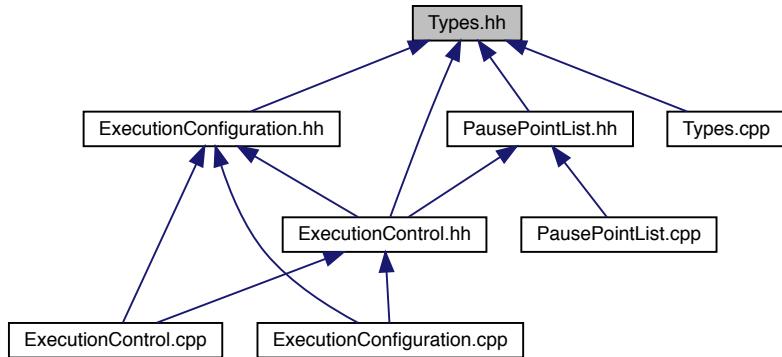
Definition of the [TrickHLA DIS](#) enumeration types and utilities.

```
#include <stdint.h>
#include <string>
#include "TrickHLA/Types.hh"
```

Include dependency graph for DIS/Types.hh:



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



## Namespaces

- [DIS](#)

## Enumerations

- enum [DIS::ExecutionModeEnum](#) {
   
`DIS::EXECUTION_MODE_FIRST_VALUE = 0, DIS::EXECUTION_MODE_UNINITIALIZED = 0, DIS::EXECUTION_MODE_INITIALIZING = 1, DIS::EXECUTION_MODE_RUNNING = 2,`
  
`DIS::EXECUTION_MODE_FREEZE = 3, DIS::EXECUTION_MODE_SHUTDOWN = 4, DIS::EXECUTION_MODE_LAST_VALUE = 4 }`

Define the [TrickHLA DIS](#) execution mode enumeration values.

- enum [DIS::MTREnum](#) {
   
`DIS::MTR_FIRST_VALUE = 0, DIS::MTR_UNINITIALIZED = 0, DIS::MTR_INITIALIZING = 1, DIS::MTR_GOTO_RUN = 2,`
  
`DIS::MTR_GOTO_FREEZE = 3, DIS::MTR_GOTO_SHUTDOWN = 4, DIS::MTR_LAST_VALUE = 4 }`

Define the [TrickHLA DIS](#) Mode Transition Request state enumeration values.

- enum [DIS::PausePointStateEnum](#) {
   
`DIS::PAUSE_POINT_STATE_FIRST_VALUE = 0, DIS::PAUSE_POINT_STATE_ERROR = 0, DIS::PAUSE_POINT_STATE_PENDING = 1,`

```
= 1, DIS::PAUSE_POINT_STATE_ACKNOWLEDGED = 2,
DIS::PAUSE_POINT_STATE_RUN = 3, DIS::PAUSE_POINT_STATE_FREEZE = 4, DIS::PAUSE_POINT_STATE_EXIT
= 5, DIS::PAUSE_POINT_STATE_RESTART = 6,
DIS::PAUSE_POINT_STATE_RECONFIG = 7, DIS::PAUSE_POINT_STATE_UNKNOWN = INT_MAX }
```

Define the *TrickHLA* synchronization point state enumeration values.

## Functions

- `std::string DIS::execution_mode_enum_to_string (ExecutionModeEnum mode)`  
*Convert an ExecutionModeEnum value into a printable string.*
- `int16_t DIS::execution_mode_enum_to_int16 (ExecutionModeEnum mode)`  
*Convert an ExecutionModeEnum value into a 16 bit integer.*
- `ExecutionModeEnum DIS::execution_mode_int16_to_enum (int16_t int_mode)`  
*Convert a 16 bit integer to an ExecutionModeEnum value.*
- `TrickHLA::ExecutionControlEnum DIS::to_execution_control_enum (ExecutionModeEnum mode)`  
*Convert an DIS::ExecutionModeEnum value to a TrickHLA::ExecutionModeEnum value.*
- `ExecutionModeEnum DIS::from_execution_contorl_enum (TrickHLA::ExecutionControlEnum mode)`  
*Convert a TrickHLA::ExecutionModeEnum value to an DIS::ExecutionModeEnum value.*
- `std::string DIS::mtr_enum_to_string (MTREnum mtr_enum)`  
*Convert a Mode Transition Request (MTR) enum value into a printable string.*
- `int16_t DIS::mtr_enum_to_int16 (MTREnum mtr_enum)`  
*Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.*
- `MTREnum DIS::mtr_int16_to_enum (int16_t mtr_int)`  
*Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.*
- `std::string DIS::pause_pnt_state_enum_to_string (PausePointStateEnum state)`  
*Convert a Pause Synchronization Point State enum value into a printable string.*
- `int16_t DIS::pause_pnt_state_enum_to_int16 (PausePointStateEnum state)`  
*Convert a Pause Synchronization Point State enum value into a 16 bit integer.*
- `PausePointStateEnum DIS::pause_pnt_state_int16_to_enum (int16_t int_state)`  
*Convert an integer value to a Pause Synchronization Point State enumeration value.*

### 8.139.1 Detailed Description

Definition of the *TrickHLA* DIS enumeration types and utilities.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.DIS*

#### Link Dependencies

- `../source/DIS/Types.cpp`

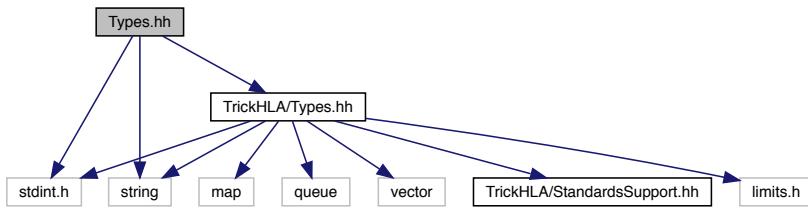
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

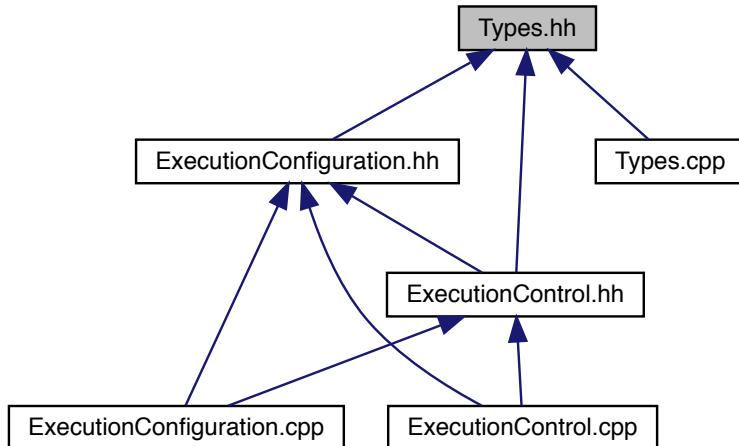
## 8.140 Types.hh File Reference

Definition of the [TrickHLA DSES](#) enumeration types and utilities.

```
#include <stdint.h>
#include <string>
#include "TrickHLA/Types.hh"
Include dependency graph for DSES/Types.hh:
```



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



## Namespaces

- [DSES](#)

## Enumerations

- enum `DSES::ExecutionModeEnum` {
 `DSES::EXECUTION_MODE_FIRST_VALUE` = 0, `DSES::EXECUTION_MODE_UNINITIALIZED` = 0, `DSES::EXECUTION_MODE_RUNNING` = 1, `DSES::EXECUTION_MODE_FREEZE` = 2, `DSES::EXECUTION_MODE_SHUTDOWN` = 4, `DSES::EXECUTION_MODE_LAST_VALUE` = 4 }
- enum `DSES::MTREnum` {
 `DSES::MTR_FIRST_VALUE` = 0, `DSES::MTR_UNINITIALIZED` = 0, `DSES::MTR_INITIALIZING` = 1, `DSES::MTR_GOTO_RUN` = 2, `DSES::MTR_GOTO_FREEZE` = 3, `DSES::MTR_GOTO_SHUTDOWN` = 4, `DSES::MTR_LAST_VALUE` = 4 }

## Functions

- `std::string DSES::execution_mode_enum_to_string (ExecutionModeEnum mode)`  
`Convert an ExecutionModeEnum value into a printable string.`
- `int16_t DSES::execution_mode_enum_to_int16 (ExecutionModeEnum mode)`  
`Convert an ExecutionModeEnum value into a 16 bit integer.`
- `ExecutionModeEnum DSES::execution_mode_int16_to_enum (int16_t int_mode)`  
`Convert a 16 bit integer to an ExecutionModeEnum value.`
- `TrickHLA::ExecutionControlEnum DSES::to_execution_control_enum (ExecutionModeEnum mode)`  
`Convert an DSES::ExecutionModeEnum value to a TrickHLA::ExecutionModeEnum value.`
- `ExecutionModeEnum DSES::from_execution_control_enum (TrickHLA::ExecutionControlEnum mode)`  
`Convert a TrickHLA::ExecutionModeEnum value to an DSES::ExecutionModeEnum value.`
- `std::string DSES::mtr_enum_to_string (MTREnum mtr_enum)`  
`Convert a Mode Transition Request (MTR) enum value into a printable string.`
- `int16_t DSES::mtr_enum_to_int16 (MTREnum mtr_enum)`  
`Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.`
- `MTREnum DSES::mtr_int16_to_enum (int16_t mtr_int)`  
`Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.`

### 8.140.1 Detailed Description

Definition of the `TrickHLA DSES` enumeration types and utilities.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.DSES`

#### Link Dependencies

- `../source/DSES/Types.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

## Revision History

### 8.141 Types.hh File Reference

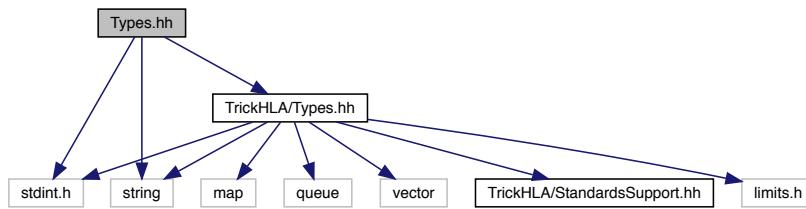
Definition of the [TrickHLA IMSim](#) enumeration types and utilities.

```
#include <stdint.h>
```

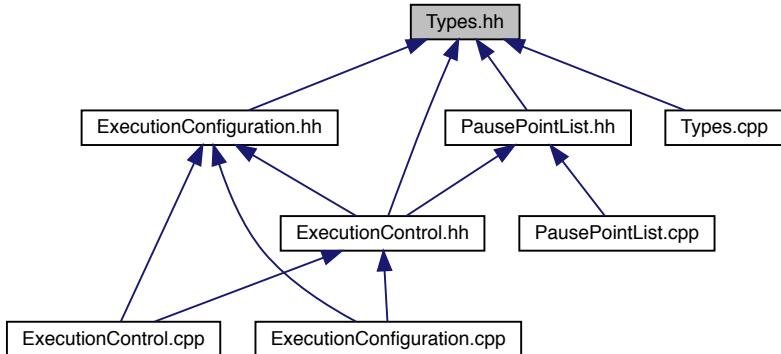
```
#include <string>
```

```
#include "TrickHLA/Types.hh"
```

Include dependency graph for IMSim/Types.hh:



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



## Namespaces

- [IMSim](#)

## Typedefs

- `typedef std::set< double > IMSim::FreezeTimeSet`

## Enumerations

- enum `IMSim::ExecutionModeEnum` {
 `IMSim::EXECUTION_MODE_FIRST_VALUE` = 0, `IMSim::EXECUTION_MODE_UNINITIALIZED` = 0,
 `IMSim::EXECUTION_MODE_INITIALIZING` = 1, `IMSim::EXECUTION_MODE_RUNNING` = 2,
 `IMSim::EXECUTION_MODE_FREEZE` = 3, `IMSim::EXECUTION_MODE_SHUTDOWN` = 4, `IMSim::EXECUTION_MODE_LAST_VALUE` = 4 }
- enum `IMSim::MTREnum` {
 `IMSim::MTR_FIRST_VALUE` = 0, `IMSim::MTR_UNINITIALIZED` = 0, `IMSim::MTR_INITIALIZING` = 1,
 `IMSim::MTR_GOTO_RUN` = 2, `IMSim::MTR_GOTO_FREEZE` = 3, `IMSim::MTR_GOTO_SHUTDOWN` = 4, `IMSim::MTR_LAST_VALUE` = 4 }
- enum `IMSim::PausePointStateEnum` {
 `IMSim::PAUSE_POINT_STATE_FIRST_VALUE` = 0, `IMSim::PAUSE_POINT_STATE_ERROR` = 0, `IMSim::PAUSE_POINT_STATE_UNINITIALIZED` = 1, `IMSim::PAUSE_POINT_STATE_ACKNOWLEDGED` = 2,
 `IMSim::PAUSE_POINT_STATE_RUN` = 3, `IMSim::PAUSE_POINT_STATE_FREEZE` = 4, `IMSim::PAUSE_POINT_STATE_EXIT` = 5, `IMSim::PAUSE_POINT_STATE_RESTART` = 6,
 `IMSim::PAUSE_POINT_STATE_RECONFIG` = 7, `IMSim::PAUSE_POINT_STATE_UNKNOWN` = `INT_MAX` }

## Functions

- `std::string IMSim::execution_mode_enum_to_string (ExecutionModeEnum mode)`  
*Convert an ExecutionModeEnum value into a printable string.*
- `int16_t IMSim::execution_mode_enum_to_int16 (ExecutionModeEnum mode)`  
*Convert an ExecutionModeEnum value into a 16 bit integer.*
- `ExecutionModeEnum IMSim::execution_mode_int16_to_enum (int16_t int_mode)`  
*Convert a 16 bit integer to an ExecutionModeEnum value.*
- `TrickHLA::ExecutionControlEnum IMSim::to_execution_control_enum (ExecutionModeEnum mode)`  
*Convert an IMSim::ExecutionModeEnum value to a TrickHLA::ExecutionModeEnum value.*
- `ExecutionModeEnum IMSim::from_execution_control_enum (TrickHLA::ExecutionControlEnum mode)`  
*Convert a TrickHLA::ExecutionModeEnum value to an IMSim::ExecutionModeEnum value.*
- `std::string IMSim::mtr_enum_to_string (MTREnum mtr_enum)`  
*Convert a Mode Transition Request (MTR) enum value into a printable string.*
- `int16_t IMSim::mtr_enum_to_int16 (MTREnum mtr_enum)`  
*Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.*
- `MTREnum IMSim::mtr_int16_to_enum (int16_t mtr_int)`  
*Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.*
- `std::string IMSim::pause_pnt_state_enum_to_string (PausePointStateEnum state)`  
*Convert a Pause Synchronization Point State enum value into a printable string.*
- `int16_t IMSim::pause_pnt_state_enum_to_int16 (PausePointStateEnum state)`  
*Convert a Pause Synchronization Point State enum value into a 16 bit integer.*
- `PausePointStateEnum IMSim::pause_pnt_state_int16_to_enum (int16_t int_state)`  
*Convert an integer value to a Pause Synchronization Point State enumeration value.*

### 8.141.1 Detailed Description

Definition of the `TrickHLA IMSim` enumeration types and utilities.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

### Python Module: `trick.IMSim`

#### Link Dependencies

- ..../source/IMSim/Types.cpp

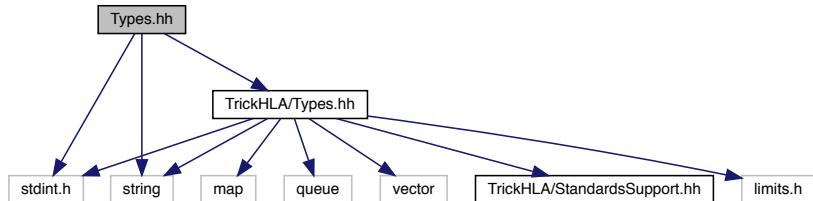
Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

#### Revision History

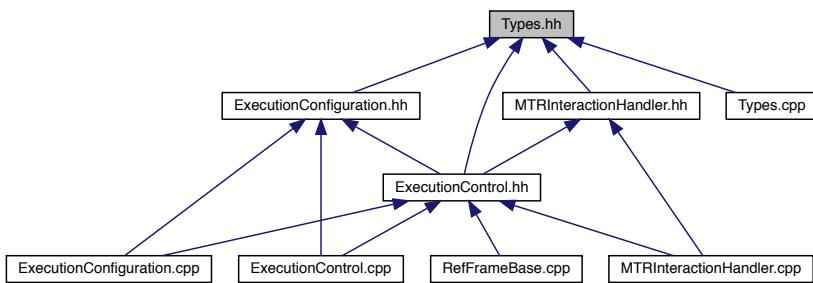
## 8.142 Types.hh File Reference

Definition of the [TrickHLA SpaceFOM](#) enumeration types and utilities.

```
#include <stdint.h>
#include <string>
#include "TrickHLA/Types.hh"
Include dependency graph for SpaceFOM/Types.hh:
```



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



## Namespaces

- [SpaceFOM](#)

## Enumerations

- enum [SpaceFOM::ExecutionModeEnum](#) {
   
SpaceFOM::EXECUTION\_MODE\_FIRST\_VALUE = 0, [SpaceFOM::EXECUTION\\_MODE\\_UNINITIALIZED](#) = 0,  
SpaceFOM::EXECUTION\_MODE\_INITIALIZING = 1, [SpaceFOM::EXECUTION\\_MODE\\_RUNNING](#) = 2,  
SpaceFOM::EXECUTION\_MODE\_FREEZE = 3, [SpaceFOM::EXECUTION\\_MODE\\_SHUTDOWN](#) = 4,  
SpaceFOM::EXECUTION\_MODE\_LAST\_VALUE = 4 }
- enum [SpaceFOM::MTREnum](#) {
   
SpaceFOM::MTR\_FIRST\_VALUE = 0, [SpaceFOM::MTR\\_UNINITIALIZED](#) = 0, [SpaceFOM::MTR\\_INITIALIZING](#) = 1, [SpaceFOM::MTR\\_GOTO\\_RUN](#) = 2,  
[SpaceFOM::MTR\\_GOTO\\_FREEZE](#) = 3, [SpaceFOM::MTR\\_GOTO\\_SHUTDOWN](#) = 4, [SpaceFOM::MTR\\_LAST\\_VALUE](#) = 4 }

## Functions

- [std::string SpaceFOM::execution\\_mode\\_enum\\_to\\_string \(ExecutionModeEnum mode\)](#)  
*Convert an ExecutionModeEnum value into a printable string.*
- [int16\\_t SpaceFOM::execution\\_mode\\_enum\\_to\\_int16 \(ExecutionModeEnum mode\)](#)  
*Convert an ExecutionModeEnum value into a 16 bit integer.*
- [ExecutionModeEnum SpaceFOM::execution\\_mode\\_int16\\_to\\_enum \(int16\\_t int\\_mode\)](#)  
*Convert a 16 bit integer to an ExecutionModeEnum value.*
- [TrickHLA::ExecutionControlEnum SpaceFOM::to\\_execution\\_control\\_enum \(ExecutionModeEnum mode\)](#)  
*Convert an SpaceFOM::ExecutionModeEnum value to a TrickHLA::ExecutionModeEnum value.*
- [ExecutionModeEnum SpaceFOM::from\\_execution\\_control\\_enum \(TrickHLA::ExecutionControlEnum mode\)](#)  
*Convert a TrickHLA::ExecutionModeEnum value to an SpaceFOM::ExecutionModeEnum value.*
- [std::string SpaceFOM::mtr\\_enum\\_to\\_string \(MTREnum mtr\\_enum\)](#)  
*Convert a Mode Transition Request (MTR) enum value into a printable string.*
- [int16\\_t SpaceFOM::mtr\\_enum\\_to\\_int16 \(MTREnum mtr\\_enum\)](#)  
*Convert a Mode Transition Request (MTR) enum value into a 16 bit integer.*
- [MTREnum SpaceFOM::mtr\\_int16\\_to\\_enum \(int16\\_t mtr\\_int\)](#)  
*Convert a 16 bit integer into a Mode Transition Request (MTR) enum value.*
- [MTREnum SpaceFOM::from\\_mode\\_transition\\_enum \(TrickHLA::ModeTransitionEnum mode\)](#)  
*Convert a TrickHLA::ModeTransitionEnum value to an SpaceFOM::MTREnum value.*

### 8.142.1 Detailed Description

Definition of the [TrickHLA SpaceFOM](#) enumeration types and utilities.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

## Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

## Python Module: *trick.SpaceFOM*

### Link Dependencies

- ..../source/SpaceFOM/Types.cpp

Author	Organization	Project	Date	Rev. ID	Description
Edwin Z. Crues	NASA ER7	TrickHLA	March 2019	–	Version 3 rewrite.

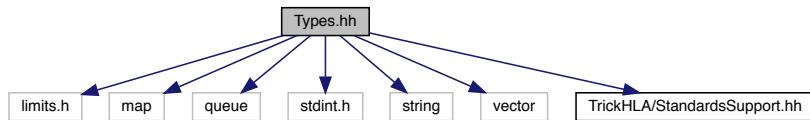
### Revision History

## 8.143 Types.hh File Reference

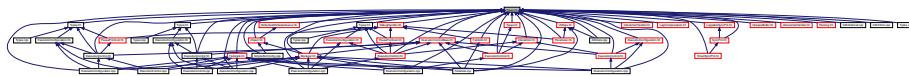
Definition of the [TrickHLA](#) enumeration types and utilities.

```
#include <limits.h>
#include <map>
#include <queue>
#include <stdint.h>
#include <string>
#include <vector>
#include "TrickHLA/StandardsSupport.hh"
```

Include dependency graph for TrickHLA/Types.hh:



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



## Namespaces

- [TrickHLA](#)

## Typedefs

- `typedef std::auto_ptr< RTI1516_NAMESPACE::RTIambassador > TrickHLA::TrickRTIAmbPtr`

- `typedef std::queue< RTI1516_NAMESPACE::AttributeHandleValueMap > TrickHLA::HLAAttributeMapQueue`
- `typedef std::map< RTI1516_NAMESPACE::ObjectInstanceHandle, std::wstring > TrickHLA::TrickHLAOBJInstanceNameMap`
- `typedef std::vector< std::string > TrickHLA::VectorOfStrings`
- `typedef std::vector< std::wstring > TrickHLA::VectorOfWstrings`

## Enumerations

- `enum TrickHLA::DataUpdateEnum { TrickHLA::CONFIG_NONE = 0x0001, TrickHLA::CONFIG_INITIALIZE = 0x0002, TrickHLA::CONFIG_INTERMITTENT = 0x0004, TrickHLA::CONFIG_CYCLIC = 0x0008, TrickHLA::CONFIG_MAX_VALUE }`

*Define the `TrickHLA` attribute update reflection type.*
- `enum TrickHLA::EncodingEnum { TrickHLA::ENCODING_FIRST_VALUE = 0, TrickHLA::ENCODING_UNKNOWN = 0, TrickHLA::ENCODING_BIG_ENDIAN = 1, TrickHLA::ENCODING_LITTLE_ENDIAN = 2, TrickHLA::ENCODING_LOGICAL_TIME = 3, TrickHLA::ENCODING_C_STRING = 4, TrickHLA::ENCODING_UNICODE_STRING = 5, TrickHLA::ENCODING_ASCII_STRING = 6, TrickHLA::ENCODING_OPAQUE_DATA = 7, TrickHLA::ENCODING_BOOLEAN = 8, TrickHLA::ENCODING_NO_ENCODING = 9, TrickHLA::ENCODING_LAST_VALUE = 9 }`

*Define the `TrickHLA` data encoding type.*
- `enum TrickHLA::TransportationEnum { TrickHLA::TRANSPORT_FIRST_VALUE = 0, TrickHLA::TRANSPORT_SPECIFIED_IN_FOM = 0, TrickHLA::TRANSPORT_TIMESTAMP = 1, TrickHLA::TRANSPORT_RECEIVE_ORDER = 2, TrickHLA::TRANSPORT_LAST_VALUE = 3 }`

*Define the `TrickHLA` data transportation type.*
- `enum TrickHLA::LagCompensationEnum { TrickHLA::LAG_COMPENSATION_FIRST_VALUE = 0, TrickHLA::LAG_COMPENSATION_NONE = 0, TrickHLA::LAG_COMPENSATION_SEND_SIDE = 1, TrickHLA::LAG_COMPENSATION_RECEIVE_SIDE = 2, TrickHLA::LAG_COMPENSATION_LAST_VALUE = 2 }`

*Define the `TrickHLA` latency (lag) compensation type.*
- `enum TrickHLA::DebugLevelEnum { TrickHLA::DEBUG_LEVEL_NO_TRACE = 0, TrickHLA::DEBUG_LEVEL_0_TRACE = 0, TrickHLA::DEBUG_LEVEL_1_TRACE = 1, TrickHLA::DEBUG_LEVEL_2_TRACE = 2, TrickHLA::DEBUG_LEVEL_3_TRACE = 3, TrickHLA::DEBUG_LEVEL_4_TRACE = 4, TrickHLA::DEBUG_LEVEL_5_TRACE = 5, TrickHLA::DEBUG_LEVEL_6_TRACE = 6, TrickHLA::DEBUG_LEVEL_7_TRACE = 7, TrickHLA::DEBUG_LEVEL_8_TRACE = 8, TrickHLA::DEBUG_LEVEL_9_TRACE = 9, TrickHLA::DEBUG_LEVEL_10_TRACE = 10, TrickHLA::DEBUG_LEVEL_11_TRACE = 11, TrickHLA::DEBUG_LEVEL_FULL_TRACE = 11 }`

*Define the `TrickHLA` level for debug messages.*
- `enum TrickHLA::DebugSourceEnum { TrickHLA::DEBUG_SOURCE_FED_AMB = 0x00000001, TrickHLA::DEBUG_SOURCE_FEDERATE = 0x00000002, TrickHLA::DEBUG_SOURCE_MANAGER = 0x00000004, TrickHLA::DEBUG_SOURCE_OBJECT = 0x00000008, TrickHLA::DEBUG_SOURCE_INTERACTION = 0x00000010, TrickHLA::DEBUG_SOURCE_ATTRIBUTE = 0x00000020, TrickHLA::DEBUG_SOURCE_PARAMETER = 0x00000040, TrickHLA::DEBUG_SOURCE_SYNCPOINT = 0x00000080, TrickHLA::DEBUG_SOURCE_OWNERSHIP = 0x00000100, TrickHLA::DEBUG_SOURCE_PACKING = 0x00000200, TrickHLA::DEBUG_SOURCE_LAG_COMPENSATION = 0x00000400, TrickHLA::DEBUG_SOURCE_ALL_MODULES = 0x7FFFFFFF }`

*Define the `TrickHLA` source for debug messages.*

- enum `TrickHLA::FederateJoinEnum` {
 `TrickHLA::FEDERATE_JOIN_FIRST_VALUE` = 0, `TrickHLA::FEDERATE_JOIN_NOMINAL` = 0, `TrickHLA::FEDERATE_JOIN_EARLY` = 0, `TrickHLA::FEDERATE_JOIN_LATE` = 1, `TrickHLA::FEDERATE_JOIN_RESTORING` = 2, `TrickHLA::FEDERATE_JOIN_UNKNOWN` = 3, `TrickHLA::FEDERATE_JOIN_LAST` = 3 }

*Define the `TrickHLA` federate join enumeration values.*

- enum `TrickHLA::ExecutionControlEnum` {
 `TrickHLA::EXECUTION_CONTROL_FIRST_VALUE` = 0, `TrickHLA::EXECUTION_CONTROL_UNINITIALIZED` = 0, `TrickHLA::EXECUTION_CONTROL_INITIALIZING` = 1, `TrickHLA::EXECUTION_CONTROL_RUNNING` = 2, `TrickHLA::EXECUTION_CONTROL_FREEZE` = 3, `TrickHLA::EXECUTION_CONTROL_RESTART` = 4, `TrickHLA::EXECUTION_CONTROL_RECONFIG` = 5, `TrickHLA::EXECUTION_CONTROL_SHUTDOWN` = 6, `TrickHLA::EXECUTION_CONTROL_LAST_VALUE` = 6 }

*Define the `TrickHLA` execution control enumeration values.*

- enum `TrickHLA::ModeTransitionEnum` {
 `TrickHLA::MODE_TRANSITION_FIRST_VALUE` = 0, `TrickHLA::MODE_TRANSITION_UNINITIALIZED` = 0, `TrickHLA::MODE_TRANSITION_INITIALIZING` = 1, `TrickHLA::MODE_TRANSITION_GOTO_RUN` = 2, `TrickHLA::MODE_TRANSITION_GOTO_FREEZE` = 3, `TrickHLA::MODE_TRANSITION_GOTO_RESTART` = 4, `TrickHLA::MODE_TRANSITION_GOTO_RECONFIG` = 5, `TrickHLA::MODE_TRANSITION_GOTO_SHUTDOWN` = 6, `TrickHLA::MODE_TRANSITION_LAST_VALUE` = 6 }

*Define the `TrickHLA` Mode Transition state enumeration values.*

- enum `TrickHLA::SyncPntStateEnum` {
 `TrickHLA::SYNC_PNT_STATE_FIRST_VALUE` = 0, `TrickHLA::SYNC_PNT_STATE_ERROR` = 0, `TrickHLA::SYNC_PNT_STATE_ERROR` = 1, `TrickHLA::SYNC_PNT_STATE_REGISTERED` = 2, `TrickHLA::SYNC_PNT_STATE_ANNOUCED` = 3, `TrickHLA::SYNC_PNT_STATE_ACHIEVED` = 4, `TrickHLA::SYNC_PNT_STATE_ACHIEVED` = 5, `TrickHLA::SYNC_PNT_STATE_LAST_VALUE` = 5, `TrickHLA::SYNC_PNT_STATE_UNKNOWN` = `INT_MAX` }

*Define the `TrickHLA` synchronization point state enumeration values.*

## Functions

- `std::string TrickHLA::execution_control_enum_to_string (ExecutionControlEnum mode)`

*Convert an `ExecutionModeEnum` value into a printable string.*
- `int16_t TrickHLA::execution_control_enum_to_int16 (ExecutionControlEnum mode)`

*Convert an `ExecutionModeEnum` value into a 16 bit integer.*
- `ExecutionControlEnum TrickHLA::execution_control_int16_to_enum (int16_t int_mode)`

*Convert a 16 bit integer to an `ExecutionModeEnum` value.*
- `std::string TrickHLA::mode_transition_enum_to_string (ModeTransitionEnum mode)`

*Convert an `ModeTransitionEnum` value into a printable string.*
- `int16_t TrickHLA::mode_transition_enum_to_int16 (ModeTransitionEnum mode)`

*Convert an `ModeTransitionEnum` value into a 16 bit integer.*
- `ModeTransitionEnum TrickHLA::mode_transition_int16_to_enum (int16_t int_mode)`

*Convert a 16 bit integer to an `ModeTransitionEnum` value.*
- `std::string TrickHLA::sync_pnt_state_enum_to_string (SyncPntStateEnum state)`

*Convert a `Synchronization Point State` enum value into a printable string.*
- `int16_t TrickHLA::sync_pnt_state_enum_to_int16 (SyncPntStateEnum state)`

*Convert a `Synchronization Point State` enum value into a 16 bit integer.*
- `SyncPntStateEnum TrickHLA::sync_pnt_state_int16_to_enum (int16_t int_state)`

*Convert an integer value to a `Synchronization Point State` enumeration value.*

### 8.143.1 Detailed Description

Definition of the [TrickHLA](#) enumeration types and utilities.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: *trick.TrickHLA*

#### Link Dependencies

- ..../source/TrickHLA/Types.cpp

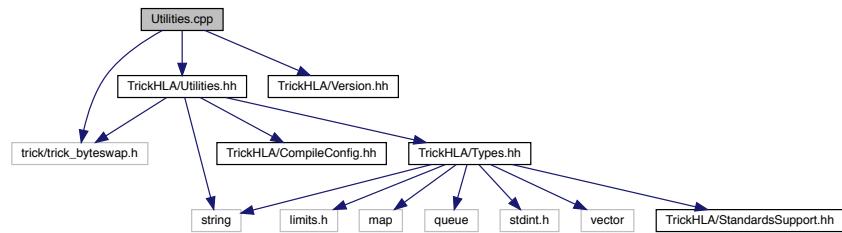
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 2 origin.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

#### Revision History

## 8.144 Utilities.cpp File Reference

Implementation of the [TrickHLA](#) utilities.

```
#include "trick/trick_bytswap.h"
#include "TrickHLA/Utilities.hh"
#include "TrickHLA/Version.hh"
Include dependency graph for Utilities.cpp:
```



### 8.144.1 Detailed Description

Implementation of the [TrickHLA](#) utilities.

**Copyright**

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

**Responsible Organization**

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

**Link Dependencies**

- [Utilities.cpp](#)

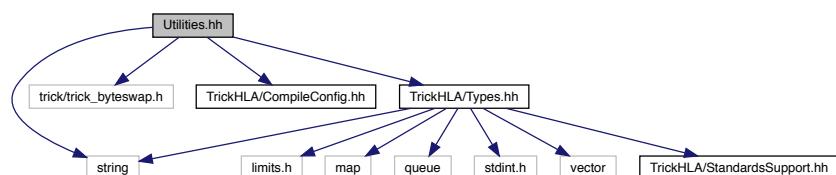
Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">TrickHLA</a>	Aug 2006	–	<a href="#">DSES TrickHLA Utilities.</a>
Dan Dexter	NASA/ER7	<a href="#">TrickHLA</a>	Sept 2010	–	Added Mac FPU control word support.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	March 2019	–	Version 3 rewrite.

**Revision History**

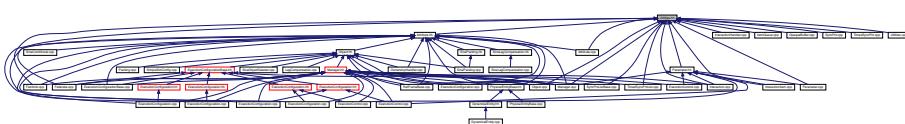
## 8.145 Utilities.hh File Reference

Definition of the [TrickHLA](#) utilities.

```
#include <string>
#include "trick/trick_bytswap.h"
#include "TrickHLA/CompileConfig.hh"
#include "TrickHLA/Types.hh"
Include dependency graph for Utilities.hh:
```



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



## Data Structures

- class [TrickHLA::Utilities](#)

## Namespaces

- [TrickHLA](#)

## Macros

- `#define _FPU_PC_MASK 0x300`
- `#define _FPU_PC_EXTENDED 0x300`
- `#define _FPU_PC_DOUBLE 0x200`
- `#define _FPU_PC_UNDEFINED 0x100`
- `#define _FPU_PC_SINGLE 0x000`
- `#define TRICKHLA_INIT_FPU_CONTROL_WORD`
- `#define TRICKHLA_SAVE_FPU_CONTROL_WORD`
- `#define TRICKHLA_RESTORE_FPU_CONTROL_WORD`
- `#define TRICKHLA_VALIDATE_FPU_CONTROL_WORD`

### 8.145.1 Detailed Description

Definition of the [TrickHLA](#) utilities.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

#### Python Module: `trick.TrickHLA`

#### Link Dependencies

- `../source/TrickHLA/Utilities.cpp`

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	L3 Titan Group	<a href="#">TrickHLA</a>	Aug 2006	–	DSES <a href="#">TrickHLA</a> Utilities
Dan Dexter	L3	<a href="#">TrickHLA</a>	May 2008	–	<a href="#">IMSim</a> : Added FPU code word protection macros
Dan Dexter	NASA/ER7	<a href="#">TrickHLA</a>	December 2008	–	<a href="#">IMSim</a> : Added MacOS X support.
Dan Dexter	NASA/ER7	<a href="#">TrickHLA</a>	Sept 2010	–	Added Mac FPU control word support.
Danny Strauss	L3	<a href="#">TrickHLA</a>	June 2012	–	Add version to THLA simobject
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

## Revision History

### 8.145.2 Macro Definition Documentation

#### 8.145.2.1 `_FPU_PC_DOUBLE`

```
#define _FPU_PC_DOUBLE 0x200
Definition at line 59 of file Utilities.hh.
```

#### 8.145.2.2 `_FPU_PC_EXTENDED`

```
#define _FPU_PC_EXTENDED 0x300
Definition at line 58 of file Utilities.hh.
```

#### 8.145.2.3 `_FPU_PC_MASK`

```
#define _FPU_PC_MASK 0x300
Definition at line 57 of file Utilities.hh.
```

#### 8.145.2.4 `_FPU_PC_SINGLE`

```
#define _FPU_PC_SINGLE 0x000
Definition at line 61 of file Utilities.hh.
```

#### 8.145.2.5 `_FPU_PC_UNDEFINED`

```
#define _FPU_PC_UNDEFINED 0x100
Definition at line 60 of file Utilities.hh.
```

#### 8.145.2.6 `TRICKHLA_INIT_FPU_CONTROL_WORD`

```
#define TRICKHLA_INIT_FPU_CONTROL_WORD
Definition at line 113 of file Utilities.hh.
```

#### 8.145.2.7 `TRICKHLA_RESTORE_FPU_CONTROL_WORD`

```
#define TRICKHLA_RESTORE_FPU_CONTROL_WORD
Definition at line 115 of file Utilities.hh.
```

#### 8.145.2.8 `TRICKHLA_SAVE_FPU_CONTROL_WORD`

```
#define TRICKHLA_SAVE_FPU_CONTROL_WORD
Definition at line 114 of file Utilities.hh.
```

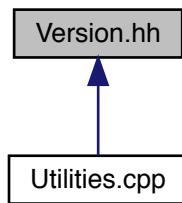
### 8.145.2.9 TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD

```
#define TRICKHLA_VALIDATE_FPU_CONTROL_WORD
Definition at line 116 of file Utilities.hh.
```

## 8.146 Version.hh File Reference

Definition of the [TrickHLA](#) version tag.

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



### Macros

- `#define TRICKHLA_VERSION "d3.0.0"`
- `#define TRICKHLA_RELEASE_DATE "2019-06-30"`

### 8.146.1 Detailed Description

Definition of the [TrickHLA](#) version tag.

#### Copyright

Copyright 2019 United States Government as represented by the Administrator of the National Aeronautics and Space Administration. No copyright is claimed in the United States under Title 17, U.S. Code. All Other Rights Reserved.

#### Responsible Organization

Simulation and Graphics Branch, Mail Code ER7  
 Software, Robotics & Simulation Division  
 NASA, Johnson Space Center  
 2101 NASA Parkway, Houston, TX 77058

Author	Organization	Project	Date	Rev. ID	Description
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	December 2012	–	Version 2 origin.
Dan Dexter	NASA ER7	<a href="#">TrickHLA</a>	January 2015	–	Added release date.
Edwin Z. Crues	NASA ER7	<a href="#">TrickHLA</a>	June 2019	–	Version 3 rewrite.

---

## Revision History

### 8.146.2 Macro Definition Documentation

#### 8.146.2.1 TRICKHLA\_RELEASE\_DATE

```
#define TRICKHLA_RELEASE_DATE "2019-06-30"
```

Definition at line 32 of file Version.hh.

#### 8.146.2.2 TRICKHLA\_VERSION

```
#define TRICKHLA_VERSION "d3.0.0"
```

Definition at line 31 of file Version.hh.

# Index

\_FPU\_PC\_DOUBLE  
    Utilities.hh, [982](#)  
\_FPU\_PC\_EXTENDED  
    Utilities.hh, [982](#)  
\_FPU\_PC\_MASK  
    Utilities.hh, [982](#)  
\_FPU\_PC\_SINGLE  
    Utilities.hh, [982](#)  
\_FPU\_PC\_UNDEFINED  
    Utilities.hh, [982](#)  
\_HLAinterval  
    TrickHLA::Int64Interval, [418](#)  
\_HLAtime  
    TrickHLA::Int64Time, [433](#)  
\_TRICKHLA\_OBJECT\_HH\_  
    Object.hh, [880](#)  
~Attribute  
    TrickHLA::Attribute, [64](#)  
~BasicClock  
    TrickHLA::BasicClock, [86](#)  
~CTETimelineBase  
    TrickHLA::CTETimelineBase, [91](#)  
~Conditional  
    TrickHLA::Conditional, [88](#)  
~DebugHandler  
    TrickHLA::DebugHandler, [94](#)  
~DynamicalEntity  
    SpaceFOM::DynamicalEntity, [100](#)  
~ExecutionConfiguration  
    DIS::ExecutionConfiguration, [105](#)  
    DSES::ExecutionConfiguration, [116](#)  
    IMSim::ExecutionConfiguration, [126](#)  
    SpaceFOM::ExecutionConfiguration, [139](#)  
    TrickHLA::ExecutionConfiguration, [150](#)  
~ExecutionConfigurationBase  
    TrickHLA::ExecutionConfigurationBase, [156](#)  
~ExecutionControl  
    DIS::ExecutionControl, [167](#)  
    DSES::ExecutionControl, [185](#)  
    IMSim::ExecutionControl, [201](#)  
    SpaceFOM::ExecutionControl, [225](#)  
    TrickHLA::ExecutionControl, [247](#)  
~ExecutionControlBase  
    TrickHLA::ExecutionControlBase, [264](#)  
~FedAmb  
    TrickHLA::FedAmb, [307](#)  
~Federate  
    TrickHLA::Federate, [337](#)  
~FreezeInteractionHandler  
    IMSim::FreezeInteractionHandler, [408](#)  
~Int64Interval  
    TrickHLA::Int64Interval, [414](#)  
~Int64Time  
    TrickHLA::Int64Time, [422](#)  
~Interaction  
    TrickHLA::Interaction, [437](#)  
~InteractionHandler  
    TrickHLA::InteractionHandler, [453](#)  
~InteractionItem  
    TrickHLA::InteractionItem, [462](#)  
~Item  
    TrickHLA::Item, [467](#)  
~ItemQueue  
    TrickHLA::ItemQueue, [470](#)  
~KnownFederate  
    TrickHLA::KnownFederate, [475](#)  
~LagCompensation  
    TrickHLA::LagCompensation, [479](#)  
~LoggableSyncPnt  
    TrickHLA::LoggableSyncPnt, [484](#)  
~LoggableTimedSyncPnt  
    TrickHLA::LoggableTimedSyncPnt, [487](#)  
~MTRInteractionHandler  
    SpaceFOM::MTRInteractionHandler, [526](#)  
~Manager  
    TrickHLA::Manager, [494](#)  
~Object  
    TrickHLA::Object, [538](#)  
~ObjectDeleted  
    TrickHLA::ObjectDeleted, [575](#)  
~OpaqueBuffer  
    TrickHLA::OpaqueBuffer, [578](#)  
~OwnershipHandler  
    TrickHLA::OwnershipHandler, [587](#)  
~OwnershipItem  
    TrickHLA::OwnershipItem, [598](#)  
~Packing  
    TrickHLA::Packing, [601](#)  
~Parameter  
    TrickHLA::Parameter, [609](#)

~ParameterItem  
     TrickHLA::ParameterItem, 625  
 ~PausePointList  
     DIS::PausePointList, 635  
     IMSim::PausePointList, 629  
 ~PhysicalEntityBase  
     SpaceFOM::PhysicalEntityBase, 641  
 ~RefFrameBase  
     SpaceFOM::RefFrameBase, 653  
 ~ReflectedAttributesQueue  
     TrickHLA::ReflectedAttributesQueue, 661  
 ~ScenarioTimeline  
     TrickHLA::ScenarioTimeline, 665  
 ~SimTimeline  
     TrickHLA::SimTimeline, 675  
 ~SimpleSimConfig  
     TrickHLAModel::SimpleSimConfig, 671  
 ~SineConditional  
     TrickHLAModel::SineConditional, 678  
 ~SineData  
     TrickHLAModel::SineData, 683  
 ~SineInteractionHandler  
     TrickHLAModel::SineInteractionHandler, 693  
 ~SineLagCompensation  
     TrickHLAModel::SineLagCompensation, 697  
 ~SineObjectDeleted  
     TrickHLAModel::SineObjectDeleted, 702  
 ~SineOwnershipHandler  
     TrickHLAModel::SineOwnershipHandler, 705  
 ~SinePacking  
     TrickHLAModel::SinePacking, 709  
 ~StringUtilities  
     TrickHLA::StringUtilities, 722  
 ~SyncPnt  
     TrickHLA::SyncPnt, 732  
 ~SyncPntList  
     TrickHLA::SyncPntList, 741  
 ~SyncPntListBase  
     TrickHLA::SyncPntListBase, 746  
 ~TimeOfDayTimeline  
     TrickHLA::TimeOfDayTimeline, 775  
 ~TimedSyncPnt  
     TrickHLA::TimedSyncPnt, 762  
 ~TimedSyncPntList  
     TrickHLA::TimedSyncPntList, 766  
 ~Timeline  
     TrickHLA::Timeline, 771  
 ~Utilities  
     TrickHLA::Utilities, 777  
  
 accel  
     SpaceFOM::PhysicalEntityBase, 643  
 achieve\_all\_multiphase\_init\_sync\_pnts  
     DSES::ExecutionControl, 185  
  
 IMSim::ExecutionControl, 201  
 TrickHLA::ExecutionControlBase, 264  
 achieve\_all\_sync\_pnts  
     TrickHLA::SyncPntListBase, 746  
     TrickHLA::TimedSyncPntList, 767  
 achieve\_and\_wait\_for\_synchronization  
     TrickHLA::Federate, 337  
     TrickHLA::SyncPntListBase, 747  
 achieve\_sync\_pnt  
     TrickHLA::SyncPntListBase, 747  
 achieve\_sync\_point  
     TrickHLA::SyncPnt, 732  
 achieve\_synchronization\_point  
     TrickHLA::Federate, 338  
 add\_a\_single\_entry\_into\_running\_feds  
     TrickHLA::Federate, 338  
 add\_federate\_instance\_id  
     TrickHLA::Federate, 338  
 add\_freeze\_scenario\_time  
     IMSim::ExecutionControl, 202  
 add\_initialization\_sync\_points  
     SpaceFOM::ExecutionControl, 225  
     TrickHLA::ExecutionControl, 247  
     TrickHLA::ExecutionControlBase, 264  
 add\_MOM\_HLAfederate\_instance\_id  
     TrickHLA::Federate, 338  
 add\_MOM\_HLAfederation\_instance\_id  
     TrickHLA::Federate, 339  
 add\_multiphase\_init\_sync\_points  
     DIS::ExecutionControl, 167  
     DSES::ExecutionControl, 185  
     IMSim::ExecutionControl, 202  
     TrickHLA::ExecutionControl, 247  
     TrickHLA::ExecutionControlBase, 264  
 add\_object\_to\_map  
     TrickHLA::ExecutionControlBase, 265  
     TrickHLA::Manager, 495  
 add\_pause  
     DIS::ExecutionControl, 167  
     IMSim::ExecutionControl, 202  
 add\_sync\_pnt  
     TrickHLA::SyncPntListBase, 748  
     TrickHLA::TimedSyncPntList, 767, 768  
 address  
     TrickHLA::Parameter, 619  
 adjust\_phase  
     TrickHLAModel::SineData, 683  
 alignment  
     TrickHLA::OpaqueBuffer, 583  
 all\_federates\_joined  
     TrickHLA::Federate, 389  
 allocate\_input\_string  
     SpaceFOM::RefFrameBase, 653  
 amp

TrickHLAModel::SineData, 689  
amp\_attr  
TrickHLAModel::SineLagCompensation, 699  
TrickHLAModel::SinePacking, 711  
ang\_vel  
SpaceTimeCoordinateData, 714  
ang\_vel\_encoder  
SpaceFOM::SpaceTimeCoordinateEncoder, 718  
angular\_velocity  
SpaceFOM::SpaceTimeCoordinateEncoder, 718  
announce\_freeze  
TrickHLA::Federate, 389  
announce\_restore  
TrickHLA::Federate, 389  
announce\_save  
TrickHLA::Federate, 389  
announce\_sync\_point  
DIS::ExecutionControl, 168  
DSES::ExecutionControl, 186  
IMSim::ExecutionControl, 202  
SpaceFOM::ExecutionControl, 226  
TrickHLA::ExecutionControl, 248  
TrickHLA::Federate, 339  
TrickHLA::SyncPntListBase, 748  
announceSynchronizationPoint  
TrickHLA::FedAmb, 307  
any\_attribute\_FOM\_specified\_order  
TrickHLA::Object, 567  
any\_attribute\_published  
TrickHLA::Object, 538  
any\_attribute\_subscribed  
TrickHLA::Object, 538  
any\_attribute\_timestamp\_order  
TrickHLA::Object, 568  
any\_locally\_owned\_attribute  
TrickHLA::Object, 539  
any\_locally\_owned\_published\_attribute  
TrickHLA::Object, 539  
any\_locally\_owned\_published\_cyclic\_data\_ready\_attribute  
TrickHLA::Object, 540  
any\_locally\_owned\_published\_init\_attribute  
TrickHLA::Object, 540  
any\_locally\_owned\_published\_requested\_attribute  
TrickHLA::Object, 540  
any\_remotely\_owned\_subscribed\_attribute  
TrickHLA::Object, 540  
any\_remotely\_owned\_subscribed\_cyclic\_attribute  
TrickHLA::Object, 541  
any\_remotely\_owned\_subscribed\_init\_attribute  
TrickHLA::Object, 541  
ask\_MOM\_for\_auto\_provide\_setting  
TrickHLA::Federate, 339  
ask\_MOM\_for\_federate\_names  
TrickHLA::Federate, 339  
attr  
TrickHLA::Parameter, 620  
attr\_count  
TrickHLA::Object, 568  
attr\_handle  
TrickHLA::Attribute, 80  
attr\_pos  
TrickHLAModel::SineConditional, 680  
attr\_update\_requested  
TrickHLA::Object, 568  
attrDIS\_\_ExecutionConfiguration  
DIS/ExecutionConfiguration.cpp, 801  
attrDSES\_\_ExecutionConfiguration  
DSES/ExecutionConfiguration.cpp, 803  
Attribute  
TrickHLA::Attribute, 64, 65  
Attribute.cpp, 783  
Attribute.hh, 784  
attribute\_FOM\_names  
TrickHLA::Object, 568  
attribute\_map\_queue  
TrickHLA::ReflectedAttributesQueue, 663  
attribute\_values\_map  
TrickHLA::Object, 569  
attributelsNotOwned  
TrickHLA::FedAmb, 307  
attributelsOwnedByRTI  
TrickHLA::FedAmb, 307  
AttributeMap  
TrickHLA, 47  
attributeOwnershipAcquisitionNotification  
TrickHLA::FedAmb, 308  
AttributeOwnershipMap  
TrickHLA, 47  
attributeOwnershipUnavailable  
TrickHLA::FedAmb, 308  
attributes  
TrickHLA::Object, 569  
attributesInScope  
TrickHLA::FedAmb, 308  
attributesOutOfScope  
TrickHLA::FedAmb, 308  
attrIMSim\_\_ExecutionConfiguration  
IMSim/ExecutionConfiguration.cpp, 804  
attrParameterItem  
InteractionItem.cpp, 857  
attrSpaceFOM\_\_ExecutionConfiguration  
SpaceFOM/ExecutionConfiguration.cpp, 806  
attrSpaceFOM\_\_MTRInteractionHandler  
SpaceFOM/ExecutionControl.cpp, 824  
attrSRFOM\_\_MTRInteractionHandler  
Manager.cpp, 871  
attrTrickHLA\_\_ExecutionConfiguration  
TrickHLA/ExecutionConfiguration.cpp, 807

attrTrickHLA\_\_ExecutionConfigurationBase  
     ExecutionConfigurationBase.cpp, 817

attrTrickHLA\_\_FreezeInteractionHandler  
     IMSim/ExecutionControl.cpp, 822

attrTrickHLA\_\_Interaction  
     Manager.cpp, 871

attrTrickHLA\_\_InteractionItem  
     Manager.cpp, 871

attrTrickHLA\_\_MTRInteractionHandler  
     TrickHLA/ExecutionControl.cpp, 825

attrTrickHLA\_\_Parameter  
     Manager.cpp, 871

auto\_provide\_setting  
     TrickHLA::Federate, 390

backup\_auto\_provide\_setting\_from\_MOM\_then\_disable  
     TrickHLA::Federate, 340

BasicClock  
     TrickHLA::BasicClock, 86

BasicClock.hh, 786

blocking\_cyclic\_read  
     TrickHLA::Object, 569

body\_wrt\_struct  
     SpaceFOM::PhysicalEntityBase, 643

buff  
     TrickHLAModel::SinePacking, 711

buff\_size  
     TrickHLAModel::SinePacking, 711

buffer  
     TrickHLA::Attribute, 81  
     TrickHLA::OpaqueBuffer, 583  
     TrickHLA::Parameter, 620

buffer\_capacity  
     TrickHLA::Attribute, 81  
     TrickHLA::Parameter, 620

build\_attribute\_map  
     TrickHLA::Object, 541

byteswap  
     TrickHLA::Attribute, 81  
     TrickHLA::Parameter, 620

byteswap\_buffer\_copy  
     TrickHLA::Attribute, 65  
     TrickHLA::OpaqueBuffer, 578  
     TrickHLA::Parameter, 609

byteswap\_double  
     TrickHLA::Utilities, 777

byteswap\_float  
     TrickHLA::Utilities, 778

byteswap\_int  
     TrickHLA::Utilities, 778

byteswap\_long  
     TrickHLA::Utilities, 778

byteswap\_long\_long  
     TrickHLA::Utilities, 779

byteswap\_short  
     TrickHLA::Utilities, 779

byteswap\_unsigned\_int  
     TrickHLA::Utilities, 779

byteswap\_unsigned\_long  
     TrickHLA::Utilities, 780

byteswap\_unsigned\_long\_long  
     TrickHLA::Utilities, 780

byteswap\_unsigned\_short  
     TrickHLA::Utilities, 780

calculate\_number\_of\_items  
     TrickHLA::Attribute, 65  
     TrickHLA::Parameter, 610

calculate\_size\_and\_number\_of\_items  
     TrickHLA::Attribute, 66  
     TrickHLA::Parameter, 610

calculate\_static\_number\_of\_items  
     TrickHLA::Attribute, 66  
     TrickHLA::Parameter, 611

can\_rejoin\_federation  
     TrickHLA::Federate, 390

capacity  
     TrickHLA::OpaqueBuffer, 583

changed  
     TrickHLA::Interaction, 448  
     TrickHLA::Object, 569

check\_data\_cycle\_ready  
     TrickHLA::Attribute, 66

check\_for\_shutdown  
     SpaceFOM::ExecutionControl, 226  
     TrickHLA::ExecutionControlBase, 265  
     TrickHLA::Federate, 340

check\_for\_shutdown\_with\_termination  
     SpaceFOM::ExecutionControl, 226  
     TrickHLA::ExecutionControlBase, 265  
     TrickHLA::Federate, 340

check\_freeze  
     DIS::ExecutionControl, 168  
     TrickHLA::Federate, 341

check\_freeze\_exit  
     DIS::ExecutionControl, 168, 169  
     DSES::ExecutionControl, 186  
     IMSim::ExecutionControl, 203  
     SpaceFOM::ExecutionControl, 227  
     TrickHLA::ExecutionControlBase, 265

check\_freeze\_time  
     IMSim::ExecutionControl, 203

check\_HLA\_save\_directory  
     TrickHLA::Federate, 341

check\_interactions  
     TrickHLA::Manager, 519

check\_interactions\_count  
     TrickHLA::Manager, 519

check\_mode\_transition\_request  
DIS::ExecutionControl, 169  
DSES::ExecutionControl, 186  
IMSim::ExecutionControl, 204  
SpaceFOM::ExecutionControl, 227

check\_pause  
IMSim::ExecutionControl, 204  
TrickHLA::ExecutionControlBase, 266

check\_pause\_at\_init  
IMSim::ExecutionControl, 204, 205  
TrickHLA::ExecutionControlBase, 266

check\_scenario\_freeze\_time  
IMSim::ExecutionControl, 205

check\_state  
DIS::PausePointList, 636  
IMSim::PausePointList, 630

check\_sync\_pnts  
TrickHLA::TimedSyncPntList, 768

check\_values  
IMSim::FreezeInteractionHandler, 408

checkpoint\_file\_name  
TrickHLA::Federate, 390

checkpoint\_queue  
TrickHLA::InteractionItem, 462

checkpoint\_rt\_itimer  
TrickHLA::Federate, 390

checktime  
DIS::ExecutionControl, 180  
IMSim::ExecutionControl, 219

class\_handle  
TrickHLA::Interaction, 449  
TrickHLA::Object, 569

clear  
TrickHLA::LoggableSyncPnt, 484  
TrickHLA::OwnershipItem, 598  
TrickHLA::ParameterItem, 625  
TrickHLA::ReflectedAttributesQueue, 661

clear\_checkpoint  
TrickHLA::OwnershipHandler, 587

clear\_init\_sync\_points  
TrickHLA::Manager, 495

clear\_interactions  
TrickHLA::Manager, 495

clear\_mode\_transition\_requested  
TrickHLA::ExecutionControlBase, 266

clear\_mode\_values  
DIS::ExecutionControl, 169  
DSES::ExecutionControl, 187  
IMSim::ExecutionControl, 205  
TrickHLA::ExecutionControlBase, 267

clear\_multiphase\_init\_sync\_points  
DIS::ExecutionControl, 169  
TrickHLA::ExecutionControl, 248  
TrickHLA::ExecutionControlBase, 267

clear\_parm\_items  
TrickHLA::InteractionItem, 462

clear\_pause  
IMSim::ExecutionControl, 205

clear\_running\_feds  
TrickHLA::Federate, 341

clear\_state  
DIS::PausePointList, 636  
IMSim::PausePointList, 630

clear\_sync\_pnt  
DIS::PausePointList, 636  
IMSim::PausePointList, 630  
TrickHLA::SyncPntListBase, 749

clear\_update\_pending  
TrickHLA::ExecutionConfigurationBase, 156

clk\_id  
TrickHLA::CTETimelineBase, 93

clock  
TrickHLA::Object, 570

clock\_init  
TrickHLA::CTETimelineBase, 91

clock\_stop  
TrickHLA::CTETimelineBase, 92

cm  
SpaceFOM::PhysicalEntityBase, 644

code\_section  
TrickHLA::DebugHandler, 97

CompileConfig.hh, 787

MIN\_TRICK\_MINOR, 788  
MIN\_TRICK\_PATCH, 788  
MIN\_TRICK\_VER, 789  
NO\_THLA\_THREAD\_WAIT\_FOR\_DATA, 789  
NO\_TRICKHLA\_ENABLE\_FOM\_DUMP, 789  
NO\_TRICKHLA\_ENABLE\_FPU\_CONTROL\_WORD\_VALIDATION, 789  
THLA\_10SEC\_TIMEOUT WHILE\_WAITING\_FOR\_DATA, 789  
THLA\_ENDL, 789  
THLA\_INTRAFRAME\_BLOCKING\_READ\_CONFIG, 789  
THLA\_NEWLINE, 789  
THLA\_QUEUE\_REFLECTED\_ATTRIBUTES, 789  
THLA\_THREAD\_TIMED\_WAIT\_FOR\_DATA, 790

complete\_initialization  
TrickHLA::Parameter, 611

complete\_restore  
TrickHLA::Federate, 341

compute\_derivative  
TrickHLAModel::SineData, 683, 684

compute\_HLT  
TrickHLA::ScenarioTimeline, 666

compute\_simulation\_time  
TrickHLA::ScenarioTimeline, 666

compute\_value

TrickHLAModel::SineData, 684  
 Conditional  
   TrickHLA::Conditional, 88  
 conditional  
   TrickHLA::Attribute, 81  
 Conditional.cpp, 790  
 Conditional.hh, 791  
 config  
   TrickHLA::Attribute, 81  
 CONFIG\_CYCLIC  
   TrickHLA, 48  
 CONFIG\_INITIALIZE  
   TrickHLA, 48  
 CONFIG\_INTERMITTENT  
   TrickHLA, 48  
 CONFIG\_MAX\_VALUE  
   TrickHLA, 48  
 CONFIG\_NONE  
   TrickHLA, 48  
 configure  
   SpaceFOM::ExecutionConfiguration, 139  
   TrickHLA::ExecutionConfiguration, 150  
   TrickHLA::ExecutionConfigurationBase, 156  
 configure\_attributes  
   DIS::ExecutionConfiguration, 106  
   DSES::ExecutionConfiguration, 116  
   IMSim::ExecutionConfiguration, 127  
   SpaceFOM::ExecutionConfiguration, 139  
   TrickHLA::ExecutionConfiguration, 151  
   TrickHLA::ExecutionConfigurationBase, 157  
 confirmAttributeOwnershipAcquisitionCancellation  
   TrickHLA::FedAmb, 308  
 confirmAttributeTransportationTypeChange  
   TrickHLA::FedAmb, 309  
 confirmInteractionTransportationTypeChange  
   TrickHLA::FedAmb, 309  
 connectionLost  
   TrickHLA::FedAmb, 309  
 Constants.hh, 792  
 contains  
   TrickHLA::SyncPntListBase, 749  
 convert  
   TrickHLA::SyncPnt, 733  
   TrickHLA::TimedSyncPnt, 762  
 convert\_FOM\_name\_to\_pos  
   TrickHLAModel::SineConditional, 678  
 convert\_loggable\_sync\_pts  
   IMSim::ExecutionControl, 206  
   TrickHLA::ExecutionControlBase, 267  
 convert\_scenario\_time\_to\_sim\_time  
   TrickHLA::ExecutionControlBase, 267  
 convert\_sim\_time\_to\_scenario\_time  
   TrickHLA::ExecutionControlBase, 268  
 convert\_sync\_pts  
   TrickHLA::Federate, 341  
   TrickHLA::SyncPntListBase, 749  
   TrickHLA::TimedSyncPntList, 768  
 copy\_data  
   TrickHLAModel::SineData, 684  
 copy\_running\_feds\_into\_known\_feds  
   TrickHLA::Federate, 342  
 count  
   TrickHLA::ItemQueue, 474  
 create\_and\_join\_federation  
   TrickHLA::Federate, 342  
 create\_attribute\_set  
   TrickHLA::Object, 541  
 create\_federation  
   TrickHLA::Federate, 342  
 create\_HLA\_instance  
   TrickHLA::Object, 570  
 create\_requested\_attribute\_set  
   TrickHLA::Object, 542  
 create\_RTI\_ambassador\_and\_connect  
   TrickHLA::Federate, 342  
 cstr\_restore\_label  
   TrickHLA::Federate, 390  
 cstr\_save\_label  
   TrickHLA::Federate, 390  
 cte\_time  
   SpaceFOM::MTRInteractionHandler, 528  
 cte\_timeline  
   TrickHLA::ExecutionControlBase, 294  
 CTETimelineBase  
   TrickHLA::CTETimelineBase, 91  
 CTETimelineBase.cpp, 793  
 CTETimelineBase.hh, 794  
 current\_execution\_control\_mode  
   TrickHLA::ExecutionControlBase, 295  
 current\_execution\_mode  
   DIS::ExecutionConfiguration, 111  
   DSES::ExecutionConfiguration, 122  
   IMSim::ExecutionConfiguration, 134  
   SpaceFOM::ExecutionConfiguration, 146  
 cycle\_cnt  
   TrickHLA::Attribute, 81  
 cycle\_ratio  
   TrickHLA::Attribute, 82  
 cycle\_time  
   TrickHLA::Attribute, 82  
 data  
   SpaceFOM::QuaternionEncoder, 649  
   SpaceFOM::SpaceTimeCoordinateEncoder, 719  
   TrickHLA::ParameterItem, 626  
 data\_changed  
   TrickHLA::Object, 570  
 DataUpdateEnum

TrickHLA, 48  
debug  
SpaceFOM::RefFrameBase, 657  
debug\_handler  
TrickHLA::Attribute, 82  
TrickHLA::Manager, 519  
TrickHLA::Parameter, 620  
TrickHLA::SyncPntListBase, 758  
debug\_level  
TrickHLA::DebugHandler, 97  
DEBUG\_LEVEL\_0\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_10\_TRACE  
TrickHLA, 49  
DEBUG\_LEVEL\_11\_TRACE  
TrickHLA, 49  
DEBUG\_LEVEL\_1\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_2\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_3\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_4\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_5\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_6\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_7\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_8\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_9\_TRACE  
TrickHLA, 48  
DEBUG\_LEVEL\_FULL\_TRACE  
TrickHLA, 49  
DEBUG\_LEVEL\_NO\_TRACE  
TrickHLA, 48  
DEBUG\_SOURCE\_ALL\_MODULES  
TrickHLA, 49  
DEBUG\_SOURCE\_ATTRIBUTE  
TrickHLA, 49  
DEBUG\_SOURCE\_FED\_AMB  
TrickHLA, 49  
DEBUG\_SOURCE\_FEDERATE  
TrickHLA, 49  
DEBUG\_SOURCE\_INTERACTION  
TrickHLA, 49  
DEBUG\_SOURCE\_LAG\_COMPENSATION  
TrickHLA, 49  
DEBUG\_SOURCE\_MANAGER  
TrickHLA, 49  
DEBUG\_SOURCE\_OBJECT  
TrickHLA, 49  
DEBUG\_SOURCE\_OWNERSHIP  
TrickHLA, 49  
DEBUG\_SOURCE\_PACKING  
TrickHLA, 49  
DEBUG\_SOURCE\_PARAMETER  
TrickHLA, 49  
DEBUG\_SOURCE\_SYNCPOINT  
TrickHLA, 49  
DebugHandler  
TrickHLA::DebugHandler, 94  
DebugHandler.hh, 796  
DebugLevelEnum  
TrickHLA, 48  
DebugSourceEnum  
TrickHLA, 49  
decode  
SpaceFOM::QuaternionEncoder, 648  
SpaceFOM::SpaceTimeCoordinateEncoder, 717  
TrickHLA::Int64Time, 422  
decode\_boolean\_from\_buffer  
TrickHLA::Attribute, 66  
TrickHLA::Parameter, 611  
decode\_logical\_time  
TrickHLA::Attribute, 66  
TrickHLA::Parameter, 611  
decode\_opaque\_data\_from\_buffer  
TrickHLA::Attribute, 67  
TrickHLA::Parameter, 611  
decode\_string\_from\_buffer  
TrickHLA::Attribute, 67  
TrickHLA::Parameter, 612  
def\_scenario\_timeline  
TrickHLA, 56  
def\_sim\_timeline  
TrickHLA, 56  
default\_data  
SpaceFOM::RefFrameBase, 653  
deleted  
TrickHLA::Object, 570  
TrickHLA::ObjectDeleted, 576  
TrickHLAModel::SineObjectDeleted, 703  
deriv  
integrator.d, 850  
destroy  
TrickHLA::Federate, 343  
destroy\_orphaned\_federation  
TrickHLA::Federate, 343  
determine\_cycle\_ratio  
TrickHLA::Attribute, 67  
determine\_federate\_MOM\_object\_instance\_names  
TrickHLA::Federate, 343  
determine\_federation\_master  
DIS::ExecutionControl, 169  
DSES::ExecutionControl, 187

determine\_if\_late\_joining\_or\_restoring\_federate  
 IMSim::ExecutionControl, 206  
 determine\_job\_cycle\_time  
 TrickHLA::Manager, 495  
 DIS, 17  
 EXECUTION\_MODE\_LAST\_VALUE, 19  
 execution\_mode\_enum\_to\_int16, 20  
 execution\_mode\_enum\_to\_string, 20  
 EXECUTION\_MODE\_FIRST\_VALUE, 18  
 EXECUTION\_MODE\_FREEZE, 19  
 EXECUTION\_MODE\_INITIALIZING, 18  
 execution\_mode\_int16\_to\_enum, 20  
 EXECUTION\_MODE\_RUNNING, 18  
 EXECUTION\_MODE\_SHUTDOWN, 19  
 EXECUTION\_MODE\_UNINITIALIZED, 18  
 ExecutionModeEnum, 18  
 from\_execution\_control\_enum, 21  
 INITIALIZE\_SYNC\_POINT, 24  
 mtr\_enum\_to\_int16, 21  
 mtr\_enum\_to\_string, 21  
 MTR\_FIRST\_VALUE, 19  
 MTR\_GOTO\_FREEZE, 19  
 MTR\_GOTO\_RUN, 19  
 MTR\_GOTO\_SHUTDOWN, 19  
 MTR\_INITIALIZING, 19  
 mtr\_int16\_to\_enum, 22  
 MTR\_LAST\_VALUE, 19  
 MTR\_UNINITIALIZED, 19  
 MTREnum, 19  
 pause\_pnt\_state\_enum\_to\_int16, 22  
 pause\_pnt\_state\_enum\_to\_string, 23  
 pause\_pnt\_state\_int16\_to\_enum, 23  
 PAUSE\_POINT\_STATE\_ACKNOWLEDGED, 19  
 PAUSE\_POINT\_STATE\_ERROR, 19  
 PAUSE\_POINT\_STATE\_EXIT, 19  
 PAUSE\_POINT\_STATE\_FIRST\_VALUE, 19  
 PAUSE\_POINT\_STATE\_FREEZE, 19  
 PAUSE\_POINT\_STATE\_PENDING, 19  
 PAUSE\_POINT\_STATE\_RECONFIG, 19  
 PAUSE\_POINT\_STATE\_RESTART, 19  
 PAUSE\_POINT\_STATE\_RUN, 19  
 PAUSE\_POINT\_STATE\_UNKNOWN, 19  
 PausePointStateEnum, 19  
 STARTUP\_FREEZE\_SYNC\_POINT, 24  
 STARTUP\_SYNC\_POINT, 24  
 to\_execution\_control\_enum, 23  
 DIS/ExecutionConfiguration.cpp  
 attrDIS\_\_ExecutionConfiguration, 801  
 DIS::ExecutionConfiguration, 103  
 ~ExecutionConfiguration, 105  
 configure\_attributes, 106  
 current\_execution\_mode, 111  
 ExecutionConfiguration, 105  
 get\_current\_execution\_mode, 106  
 get\_next\_execution\_mode, 106  
 get\_next\_mode\_cte\_time, 106  
 get\_next\_mode\_scenario\_time, 107  
 get\_root\_frame\_name, 107  
 get\_scenario\_time\_epoch, 107  
 init\_attrDIS\_\_ExecutionConfiguration, 111  
 InputProcessor, 111  
 next\_execution\_mode, 112  
 next\_mode\_cte\_time, 112  
 next\_mode\_scenario\_time, 112  
 operator=, 107  
 pack, 107  
 print\_execution\_configuration, 108  
 root\_frame\_name, 112  
 scenario\_time\_epoch, 112  
 set\_current\_execution\_mode, 108  
 set\_next\_execution\_mode, 108, 109  
 set\_next\_mode\_cte\_time, 109  
 set\_next\_mode\_scenario\_time, 109  
 set\_root\_frame\_name, 110  
 set\_scenario\_time\_epoch, 110  
 setup\_ref\_attributes, 110  
 unpack, 110  
 wait\_on\_update, 111  
 DIS::ExecutionControl, 163  
 ~ExecutionControl, 167  
 add\_multiphase\_init\_sync\_points, 167  
 add\_pause, 167  
 announce\_sync\_point, 168  
 check\_freeze, 168  
 check\_freeze\_exit, 168, 169  
 check\_mode\_transition\_request, 169  
 checktime, 180  
 clear\_mode\_values, 169  
 clear\_multiphase\_init\_sync\_points, 169  
 determine\_federation\_master, 169  
 earlyJoinerHlaInitProcess, 170  
 enter\_freeze, 170  
 ExecutionControl, 167  
 exit\_freeze, 170  
 freeze\_mode\_announce, 170  
 freeze\_mode\_transition, 170  
 get\_execution\_configuration, 171  
 get\_type, 171  
 init\_attrDIS\_\_ExecutionControl, 180  
 initialize, 171  
 InputProcessor, 180  
 is\_mtr\_valid, 172  
 is\_save\_and\_restore\_supported, 172  
 is\_save\_initiated, 172  
 join\_federation\_process, 172  
 lateJoinerHlaInitProcess, 173  
 object\_instance\_name\_reservation\_failed, 173  
 operator=, 173

pause\_sync\_pts, 181  
pending\_mtr, 181  
post\_multi\_phase\_init\_process, 173  
pre\_multi\_phase\_init\_processes, 173, 174  
process\_execution\_control\_updates, 174  
process\_mode\_interaction, 175  
process\_mode\_transition\_request, 175  
publish, 175  
role\_determination\_process, 176  
run\_mode\_transition, 176  
set\_mode\_request\_from\_mtr, 176  
set\_next\_execution\_control\_mode, 176  
set\_pending\_mtr, 177  
set\_time\_padding, 177  
setup\_interaction\_ref\_attributes, 177  
setup\_interaction\_RTI\_handles, 178  
setup\_object\_ref\_attributes, 178  
setup\_object\_RTI\_handles, 178  
shutdown, 178  
shutdown\_mode\_announce, 178  
shutdown\_mode\_transition, 178  
subscribe, 179  
sync\_point\_registration\_failed, 179  
sync\_point\_registration\_succeeded, 179  
type, 181  
unpublish, 180  
unsubscribe, 180  
wait\_on\_init\_sync\_point, 180

DIS::PausePointList, 633  
~PausePointList, 635  
check\_state, 636  
clear\_state, 636  
clear\_sync\_pnt, 636  
get\_state, 636  
init\_attrTrickHLA\_\_PausePointList, 638  
InputProcessor, 638  
operator=, 637  
PausePointList, 635  
print\_sync\_pnts, 637  
should\_exit, 637  
should\_freeze, 637  
should\_reconfig, 637  
should\_restart, 637  
should\_run, 637  
state, 638  
to\_string, 637

discover\_object\_instance  
TrickHLA::Manager, 495

discoverObjectInstance  
TrickHLA::FedAmb, 309

divest\_requested  
TrickHLA::Attribute, 82  
TrickHLA::Object, 570

does\_cte\_timeline\_exist

TrickHLA::ExecutionControlBase, 268  
does\_init\_complete\_sync\_point\_exist  
TrickHLA::ExecutionControlBase, 268  
does\_scenario\_timeline\_exist  
TrickHLA::ExecutionControlBase, 269  
does\_sim\_timeline\_exist  
TrickHLA::ExecutionControlBase, 269

DSES, 24

EXECUTION\_MODE\_LAST\_VALUE, 25  
execution\_mode\_enum\_to\_int16, 26  
execution\_mode\_enum\_to\_string, 26  
EXECUTION\_MODE\_FIRST\_VALUE, 25  
EXECUTION\_MODE\_FREEZE, 25  
EXECUTION\_MODE\_INITIALIZING, 25  
execution\_mode\_int16\_to\_enum, 27  
EXECUTION\_MODE\_RUNNING, 25  
EXECUTION\_MODE\_SHUTDOWN, 25  
EXECUTION\_MODE\_UNINITIALIZED, 25  
ExecutionModeEnum, 25  
from\_execution\_control\_enum, 27  
INITIALIZE\_SYNC\_POINT, 29  
mtr\_enum\_to\_int16, 27  
mtr\_enum\_to\_string, 28  
MTR\_FIRST\_VALUE, 26  
MTR\_GOTO\_FREEZE, 26  
MTR\_GOTO\_RUN, 26  
MTR\_GOTO\_SHUTDOWN, 26  
MTR\_INITIALIZING, 26  
mtr\_int16\_to\_enum, 28  
MTR\_LAST\_VALUE, 26  
MTR\_UNINITIALIZED, 26  
MTERenum, 25  
SIM\_CONFIG\_SYNC\_POINT, 29  
STARTUP\_SYNC\_POINT, 29  
to\_execution\_control\_enum, 28

DSES/ExecutionConfiguration.cpp  
atrdSES\_\_ExecutionConfiguration, 803

DSES::ExecutionConfiguration, 113  
~ExecutionConfiguration, 116  
configure\_attributes, 116  
current\_execution\_mode, 122  
ExecutionConfiguration, 116  
get\_current\_execution\_mode, 116  
get\_next\_execution\_mode, 117  
get\_next\_mode\_cte\_time, 117  
get\_next\_mode\_scenario\_time, 117  
get\_root\_frame\_name, 117  
get\_scenario\_time\_epoch, 117  
init\_attrDSES\_\_ExecutionConfiguration, 122  
InputProcessor, 122  
next\_execution\_mode, 122  
next\_mode\_cte\_time, 122  
next\_mode\_scenario\_time, 122  
operator=, 118

pack, 118  
 print\_execution\_configuration, 118  
 root\_frame\_name, 123  
 scenario\_time\_epoch, 123  
 set\_current\_execution\_mode, 118, 119  
 set\_next\_execution\_mode, 119  
 set\_next\_mode\_cte\_time, 119  
 set\_next\_mode\_scenario\_time, 120  
 set\_root\_frame\_name, 120  
 set\_scenario\_time\_epoch, 120  
 setup\_ref\_attributes, 121  
 unpack, 121  
 wait\_on\_update, 121  
**DSES::ExecutionControl**, 181  
 ~ExecutionControl, 185  
 achieve\_all\_multiphase\_init\_sync\_pnts, 185  
 add\_multiphase\_init\_sync\_points, 185  
 announce\_sync\_point, 186  
 check\_freeze\_exit, 186  
 check\_mode\_transition\_request, 186  
 clear\_mode\_values, 187  
 determine\_federation\_master, 187  
 ExecutionControl, 184, 185  
 freeze\_mode\_announce, 187  
 freeze\_mode\_transition, 187  
 get\_execution\_configuration, 188  
 get\_type, 188  
 init\_attrDSES\_\_ExecutionControl, 195  
 initialize, 188  
 InputProcessor, 196  
 is\_mtr\_valid, 188  
 join\_federation\_process, 189  
 object\_instance\_name\_reservation\_failed, 189  
 operator=, 189  
 pending\_mtr, 196  
 post\_multi\_phase\_init\_process, 190  
 pre\_multi\_phase\_init\_processes, 190  
 process\_execution\_control\_updates, 190  
 process\_mode\_interaction, 191  
 process\_mode\_transition\_request, 191  
 publish, 192  
 run\_mode\_transition, 192  
 set\_mode\_request\_from\_mtr, 192  
 set\_next\_execution\_control\_mode, 193  
 set\_pending\_mtr, 193  
 set\_time\_padding, 193  
 setup\_interaction\_ref\_attributes, 194  
 setup\_interaction\_RTI\_handles, 194  
 setup\_object\_ref\_attributes, 194  
 setup\_object\_RTI\_handles, 194  
 shutdown, 194  
 shutdown\_mode\_announce, 194  
 shutdown\_mode\_transition, 195  
 subscribe, 195  
 type, 196  
 unpublish, 195  
 unsubscribe, 195  
 wait\_for\_all\_multiphase\_init\_sync\_pnts, 195  
**dump\_head\_pointers**  
 TrickHLA::ItemQueue, 471  
**dump\_interactions**  
 TrickHLA::Manager, 496  
**dvdt**  
 TrickHLAModel::SineData, 689  
**dvdt\_attr**  
 TrickHLAModel::SineLagCompensation, 700  
 TrickHLAModel::SinePacking, 711  
**DynamicalEntity**  
 SpaceFOM::DynamicalEntity, 100  
**DynamicalEntity.cpp**, 797  
**DynamicalEntity.hh**, 798  
**early\_joiner\_hla\_init\_process**  
 DIS::ExecutionControl, 170  
 SpaceFOM::ExecutionControl, 227  
**empty**  
 TrickHLA::ItemQueue, 471  
 TrickHLA::ReflectedAttributesQueue, 661  
**enable\_async\_delivery**  
 TrickHLA::Federate, 343  
**enable\_known\_feds**  
 TrickHLA::Federate, 391  
**enable\_MOM\_auto\_provide\_setting**  
 TrickHLA::Federate, 344  
**encode**  
 SpaceFOM::QuaternionEncoder, 648  
 SpaceFOM::SpaceTimeCoordinateEncoder, 717  
**encode\_boolean\_to\_buffer**  
 TrickHLA::Attribute, 67  
 TrickHLA::Parameter, 612  
**encode\_logical\_time**  
 TrickHLA::Attribute, 68  
 TrickHLA::Parameter, 612  
**encode\_opaque\_data\_to\_buffer**  
 TrickHLA::Attribute, 68  
 TrickHLA::Parameter, 612  
**encode\_string\_to\_buffer**  
 TrickHLA::Attribute, 68  
 TrickHLA::Parameter, 612  
**encoder**  
 SpaceFOM::QuaternionEncoder, 649  
 SpaceFOM::SpaceTimeCoordinateEncoder, 719  
**ENCODING\_ASCII\_STRING**  
 TrickHLA, 49  
**ENCODING\_BIG\_ENDIAN**  
 TrickHLA, 49  
**ENCODING\_BOOLEAN**  
 TrickHLA, 50

ENCODING\_C\_STRING  
TrickHLA, 49

ENCODING\_FIRST\_VALUE  
TrickHLA, 49

ENCODING\_LAST\_VALUE  
TrickHLA, 50

ENCODING\_LITTLE\_ENDIAN  
TrickHLA, 49

ENCODING\_LOGICAL\_TIME  
TrickHLA, 49

ENCODING\_NO\_ENCODING  
TrickHLA, 50

ENCODING\_OPAQUE\_DATA  
TrickHLA, 49

ENCODING\_UNICODE\_STRING  
TrickHLA, 49

ENCODING\_UNKNOWN  
TrickHLA, 49

EncodingEnum  
TrickHLA, 49

ensure\_buffer\_capacity  
TrickHLA::Attribute, 68  
TrickHLA::OpaqueBuffer, 579  
TrickHLA::Parameter, 613

enter\_freeze  
DIS::ExecutionControl, 170  
IMSim::ExecutionControl, 206  
SpaceFOM::ExecutionControl, 228  
TrickHLA::ExecutionControlBase, 269  
TrickHLA::Federate, 344

epoch  
TrickHLA::Timeline, 773

epoch\_and\_root\_frame\_discovery\_process  
SpaceFOM::ExecutionControl, 228

EXECUITON\_MODE\_LAST\_VALUE  
DIS, 19  
DSES, 25  
IMSim, 31  
SpaceFOM, 39

execution\_configuration  
TrickHLA::ExecutionControlBase, 295

execution\_control  
TrickHLA::ExecutionConfigurationBase, 162  
TrickHLA::Federate, 391  
TrickHLA::Manager, 520

execution\_control\_enum\_to\_int16  
TrickHLA, 53

execution\_control\_enum\_to\_string  
TrickHLA, 53

EXECUTION\_CONTROL\_FIRST\_VALUE  
TrickHLA, 50

EXECUTION\_CONTROL\_FREEZE  
TrickHLA, 50

EXECUTION\_CONTROL\_INITIALIZING  
TrickHLA, 50

EXECUTION\_CONTROL\_SHUTDOWN  
TrickHLA, 50

execution\_control\_int16\_to\_enum  
TrickHLA, 53

EXECUTION\_CONTROL\_LAST\_VALUE  
TrickHLA, 50

EXECUTION\_CONTROL\_RECONFIG  
TrickHLA, 50

EXECUTION\_CONTROL\_RESTART  
TrickHLA, 50

EXECUTION\_CONTROL\_RUNNING  
TrickHLA, 50

EXECUTION\_CONTROL\_SHUTDOWN  
TrickHLA, 50

EXECUTION\_CONTROL\_UNINITIALIZED  
TrickHLA, 50

execution\_has\_begun  
TrickHLA::Federate, 391

execution\_mode\_enum\_to\_int16  
DIS, 20  
DSES, 26  
IMSim, 32  
SpaceFOM, 39

execution\_mode\_enum\_to\_string  
DIS, 20  
DSES, 26  
IMSim, 32  
SpaceFOM, 39

EXECUTION\_MODE\_FIRST\_VALUE  
DIS, 18  
DSES, 25  
IMSim, 31  
SpaceFOM, 38

EXECUTION\_MODE\_FREEZE  
DIS, 19  
DSES, 25  
IMSim, 31  
SpaceFOM, 39

EXECUTION\_MODE\_INITIALIZING  
DIS, 18  
DSES, 25  
IMSim, 31  
SpaceFOM, 38

execution\_mode\_int16\_to\_enum  
DIS, 20  
DSES, 27  
IMSim, 33  
SpaceFOM, 40

EXECUTION\_MODE\_RUNNING  
DIS, 18  
DSES, 25  
IMSim, 31  
SpaceFOM, 39

EXECUTION\_MODE\_SHUTDOWN  
DIS, 19

DSES, 25  
 IMSim, 31  
 SpaceFOM, 39  
**EXECUTION\_MODE\_UNINITIALIZED**  
 DIS, 18  
 DSES, 25  
 IMSim, 31  
 SpaceFOM, 38  
**ExecutionConfiguration**  
 DIS::ExecutionConfiguration, 105  
 DSES::ExecutionConfiguration, 116  
 IMSim::ExecutionConfiguration, 126  
 SpaceFOM::ExecutionConfiguration, 138, 139  
 TrickHLA::ExecutionConfiguration, 150  
**ExecutionConfiguration.cpp**, 800, 801, 803, 804, 806  
**ExecutionConfiguration.hh**, 807, 809, 811, 812, 814  
**ExecutionConfigurationBase**  
 TrickHLA::ExecutionConfigurationBase, 155, 156  
**ExecutionConfigurationBase.cpp**, 815  
 attrTrickHLA\_\_ExecutionConfigurationBase, 817  
**ExecutionConfigurationBase.hh**, 817  
**ExecutionControl**  
 DIS::ExecutionControl, 167  
 DSES::ExecutionControl, 184, 185  
 IMSim::ExecutionControl, 201  
 SpaceFOM::ExecutionControl, 225  
 TrickHLA::ExecutionControl, 246, 247  
**ExecutionControl.cpp**, 818, 819, 821, 822, 824  
**ExecutionControl.hh**, 825, 827, 828, 830, 831  
**ExecutionControlBase**  
 TrickHLA::ExecutionControlBase, 263, 264  
**ExecutionControlBase.cpp**, 832  
**ExecutionControlBase.hh**, 834  
**ExecutionControlEnum**  
 TrickHLA, 50  
**ExecutionModeEnum**  
 DIS, 18  
 DSES, 25  
 IMSim, 31  
 SpaceFOM, 38  
**exists**  
 TrickHLA::SyncPnt, 733  
**exit\_freeze**  
 DIS::ExecutionControl, 170  
 IMSim::ExecutionControl, 207  
 SpaceFOM::ExecutionControl, 228  
 TrickHLA::ExecutionControlBase, 269  
 TrickHLA::Federate, 344  
**extract\_data**  
 TrickHLA::Attribute, 68  
 TrickHLA::Interaction, 438  
 TrickHLA::Object, 542  
 TrickHLA::Parameter, 613  
**FedAmb**  
 TrickHLA::FedAmb, 306, 307  
**FedAmb.cpp**, 835  
**FedAmb.hh**, 836  
**Federate**  
 TrickHLA::Federate, 336, 337  
 TrickHLA::Manager, 519  
**federate**  
 TrickHLA::ExecutionControlBase, 296  
 TrickHLA::FedAmb, 321  
 TrickHLA::Manager, 520  
**Federate.cpp**, 838  
 ip\_mutex, 840  
 the\_cpr, 840  
 THLA\_TAG\_USE\_USLEEP, 840  
**Federate.hh**, 841  
**federate\_ambassador**  
 TrickHLA::Federate, 391  
**federate\_can\_rejoin\_federation**  
 TrickHLA::Federate, 344  
**federate\_has\_been\_restarted**  
 TrickHLA::Federate, 391  
**federate\_has\_been\_restored**  
 TrickHLA::Manager, 520  
**federate\_id**  
 TrickHLA::Federate, 392  
**FEDERATE\_JOIN\_EARLY**  
 TrickHLA, 50  
**FEDERATE\_JOIN\_FIRST\_VALUE**  
 TrickHLA, 50  
**FEDERATE\_JOIN\_LAST\_VALUE**  
 TrickHLA, 50  
**FEDERATE\_JOIN\_LATE**  
 TrickHLA, 50  
**FEDERATE\_JOIN\_NOMINAL**  
 TrickHLA, 50  
**FEDERATE\_JOIN\_RESTORING**  
 TrickHLA, 50  
**FEDERATE\_JOIN\_UNKNOWN**  
 TrickHLA, 50  
**FederateJoinEnum**  
 TrickHLA, 50  
**federation\_created\_by\_federate**  
 TrickHLA::Federate, 392  
**federation\_exists**  
 TrickHLA::Federate, 392  
**federation\_joined**  
 TrickHLA::Federate, 392  
**federation\_name**  
 TrickHLA::Federate, 392  
**federation\_restore\_failed\_callback\_complete**  
 TrickHLA::Federate, 392  
**federation\_restore\_status\_response\_context\_switch**  
 TrickHLA::FedAmb, 321

federation\_restored  
TrickHLA::Federate, 345

federation\_restored\_rebuild\_federate\_handle\_set  
TrickHLA::FedAmb, 321

federation\_saved  
TrickHLA::Federate, 345

federation\_synchronized  
TrickHLA::Federate, 345

federationNotRestored  
TrickHLA::FedAmb, 310

federationNotSaved  
TrickHLA::FedAmb, 310

federationRestoreBegin  
TrickHLA::FedAmb, 310

federationRestored  
TrickHLA::FedAmb, 310

federationRestoreStatusResponse  
TrickHLA::FedAmb, 310

federationSaved  
TrickHLA::FedAmb, 310

federationSaveStatusResponse  
TrickHLA::FedAmb, 310

federationSynchronized  
TrickHLA::FedAmb, 311

FEDRUN\_SYNC\_POINT  
IMSim, 36

FEDSAVE\_SYNC\_POINT  
IMSim, 36

first\_pass  
TrickHLA::BasicClock, 87

first\_step\_deriv  
integrator.d, 850

fix\_FPU\_control\_word  
TrickHLA::Federate, 345

FOM\_modules  
TrickHLA::Federate, 393

FOM\_name  
TrickHLA::Attribute, 82  
TrickHLA::Interaction, 449  
TrickHLA::Object, 570  
TrickHLA::OwnershipItem, 599  
TrickHLA::Parameter, 621

force  
SpaceFOM::DynamicalEntity, 101

freeze\_delay\_frames  
TrickHLA::Federate, 393

freeze\_init  
SpaceFOM::ExecutionControl, 228  
TrickHLA::ExecutionControlBase, 269  
TrickHLA::Federate, 346

freeze\_inter\_count  
IMSim::ExecutionControl, 219

freeze\_interaction  
IMSim::ExecutionControl, 219

freeze\_interaction\_handler  
IMSim::ExecutionControl, 220

freeze\_mode\_announce  
DIS::ExecutionControl, 170  
DSES::ExecutionControl, 187  
IMSim::ExecutionControl, 207  
SpaceFOM::ExecutionControl, 229  
TrickHLA::ExecutionControl, 248  
TrickHLA::ExecutionControlBase, 270

freeze\_mode\_transition  
DIS::ExecutionControl, 170  
DSES::ExecutionControl, 187  
IMSim::ExecutionControl, 207  
SpaceFOM::ExecutionControl, 229  
TrickHLA::ExecutionControl, 248  
TrickHLA::ExecutionControlBase, 270

freeze\_scenario\_times  
IMSim::ExecutionControl, 220

freeze\_the\_federation  
TrickHLA::Federate, 393

FreezeInteractionHandler  
IMSim::FreezeInteractionHandler, 408

FreezeInteractionHandler.cpp, 842

THLA\_FREEZE\_INTERACTION\_DEBUG, 843

FreezeInteractionHandler.hh, 844

FreezeTimeSet  
IMSim, 31

freq  
TrickHLAModel::SineData, 690

freq\_attr  
TrickHLAModel::SineLagCompensation, 700  
TrickHLAModel::SinePacking, 711

from\_execution\_control\_enum  
DIS, 21  
DSES, 27  
IMSim, 33

from\_execution\_control\_enum  
SpaceFOM, 40

from\_mode\_transition\_enum  
SpaceFOM, 41

front  
TrickHLA::ItemQueue, 471  
TrickHLA::ReflectedAttributesQueue, 662

get  
TrickHLA::Int64Interval, 414  
TrickHLA::Int64Time, 423

get\_address\_of\_interaction\_mode  
SpaceFOM::MTRInteractionHandler, 526

get\_address\_of\_interaction\_time  
IMSim::FreezeInteractionHandler, 408

get\_amplitude  
TrickHLAModel::SineData, 685

get\_attribute

TrickHLA::LagCompensation, 479  
 TrickHLA::Object, 542, 543  
 TrickHLA::OwnershipHandler, 588  
 TrickHLA::Packing, 601  
 get\_attribute\_and\_validate  
     TrickHLA::LagCompensation, 480  
     TrickHLA::Packing, 602  
 get\_attribute\_count  
     TrickHLA::Object, 543  
     TrickHLA::OwnershipHandler, 588  
 get\_attribute\_FOM\_names  
     TrickHLA::Object, 544  
     TrickHLA::OwnershipHandler, 588  
 get\_attribute\_handle  
     TrickHLA::Attribute, 69  
 get\_attribute\_size  
     TrickHLA::Attribute, 69  
 get\_attribute\_value  
     TrickHLA::Attribute, 69  
 get\_attributes  
     TrickHLA::Object, 544  
 get\_byte\_alignment  
     TrickHLA::OpaqueBuffer, 579  
 get\_capacity  
     TrickHLA::OpaqueBuffer, 580  
 get\_class\_handle  
     TrickHLA::Interaction, 438  
     TrickHLA::Object, 544  
 get\_clock\_ID  
     TrickHLA::CTETimelineBase, 92  
 get\_code\_section  
     TrickHLA::DebugHandler, 95  
 get\_conditional  
     TrickHLA::Attribute, 69  
 get\_configuration  
     TrickHLA::Attribute, 70  
 get\_cte\_time  
     TrickHLA::ExecutionControlBase, 270  
     TrickHLA::InteractionHandler, 454  
     TrickHLA::LagCompensation, 480  
     TrickHLA::OwnershipHandler, 588  
     TrickHLA::Packing, 602  
 get\_current\_execution\_control\_mode  
     TrickHLA::ExecutionControlBase, 270  
 get\_current\_execution\_mode  
     DIS::ExecutionConfiguration, 106  
     DSES::ExecutionConfiguration, 116  
     IMSim::ExecutionConfiguration, 127  
     SpaceFOM::ExecutionConfiguration, 139  
 get\_data  
     SpaceFOM::QuaternionEncoder, 648  
     SpaceFOM::SpaceTimeCoordinateEncoder, 718  
 get\_debug\_handler  
     TrickHLA::Manager, 496  
 get\_debug\_level  
     TrickHLA::DebugHandler, 95  
 get\_debug\_level\_as\_int  
     TrickHLA::DebugHandler, 95  
 get\_derivative  
     TrickHLAModel::SineData, 685  
 get\_elapsed\_time  
     TrickHLA::Timeline, 771  
 get\_encoded\_parameter\_value  
     TrickHLA::Parameter, 613  
 get\_endianness  
     TrickHLA::Utilities, 781  
 get\_epoch  
     TrickHLA::Timeline, 771  
 get\_execution\_configuration  
     DIS::ExecutionControl, 171  
     DSES::ExecutionControl, 188  
     IMSim::ExecutionControl, 207  
     SpaceFOM::ExecutionControl, 229  
     TrickHLA::ExecutionControl, 249  
     TrickHLA::ExecutionControlBase, 271  
     TrickHLA::Manager, 496  
 get\_execution\_control  
     TrickHLA::ExecutionConfigurationBase, 157  
     TrickHLA::Federate, 346  
     TrickHLA::Manager, 497  
 get\_fed\_ambassador  
     TrickHLA::Federate, 346  
 get\_fed\_lookahead  
     TrickHLA::Interaction, 438  
     TrickHLA::InteractionHandler, 454  
     TrickHLA::LagCompensation, 480  
     TrickHLA::Manager, 497  
     TrickHLA::Object, 544  
     TrickHLA::OwnershipHandler, 589  
 get\_federate  
     TrickHLA::ExecutionControlBase, 271  
     TrickHLA::Interaction, 439  
     TrickHLA::Manager, 497  
     TrickHLA::Object, 545  
 get\_federate\_name  
     TrickHLA::Federate, 346  
 get\_federate\_type  
     TrickHLA::Federate, 346  
 get\_federation\_name  
     TrickHLA::Federate, 347  
 get\_FOM\_name  
     TrickHLA::Attribute, 70  
     TrickHLA::Interaction, 439  
     TrickHLA::Object, 545  
     TrickHLA::Parameter, 614  
 get\_freeze\_announced  
     TrickHLA::Federate, 347  
 get\_freeze\_pending

TrickHLA::Federate, 347  
get\_frequency  
    TrickHLAModel::SineData, 685  
get\_granted\_fed\_time  
    TrickHLA::Federate, 347  
    TrickHLA::Interaction, 439  
    TrickHLA::InteractionHandler, 454  
    TrickHLA::LagCompensation, 480  
    TrickHLA::Manager, 497  
    TrickHLA::Object, 545  
    TrickHLA::OwnershipHandler, 589  
get\_granted\_time  
    TrickHLA::Federate, 348  
    TrickHLA::Manager, 498  
    TrickHLA::Object, 546  
get\_handler  
    TrickHLA::Interaction, 439  
get\_HLT\_offset  
    TrickHLA::ScenarioTimeline, 666  
get\_instance\_handle  
    TrickHLA::Object, 546  
get\_interaction\_count  
    TrickHLA::Manager, 498  
get\_interactions  
    TrickHLA::Manager, 498  
get\_joined\_federate\_handles  
    TrickHLA::Federate, 348  
get\_label  
    TrickHLA::SyncPnt, 733  
get\_lag\_compensation\_type  
    TrickHLA::Object, 546  
get\_last\_update\_time  
    TrickHLA::Object, 546  
get\_least\_common\_time\_step  
    IMSim::ExecutionConfiguration, 127  
    SpaceFOM::ExecutionConfiguration, 140  
    TrickHLA::ExecutionControlBase, 271  
get\_looking\_ahead  
    TrickHLA::Federate, 348  
get\_looking\_ahead\_time  
    TrickHLA::Federate, 349  
get\_manager  
    TrickHLA::ExecutionControlBase, 271  
    TrickHLA::FedAmb, 311  
    TrickHLA::Federate, 349  
    TrickHLA::Interaction, 440  
get\_MOM\_HLAfederate\_class\_handle  
    TrickHLA::Federate, 349  
get\_name  
    SpaceFOM::PhysicalEntityBase, 641  
    SpaceFOM::RefFrameBase, 654  
    TrickHLA::Object, 547  
    TrickHLAModel::SineData, 685  
get\_name\_string

TrickHLA::Object, 547  
get\_next\_execution\_mode  
    DIS::ExecutionConfiguration, 106  
    DSES::ExecutionConfiguration, 117  
    IMSim::ExecutionConfiguration, 127  
    SpaceFOM::ExecutionConfiguration, 140  
get\_next\_mode\_cte\_time  
    DIS::ExecutionConfiguration, 106  
    DSES::ExecutionConfiguration, 117  
    IMSim::ExecutionConfiguration, 128  
    SpaceFOM::ExecutionConfiguration, 140  
get\_next\_mode\_scenario\_time  
    DIS::ExecutionConfiguration, 107  
    DSES::ExecutionConfiguration, 117  
    IMSim::ExecutionConfiguration, 128  
    SpaceFOM::ExecutionConfiguration, 140  
get\_object  
    SpaceFOM::RefFrameBase, 654  
get\_object\_count  
    TrickHLA::Manager, 498  
get\_object\_FOM\_name  
    TrickHLA::OwnershipHandler, 589  
get\_object\_name  
    TrickHLA::OwnershipHandler, 589  
get\_objects  
    TrickHLA::Manager, 498  
get\_parameter\_count  
    TrickHLA::Interaction, 440  
get\_parameter\_handle  
    TrickHLA::Parameter, 614  
get\_parameter\_size  
    TrickHLA::Parameter, 614  
get\_parameters  
    TrickHLA::Interaction, 440  
get\_parent\_name  
    SpaceFOM::RefFrameBase, 654  
get\_parent\_ref\_frame  
    SpaceFOM::PhysicalEntityBase, 641  
get\_phase  
    TrickHLAModel::SineData, 686  
get\_preferred\_order  
    TrickHLA::Attribute, 70  
    TrickHLA::Interaction, 440  
get\_reconfig\_name  
    TrickHLA::SyncPntListBase, 750  
get\_ref2\_attributes  
    TrickHLA::Attribute, 70  
    TrickHLA::Parameter, 614  
get\_release\_date  
    TrickHLA::Utilities, 781  
get\_requested\_execution\_control\_mode  
    TrickHLA::ExecutionControlBase, 272  
get\_requested\_fed\_time  
    TrickHLA::Federate, 349

get\_requested\_time  
     TrickHLA::Federate, 350

get\_restart  
     TrickHLA::Federate, 350

get\_restart\_cfg  
     TrickHLA::Federate, 350

get\_root\_frame\_name  
     DIS::ExecutionConfiguration, 107  
     DSES::ExecutionConfiguration, 117  
     IMSim::ExecutionConfiguration, 128  
     SpaceFOM::ExecutionConfiguration, 141

get\_RTI\_ambassador  
     TrickHLA::Federate, 350  
     TrickHLA::Interaction, 441  
     TrickHLA::Manager, 499  
     TrickHLA::Object, 548

get\_rti\_encoding  
     TrickHLA::Attribute, 71  
     TrickHLA::Parameter, 615

get\_running\_feds\_count  
     TrickHLA::Federate, 351

get\_S\_define\_name  
     TrickHLA::ExecutionConfigurationBase, 157

get\_scenario\_freeze\_time  
     TrickHLA::ExecutionControlBase, 272

get\_scenario\_time  
     TrickHLA::ExecutionControlBase, 272  
     TrickHLA::InteractionHandler, 455  
     TrickHLA::LagCompensation, 481  
     TrickHLA::OwnershipHandler, 589  
     TrickHLA::Packing, 602

get\_scenario\_time\_epoch  
     DIS::ExecutionConfiguration, 107  
     DSES::ExecutionConfiguration, 117  
     IMSim::ExecutionConfiguration, 128  
     SpaceFOM::ExecutionConfiguration, 141

get\_sim\_offset  
     TrickHLA::ScenarioTimeline, 666

get\_sim\_time  
     TrickHLA::ExecutionControlBase, 273  
     TrickHLA::InteractionHandler, 455

get\_sim\_variable\_address  
     TrickHLA::Attribute, 71

get\_simulation\_freeze\_time  
     TrickHLA::ExecutionControlBase, 273

get\_size  
     TrickHLA::SyncPntListBase, 750

get\_stale\_data\_counter  
     TrickHLA::Federate, 351

get\_state  
     DIS::PausePointList, 636  
     IMSim::PausePointList, 630  
     TrickHLA::SyncPnt, 733

get\_status

SpaceFOM::PhysicalEntityBase, 641

get\_sync\_pnt  
     TrickHLA::SyncPntListBase, 750

get\_sync\_pnt\_state  
     TrickHLA::SyncPntListBase, 750

get\_time  
     SpaceFOM::PhysicalEntityBase, 641  
     TrickHLA::BasicClock, 87  
     TrickHLA::CTETimelineBase, 92  
     TrickHLA::ScenarioTimeline, 667  
     TrickHLA::SimTimeline, 675  
     TrickHLA::TimedSyncPnt, 762  
     TrickHLA::Timeline, 772  
     TrickHLA::TimeOfDayTimeline, 775  
     TrickHLAModel::SineData, 686

get\_time\_padding  
     TrickHLA::ExecutionControlBase, 273

get\_tolerance  
     TrickHLAModel::SineData, 686

get\_trick\_name  
     TrickHLA::Attribute, 71  
     TrickHLA::Parameter, 615

get\_trickhla\_object  
     TrickHLA::ExecutionControlBase, 273  
     TrickHLA::Manager, 499

get\_type  
     DIS::ExecutionControl, 171  
     DSES::ExecutionControl, 188  
     IMSim::ExecutionControl, 208  
     SpaceFOM::ExecutionControl, 230  
     SpaceFOM::PhysicalEntityBase, 642  
     TrickHLA::ExecutionControl, 249  
     TrickHLA::ExecutionControlBase, 274

get\_unregistered\_object  
     TrickHLA::ExecutionControlBase, 274  
     TrickHLA::Manager, 500

get\_unregistered\_remote\_object  
     TrickHLA::ExecutionControlBase, 274  
     TrickHLA::Manager, 500

get\_update\_time\_plus\_lookahead  
     TrickHLA::Object, 548

get\_value  
     TrickHLAModel::SineData, 686

get\_version  
     TrickHLA::Utilities, 781

getDoubleTime  
     TrickHLA::Int64Interval, 414  
     TrickHLA::Int64Time, 423

getMicros  
     TrickHLA::Int64Interval, 414  
     TrickHLA::Int64Time, 423

getSeconds  
     TrickHLA::Int64Interval, 414  
     TrickHLA::Int64Time, 424

getTimelnMicros  
TrickHLA::Int64Interval, 415  
TrickHLA::Int64Time, 424

got\_startup\_sp  
TrickHLA::Federate, 393

grant\_pull\_request  
TrickHLA::Manager, 501  
TrickHLA::Object, 548

grant\_push\_pthread\_function  
Object.cpp, 877

grant\_push\_request  
TrickHLA::Object, 548

grant\_push\_request\_pthread  
TrickHLA::Object, 549

granted\_time  
SpaceFOM::MTRInteractionHandler, 528  
TrickHLA::Federate, 393

handle\_set  
TrickHLA::DivestThreadArgs, 98

handler  
TrickHLA::Interaction, 449

has\_conditional  
TrickHLA::Attribute, 71

has\_federate\_been\_restored  
TrickHLA::Manager, 501

has\_restore\_been\_announced  
TrickHLA::Federate, 351

has\_restore\_process\_restore\_request\_failed  
TrickHLA::Federate, 352

has\_restore\_process\_restore\_request\_succeeded  
TrickHLA::Federate, 352

has\_restore\_request\_failed  
TrickHLA::Federate, 352

has\_restore\_request\_succeeded  
TrickHLA::Federate, 352

head  
TrickHLA::ItemQueue, 474

HLA\_save\_directory  
TrickHLA::Federate, 393

HLA\_time  
TrickHLA::Federate, 394

HLAAttributeMapQueue  
TrickHLA, 47

HLAtrue  
TrickHLA::Attribute, 83  
TrickHLA::Parameter, 621

hlt\_offset  
TrickHLA::ScenarioTimeline, 669

IEEE\_1516\_2010  
StandardsSupport.hh, 944

IMSim, 29

EXECUTION\_MODE\_LAST\_VALUE, 31

execution\_mode\_enum\_to\_int16, 32

execution\_mode\_enum\_to\_string, 32

EXECUTION\_MODE\_FIRST\_VALUE, 31

EXECUTION\_MODE\_FREEZE, 31

EXECUTION\_MODE\_INITIALIZING, 31

execution\_mode\_int16\_to\_enum, 33

EXECUTION\_MODE\_RUNNING, 31

EXECUTION\_MODE\_SHUTDOWN, 31

EXECUTION\_MODE\_UNINITIALIZED, 31

ExecutionModeEnum, 31

FEDRUN\_SYNC\_POINT, 36

FEDSAVE\_SYNC\_POINT, 36

FreezeTimeSet, 31

from\_execution\_control\_enum, 33

INIT\_COMPLETE\_SYNC\_POINT, 36

INITIALIZE\_SYNC\_POINT, 37

mtr\_enum\_to\_int16, 34

mtr\_enum\_to\_string, 34

MTR\_FIRST\_VALUE, 31

MTR\_GOTO\_FREEZE, 31

MTR\_GOTO\_RUN, 31

MTR\_GOTO\_SHUTDOWN, 31

MTR\_INITIALIZING, 31

mtr\_int16\_to\_enum, 34

MTR\_LAST\_VALUE, 31

MTR\_UNINITIALIZED, 31

MTREnum, 31

pause\_pnt\_state\_enum\_to\_int16, 35

pause\_pnt\_state\_enum\_to\_string, 35

pause\_pnt\_state\_int16\_to\_enum, 35

PAUSE\_POINT\_STATE\_ACKNOWLEDGED, 32

PAUSE\_POINT\_STATE\_ERROR, 32

PAUSE\_POINT\_STATE\_EXIT, 32

PAUSE\_POINT\_STATE\_FIRST\_VALUE, 32

PAUSE\_POINT\_STATE\_FREEZE, 32

PAUSE\_POINT\_STATE\_PENDING, 32

PAUSE\_POINT\_STATE\_RECONFIG, 32

PAUSE\_POINT\_STATE\_RESTART, 32

PAUSE\_POINT\_STATE\_RUN, 32

PAUSE\_POINT\_STATE\_UNKNOWN, 32

PausePointStateEnum, 32

SIM\_CONFIG\_SYNC\_POINT, 37

STARTUP\_FREEZE\_SYNC\_POINT, 37

STARTUP\_SYNC\_POINT, 37

to\_execution\_control\_enum, 36

IMSim/ExecutionConfiguration.cpp  
attrIMSim\_\_ExecutionConfiguration, 804

IMSim/ExecutionControl.cpp  
attrTrickHLA\_\_FreezeInteractionHandler, 822

IMSim::ExecutionConfiguration, 123

~ExecutionConfiguration, 126

configure\_attributes, 127

current\_execution\_mode, 134

ExecutionConfiguration, 126

get\_current\_execution\_mode, 127

get\_least\_common\_time\_step, 127  
 get\_next\_execution\_mode, 127  
 get\_next\_mode\_cte\_time, 128  
 get\_next\_mode\_scenario\_time, 128  
 get\_root\_frame\_name, 128  
 get\_scenario\_time\_epoch, 128  
 init\_attrIMSim\_\_ExecutionConfiguration, 134  
 InputProcessor, 134  
 least\_common\_time\_step, 134  
 next\_execution\_mode, 134  
 next\_mode\_cte\_time, 134  
 next\_mode\_scenario\_time, 135  
 operator=, 128  
 pack, 129  
 print\_execution\_configuration, 129  
 root\_frame\_name, 135  
 scenario\_time\_epoch, 135  
 set\_current\_execution\_mode, 129  
 set\_least\_common\_time\_step, 130  
 set\_next\_execution\_mode, 130  
 set\_next\_mode\_cte\_time, 131  
 set\_next\_mode\_scenario\_time, 131  
 set\_root\_frame\_name, 131  
 set\_scenario\_time\_epoch, 131  
 setup\_ref\_attributes, 133  
 unpack, 133  
 wait\_on\_update, 133  
**IMSim::ExecutionControl**, 196  
 ~ExecutionControl, 201  
 achieve\_all\_multiphase\_init\_sync\_pnts, 201  
 add\_freeze\_scenario\_time, 202  
 add\_multiphase\_init\_sync\_points, 202  
 add\_pause, 202  
 announce\_sync\_point, 202  
 check\_freeze\_exit, 203  
 check\_freeze\_time, 203  
 check\_mode\_transition\_request, 204  
 check\_pause, 204  
 check\_pause\_at\_init, 204, 205  
 check\_scenario\_freeze\_time, 205  
 checktime, 219  
 clear\_mode\_values, 205  
 clear\_pause, 205  
 convert\_loggable\_sync\_pts, 206  
 determine\_if\_late\_joining\_or\_restoring\_federate, 206  
 enter\_freeze, 206  
 ExecutionControl, 201  
 exit\_freeze, 207  
 freeze\_inter\_count, 219  
 freeze\_interaction, 219  
 freeze\_interaction\_handler, 220  
 freeze\_mode\_announce, 207  
 freeze\_mode\_transition, 207  
 freeze\_scenario\_times, 220  
 get\_execution\_configuration, 207  
 get\_type, 208  
 init\_attrIMSim\_\_ExecutionControl, 219  
 initialize, 208  
 InputProcessor, 219  
 is\_mtr\_valid, 208  
 is\_save\_and\_restore\_supported, 209  
 is\_save\_initiated, 209  
 join\_federation\_process, 209  
 mark\_synchronized, 209  
 operator=, 210  
 pause\_sync\_pts, 220  
 pending\_mtr, 220  
 perform\_save, 210  
 post\_multi\_phase\_init\_process, 210  
 pre\_multi\_phase\_init\_processes, 210  
 process\_execution\_control\_updates, 211  
 process\_mode\_interaction, 212  
 process\_mode\_transition\_request, 212  
 publish, 213  
 receive\_interaction, 213  
 reinstate\_logged\_sync\_pts, 214  
 run\_mode\_transition, 214  
 send\_mode\_transition\_interaction, 214  
 set\_mode\_request\_from\_mtr, 214  
 set\_next\_execution\_control\_mode, 215  
 set\_pending\_mtr, 215  
 set\_time\_padding, 215  
 setup\_interaction\_ref\_attributes, 216  
 setup\_interaction\_RTI\_handles, 216  
 setup\_object\_ref\_attributes, 216  
 setup\_object\_RTI\_handles, 216  
 shutdown, 217  
 shutdown\_mode\_announce, 217  
 shutdown\_mode\_transition, 217  
 start\_federation\_save\_at\_scenario\_time, 217  
 subscribe, 218  
 trigger\_freeze\_interaction, 218  
 type, 220  
 un\_freeze, 218  
 unpublish, 218  
 unsubscribe, 218  
 wait\_for\_all\_multiphase\_init\_sync\_pnts, 219  
**IMSim::FreezeInteractionHandler**, 406  
 ~FreezeInteractionHandler, 408  
 check\_values, 408  
 FreezeInteractionHandler, 408  
 get\_address\_of\_interaction\_time, 408  
 init\_attrTrickHLA\_\_FreezeInteractionHandler, 410  
 InputProcessor, 410  
 operator=, 409  
 receive\_interaction, 409  
 send\_scenario\_freeze\_interaction, 409  
 time, 410

IMSim::PausePointList, 626  
~PausePointList, 629  
check\_state, 630  
clear\_state, 630  
clear\_sync\_pnt, 630  
get\_state, 630  
init\_attrTrickHLA\_\_PausePointList, 632  
InputProcessor, 632  
operator=, 631  
PausePointList, 629  
print\_sync\_pnts, 631  
should\_exit, 631  
should\_freeze, 631  
should\_reconfig, 631  
should\_restart, 631  
should\_run, 631  
state, 632  
to\_string, 632  
in\_time\_regulating\_state  
    TrickHLA::Federate, 352  
index  
    TrickHLA::InteractionItem, 464  
    TrickHLA::ParameterItem, 626  
inertia  
    SpaceFOM::DynamicalEntity, 102  
inertia\_rate  
    SpaceFOM::DynamicalEntity, 102  
inform\_RTI\_of\_restore\_completion  
    TrickHLA::Federate, 353  
informAttributeOwnership  
    TrickHLA::FedAmb, 311  
init  
    integrator.d, 850  
init\_attrDIS\_\_ExecutionConfiguration  
    DIS::ExecutionConfiguration, 111  
init\_attrDIS\_\_ExecutionControl  
    DIS::ExecutionControl, 180  
init\_attrDSES\_\_ExecutionConfiguration  
    DSES::ExecutionConfiguration, 122  
init\_attrDSES\_\_ExecutionControl  
    DSES::ExecutionControl, 195  
init\_attrIMSim\_\_ExecutionConfiguration  
    IMSim::ExecutionConfiguration, 134  
init\_attrIMSim\_\_ExecutionControl  
    IMSim::ExecutionControl, 219  
init\_attrSpaceFOM\_\_DynamicalEntity  
    SpaceFOM::DynamicalEntity, 101  
init\_attrSpaceFOM\_\_ExecutionConfiguration  
    SpaceFOM::ExecutionConfiguration, 146  
init\_attrSpaceFOM\_\_ExecutionControl  
    SpaceFOM::ExecutionControl, 242  
init\_attrSpaceFOM\_\_MTRInteractionHandler  
    SpaceFOM::MTRInteractionHandler, 528  
init\_attrSpaceFOM\_\_PhysicalEntityBase  
    SpaceFOM::PhysicalEntityBase, 643  
init\_attrSpaceFOM\_\_QuaternionEncoder  
    SpaceFOM::QuaternionEncoder, 649  
init\_attrSpaceFOM\_\_RefFrameBase  
    SpaceFOM::RefFrameBase, 656  
init\_attrSpaceFOM\_\_SpaceTimeCoordinateEncoder  
    SpaceFOM::SpaceTimeCoordinateEncoder, 718  
init\_attrTrickHLA\_\_Attribute  
    TrickHLA::Attribute, 80  
init\_attrTrickHLA\_\_BasicClock  
    TrickHLA::BasicClock, 87  
init\_attrTrickHLA\_\_Conditional  
    TrickHLA::Conditional, 89  
init\_attrTrickHLA\_\_CTETimelineBase  
    TrickHLA::CTETimelineBase, 93  
init\_attrTrickHLA\_\_DebugHandler  
    TrickHLA::DebugHandler, 97  
init\_attrTrickHLA\_\_ExecutionConfiguration  
    TrickHLA::ExecutionConfiguration, 152  
init\_attrTrickHLA\_\_ExecutionConfigurationBase  
    TrickHLA::ExecutionConfigurationBase, 162  
init\_attrTrickHLA\_\_ExecutionControl  
    TrickHLA::ExecutionControl, 255  
init\_attrTrickHLA\_\_ExecutionControlBase  
    TrickHLA::ExecutionControlBase, 294  
init\_attrTrickHLA\_\_FedAmb  
    TrickHLA::FedAmb, 321  
init\_attrTrickHLA\_\_Federate  
    TrickHLA::Federate, 389  
init\_attrTrickHLA\_\_FreezeInteractionHandler  
    IMSim::FreezeInteractionHandler, 410  
init\_attrTrickHLA\_\_Int64Interval  
    TrickHLA::Int64Interval, 418  
init\_attrTrickHLA\_\_Int64Time  
    TrickHLA::Int64Time, 433  
init\_attrTrickHLA\_\_Interaction  
    TrickHLA::Interaction, 448  
init\_attrTrickHLA\_\_InteractionHandler  
    TrickHLA::InteractionHandler, 458  
init\_attrTrickHLA\_\_InteractionItem  
    TrickHLA::InteractionItem, 464  
init\_attrTrickHLA\_\_Item  
    TrickHLA::Item, 468  
init\_attrTrickHLA\_\_ItemQueue  
    TrickHLA::ItemQueue, 473  
init\_attrTrickHLA\_\_KnownFederate  
    TrickHLA::KnownFederate, 476  
init\_attrTrickHLA\_\_LagCompensation  
    TrickHLA::LagCompensation, 482  
init\_attrTrickHLA\_\_LoggableSyncPnt  
    TrickHLA::LoggableSyncPnt, 485  
init\_attrTrickHLA\_\_LoggableTimedSyncPnt  
    TrickHLA::LoggableTimedSyncPnt, 487  
init\_attrTrickHLA\_\_Manager

TrickHLA::Manager, 519  
 init\_attrTrickHLA\_\_Object  
   TrickHLA::Object, 567  
 init\_attrTrickHLA\_\_ObjectDeleted  
   TrickHLA::ObjectDeleted, 576  
 init\_attrTrickHLA\_\_OpaqueBuffer  
   TrickHLA::OpaqueBuffer, 583  
 init\_attrTrickHLA\_\_OwnershipHandler  
   TrickHLA::OwnershipHandler, 594  
 init\_attrTrickHLA\_\_OwnershipItem  
   TrickHLA::OwnershipItem, 598  
 init\_attrTrickHLA\_\_Packing  
   TrickHLA::Packing, 605  
 init\_attrTrickHLA\_\_Parameter  
   TrickHLA::Parameter, 619  
 init\_attrTrickHLA\_\_ParameterItem  
   TrickHLA::ParameterItem, 625  
 init\_attrTrickHLA\_\_PausePointList  
   DIS::PausePointList, 638  
   IMSim::PausePointList, 632  
 init\_attrTrickHLA\_\_ReflectedAttributesQueue  
   TrickHLA::ReflectedAttributesQueue, 662  
 init\_attrTrickHLA\_\_ScenarioTimeline  
   TrickHLA::ScenarioTimeline, 669  
 init\_attrTrickHLA\_\_SimTimeline  
   TrickHLA::SimTimeline, 676  
 init\_attrTrickHLA\_\_StringUtilities  
   TrickHLA::StringUtilities, 729  
 init\_attrTrickHLA\_\_SyncPnt  
   TrickHLA::SyncPnt, 738  
 init\_attrTrickHLA\_\_SyncPntList  
   TrickHLA::SyncPntList, 742  
 init\_attrTrickHLA\_\_SyncPntListBase  
   TrickHLA::SyncPntListBase, 758  
 init\_attrTrickHLA\_\_TimedSyncPnt  
   TrickHLA::TimedSyncPnt, 763  
 init\_attrTrickHLA\_\_TimedSyncPntList  
   TrickHLA::TimedSyncPntList, 769  
 init\_attrTrickHLA\_\_Timeline  
   TrickHLA::Timeline, 773  
 init\_attrTrickHLA\_\_TimeOfDayTimeline  
   TrickHLA::TimeOfDayTimeline, 775  
 init\_attrTrickHLA\_\_Utilities  
   TrickHLA::Utilities, 782  
 init\_attrTrickHLAModel\_\_SimpleSimConfig  
   TrickHLAModel::SimpleSimConfig, 673  
 init\_attrTrickHLAModel\_\_SineConditional  
   TrickHLAModel::SineConditional, 679  
 init\_attrTrickHLAModel\_\_SineData  
   TrickHLAModel::SineData, 689  
 init\_attrTrickHLAModel\_\_SineInteractionHandler  
   TrickHLAModel::SineInteractionHandler, 694  
 init\_attrTrickHLAModel\_\_SineLagCompensation  
   TrickHLAModel::SineLagCompensation, 699  
 init\_attrTrickHLAModel\_\_SineObjectDeleted  
   TrickHLAModel::SineObjectDeleted, 703  
 init\_attrTrickHLAModel\_\_SineOwnershipHandler  
   TrickHLAModel::SineOwnershipHandler, 706  
 init\_attrTrickHLAModel\_\_SinePacking  
   TrickHLAModel::SinePacking, 710  
 init\_complete\_sp\_exists  
   TrickHLA::ExecutionControlBase, 296  
 INIT\_COMPLETE\_SYNC\_POINT  
   IMSim, 36  
 INIT\_COMPLETED\_SYNC\_POINT  
   SpaceFOM, 43  
 INIT\_STARTED\_SYNC\_POINT  
   SpaceFOM, 43  
 initialization\_execution\_control  
   TrickHLA::Manager, 501  
 initialize  
   DIS::ExecutionControl, 171  
   DSES::ExecutionControl, 188  
   IMSim::ExecutionControl, 208  
   SpaceFOM::ExecutionControl, 230  
   SpaceFOM::PhysicalEntityBase, 642  
   SpaceFOM::RefFrameBase, 654  
   TrickHLA::Attribute, 71  
   TrickHLA::ExecutionControl, 249  
   TrickHLA::ExecutionControlBase, 275  
   TrickHLA::FedAmb, 311  
   TrickHLA::Federate, 353  
   TrickHLA::Interaction, 441  
   TrickHLA::InteractionItem, 463  
   TrickHLA::Manager, 501  
   TrickHLA::Object, 549  
   TrickHLA::Parameter, 615  
   TrickHLAModel::SimpleSimConfig, 672  
   TrickHLAModel::SineConditional, 679  
   TrickHLAModel::SineLagCompensation, 698  
   TrickHLAModel::SinePacking, 709  
 initialize\_callback  
   SpaceFOM::RefFrameBase, 655  
   TrickHLA::InteractionHandler, 455  
   TrickHLA::LagCompensation, 481  
   TrickHLA::OwnershipHandler, 590  
   TrickHLA::Packing, 602  
   TrickHLAModel::SineLagCompensation, 698  
   TrickHLAModel::SineOwnershipHandler, 705  
   TrickHLAModel::SinePacking, 709  
 initialize\_MOM\_handles  
   TrickHLA::Federate, 353  
 INITIALIZE\_SYNC\_POINT  
   DIS, 24  
   DSES, 29  
   IMSim, 37  
 initialized  
   SpaceFOM::RefFrameBase, 657

TrickHLA::Attribute, 83  
TrickHLAModel::SinePacking, 711  
initiate\_federation\_save  
    TrickHLA::Manager, 501  
Initiate\_Restore  
    TrickHLA, 52  
initiate\_restore\_announce  
    TrickHLA::Federate, 354  
initiate\_restore\_flag  
    TrickHLA::Federate, 394  
initiate\_save\_announce  
    TrickHLA::Federate, 354  
initiate\_save\_flag  
    TrickHLA::Federate, 394  
initiated\_a\_federation\_save  
    TrickHLA::Manager, 520  
initiateFederateRestore  
    TrickHLA::FedAmb, 311  
initiateFederateSave  
    TrickHLA::FedAmb, 312  
InputProcessor  
    DIS::ExecutionConfiguration, 111  
    DIS::ExecutionControl, 180  
    DIS::PausePointList, 638  
    DSES::ExecutionConfiguration, 122  
    DSES::ExecutionControl, 196  
    IMSim::ExecutionConfiguration, 134  
    IMSim::ExecutionControl, 219  
    IMSim::FreezeInteractionHandler, 410  
    IMSim::PausePointList, 632  
    SpaceFOM::DynamicalEntity, 101  
    SpaceFOM::ExecutionConfiguration, 146  
    SpaceFOM::ExecutionControl, 242  
    SpaceFOM::MTRInteractionHandler, 528  
    SpaceFOM::PhysicalEntityBase, 643  
    SpaceFOM::QuaternionEncoder, 649  
    SpaceFOM::RefFrameBase, 656  
    SpaceFOM::SpaceTimeCoordinateEncoder, 718  
    TrickHLA::Attribute, 80  
    TrickHLA::BasicClock, 87  
    TrickHLA::Conditional, 89  
    TrickHLA::CTETimelineBase, 93  
    TrickHLA::DebugHandler, 97  
    TrickHLA::ExecutionConfiguration, 152  
    TrickHLA::ExecutionConfigurationBase, 162  
    TrickHLA::ExecutionControl, 255  
    TrickHLA::ExecutionControlBase, 294  
    TrickHLA::FedAmb, 321  
    TrickHLA::Federate, 389  
    TrickHLA::Int64Interval, 418  
    TrickHLA::Int64Time, 433  
    TrickHLA::Interaction, 448  
    TrickHLA::InteractionHandler, 458  
    TrickHLA::InteractionItem, 464  
TrickHLA::Item, 468  
TrickHLA::ItemQueue, 473  
TrickHLA::KnownFederate, 476  
TrickHLA::LagCompensation, 482  
TrickHLA::LoggableSyncPnt, 485  
TrickHLA::LoggableTimedSyncPnt, 487  
TrickHLA::Manager, 519  
TrickHLA::Object, 567  
TrickHLA::ObjectDeleted, 576  
TrickHLA::OpaqueBuffer, 583  
TrickHLA::OwnershipHandler, 594  
TrickHLA::OwnershipItem, 598  
TrickHLA::Packing, 605  
TrickHLA::Parameter, 619  
TrickHLA::ParameterItem, 625  
TrickHLA::ReflectedAttributesQueue, 663  
TrickHLA::ScenarioTimeline, 669  
TrickHLA::SimTimeline, 676  
TrickHLA::StringUtilities, 729  
TrickHLA::SyncPnt, 738  
TrickHLA::SyncPntList, 742  
TrickHLA::SyncPntListBase, 758  
TrickHLA::TimedSyncPnt, 763  
TrickHLA::TimedSyncPntList, 769  
TrickHLA::Timeline, 773  
TrickHLA::TimeOfDayTimeline, 775  
TrickHLA::Utilities, 782  
TrickHLAModel::SimpleSimConfig, 673  
TrickHLAModel::SineConditional, 680  
TrickHLAModel::SineData, 689  
TrickHLAModel::SineInteractionHandler, 694  
TrickHLAModel::SineLagCompensation, 699  
TrickHLAModel::SineObjectDeleted, 703  
TrickHLAModel::SineOwnershipHandler, 706  
TrickHLAModel::SinePacking, 711  
instance\_handle  
    TrickHLA::Object, 571  
Int64Interval  
    TrickHLA::Int64Interval, 412, 413  
Int64Interval.cpp, 845  
Int64Interval.hh, 846  
Int64Time  
    TrickHLA::Int64Time, 421, 422  
Int64Time.cpp, 848  
Int64Time.hh, 848  
integration  
    TrickHLAModel::SineData, 687  
integrator.d, 850  
    deriv, 850  
    first\_step\_deriv, 850  
    init, 850  
    num\_state, 850  
    option, 850  
    state, 850

state\_ws, 850  
 inter\_count  
     TrickHLA::Manager, 521  
 Interaction  
     TrickHLA::Interaction, 437  
 interaction  
     TrickHLA::InteractionHandler, 458  
 Interaction.cpp, 851  
 Interaction.hh, 852  
 interaction\_FOM\_name  
     TrickHLA::Parameter, 621  
 interaction\_type  
     TrickHLA::InteractionItem, 464  
 InteractionHandler  
     TrickHLA::InteractionHandler, 453, 454  
 InteractionHandler.cpp, 853  
 InteractionHandler.hh, 855  
 InteractionItem  
     TrickHLA::InteractionItem, 461, 462  
 InteractionItem.cpp, 856  
     attrParameterItem, 857  
 InteractionItem.hh, 857  
 interactions  
     TrickHLA::Manager, 521  
 interactions\_queue  
     TrickHLA::Manager, 521  
 ip\_mutex  
     Federate.cpp, 840  
 ip\_strdup\_wstring  
     TrickHLA::StringUtilities, 722  
 is\_a\_required\_startup\_federate  
     TrickHLA::Federate, 354  
 is\_achieved  
     TrickHLA::SyncPnt, 734  
 is\_announced  
     TrickHLA::SyncPnt, 734  
 is\_attribute\_update\_requested  
     TrickHLA::Object, 549  
 is\_byteswap  
     TrickHLA::Attribute, 72  
     TrickHLA::Parameter, 616  
 is\_changed  
     TrickHLA::Attribute, 72  
     TrickHLA::Interaction, 441  
     TrickHLA::Object, 550  
     TrickHLA::Parameter, 616  
 is\_create\_HLA\_instance  
     TrickHLA::Object, 550  
 is\_data\_cycle\_ready  
     TrickHLA::Attribute, 72  
 is\_divest\_requested  
     TrickHLA::Attribute, 73  
 is\_error  
     TrickHLA::SyncPnt, 734  
 is\_execution\_configuration\_used  
     TrickHLA::ExecutionControlBase, 275  
     TrickHLA::Manager, 502  
 is\_execution\_member  
     TrickHLA::Federate, 355  
 is\_federate\_executing  
     TrickHLA::Federate, 355  
 is\_federate\_instance\_id  
     TrickHLA::Federate, 355  
 is\_federation\_created\_by\_federate  
     TrickHLA::Federate, 356  
 is\_HLA\_save\_and\_restore\_supported  
     TrickHLA::Federate, 356  
 is\_in\_freeze  
     TrickHLA::ExecutionControlBase, 275  
 is\_in\_reconfig  
     TrickHLA::ExecutionControlBase, 275  
 is\_in\_restart  
     TrickHLA::ExecutionControlBase, 276  
 is\_initializing  
     TrickHLA::ExecutionControlBase, 276  
 is\_instance\_handle\_valid  
     TrickHLA::Object, 550  
 is\_joined\_federate  
     TrickHLA::Federate, 356, 357  
 is\_lateJoiner  
     TrickHLA::ExecutionControlBase, 276  
 is\_lateJoiner\_determined  
     TrickHLA::ExecutionControlBase, 276  
 is\_lateJoining\_federate  
     TrickHLA::Manager, 502  
 is\_locally\_owned  
     TrickHLA::Attribute, 73  
     TrickHLA::OwnershipHandler, 590  
 is\_master  
     TrickHLA::ExecutionControlBase, 277  
 is\_master\_preset  
     TrickHLA::ExecutionControlBase, 278  
 is\_mode\_transition\_requested  
     TrickHLA::ExecutionControlBase, 278  
 is\_MOM\_HLAfederate\_class  
     TrickHLA::Federate, 357  
 is\_MOM\_HLAfederation\_class  
     TrickHLA::Federate, 357  
 is\_MOM\_HLAfederation\_instance\_id  
     TrickHLA::Federate, 358  
 is\_mtr\_valid  
     DIS::ExecutionControl, 172  
     DSES::ExecutionControl, 188  
     IMSim::ExecutionControl, 208  
     SpaceFOM::ExecutionControl, 230  
 is\_name\_registered  
     TrickHLA::Object, 551  
 is\_name\_required

TrickHLA::Object, 551  
is\_object\_deleted\_from\_RTI  
    TrickHLA::Object, 551  
is\_pacing  
    SpaceFOM::ExecutionControl, 231  
is\_publish  
    TrickHLA::Attribute, 73  
    TrickHLA::Interaction, 442  
is\_published  
    TrickHLA::OwnershipHandler, 590  
is\_pull\_requested  
    TrickHLA::Attribute, 73  
is\_push\_requested  
    TrickHLA::Attribute, 74  
is\_receive\_order  
    TrickHLA::InteractionItem, 463  
is\_received  
    TrickHLA::Attribute, 74  
is\_registered  
    TrickHLA::SyncPnt, 735  
is\_remotely\_owned  
    TrickHLA::Attribute, 74  
    TrickHLA::OwnershipHandler, 591  
is\_required  
    TrickHLA::Object, 551  
is\_required\_federate  
    TrickHLA::Federate, 358  
is\_restore\_federate  
    TrickHLA::Manager, 502  
is\_root\_frame\_publisher  
    SpaceFOM::ExecutionControl, 231  
is\_RTI\_ready  
    TrickHLA::Manager, 502  
is\_running  
    TrickHLA::ExecutionControlBase, 278  
is\_save\_and\_restore\_supported  
    DIS::ExecutionControl, 172  
    IMSim::ExecutionControl, 209  
    TrickHLA::ExecutionControlBase, 279  
is\_save\_initiated  
    DIS::ExecutionControl, 172  
    IMSim::ExecutionControl, 209  
    TrickHLA::ExecutionControlBase, 279  
is\_shutdown  
    TrickHLA::ExecutionControlBase, 279  
    TrickHLA::Manager, 503  
is\_start\_to\_restore  
    TrickHLA::Federate, 358  
is\_static\_in\_size  
    TrickHLA::Attribute, 74  
    TrickHLA::Parameter, 616  
is\_subscribe  
    TrickHLA::Attribute, 75  
    TrickHLA::Interaction, 442  
is\_subscribed  
    TrickHLA::OwnershipHandler, 591  
is\_supported\_attribute\_type  
    TrickHLA::Attribute, 75  
is\_supported\_parameter\_type  
    TrickHLA::Parameter, 616  
is\_sync\_pnt\_announced  
    TrickHLA::SyncPntListBase, 751  
is\_synchronized  
    TrickHLA::SyncPnt, 735  
is\_this\_a\_rejoining\_federate  
    TrickHLA::Manager, 503  
is\_time\_advance\_granted  
    TrickHLA::Federate, 359  
is\_time\_management\_enabled  
    TrickHLA::Federate, 359  
is\_timestamp\_order  
    TrickHLA::InteractionItem, 463  
is\_transmission\_byteswap  
    TrickHLA::Utilities, 781  
is\_update\_requested  
    TrickHLA::Attribute, 75  
is\_valid  
    TrickHLA::SyncPnt, 735  
is\_zero\_lookahead\_time  
    TrickHLA::Federate, 359  
Item  
    TrickHLA::Item, 467  
Item.cpp, 859  
Item.hh, 860  
ItemQueue  
    TrickHLA::ItemQueue, 470, 471  
ItemQueue.cpp, 861  
ItemQueue.hh, 862  
job\_cycle\_time  
    TrickHLA::Manager, 521  
join\_federation  
    TrickHLA::Federate, 359  
join\_federation\_process  
    DIS::ExecutionControl, 172  
    DSES::ExecutionControl, 189  
    IMSim::ExecutionControl, 209  
    SpaceFOM::ExecutionControl, 231  
    TrickHLA::ExecutionControl, 249  
    TrickHLA::ExecutionControlBase, 279  
joined\_federate\_handles  
    TrickHLA::Federate, 394  
joined\_federate\_name\_map  
    TrickHLA::Federate, 394  
joined\_federate\_names  
    TrickHLA::Federate, 395  
known\_feds  
    TrickHLA::Federate, 395

known\_feds\_count  
     TrickHLA::Federate, 395

KnownFederate  
     TrickHLA::KnownFederate, 475

KnownFederate.hh, 863

label  
     TrickHLA::LoggableSyncPnt, 485

lag\_comp  
     TrickHLA::Object, 571

lag\_comp\_data  
     TrickHLAModel::SineLagCompensation, 700

lag\_comp\_type  
     TrickHLA::Object, 571

LAG\_COMPENSATION\_FIRST\_VALUE  
     TrickHLA, 51

LAG\_COMPENSATION\_LAST\_VALUE  
     TrickHLA, 51

LAG\_COMPENSATION\_NONE  
     TrickHLA, 51

LAG\_COMPENSATION\_RECEIVE\_SIDE  
     TrickHLA, 51

LAG\_COMPENSATION\_SEND\_SIDE  
     TrickHLA, 51

LagCompensation  
     TrickHLA::LagCompensation, 479

LagCompensation.cpp, 864

LagCompensation.hh, 865

LagCompensationEnum  
     TrickHLA, 50

last\_update\_time  
     TrickHLA::Object, 571

late\_joiner  
     TrickHLA::ExecutionControlBase, 297

late\_joiner\_determined  
     TrickHLA::ExecutionControlBase, 297

late\_joiner\_hla\_init\_process  
     DIS::ExecutionControl, 173

    SpaceFOM::ExecutionControl, 231

least\_common\_time\_step  
     IMSim::ExecutionConfiguration, 134

    SpaceFOM::ExecutionConfiguration, 146

    TrickHLA::ExecutionControlBase, 297

load\_and\_print\_running\_federate\_names  
     TrickHLA::Federate, 361

local\_settings  
     TrickHLA::Federate, 395

locally\_owned  
     TrickHLA::Attribute, 83

lock  
     TrickHLA::Interaction, 442

    TrickHLA::ItemQueue, 471

    TrickHLA::Object, 551

lock\_read\_only  
     TrickHLA::SyncPntListBase, 751

lock\_read\_write  
     TrickHLA::SyncPntListBase, 751

loggable\_sync\_pts  
     TrickHLA::ExecutionControlBase, 297

LoggableSyncPnt  
     TrickHLA::LoggableSyncPnt, 484

LoggableSyncPnt.hh, 867

LoggableTimedSyncPnt  
     TrickHLA::LoggableTimedSyncPnt, 487

LoggableTimedSyncPnt.hh, 868

logged\_sync\_pts\_count  
     TrickHLA::ExecutionControlBase, 298

lookahead  
     TrickHLA::Federate, 395

lookahead\_time  
     TrickHLA::Federate, 396

make\_copy\_of\_run\_directory  
     TrickHLA::Federate, 396

Manager  
     TrickHLA::ExecutionControlBase, 294

    TrickHLA::Manager, 494

manager  
     TrickHLA::ExecutionControlBase, 298

    TrickHLA::FedAmb, 322

    TrickHLA::Federate, 396

    TrickHLA::Interaction, 449

    TrickHLA::Object, 571

Manager.cpp, 869

    attrSRFOM\_\_MTRInteractionHandler, 871

    attrTrickHLA\_\_Interaction, 871

    attrTrickHLA\_\_InteractionItem, 871

    attrTrickHLA\_\_Parameter, 871

Manager.hh, 871

ManagerTypeOfInteractionEnum  
     TrickHLA, 51

mark\_all\_attributes\_as\_nonlocal  
     TrickHLA::Object, 552

mark\_announced  
     TrickHLA::SyncPntListBase, 752

mark\_changed  
     TrickHLA::Attribute, 75

    TrickHLA::Interaction, 442

    TrickHLA::Object, 552

    TrickHLA::Parameter, 617

mark\_locally\_owned  
     TrickHLA::Attribute, 76

mark\_object\_as\_deleted\_from\_federation  
     TrickHLA::ExecutionControlBase, 279

    TrickHLA::Manager, 503

mark\_registered  
     TrickHLA::SyncPntListBase, 752

mark\_remotely\_owned  
TrickHLA::Attribute, 76

mark\_required  
TrickHLA::Object, 552

mark\_synchronized  
IMSim::ExecutionControl, 209  
TrickHLA::SyncPntListBase, 752

mark\_unchanged  
TrickHLA::Attribute, 76  
TrickHLA::Interaction, 442  
TrickHLA::Object, 552  
TrickHLA::Parameter, 617

mass  
SpaceFOM::DynamicalEntity, 102

mass\_rate  
SpaceFOM::DynamicalEntity, 102

master  
TrickHLA::ExecutionControlBase, 298

MAX\_LOGICAL\_TIME\_SECONDS  
TrickHLA, 56

MAX\_VALUE\_IN\_MICROS  
TrickHLA, 56

message  
TrickHLAModel::SineInteractionHandler, 694

mgr\_initialized  
TrickHLA::Manager, 521

MICROS\_MULTIPLIER  
TrickHLA, 57

MIM\_module  
TrickHLA::Federate, 396

MIN\_TRICK\_MINOR  
CompileConfig.hh, 788

MIN\_TRICK\_PATCH  
CompileConfig.hh, 788

MIN\_TRICK\_VER  
CompileConfig.hh, 789

mode\_transition\_enum\_to\_int16  
TrickHLA, 54

mode\_transition\_enum\_to\_string  
TrickHLA, 54

MODE\_TRANSITION\_FIRST\_VALUE  
TrickHLA, 51

MODE\_TRANSITION\_GOTO\_FREEZE  
TrickHLA, 51

MODE\_TRANSITION\_GOTO\_RECONFIG  
TrickHLA, 51

MODE\_TRANSITION\_GOTO\_RESTART  
TrickHLA, 51

MODE\_TRANSITION\_GOTO\_RUN  
TrickHLA, 51

MODE\_TRANSITION\_GOTO\_SHUTDOWN  
TrickHLA, 51

MODE\_TRANSITION\_INITIALIZING  
TrickHLA, 51

mode\_transition\_int16\_to\_enum  
TrickHLA, 54

MODE\_TRANSITION\_LAST\_VALUE  
TrickHLA, 51

mode\_transition\_requested  
TrickHLA::ExecutionControlBase, 298

MODE\_TRANSITION\_UNINITIALIZED  
TrickHLA, 51

ModeTransitionEnum  
TrickHLA, 51

MOM\_HLAautoProvide\_handle  
TrickHLA::Federate, 396

MOM\_HLAautoProvide\_param\_handle  
TrickHLA::Federate, 396

MOM\_HLAfederate\_class\_handle  
TrickHLA::Federate, 397

MOM\_HLAfederate\_handle  
TrickHLA::Federate, 397

mom\_HLAfederate\_inst\_name\_map  
TrickHLA::Federate, 397

MOM\_HLAfederateName\_handle  
TrickHLA::Federate, 397

MOM\_HLAfederatesInFederation\_handle  
TrickHLA::Federate, 397

MOM\_HLAfederateType\_handle  
TrickHLA::Federate, 398

MOM\_HLAfederation\_class\_handle  
TrickHLA::Federate, 398

mom\_HLAfederation\_instance\_name\_map  
TrickHLA::Federate, 398

MOM\_HLAsetSwitches\_class\_handle  
TrickHLA::Federate, 398

MOM\_instance\_name  
TrickHLA::KnownFederate, 476

mtr\_enum\_to\_int16  
DIS, 21  
DSES, 27  
IMSim, 34  
SpaceFOM, 41

mtr\_enum\_to\_string  
DIS, 21  
DSES, 28  
IMSim, 34  
SpaceFOM, 41

MTR\_FIRST\_VALUE  
DIS, 19  
DSES, 26  
IMSim, 31  
SpaceFOM, 39

MTR\_GOTO\_FREEZE  
DIS, 19  
DSES, 26  
IMSim, 31  
SpaceFOM, 39

MTR\_GOTO\_RUN  
 DIS, 19  
 DSES, 26  
 IMSim, 31  
 SpaceFOM, 39

MTR\_GOTO\_SHUTDOWN  
 DIS, 19  
 DSES, 26  
 IMSim, 31  
 SpaceFOM, 39

MTR\_INITIALIZING  
 DIS, 19  
 DSES, 26  
 IMSim, 31  
 SpaceFOM, 39

mtr\_int16\_to\_enum  
 DIS, 22  
 DSES, 28  
 IMSim, 34  
 SpaceFOM, 42

mtr\_interaction  
 SpaceFOM::ExecutionControl, 242

mtr\_interaction\_handler  
 SpaceFOM::ExecutionControl, 243

MTR\_LAST\_VALUE  
 DIS, 19  
 DSES, 26  
 IMSim, 31  
 SpaceFOM, 39

mtr\_mode  
 SpaceFOM::MTRInteractionHandler, 528

mtr\_mode\_int  
 SpaceFOM::MTRInteractionHandler, 529

MTR\_UNINITIALIZED  
 DIS, 19  
 DSES, 26  
 IMSim, 31  
 SpaceFOM, 39

MTREnum  
 DIS, 19  
 DSES, 25  
 IMSim, 31  
 SpaceFOM, 39

MTRInteractionHandler  
 SpaceFOM::MTRInteractionHandler, 526

MTRInteractionHandler.cpp, 872

MTRInteractionHandler.hh, 874

multiphase\_init\_sync\_pnt\_list  
 TrickHLA::ExecutionControlBase, 298

multiphase\_init\_sync\_points  
 TrickHLA::ExecutionControlBase, 299

multipleObjectNameReservationFailed  
 TrickHLA::FedAmb, 312

multipleObjectNameReservationSucceeded

TrickHLA::FedAmb, 312

mutex  
 TrickHLA::Interaction, 449  
 TrickHLA::ItemQueue, 474  
 TrickHLA::Object, 572

name  
 RefFrameData, 659  
 SpaceFOM::MTRInteractionHandler, 529  
 SpaceFOM::PhysicalEntityBase, 644  
 SpaceFOM::RefFrameBase, 657  
 TrickHLA::Federate, 398  
 TrickHLA::KnownFederate, 476  
 TrickHLA::Object, 572  
 TrickHLAModel::SineData, 690  
 TrickHLAModel::SineInteractionHandler, 694

name\_attr  
 TrickHLAModel::SineLagCompensation, 700  
 TrickHLAModel::SinePacking, 712

name\_registered  
 TrickHLA::Object, 572

name\_required  
 TrickHLA::Object, 572

negotiated\_attribute\_ownership\_divestiture  
 TrickHLA::Object, 552

next  
 TrickHLA::Item, 468  
 TrickHLA::ItemQueue, 472

next\_execution\_mode  
 DIS::ExecutionConfiguration, 112  
 DSES::ExecutionConfiguration, 122  
 IMSim::ExecutionConfiguration, 134  
 SpaceFOM::ExecutionConfiguration, 146

next\_mode\_cte\_time  
 DIS::ExecutionConfiguration, 112  
 DSES::ExecutionConfiguration, 122  
 IMSim::ExecutionConfiguration, 134  
 SpaceFOM::ExecutionConfiguration, 147  
 TrickHLA::ExecutionControlBase, 299

next\_mode\_scenario\_time  
 DIS::ExecutionConfiguration, 112  
 DSES::ExecutionConfiguration, 122  
 IMSim::ExecutionConfiguration, 135  
 SpaceFOM::ExecutionConfiguration, 147  
 TrickHLA::ExecutionControlBase, 299

No\_Restore  
 TrickHLA, 52

NO\_THREAD\_WAIT\_FOR\_DATA  
 CompileConfig.hh, 789

NO\_TRICKHLA\_ENABLE\_FOM\_DUMP  
 CompileConfig.hh, 789

NO\_TRICKHLA\_ENABLE\_FPU\_CONTROL\_WORD\_VALIDATION  
 CompileConfig.hh, 789

notify\_attribute\_ownership\_changed

TrickHLA::Object, 553  
num\_federates  
TrickHLA::ExecutionConfiguration, 152  
TrickHLAModel::SimpleSimConfig, 673  
num\_items  
TrickHLA::Attribute, 83  
TrickHLA::Parameter, 621  
num\_state  
integrator.d, 850

obj\_count  
TrickHLA::Manager, 521  
Object  
TrickHLA::Object, 538  
TrickHLA::OwnershipHandler, 594  
object  
TrickHLA::LagCompensation, 482  
TrickHLA::OwnershipHandler, 595  
TrickHLA::Packing, 605  
Object.cpp, 875  
grant\_push\_pthread\_function, 877  
ownership\_divestiture\_pthread\_function, 878  
THLA\_OBJ\_DEBUG, 877  
THLA\_OBJ\_DEBUG\_RECEIVE, 877  
THLA\_OBJ\_DEBUG\_SEND, 877  
THLA\_OBJ\_DEBUG\_VALID\_OBJECT\_RECEIVE,  
877  
THLA\_OBJ\_OWNERSHIP\_DEBUG, 877  
Object.hh, 878  
\_TRICKHLA\_OBJECT\_HH\_, 880  
THLA\_USE\_ATTRIBUTE\_MAP\_CLASS\_INSTANCE,  
880  
object\_deleted\_from\_RTI  
TrickHLA::Object, 572  
object\_instance\_name\_reservation\_failed  
DIS::ExecutionControl, 173  
DSES::ExecutionControl, 189  
TrickHLA::ExecutionControlBase, 280  
TrickHLA::Manager, 504  
object\_instance\_name\_reservation\_succeeded  
TrickHLA::ExecutionControlBase, 280  
TrickHLA::Manager, 504  
object\_map  
TrickHLA::Manager, 522  
ObjectDeleted  
TrickHLA::ObjectDeleted, 575  
ObjectDeleted.hh, 880  
ObjectInstanceMap  
TrickHLA, 47  
objectInstanceNameReservationFailed  
TrickHLA::FedAmb, 312  
objectInstanceNameReservationSucceeded  
TrickHLA::FedAmb, 312  
objects  
TrickHLA::Manager, 522  
OBJECTS\_DISCOVERED\_SYNC\_POINT  
SpaceFOM, 43  
OpaqueBuffer  
TrickHLA::OpaqueBuffer, 578  
OpaqueBuffer.cpp, 881  
OpaqueBuffer.hh, 882  
operator!=  
TrickHLA::Int64Time, 424, 425  
operator<  
TrickHLA::Int64Time, 428  
operator<=

TrickHLA::Int64Time, 428  
operator>  
TrickHLA::Int64Interval, 416  
TrickHLA::Int64Time, 430, 431  
operator>=

TrickHLA::Int64Time, 431  
operator\*  
TrickHLA::Int64Time, 425  
operator+  
TrickHLA::Int64Time, 426  
operator+=

TrickHLA::Int64Time, 426, 427  
operator-  
TrickHLA::Int64Time, 427  
operator/  
TrickHLA::Int64Time, 428  
operator=

DIS::ExecutionConfiguration, 107  
DIS::ExecutionControl, 173  
DIS::PausePointList, 637  
DSES::ExecutionConfiguration, 118  
DSES::ExecutionControl, 189  
IMSim::ExecutionConfiguration, 128  
IMSim::ExecutionControl, 210  
IMSim::FreezeInteractionHandler, 409  
IMSim::PausePointList, 631  
SpaceFOM::DynamicalEntity, 101  
SpaceFOM::ExecutionConfiguration, 141  
SpaceFOM::ExecutionControl, 232  
SpaceFOM::MTRInteractionHandler, 526  
SpaceFOM::PhysicalEntityBase, 642  
SpaceFOM::QuaternionEncoder, 649  
SpaceFOM::RefFrameBase, 655  
SpaceFOM::SpaceTimeCoordinateEncoder, 718  
TrickHLA::Attribute, 76  
TrickHLA::CTETimelineBase, 92  
TrickHLA::ExecutionConfiguration, 151  
TrickHLA::ExecutionConfigurationBase, 157  
TrickHLA::ExecutionControl, 249  
TrickHLA::ExecutionControlBase, 281  
TrickHLA::FedAmb, 313  
TrickHLA::Federate, 361

TrickHLA::Int64Interval, 415, 416  
 TrickHLA::Int64Time, 429  
 TrickHLA::Interaction, 443  
 TrickHLA::InteractionHandler, 455  
 TrickHLA::InteractionItem, 463  
 TrickHLA::Item, 467  
 TrickHLA::ItemQueue, 472  
 TrickHLA::KnownFederate, 476  
 TrickHLA::LagCompensation, 481  
 TrickHLA::LoggableSyncPnt, 484  
 TrickHLA::LoggableTimedSyncPnt, 487  
 TrickHLA::Manager, 504  
 TrickHLA::Object, 553  
 TrickHLA::OwnershipHandler, 591  
 TrickHLA::OwnershipItem, 598  
 TrickHLA::Packing, 604  
 TrickHLA::Parameter, 617  
 TrickHLA::ParameterItem, 625  
 TrickHLA::ReflectedAttributesQueue, 662  
 TrickHLA::ScenarioTimeline, 667  
 TrickHLA::SimTimeline, 676  
 TrickHLA::StringUtilities, 723  
 TrickHLA::SyncPntList, 741  
 TrickHLA::SyncPntListBase, 753  
 TrickHLA::TimedSyncPntList, 769  
 TrickHLA::Timeline, 772  
 TrickHLA::TimeOfDayTimeline, 775  
 TrickHLA::Utilities, 782  
 TrickHLAModel::SimpleSimConfig, 672  
 TrickHLAModel::SineInteractionHandler, 693  
 TrickHLAModel::SineLagCompensation, 698  
 TrickHLAModel::SineObjectDeleted, 703  
 TrickHLAModel::SineOwnershipHandler, 706  
 TrickHLAModel::SinePacking, 710  
 operator==  
     TrickHLA::Int64Time, 430  
 operator%  
     TrickHLA::Int64Time, 425  
 option  
     integrator.d, 850  
 order\_is\_TSO  
     TrickHLA::InteractionItem, 464  
 orig\_auto\_provide\_setting  
     TrickHLA::Federate, 399  
 original\_head  
     TrickHLA::ItemQueue, 474  
 owner  
     TrickHLA::ExecutionConfiguration, 152  
     TrickHLAModel::SimpleSimConfig, 673  
 ownership  
     TrickHLA::Object, 573  
 ownership\_divestiture\_pthread\_function  
     Object.cpp, 878  
 ownership\_lock

TrickHLA::Object, 553  
 ownership\_mutex  
     TrickHLA::Object, 573  
 ownership\_unlock  
     TrickHLA::Object, 553  
 OwnershipHandler  
     TrickHLA::OwnershipHandler, 587  
 OwnershipHandler.cpp, 884  
 OwnershipHandler.hh, 885  
 OwnershipItem  
     TrickHLA::OwnershipItem, 598  
 OwnershipItem.h, 886

pacing  
     SpaceFOM::ExecutionControl, 243

pack  
     DIS::ExecutionConfiguration, 107  
     DSES::ExecutionConfiguration, 118  
     IMSim::ExecutionConfiguration, 129  
     SpaceFOM::DynamicalEntity, 101  
     SpaceFOM::ExecutionConfiguration, 141  
     SpaceFOM::PhysicalEntityBase, 642  
     SpaceFOM::RefFrameBase, 655  
     TrickHLA::ExecutionConfiguration, 151  
     TrickHLA::ExecutionConfigurationBase, 157  
     TrickHLA::Packing, 604  
     TrickHLAModel::SimpleSimConfig, 672  
     TrickHLAModel::SinePacking, 710

pack\_attribute\_buffer  
     TrickHLA::Attribute, 76

pack\_attribute\_buffers  
     TrickHLA::Object, 553

pack\_count  
     TrickHLAModel::SinePacking, 712

pack\_cyclic\_attribute\_buffers  
     TrickHLA::Object, 554

pack\_init\_attribute\_buffers  
     TrickHLA::Object, 554

pack\_parameter\_buffer  
     TrickHLA::Parameter, 617

pack\_requested\_attribute\_buffers  
     TrickHLA::Object, 554

Packing  
     TrickHLA::Packing, 601

packing  
     TrickHLA::Object, 573

Packing.cpp, 888  
 Packing.hh, 889

param\_count  
     TrickHLA::Interaction, 449

param\_handle  
     TrickHLA::Parameter, 621

Parameter  
     TrickHLA::Parameter, 609

Parameter.cpp, 890  
Parameter.hh, 891  
parameter\_queue  
    TrickHLA::InteractionItem, 465  
ParameterItem  
    TrickHLA::ParameterItem, 624, 625  
ParameterItem.cpp, 893  
ParameterItem.hh, 894  
parameters  
    TrickHLA::Interaction, 450  
parent\_name  
    RefFrameData, 659  
    SpaceFOM::RefFrameBase, 657  
parent\_ref\_frame  
    SpaceFOM::PhysicalEntityBase, 644  
parm\_items  
    TrickHLA::InteractionItem, 465  
parm\_items\_count  
    TrickHLA::InteractionItem, 465  
pause\_pnt\_state\_enum\_to\_int16  
    DIS, 22  
    IMSim, 35  
pause\_pnt\_state\_enum\_to\_string  
    DIS, 23  
    IMSim, 35  
pause\_pnt\_state\_int16\_to\_enum  
    DIS, 23  
    IMSim, 35  
PAUSE\_POINT\_STATE\_ACKNOWLEDGED  
    DIS, 19  
    IMSim, 32  
PAUSE\_POINT\_STATE\_ERROR  
    DIS, 19  
    IMSim, 32  
PAUSE\_POINT\_STATE\_EXIT  
    DIS, 19  
    IMSim, 32  
PAUSE\_POINT\_STATE\_FIRST\_VALUE  
    DIS, 19  
    IMSim, 32  
PAUSE\_POINT\_STATE\_FREEZE  
    DIS, 19  
    IMSim, 32  
PAUSE\_POINT\_STATE\_PENDING  
    DIS, 19  
    IMSim, 32  
PAUSE\_POINT\_STATE\_RECONFIG  
    DIS, 19  
    IMSim, 32  
PAUSE\_POINT\_STATE\_RESTART  
    DIS, 19  
    IMSim, 32  
PAUSE\_POINT\_STATE\_RUN  
    DIS, 19  
IMSim, 32  
PAUSE\_POINT\_STATE\_UNKNOWN  
    DIS, 19  
    IMSim, 32  
pause\_sync\_pts  
    DIS::ExecutionControl, 181  
    IMSim::ExecutionControl, 220  
PausePointList  
    DIS::PausePointList, 635  
    IMSim::PausePointList, 629  
PausePointList.cpp, 895, 896  
PausePointList.hh, 897, 899  
PausePointStateEnum  
    DIS, 19  
    IMSim, 32  
pending\_mtr  
    DIS::ExecutionControl, 181  
    DSES::ExecutionControl, 196  
    IMSim::ExecutionControl, 220  
    SpaceFOM::ExecutionControl, 243  
pending\_update  
    TrickHLA::ExecutionConfigurationBase, 163  
perform\_checkpoint  
    TrickHLA::Federate, 361  
perform\_restore  
    TrickHLA::Federate, 361  
perform\_save  
    IMSim::ExecutionControl, 210  
    TrickHLA::ExecutionControlBase, 281  
perform\_time\_advance\_request  
    TrickHLA::Federate, 362  
phase  
    TrickHLAModel::SineData, 690  
phase\_attr  
    TrickHLAModel::SineLagCompensation, 700  
    TrickHLAModel::SinePacking, 712  
phase\_deg  
    TrickHLAModel::SinePacking, 712  
PhysicalEntityBase  
    SpaceFOM::PhysicalEntityBase, 641  
PhysicalEntityBase.cpp, 901  
    trick\_MM, 902  
PhysicalEntityBase.hh, 902  
pop  
    TrickHLA::ItemQueue, 472  
    TrickHLA::ReflectedAttributesQueue, 662  
pos  
    SpaceTimeCoordinateData, 714  
position  
    SpaceFOM::SpaceTimeCoordinateEncoder, 719  
position\_encoder  
    SpaceFOM::SpaceTimeCoordinateEncoder, 719  
post\_checkpoint  
    TrickHLA::Federate, 362

post\_multi\_phase\_init\_process  
 DIS::ExecutionControl, 173  
 DSES::ExecutionControl, 190  
 IMSim::ExecutionControl, 210  
 post\_multi\_phase\_init\_processes  
 SpaceFOM::ExecutionControl, 232  
 TrickHLA::ExecutionControl, 250  
 TrickHLA::ExecutionControlBase, 281  
 post\_multiphase\_initialization  
 TrickHLA::Federate, 362  
 post\_restore  
 TrickHLA::Federate, 362  
 pre\_multi\_phase\_init\_processes  
 DIS::ExecutionControl, 173, 174  
 DSES::ExecutionControl, 190  
 IMSim::ExecutionControl, 210  
 SpaceFOM::ExecutionControl, 232  
 TrickHLA::ExecutionControl, 250  
 TrickHLA::ExecutionControlBase, 281  
 pre\_multiphase\_initialization  
 TrickHLA::Federate, 363  
 preferred\_order  
 TrickHLA::Attribute, 83  
 TrickHLA::Interaction, 450  
 prev\_restore\_process  
 TrickHLA::Federate, 399  
 prev\_sim\_data  
 TrickHLAModel::SineConditional, 680  
 print\_buffer  
 TrickHLA::Attribute, 77  
 TrickHLA::Parameter, 618  
 print\_execution\_configuration  
 DIS::ExecutionConfiguration, 108  
 DSES::ExecutionConfiguration, 118  
 IMSim::ExecutionConfiguration, 129  
 SpaceFOM::ExecutionConfiguration, 141  
 TrickHLA::ExecutionConfiguration, 151  
 TrickHLA::ExecutionConfigurationBase, 157  
 print\_requested\_federation\_restore\_status  
 TrickHLA::Federate, 363  
 print\_restore\_failure\_reason  
 TrickHLA::Federate, 363  
 print\_save\_failure\_reason  
 TrickHLA::Federate, 364  
 print\_sync\_pnts  
 DIS::PausePointList, 637  
 IMSim::PausePointList, 631  
 TrickHLA::SyncPntListBase, 753  
 TrickHLA::TimedSyncPntList, 769  
 print\_version  
 TrickHLA::Federate, 364  
 process\_deleted\_object  
 TrickHLA::Object, 554  
 process\_deleted\_objects

TrickHLA::ExecutionControlBase, 281  
 TrickHLA::Manager, 504  
 process\_execution\_control\_updates  
 DIS::ExecutionControl, 174  
 DSES::ExecutionControl, 190  
 IMSim::ExecutionControl, 211  
 SpaceFOM::ExecutionControl, 233  
 TrickHLA::ExecutionControl, 250  
 TrickHLA::ExecutionControlBase, 281  
 process\_interaction  
 TrickHLA::Interaction, 443  
 process\_interactions  
 TrickHLA::Manager, 505  
 process\_mode\_interaction  
 DIS::ExecutionControl, 175  
 DSES::ExecutionControl, 191  
 IMSim::ExecutionControl, 212  
 SpaceFOM::ExecutionControl, 234  
 TrickHLA::ExecutionControl, 250  
 TrickHLA::ExecutionControlBase, 282  
 process\_mode\_transition\_request  
 DIS::ExecutionControl, 175  
 DSES::ExecutionControl, 191  
 IMSim::ExecutionControl, 212  
 SpaceFOM::ExecutionControl, 234  
 process\_ownership  
 TrickHLA::Manager, 505  
 process\_requested\_federation\_restore\_status  
 TrickHLA::Federate, 364  
 process\_requested\_federation\_save\_status  
 TrickHLA::Federate, 364  
 provide\_attribute\_update  
 TrickHLA::ExecutionControlBase, 282  
 TrickHLA::Manager, 505  
 TrickHLA::Object, 554  
 provideAttributeValueUpdate  
 TrickHLA::FedAmb, 313  
 publish  
 DIS::ExecutionControl, 175  
 DSES::ExecutionControl, 192  
 IMSim::ExecutionControl, 213  
 SpaceFOM::ExecutionControl, 234  
 TrickHLA::Attribute, 83  
 TrickHLA::ExecutionControl, 251  
 TrickHLA::ExecutionControlBase, 282  
 TrickHLA::Interaction, 450  
 TrickHLA::Manager, 505  
 publish\_and\_subscribe  
 TrickHLA::Manager, 505  
 publish\_data  
 TrickHLA::Federate, 399  
 publish\_interaction  
 TrickHLA::Interaction, 443  
 publish\_interaction\_class

TrickHLA::Federate, 365  
publish\_object\_attributes  
    TrickHLA::Object, 555  
pull\_from\_buffer  
    TrickHLA::OpaqueBuffer, 580  
pull\_items  
    TrickHLA::OwnershipHandler, 595  
pull\_items\_cnt  
    TrickHLA::OwnershipHandler, 595  
pull\_ownership  
    TrickHLA::Manager, 506  
    TrickHLA::Object, 555  
    TrickHLA::OwnershipHandler, 591, 592  
pull\_ownership\_upon\_rejoin  
    TrickHLA::Manager, 506  
    TrickHLA::Object, 555  
pull\_pad\_from\_buffer  
    TrickHLA::OpaqueBuffer, 581  
pull\_pos  
    TrickHLA::OpaqueBuffer, 583  
pull\_requested  
    TrickHLA::Attribute, 84  
    TrickHLA::Object, 573  
pull\_requests  
    TrickHLA::OwnershipHandler, 595  
push  
    TrickHLA::ItemQueue, 472  
    TrickHLA::ReflectedAttributesQueue, 662  
push\_items  
    TrickHLA::OwnershipHandler, 595  
push\_items\_cnt  
    TrickHLA::OwnershipHandler, 595  
push\_ownership  
    TrickHLA::Manager, 506  
    TrickHLA::Object, 556  
    TrickHLA::OwnershipHandler, 592, 593  
push\_pad\_to\_buffer  
    TrickHLA::OpaqueBuffer, 581  
push\_pos  
    TrickHLA::OpaqueBuffer, 584  
push\_requested  
    TrickHLA::Attribute, 84  
push\_requests  
    TrickHLA::OwnershipHandler, 596  
push\_to\_buffer  
    TrickHLA::OpaqueBuffer, 581

quat\_encoder  
    SpaceFOM::PhysicalEntityBase, 644  
    SpaceFOM::SpaceTimeCoordinateEncoder, 719  
quat\_scalar  
    SpaceTimeCoordinateData, 714  
quat\_scalar\_encoder  
    SpaceFOM::SpaceTimeCoordinateEncoder, 719

quat\_vector  
    SpaceFOM::SpaceTimeCoordinateEncoder, 720  
    SpaceTimeCoordinateData, 714  
quat\_vector\_encoder  
    SpaceFOM::SpaceTimeCoordinateEncoder, 720  
QuaternionData, 645  
    scalar, 646  
    vector, 646  
QuaternionData.h, 904  
QuaternionEncoder  
    SpaceFOM::QuaternionEncoder, 648  
QuaternionEncoder.cpp, 905  
QuaternionEncoder.hh, 906  
queue\_mutex  
    TrickHLA::ReflectedAttributesQueue, 663

read\_locks  
    TrickHLA::SyncPntListBase, 758  
read\_running\_feds\_file  
    TrickHLA::Federate, 365  
README.md, 908  
rebuild\_federate\_handles  
    TrickHLA::Federate, 365  
receive\_cnt  
    SpaceFOM::MTRInteractionHandler, 529  
    TrickHLAModel::SineInteractionHandler, 694  
receive\_cyclic\_data  
    TrickHLA::ExecutionControlBase, 282  
    TrickHLA::Manager, 506  
    TrickHLA::Object, 556  
receive\_execution\_configuration  
    TrickHLA::ExecutionControlBase, 283  
receive\_init\_data  
    TrickHLA::Manager, 507  
    TrickHLA::Object, 556  
receive\_init\_root\_ref\_frame  
    SpaceFOM::ExecutionControl, 234  
receive\_interaction  
    IMSim::ExecutionControl, 213  
    IMSim::FreezeInteractionHandler, 409  
    SpaceFOM::ExecutionControl, 235  
    SpaceFOM::MTRInteractionHandler, 527  
    TrickHLA::ExecutionControl, 251  
    TrickHLA::ExecutionControlBase, 283  
    TrickHLA::InteractionHandler, 456  
    TrickHLA::Manager, 507  
    TrickHLAModel::SineInteractionHandler, 693  
receive\_lag\_compensation  
    TrickHLA::LagCompensation, 481  
    TrickHLAModel::SineLagCompensation, 698  
receive\_root\_ref\_frame  
    SpaceFOM::ExecutionControl, 235  
received\_as\_TSO  
    TrickHLA::Interaction, 450

receiveInteraction  
     TrickHLA::FedAmb, 313, 314  
 reconfig\_name  
     TrickHLA::SyncPntListBase, 759  
 ref2  
     TrickHLA::Attribute, 84  
 ref\_frame\_attr  
     SpaceFOM::RefFrameBase, 657  
 ref\_frame\_data  
     SpaceFOM::RefFrameBase, 657  
 REF\_FRAME\_PACKING\_DEBUG  
     RefFrameBase.cpp, 909  
 REF\_FRAME\_PACKING\_EXTRA\_DEBUG  
     RefFrameBase.cpp, 909  
 RefFrameBase  
     SpaceFOM::RefFrameBase, 652, 653  
 RefFrameBase.cpp, 908  
     REF\_FRAME\_PACKING\_DEBUG, 909  
     REF\_FRAME\_PACKING\_EXTRA\_DEBUG, 909  
     trick\_MM, 909  
 RefFrameBase.hh, 909  
 RefFrameData, 658  
     name, 659  
     parent\_name, 659  
     state, 659  
 RefFrameData.h, 911  
 reflectAttributeValues  
     TrickHLA::FedAmb, 314, 315  
 ReflectedAttributesQueue  
     TrickHLA::ReflectedAttributesQueue, 661  
 ReflectedAttributesQueue.cpp, 912  
 ReflectedAttributesQueue.hh, 913  
 register\_all\_sync\_pnts  
     TrickHLA::SyncPntListBase, 753  
 register\_generic\_sync\_point  
     TrickHLA::Federate, 366  
 register\_interactions\_with\_RTI  
     TrickHLA::ExecutionControlBase, 283  
 register\_object\_with\_RTI  
     TrickHLA::Object, 557  
 register\_objects\_with\_RTI  
     TrickHLA::ExecutionControlBase, 284  
     TrickHLA::Manager, 508  
 register\_sync\_pnt  
     TrickHLA::SyncPntListBase, 754  
 register\_sync\_point  
     TrickHLA::SyncPnt, 735, 736  
 reinstate\_logged\_sync\_pts  
     IMSim::ExecutionControl, 214  
     TrickHLA::ExecutionControlBase, 284  
     TrickHLA::Federate, 366  
 rejoining\_federate  
     TrickHLA::Manager, 522  
 release\_ownership  
     TrickHLA::Manager, 508  
     TrickHLA::Object, 557  
 remove  
     TrickHLA::Interaction, 443  
     TrickHLA::Object, 557  
 remove\_execution\_configuration  
     TrickHLA::ExecutionControlBase, 284  
 remove\_federate\_instance\_id  
     TrickHLA::Federate, 366  
 remove\_MOM\_HLAfederate\_instance\_id  
     TrickHLA::Federate, 366  
 remove\_MOM\_HLAfederation\_instance\_id  
     TrickHLA::Federate, 367  
 remove\_object\_instance  
     TrickHLA::Object, 558  
 removed\_instance  
     TrickHLA::Object, 573  
 removeObjectInstance  
     TrickHLA::FedAmb, 315  
 reportAttributeTransportationType  
     TrickHLA::FedAmb, 316  
 reportFederationExecutions  
     TrickHLA::FedAmb, 316  
 reportInteractionTransportationType  
     TrickHLA::FedAmb, 316  
 request\_attribute\_update  
     TrickHLA::Federate, 367  
 request\_attribute\_value\_update  
     TrickHLA::Object, 558  
 request\_data\_update  
     TrickHLA::Manager, 508, 509  
 request\_federation\_restore\_status  
     TrickHLA::Federate, 367  
 request\_federation\_save  
     TrickHLA::Federate, 368  
 request\_federation\_save\_status  
     TrickHLA::Federate, 368  
 requestAttributeOwnershipAssumption  
     TrickHLA::FedAmb, 316  
 requestAttributeOwnershipRelease  
     TrickHLA::FedAmb, 316  
 requestDivestitureConfirmation  
     TrickHLA::FedAmb, 317  
 requested\_execution\_control\_mode  
     TrickHLA::ExecutionControlBase, 299  
 requested\_federation\_restore\_status  
     TrickHLA::Federate, 368  
 requested\_time  
     TrickHLA::Federate, 399  
 requestFederationRestoreFailed  
     TrickHLA::FedAmb, 317  
 requestFederationRestoreSucceeded  
     TrickHLA::FedAmb, 317  
 requestRetraction

TrickHLA::FedAmb, 317  
required  
TrickHLA::KnownFederate, 477  
TrickHLA::Object, 574  
required\_federates  
TrickHLA::ExecutionConfiguration, 152  
TrickHLAModel::SimpleSimConfig, 673  
reserve\_object\_name\_with\_RTI  
TrickHLA::Object, 558  
reserve\_object\_names\_with\_RTI  
TrickHLA::Manager, 509  
reset  
TrickHLA::SyncPntListBase, 755  
reset\_buffer\_positions  
TrickHLA::OpaqueBuffer, 582  
reset\_federation\_restored\_rebuild\_federate\_handle\_set  
TrickHLA::FedAmb, 317  
reset\_mgr\_initialized  
TrickHLA::Manager, 509  
reset\_ownership\_states  
TrickHLA::ExecutionConfigurationBase, 158  
reset\_preferred\_order  
TrickHLA::ExecutionConfigurationBase, 158  
reset\_pull\_position  
TrickHLA::OpaqueBuffer, 582  
reset\_push\_position  
TrickHLA::OpaqueBuffer, 582  
resign  
TrickHLA::Federate, 368  
resign\_so\_we\_can\_rejoin  
TrickHLA::Federate, 369  
restart\_cfg\_flag  
TrickHLA::Federate, 399  
restart\_checkpoint  
TrickHLA::Federate, 369  
restart\_flag  
TrickHLA::Federate, 399  
restart\_initialization  
TrickHLA::Federate, 369  
TrickHLA::Manager, 509  
restore\_begun  
TrickHLA::Federate, 400  
restore\_checkpoint  
TrickHLA::Federate, 370  
Restore\_Complete  
TrickHLA, 52  
restore\_completed  
TrickHLA::Federate, 400  
restore\_determined  
TrickHLA::Manager, 522  
Restore\_Failed  
TrickHLA, 52  
restore\_failed  
TrickHLA::Federate, 400  
restore\_federate  
TrickHLA::Manager, 523  
restore\_federate\_handles\_from\_MOM  
TrickHLA::Federate, 370  
restore\_federation  
TrickHLA::Manager, 523  
restore\_file\_name  
TrickHLA::Manager, 523  
Restore\_In\_Progress  
TrickHLA, 52  
restore\_in\_progress  
TrickHLA::Federate, 400  
restore\_interactions  
TrickHLA::Manager, 510  
restore\_is\_imminent  
TrickHLA::Federate, 400  
restore\_label\_generated  
TrickHLA::Federate, 400  
restore\_name  
TrickHLA::Federate, 401  
restore\_orig\_MOM\_auto\_provide\_setting  
TrickHLA::Federate, 370  
restore\_ownership\_transfer\_checkpointed\_data  
TrickHLA::Object, 558  
restore\_process  
TrickHLA::Federate, 401  
restore\_queue  
TrickHLA::InteractionItem, 464  
restore\_request\_complete  
TrickHLA::Federate, 401  
Restore\_Request\_Failed  
TrickHLA, 52  
Restore\_Request\_Succeeded  
TrickHLA, 52  
restore\_requests  
TrickHLA::OwnershipHandler, 593  
rewind  
TrickHLA::ItemQueue, 473  
role\_determination\_process  
DIS::ExecutionControl, 176  
SpaceFOM::ExecutionControl, 236  
ROOT\_FRAME\_DISCOVERED\_SYNC\_POINT  
SpaceFOM, 43  
root\_frame\_name  
DIS::ExecutionConfiguration, 112  
DSES::ExecutionConfiguration, 123  
IMSim::ExecutionConfiguration, 135  
SpaceFOM::ExecutionConfiguration, 147  
root\_frame\_pub  
SpaceFOM::ExecutionControl, 243  
root\_ref\_frame  
SpaceFOM::ExecutionControl, 243  
rot\_accel  
SpaceFOM::PhysicalEntityBase, 644

rot\_state\_encoder  
     SpaceFOM::SpaceTimeCoordinateEncoder, 720

RTI1516\_EXCEPTION  
     StandardsSupport.hh, 944

RTI1516\_HEADER  
     StandardsSupport.hh, 944

RTI1516\_NAMESPACE  
     StandardsSupport.hh, 944

RTI1516\_USERDATA  
     StandardsSupport.hh, 944

RTI\_ambassador  
     TrickHLA::Federate, 401

rti\_ambassador  
     TrickHLA::Object, 574

rti\_encoding  
     TrickHLA::Attribute, 84

run\_duration  
     TrickHLA::ExecutionConfiguration, 153

run\_duration\_microsec  
     TrickHLA::ExecutionConfiguration, 153

run\_mode\_transition  
     DIS::ExecutionControl, 176

running\_feds  
     TrickHLA::Federate, 402

running\_feds\_count  
     TrickHLA::Federate, 402

running\_feds\_count\_at\_time\_of\_restore  
     TrickHLA::Federate, 402

S\_define\_name  
     TrickHLA::ExecutionConfigurationBase, 163

save\_completed  
     TrickHLA::Federate, 402

save\_label\_generated  
     TrickHLA::Federate, 402

save\_name  
     TrickHLA::Federate, 402

save\_request\_complete  
     TrickHLA::Federate, 403

scalar  
     QuaternionData, 646

scalar\_encoder  
     SpaceFOM::QuaternionEncoder, 649

scenario\_freeze\_time  
     TrickHLA::ExecutionControlBase, 299

scenario\_time  
     SpaceFOM::MTRInteractionHandler, 529

scenario\_time\_epoch  
     DIS::ExecutionConfiguration, 112

send  
     TrickHLA::Interaction, 443, 444

send\_cnt  
     SpaceFOM::MTRInteractionHandler, 529

send\_cyclic\_data  
     TrickHLA::Manager, 510

send\_execution\_configuration  
     TrickHLA::ExecutionControlBase, 284

send\_init\_data  
     TrickHLA::Manager, 510

send\_init\_root\_ref\_frame  
     SpaceFOM::ExecutionControl, 236

send\_interaction  
     SpaceFOM::MTRInteractionHandler, 527

send\_lag\_compensation  
     TrickHLA::LagCompensation, 481

send\_mode\_transition\_interaction  
     IMSim::ExecutionControl, 214

send\_MTR\_interaction  
     SpaceFOM::ExecutionControl, 237

send\_requested\_data  
     TrickHLA::ExecutionControlBase, 285

send\_root\_ref\_frame  
     SpaceFOM::ExecutionControl, 237

send\_scenario\_freeze\_interaction  
     IMSim::FreezeInteractionHandler, 409

send\_sine\_interaction  
     TrickHLAModel::SineInteractionHandler, 693

set  
     TrickHLA::DebugHandler, 95

set\_all\_federate\_MOM\_instance\_handles\_by\_name

TrickHLA::Federate, 371  
set\_all\_object\_instance\_handles\_by\_name  
    TrickHLA::Manager, 511  
set\_amplitude  
    TrickHLAModel::SineData, 687  
set\_announce\_save  
    TrickHLA::Federate, 371  
set\_attribute\_handle  
    TrickHLA::Attribute, 77  
set\_byte\_alignment  
    TrickHLA::OpaqueBuffer, 582  
set\_checkpoint\_file\_name  
    TrickHLA::Federate, 371  
set\_class\_handle  
    TrickHLA::Interaction, 444  
    TrickHLA::Object, 560  
set\_clock\_ID  
    TrickHLA::CTETimelineBase, 93  
set\_configuration  
    TrickHLA::Attribute, 77  
set\_core\_job\_cycle\_time  
    TrickHLA::Object, 561  
set\_create\_HLA\_instance  
    TrickHLA::Object, 561  
set\_current\_execution\_control\_mode  
    TrickHLA::ExecutionControlBase, 285  
set\_current\_execution\_mode  
    DIS::ExecutionConfiguration, 108  
    DSES::ExecutionConfiguration, 118, 119  
    IMSim::ExecutionConfiguration, 129  
    SpaceFOM::ExecutionConfiguration, 142  
set\_debug\_level  
    TrickHLA::Attribute, 77  
    TrickHLA::Parameter, 618  
    TrickHLA::SyncPntListBase, 755  
set\_derivative  
    TrickHLAModel::SineData, 687  
set\_divest\_requested  
    TrickHLA::Attribute, 77  
    TrickHLA::Object, 561  
set\_encoding  
    TrickHLA::Attribute, 78  
    TrickHLA::Parameter, 618  
set\_epoch  
    TrickHLA::Timeline, 772  
set\_execution\_configuration  
    TrickHLA::ExecutionControlBase, 285  
    TrickHLA::Manager, 511  
set\_execution\_control  
    TrickHLA::ExecutionConfigurationBase, 158  
set\_federate  
    TrickHLA::ExecutionControlBase, 286  
set\_federate\_has\_begun\_execution  
    TrickHLA::Federate, 371  
set\_federation\_name  
    TrickHLA::Federate, 371  
set\_federation\_restore\_status\_response\_to\_echo  
    TrickHLA::FedAmb, 318  
set\_federation\_restore\_status\_response\_to\_process  
    TrickHLA::FedAmb, 318  
set\_federation\_restored\_rebuild\_federate\_handle\_set  
    TrickHLA::FedAmb, 318  
set\_FOM\_name  
    TrickHLA::Interaction, 445  
    TrickHLA::Parameter, 618  
set\_freeze\_announced  
    TrickHLA::Federate, 372  
set\_frequency  
    TrickHLAModel::SineData, 687  
set\_granted\_time  
    TrickHLA::Federate, 372  
set\_handler  
    TrickHLA::Interaction, 445  
set\_HTML\_offset  
    TrickHLA::ScenarioTimeline, 667  
set\_instance\_handle  
    TrickHLA::Object, 561  
set\_instance\_handle\_and\_name  
    TrickHLA::Object, 562  
set\_label  
    TrickHLA::SyncPnt, 736  
set\_lag\_compensation\_type  
    TrickHLA::Object, 562  
set\_last\_update\_time  
    TrickHLA::Object, 562  
set\_least\_common\_time\_step  
    IMSim::ExecutionConfiguration, 130  
    SpaceFOM::ExecutionConfiguration, 142  
    SpaceFOM::ExecutionControl, 237  
    TrickHLA::ExecutionControl, 252  
    TrickHLA::ExecutionControlBase, 286  
set\_looking\_ahead  
    TrickHLA::Federate, 373  
set\_master  
    TrickHLA::ExecutionConfigurationBase, 158  
    TrickHLA::ExecutionControlBase, 286  
set\_mode\_request\_from\_mtr  
    DIS::ExecutionControl, 176  
    DSES::ExecutionControl, 192  
    IMSim::ExecutionControl, 214  
    SpaceFOM::ExecutionControl, 238  
set\_mode\_transition\_requested  
    TrickHLA::ExecutionControlBase, 287  
set\_MOM\_HLAfederate\_instance\_attributes  
    TrickHLA::Federate, 373  
set\_MOM\_HLAfederation\_instance\_attributes  
    TrickHLA::Federate, 373  
set\_name

SpaceFOM::MTRInteractionHandler, 527  
 SpaceFOM::PhysicalEntityBase, 642  
 SpaceFOM::RefFrameBase, 655  
 TrickHLA::Object, 563  
 TrickHLAModel::SineData, 688  
 set\_name\_and\_mark\_changed  
     TrickHLA::Object, 563  
 set\_name\_registered  
     TrickHLA::Object, 563  
 set\_name\_unregisterd  
     TrickHLA::Object, 563  
 set\_next\_execution\_control\_mode  
     DIS::ExecutionControl, 176  
     DSES::ExecutionControl, 193  
     IMSim::ExecutionControl, 215  
     SpaceFOM::ExecutionControl, 238  
     TrickHLA::ExecutionControl, 252  
     TrickHLA::ExecutionControlBase, 287  
 set\_next\_execution\_mode  
     DIS::ExecutionConfiguration, 108, 109  
     DSES::ExecutionConfiguration, 119  
     IMSim::ExecutionConfiguration, 130  
     SpaceFOM::ExecutionConfiguration, 143  
 set\_next\_mode\_cte\_time  
     DIS::ExecutionConfiguration, 109  
     DSES::ExecutionConfiguration, 119  
     IMSim::ExecutionConfiguration, 131  
     SpaceFOM::ExecutionConfiguration, 143  
 set\_next\_mode\_scenario\_time  
     DIS::ExecutionConfiguration, 109  
     DSES::ExecutionConfiguration, 120  
     IMSim::ExecutionConfiguration, 131  
     SpaceFOM::ExecutionConfiguration, 144  
 set\_object\_instance\_handles\_by\_name  
     TrickHLA::Manager, 511  
 set\_parameter\_count  
     TrickHLA::Interaction, 445  
 set\_parameter\_handle  
     TrickHLA::Parameter, 619  
 set\_parameters  
     TrickHLA::Interaction, 446  
 set\_parent\_name  
     SpaceFOM::RefFrameBase, 656  
 set\_parent\_ref\_frame  
     SpaceFOM::PhysicalEntityBase, 642  
 set\_pending\_mtr  
     DIS::ExecutionControl, 177  
     DSES::ExecutionControl, 193  
     IMSim::ExecutionControl, 215  
     SpaceFOM::ExecutionControl, 239  
 set\_phase  
     TrickHLAModel::SineData, 688  
 set\_preferred\_order  
     TrickHLA::Attribute, 78  
 set\_publish  
     TrickHLA::Attribute, 78  
     TrickHLA::Interaction, 446  
 set\_pull\_requested  
     TrickHLA::Attribute, 78  
     TrickHLA::Object, 563  
 set\_push\_requested  
     TrickHLA::Attribute, 79  
 set\_requested\_execution\_control\_mode  
     TrickHLA::ExecutionControlBase, 287  
 set\_requested\_time  
     TrickHLA::Federate, 374  
 set\_restart  
     TrickHLA::Federate, 374  
 set\_restart\_cfg  
     TrickHLA::Federate, 374  
 set\_restore\_begun  
     TrickHLA::Federate, 375  
 set\_restore\_completed  
     TrickHLA::Federate, 375  
 set\_restore\_failed  
     TrickHLA::Federate, 375  
 set\_restore\_is\_imminent  
     TrickHLA::Federate, 375  
 set\_restore\_name  
     TrickHLA::Federate, 375  
 set\_restore\_request\_failed  
     TrickHLA::Federate, 376  
 set\_restore\_request\_succeeded  
     TrickHLA::Federate, 376  
 set\_root\_frame\_name  
     DIS::ExecutionConfiguration, 110  
     DSES::ExecutionConfiguration, 120  
     IMSim::ExecutionConfiguration, 131  
     SpaceFOM::ExecutionConfiguration, 144  
 set\_S\_define\_name  
     TrickHLA::ExecutionConfigurationBase, 159  
 set\_save\_completed  
     TrickHLA::Federate, 376  
 set\_save\_name  
     TrickHLA::Federate, 376  
 set\_scenario\_freeze\_time  
     TrickHLA::ExecutionControlBase, 288  
 set\_scenario\_time\_epoch  
     DIS::ExecutionConfiguration, 110  
     DSES::ExecutionConfiguration, 120  
     IMSim::ExecutionConfiguration, 131  
     SpaceFOM::ExecutionConfiguration, 144  
 set\_scenario\_timeline  
     TrickHLA::ExecutionControlBase, 288  
 set\_sim\_offset  
     TrickHLA::ScenarioTimeline, 668  
 set\_simulation\_freeze\_time  
     TrickHLA::ExecutionControlBase, 288

set\_start\_to\_restore  
    TrickHLA::Federate, 376

set\_start\_to\_save  
    TrickHLA::Federate, 377

set\_startup  
    TrickHLA::Federate, 377

set\_state  
    TrickHLA::SyncPnt, 736

set\_status  
    SpaceFOM::PhysicalEntityBase, 642

set\_subscribe  
    TrickHLA::Attribute, 79  
    TrickHLA::Interaction, 446

set\_time  
    TrickHLA::TimedSyncPnt, 763  
    TrickHLAModel::SineData, 688

set\_time\_advance\_grant  
    TrickHLA::Federate, 377

set\_time\_constrained\_state  
    TrickHLA::Federate, 378

set\_time\_padding  
    DIS::ExecutionControl, 177  
    DSES::ExecutionControl, 193  
    IMSim::ExecutionControl, 215  
    SpaceFOM::ExecutionControl, 239  
    TrickHLA::ExecutionControl, 252  
    TrickHLA::ExecutionControlBase, 288

set\_time\_regulation\_state  
    TrickHLA::Federate, 378

set\_to\_unblocking\_cyclic\_reads  
    TrickHLA::Object, 564

set\_tolerance  
    TrickHLAModel::SineData, 689

set\_type  
    SpaceFOM::PhysicalEntityBase, 643

set\_update\_requested  
    TrickHLA::Attribute, 79

set\_user\_supplied\_tag  
    TrickHLA::Interaction, 446

set\_value  
    TrickHLAModel::SineData, 689

set\_wait\_sleep  
    TrickHLA::ExecutionControlBase, 289  
    TrickHLA::SyncPnt, 737

set\_wait\_timeout  
    TrickHLA::ExecutionControlBase, 289  
    TrickHLA::SyncPnt, 737

setTo  
    TrickHLA::Int64Interval, 417  
    TrickHLA::Int64Time, 432

setup  
    TrickHLA::ExecutionConfigurationBase, 159  
    TrickHLA::ExecutionControlBase, 289, 290  
    TrickHLA::FedAmb, 318

    TrickHLA::Federate, 378  
    TrickHLA::Manager, 512

setup\_all\_ref\_attributes  
    TrickHLA::Manager, 512

setup\_all\_RTI\_handles  
    TrickHLA::Manager, 513

setup\_checkpoint  
    TrickHLA::ExecutionControlBase, 290  
    TrickHLA::Federate, 379  
    TrickHLA::Manager, 513

setup\_checkpoint\_interactions  
    TrickHLA::Manager, 513

setup\_checkpoint\_requests  
    TrickHLA::OwnershipHandler, 594

setup\_interaction\_ref\_attributes  
    DIS::ExecutionControl, 177  
    DSES::ExecutionControl, 194  
    IMSim::ExecutionControl, 216  
    SpaceFOM::ExecutionControl, 239  
    TrickHLA::ExecutionControl, 253  
    TrickHLA::ExecutionControlBase, 290  
    TrickHLA::Manager, 513

setup\_interaction\_RTI\_handles  
    DIS::ExecutionControl, 178  
    DSES::ExecutionControl, 194  
    IMSim::ExecutionControl, 216  
    SpaceFOM::ExecutionControl, 240  
    TrickHLA::ExecutionControl, 253  
    TrickHLA::ExecutionControlBase, 290  
    TrickHLA::Manager, 514

setup\_object\_ref\_attributes  
    DIS::ExecutionControl, 178  
    DSES::ExecutionControl, 194  
    IMSim::ExecutionControl, 216  
    SpaceFOM::ExecutionControl, 240  
    TrickHLA::ExecutionControl, 253  
    TrickHLA::ExecutionControlBase, 291  
    TrickHLA::Manager, 514

setup\_object\_RTI\_handles  
    DIS::ExecutionControl, 178  
    DSES::ExecutionControl, 194  
    IMSim::ExecutionControl, 216  
    SpaceFOM::ExecutionControl, 240  
    TrickHLA::ExecutionControl, 253  
    TrickHLA::ExecutionControlBase, 291  
    TrickHLA::Manager, 515

setup\_ownership\_transfer\_checkpointed\_data  
    TrickHLA::Object, 564

setup\_preferred\_order\_with\_RTI  
    TrickHLA::Interaction, 447  
    TrickHLA::Manager, 515  
    TrickHLA::Object, 564

setup\_ref\_attributes  
    DIS::ExecutionConfiguration, 110

DSES::ExecutionConfiguration, 121  
 IMSim::ExecutionConfiguration, 133  
 SpaceFOM::ExecutionConfiguration, 144  
 TrickHLA::ExecutionConfiguration, 151  
 TrickHLA::ExecutionConfigurationBase, 159  
 setup\_restore  
     TrickHLA::Federate, 379  
 setup\_time\_constrained  
     TrickHLA::Federate, 379  
 setup\_time\_management  
     TrickHLA::Federate, 380  
 setup\_time\_regulation  
     TrickHLA::Federate, 380  
 should\_exit  
     DIS::PausePointList, 637  
     IMSim::PausePointList, 631  
 should\_freeze  
     DIS::PausePointList, 637  
     IMSim::PausePointList, 631  
 should\_print  
     TrickHLA::DebugHandler, 96  
     TrickHLA::ExecutionControlBase, 291  
     TrickHLA::FedAmb, 318  
     TrickHLA::Federate, 380  
     TrickHLA::Interaction, 447  
     TrickHLA::InteractionHandler, 457  
     TrickHLA::LagCompensation, 482  
     TrickHLA::Manager, 515  
     TrickHLA::Object, 564  
     TrickHLA::OwnershipHandler, 594  
     TrickHLA::Packing, 604  
 should\_publish\_data  
     TrickHLA::Federate, 381  
 should\_reconfig  
     DIS::PausePointList, 637  
     IMSim::PausePointList, 631  
 should\_restart  
     DIS::PausePointList, 637  
     IMSim::PausePointList, 631  
 should\_run  
     DIS::PausePointList, 637  
     IMSim::PausePointList, 631  
 should\_send  
     TrickHLA::Conditional, 89  
     TrickHLAModel::SineConditional, 679  
 shutdown  
     DIS::ExecutionControl, 178  
     DSES::ExecutionControl, 194  
     IMSim::ExecutionControl, 217  
     SpaceFOM::ExecutionControl, 240  
     TrickHLA::ExecutionControl, 253  
     TrickHLA::ExecutionControlBase, 292  
     TrickHLA::Federate, 381  
     TrickHLA::Manager, 516  
 shutdown\_called  
     TrickHLA::Federate, 403  
     TrickHLA::Manager, 523  
 shutdown\_mode\_announce  
     DIS::ExecutionControl, 178  
     DSES::ExecutionControl, 194  
     IMSim::ExecutionControl, 217  
     SpaceFOM::ExecutionControl, 240  
     TrickHLA::ExecutionControl, 253  
     TrickHLA::ExecutionControlBase, 292  
 shutdown\_mode\_transition  
     DIS::ExecutionControl, 178  
     DSES::ExecutionControl, 195  
     IMSim::ExecutionControl, 217  
     SpaceFOM::ExecutionControl, 241  
     TrickHLA::ExecutionControl, 254  
     TrickHLA::ExecutionControlBase, 292  
 shutdown\_time\_constrained  
     TrickHLA::Federate, 382  
 shutdown\_time\_management  
     TrickHLA::Federate, 382  
 shutdown\_time\_regulating  
     TrickHLA::Federate, 382  
 SIM\_CONFIG\_SYNC\_POINT  
     DSES, 29  
     IMSim, 37  
 sim\_data  
     TrickHLAModel::SineConditional, 680  
     TrickHLAModel::SineLagCompensation, 700  
     TrickHLAModel::SinePacking, 712  
 sim\_offset  
     TrickHLA::ScenarioTimeline, 669  
 sim\_time  
     SpaceFOM::MTRInteractionHandler, 529  
 sim\_timeline  
     TrickHLA::ExecutionControlBase, 300  
     TrickHLA::ScenarioTimeline, 669  
 SimpleSimConfig  
     TrickHLAModel::SimpleSimConfig, 671  
 SimpleSimConfig.cpp, 917  
 SimpleSimConfig.hh, 918  
 SimTimeline  
     TrickHLA::SimTimeline, 675  
 SimTimeline.cpp, 920  
 SimTimeline.hh, 921  
 simulation\_freeze\_time  
     TrickHLA::ExecutionControlBase, 300  
 SINE\_MSG\_SIZE  
     SineInteractionHandler.hh, 930  
 SINE\_SEND\_INTERACTION\_TSO  
     SineInteractionHandler.cpp, 928  
 SineConditional  
     TrickHLAModel::SineConditional, 678  
 SineConditional.cpp, 922

SineConditional.hh, 923  
SineData  
    TrickHLAModel::SineData, 682  
SineData.cpp, 925  
SineData.hh, 926  
SineInteractionHandler  
    TrickHLAModel::SineInteractionHandler, 692, 693  
SineInteractionHandler.cpp, 927  
    SINE\_SEND\_INTERACTION\_TSO, 928  
SineInteractionHandler.hh, 928  
    SINE\_MSG\_SIZE, 930  
SineLagCompensation  
    TrickHLAModel::SineLagCompensation, 697  
SineLagCompensation.cpp, 930  
SineLagCompensation.hh, 931  
SineObjectDeleted  
    TrickHLAModel::SineObjectDeleted, 702, 703  
SineObjectDeleted.cpp, 933  
SineObjectDeleted.hh, 934  
SineOwnershipHandler  
    TrickHLAModel::SineOwnershipHandler, 705  
SineOwnershipHandler.cpp, 935  
SineOwnershipHandler.hh, 936  
SinePacking  
    TrickHLAModel::SinePacking, 708, 709  
SinePacking.cpp, 938  
SinePacking.hh, 939  
size  
    TrickHLA::Attribute, 84  
    TrickHLA::ItemQueue, 473  
    TrickHLA::Parameter, 622  
    TrickHLA::ParameterItem, 626  
size\_is\_static  
    TrickHLA::Attribute, 85  
    TrickHLA::Parameter, 622  
SpaceFOM, 37  
    EXECUTION\_MODE\_LAST\_VALUE, 39  
    execution\_mode\_enum\_to\_int16, 39  
    execution\_mode\_enum\_to\_string, 39  
    EXECUTION\_MODE\_FIRST\_VALUE, 38  
    EXECUTION\_MODE\_FREEZE, 39  
    EXECUTION\_MODE\_INITIALIZING, 38  
    execution\_mode\_int16\_to\_enum, 40  
    EXECUTION\_MODE\_RUNNING, 39  
    EXECUTION\_MODE\_SHUTDOWN, 39  
    EXECUTION\_MODE\_UNINITIALIZED, 38  
    ExecutionModeEnum, 38  
    from\_execution\_control\_enum, 40  
    from\_mode\_transition\_enum, 41  
    INIT\_COMPLETED\_SYNC\_POINT, 43  
    INIT\_STARTED\_SYNC\_POINT, 43  
    mtr\_enum\_to\_int16, 41  
    mtr\_enum\_to\_string, 41  
    MTR\_FIRST\_VALUE, 39  
    MTR\_GOTO\_FREEZE, 39  
    MTR\_GOTO\_RUN, 39  
    MTR\_GOTO\_SHUTDOWN, 39  
    MTR\_INITIALIZING, 39  
    mtr\_int16\_to\_enum, 42  
    MTR\_LAST\_VALUE, 39  
    MTR\_UNINITIALIZED, 39  
    MTERenum, 39  
    OBJECTS\_DISCOVERED\_SYNC\_POINT, 43  
    ROOT\_FRAME\_DISCOVERED\_SYNC\_POINT, 43  
    to\_execution\_control\_enum, 42  
SpaceFOM/ExecutionConfiguration.cpp  
    attrSpaceFOM\_\_ExecutionConfiguration, 806  
SpaceFOM/ExecutionControl.cpp  
    attrSpaceFOM\_\_MTRInteractionHandler, 824  
    the\_clock, 824  
SpaceFOM::DynamicalEntity, 99  
    ~DynamicalEntity, 100  
    DynamicalEntity, 100  
    force, 101  
    inertia, 102  
    inertia\_rate, 102  
    init\_attrSpaceFOM\_\_DynamicalEntity, 101  
    InputProcessor, 101  
    mass, 102  
    mass\_rate, 102  
    operator=, 101  
    pack, 101  
    torque, 102  
    unpack, 101  
SpaceFOM::ExecutionConfiguration, 135  
    ~ExecutionConfiguration, 139  
    configure, 139  
    configure\_attributes, 139  
    current\_execution\_mode, 146  
    ExecutionConfiguration, 138, 139  
    get\_current\_execution\_mode, 139  
    get\_least\_common\_time\_step, 140  
    get\_next\_execution\_mode, 140  
    get\_next\_mode\_cte\_time, 140  
    get\_next\_mode\_scenario\_time, 140  
    get\_root\_frame\_name, 141  
    get\_scenario\_time\_epoch, 141  
    init\_attrSpaceFOM\_\_ExecutionConfiguration, 146  
    InputProcessor, 146  
    least\_common\_time\_step, 146  
    next\_execution\_mode, 146  
    next\_mode\_cte\_time, 147  
    next\_mode\_scenario\_time, 147  
    operator=, 141  
    pack, 141  
    print\_execution\_configuration, 141  
    root\_frame\_name, 147  
    scenario\_time\_epoch, 147

set\_current\_execution\_mode, 142  
 set\_least\_common\_time\_step, 142  
 set\_next\_execution\_mode, 143  
 set\_next\_mode\_cte\_time, 143  
 set\_next\_mode\_scenario\_time, 144  
 set\_root\_frame\_name, 144  
 set\_scenario\_time\_epoch, 144  
 setup\_ref\_attributes, 144  
 unpack, 145  
 wait\_on\_update, 145  
**SpaceFOM::ExecutionControl**, 221  
 ~ExecutionControl, 225  
 add\_initialization\_sync\_points, 225  
 announce\_sync\_point, 226  
 check\_for\_shutdown, 226  
 check\_for\_shutdown\_with\_termination, 226  
 check\_freeze\_exit, 227  
 check\_mode\_transition\_request, 227  
 earlyJoinerHlaInitProcess, 227  
 enter\_freeze, 228  
 epoch\_and\_root\_frame\_discovery\_process, 228  
**ExecutionControl**, 225  
 exit\_freeze, 228  
 freeze\_init, 228  
 freeze\_mode\_announce, 229  
 freeze\_mode\_transition, 229  
 get\_execution\_configuration, 229  
 get\_type, 230  
 init\_attrSpaceFOM\_\_ExecutionControl, 242  
 initialize, 230  
**InputProcessor**, 242  
 is\_mtr\_valid, 230  
 is\_pacing, 231  
 is\_root\_frame\_publisher, 231  
 join\_federation\_process, 231  
 lateJoinerHlaInitProcess, 231  
 mtr\_interaction, 242  
 mtr\_interaction\_handler, 243  
 operator=, 232  
 pacing, 243  
 pending\_mtr, 243  
 post\_multi\_phase\_init\_processes, 232  
 pre\_multi\_phase\_init\_processes, 232  
 process\_execution\_control\_updates, 233  
 process\_mode\_interaction, 234  
 process\_mode\_transition\_request, 234  
 publish, 234  
 receive\_init\_root\_ref\_frame, 234  
 receive\_interaction, 235  
 receive\_root\_ref\_frame, 235  
 role\_determination\_process, 236  
 root\_frame\_pub, 243  
 root\_ref\_frame, 243  
 run\_mode\_transition, 236  
 send\_init\_root\_ref\_frame, 236  
 send\_mode\_transition\_interaction, 237  
 send\_MTR\_interaction, 237  
 send\_root\_ref\_frame, 237  
 set\_least\_common\_time\_step, 237  
 set\_mode\_request\_from\_mtr, 238  
 set\_next\_execution\_control\_mode, 238  
 set\_pending\_mtr, 239  
 set\_time\_padding, 239  
 setup\_interaction\_ref\_attributes, 239  
 setup\_interaction\_RTI\_handles, 240  
 setup\_object\_ref\_attributes, 240  
 setup\_object\_RTI\_handles, 240  
 shutdown, 240  
 shutdown\_mode\_announce, 240  
 shutdown\_mode\_transition, 241  
 start\_federation\_save\_at\_scenario\_time, 241  
 subscribe, 241  
 type, 243  
 unpublish, 241  
 unsubscribe, 242  
 wait\_on\_root\_frame\_discovered\_synchronization, 242  
**SpaceFOM::MTRInteractionHandler**, 523  
 ~MTRInteractionHandler, 526  
 cte\_time, 528  
 get\_address\_of\_interaction\_mode, 526  
 granted\_time, 528  
 init\_attrSpaceFOM\_\_MTRInteractionHandler, 528  
**InputProcessor**, 528  
 mtr\_mode, 528  
 mtr\_mode\_int, 529  
**MTRInteractionHandler**, 526  
 name, 529  
 operator=, 526  
 receive\_cnt, 529  
 receive\_interaction, 527  
 scenario\_time, 529  
 send\_cnt, 529  
 send\_interaction, 527  
 set\_name, 527  
 sim\_time, 529  
**SpaceFOM::PhysicalEntityBase**, 638  
 ~PhysicalEntityBase, 641  
 accel, 643  
 body\_wrt\_struct, 643  
 cm, 644  
 get\_name, 641  
 get\_parent\_ref\_frame, 641  
 get\_status, 641  
 get\_time, 641  
 get\_type, 642  
 init\_attrSpaceFOM\_\_PhysicalEntityBase, 643  
 initialize, 642

InputProcessor, 643  
name, 644  
operator=, 642  
pack, 642  
parent\_ref\_frame, 644  
PhysicalEntityBase, 641  
quat\_encoder, 644  
rot\_accel, 644  
set\_name, 642  
set\_parent\_ref\_frame, 642  
set\_status, 642  
set\_type, 643  
state, 644  
status, 645  
stc\_encoder, 645  
type, 645  
unpack, 643  
SpaceFOM::QuaternionEncoder, 646  
data, 649  
decode, 648  
encode, 648  
encoder, 649  
get\_data, 648  
init\_attrSpaceFOM\_\_QuaternionEncoder, 649  
InputProcessor, 649  
operator=, 649  
QuaternionEncoder, 648  
scalar\_encoder, 649  
vector, 650  
vector\_encoder, 650  
SpaceFOM::RefFrameBase, 650  
~RefFrameBase, 653  
allocate\_input\_string, 653  
debug, 657  
default\_data, 653  
get\_name, 654  
get\_object, 654  
get\_parent\_name, 654  
init\_attrSpaceFOM\_\_RefFrameBase, 656  
initialize, 654  
initialize\_callback, 655  
initialized, 657  
InputProcessor, 656  
name, 657  
operator=, 655  
pack, 655  
parent\_name, 657  
ref\_frame\_attr, 657  
ref\_frame\_data, 657  
RefFrameBase, 652, 653  
set\_name, 655  
set\_parent\_name, 656  
stc\_data, 658  
stc\_encoder, 658  
unpack, 656  
SpaceFOM::SpaceTimeCoordinateEncoder, 715  
ang\_vel\_encoder, 718  
angular\_velocity, 718  
data, 719  
decode, 717  
encode, 717  
encoder, 719  
get\_data, 718  
init\_attrSpaceFOM\_\_SpaceTimeCoordinateEncoder, 718  
InputProcessor, 718  
operator=, 718  
position, 719  
position\_encoder, 719  
quat\_encoder, 719  
quat\_scalar\_encoder, 719  
quat\_vector, 720  
quat\_vector\_encoder, 720  
rot\_state\_encoder, 720  
SpaceTimeCoordinateEncoder, 717  
time\_encoder, 720  
trans\_state\_encoder, 720  
velocity, 720  
velocity\_encoder, 720  
SpaceTimeCoordinateData, 713  
ang\_vel, 714  
pos, 714  
quat\_scalar, 714  
quat\_vector, 714  
time, 714  
vel, 714  
SpaceTimeCoordinateData.h, 940  
SpaceTimeCoordinateEncoder  
    SpaceFOM::SpaceTimeCoordinateEncoder, 717  
SpaceTimeCoordinateEncoder.cpp, 941  
SpaceTimeCoordinateEncoder.hh, 942  
stale\_data\_counter  
    TrickHLA::Federate, 403  
StandardsSupport.hh, 943  
    IEEE\_1516\_2010, 944  
    RTI1516\_EXCEPTION, 944  
    RTI1516\_HEADER, 944  
    RTI1516\_NAMESPACE, 944  
    RTI1516\_USERDATA, 944  
start\_federation\_save  
    TrickHLA::Manager, 516  
start\_federation\_save\_at\_scenario\_time  
    IMSim::ExecutionControl, 217  
    SpaceFOM::ExecutionControl, 241  
    TrickHLA::ExecutionControl, 254  
    TrickHLA::ExecutionControlBase, 292  
    TrickHLA::Manager, 516  
start\_federation\_save\_at\_sim\_time

TrickHLA::Manager, 517  
 start\_to\_restore  
     TrickHLA::Federate, 403  
 start\_to\_save  
     TrickHLA::Federate, 403  
 startRegistrationForObjectClass  
     TrickHLA::FedAmb, 319  
 STARTUP\_FREEZE\_SYNC\_POINT  
     DIS, 24  
     IMSim, 37  
 STARTUP\_SYNC\_POINT  
     DIS, 24  
     DSES, 29  
     IMSim, 37  
 state  
     DIS::PausePointList, 638  
     IMSim::PausePointList, 632  
     integrator.d, 850  
     RefFrameData, 659  
     SpaceFOM::PhysicalEntityBase, 644  
     TrickHLA::LoggableSyncPnt, 485  
     TrickHLA::SyncPnt, 739  
 state\_ws  
     integrator.d, 850  
 status  
     SpaceFOM::PhysicalEntityBase, 645  
 stc\_data  
     SpaceFOM::RefFrameBase, 658  
 stc\_encoder  
     SpaceFOM::PhysicalEntityBase, 645  
     SpaceFOM::RefFrameBase, 658  
 stop\_publishing\_attributes  
     TrickHLA::Object, 565  
 stop\_subscribing\_attributes  
     TrickHLA::Object, 565  
 stopRegistrationForObjectClass  
     TrickHLA::FedAmb, 319  
 str\_restore\_label  
     TrickHLA::Federate, 404  
 str\_save\_label  
     TrickHLA::Federate, 404  
 StringUtilities  
     TrickHLA::StringUtilities, 722  
 StringUtilities.hh, 945  
     WHITESPACE\_CHARS, 946  
 subscribe  
     DIS::ExecutionControl, 179  
     DSES::ExecutionControl, 195  
     IMSim::ExecutionControl, 218  
     SpaceFOM::ExecutionControl, 241  
     TrickHLA::Attribute, 85  
     TrickHLA::ExecutionControl, 254  
     TrickHLA::ExecutionControlBase, 292  
     TrickHLA::Interaction, 450  
 TrickHLA::Manager, 517  
 subscribe\_attributes  
     TrickHLA::Federate, 382  
 subscribe\_to\_interaction  
     TrickHLA::Interaction, 447  
 subscribe\_to\_object\_attributes  
     TrickHLA::Object, 565  
 SYNC\_PNT\_STATE\_ACHIEVED  
     TrickHLA, 52  
 SYNC\_PNT\_STATE\_ANNOUNCED  
     TrickHLA, 52  
 sync\_pnt\_state\_enum\_to\_int16  
     TrickHLA, 55  
 sync\_pnt\_state\_enum\_to\_string  
     TrickHLA, 55  
 SYNC\_PNT\_STATE\_ERROR  
     TrickHLA, 52  
 SYNC\_PNT\_STATE\_EXISTS  
     TrickHLA, 52  
 SYNC\_PNT\_STATE\_FIRST\_VALUE  
     TrickHLA, 52  
 sync\_pnt\_state\_int16\_to\_enum  
     TrickHLA, 55  
 SYNC\_PNT\_STATE\_LAST\_VALUE  
     TrickHLA, 52  
 SYNC\_PNT\_STATE\_REGISTERED  
     TrickHLA, 52  
 SYNC\_PNT\_STATE\_SYNCHRONIZED  
     TrickHLA, 52  
 SYNC\_PNT\_STATE\_UNKNOWN  
     TrickHLA, 52  
 sync\_point\_list  
     TrickHLA::SyncPntListBase, 759  
 sync\_point\_registration\_failed  
     DIS::ExecutionControl, 179  
     TrickHLA::Federate, 383  
     TrickHLA::SyncPntListBase, 755  
 sync\_point\_registration\_succeeded  
     DIS::ExecutionControl, 179  
     TrickHLA::Federate, 383  
     TrickHLA::SyncPntListBase, 756  
 synchronizationPointRegistrationFailed  
     TrickHLA::FedAmb, 319  
 synchronizationPointRegistrationSucceeded  
     TrickHLA::FedAmb, 319  
 SyncPnt  
     TrickHLA::SyncPnt, 732  
 SyncPnt.cpp, 946  
 SyncPnt.hh, 947  
 SyncPntList  
     TrickHLA::SyncPntList, 741  
 SyncPntList.hh, 948  
 SyncPntListBase  
     TrickHLA::SyncPntListBase, 746

SyncPntListBase.cpp, 950  
SyncPntListBase.hh, 951  
SyncPntStateEnum  
    TrickHLA, 51

tail  
    TrickHLA::ItemQueue, 474

the\_clock  
    SpaceFOM/ExecutionControl.cpp, 824

the\_cpr  
    Federate.cpp, 840

THLA\_10SEC\_TIMEOUT\_WHILE\_WAITING\_FOR\_DATA  
    CompileConfig.hh, 789

thla\_attribute\_map  
    TrickHLA::Object, 574

THLA\_ENDL  
    CompileConfig.hh, 789

THLA\_FREEZE\_INTERACTION\_DEBUG  
    FreezeInteractionHandler.cpp, 843

THLA\_INTRAFRAME\_BLOCKING\_READ\_CONFIG  
    CompileConfig.hh, 789

THLA\_NEWLINE  
    CompileConfig.hh, 789

THLA\_OBJ\_DEBUG  
    Object.cpp, 877

THLA\_OBJ\_DEBUG\_RECEIVE  
    Object.cpp, 877

THLA\_OBJ\_DEBUG\_SEND  
    Object.cpp, 877

THLA\_OBJ\_DEBUG\_VALID\_OBJECT\_RECEIVE  
    Object.cpp, 877

THLA\_OBJ\_OWNERSHIP\_DEBUG  
    Object.cpp, 877

THLA\_QUEUE\_REFLECTED\_ATTRIBUTES  
    CompileConfig.hh, 789

thla\_reflected\_attributes\_queue  
    TrickHLA::Object, 574

THLA\_TAG\_USE\_USLEEP  
    Federate.cpp, 840

THLA\_THREAD\_TIMED\_WAIT\_FOR\_DATA  
    CompileConfig.hh, 790

THLA\_USE\_ATTRIBUTE\_MAP\_CLASS\_INSTANCE  
    Object.hh, 880

THLAAttributeMap  
    TrickHLA, 47

THLASaveRestoreProcEnum  
    TrickHLA, 52

time  
    IMSim::FreezeInteractionHandler, 410  
    SpaceTimeCoordinateData, 714  
    TrickHLA::Interaction, 450  
    TrickHLA::InteractionItem, 465  
    TrickHLA::LoggableTimedSyncPnt, 488  
    TrickHLA::OwnershipItem, 599

    TrickHLA::TimedSyncPnt, 763  
    TrickHLAModel::SineData, 690  
    TrickHLAModel::SineInteractionHandler, 695

time\_adv\_grant  
    TrickHLA::Federate, 404

time\_advance\_request  
    TrickHLA::Federate, 383

time\_advance\_request\_to\_GALT  
    TrickHLA::Federate, 384

time\_advance\_request\_to\_GALT\_LCTS\_multiple  
    TrickHLA::Federate, 384

time\_attr  
    TrickHLAModel::SineLagCompensation, 701  
    TrickHLAModel::SinePacking, 712

time\_constrained  
    TrickHLA::Federate, 404

time\_constrained\_state  
    TrickHLA::Federate, 404

time\_encoder  
    SpaceFOM::SpaceTimeCoordinateEncoder, 720

time\_from\_HLT  
    TrickHLA::ScenarioTimeline, 668

time\_from\_simulation\_time  
    TrickHLA::ScenarioTimeline, 668

time\_management  
    TrickHLA::Federate, 404

time\_padding  
    TrickHLA::ExecutionControlBase, 301

time\_plus\_looking\_ahead  
    TrickHLA::Object, 574

time\_regulating  
    TrickHLA::Federate, 405

time\_regulating\_state  
    TrickHLA::Federate, 405

timeAdvanceGrant  
    TrickHLA::FedAmb, 319

timeConstrainedEnabled  
    TrickHLA::FedAmb, 320

TimedSyncPnt  
    TrickHLA::TimedSyncPnt, 761

TimedSyncPnt.cpp, 952  
TimedSyncPnt.hh, 953  
TimedSyncPntList  
    TrickHLA::TimedSyncPntList, 766

TimedSyncPntList.cpp, 955  
TimedSyncPntList.hh, 956

Timeline  
    TrickHLA::Timeline, 771

Timeline.cpp, 958  
Timeline.hh, 959

TimeOfDayTimeline  
    TrickHLA::TimeOfDayTimeline, 774, 775

TimeOfDayTimeline.cpp, 960  
TimeOfDayTimeline.hh, 961

timeRegulationEnabled  
     TrickHLA::FedAmb, 320

to\_execution\_control\_enum  
     DIS, 23  
     DSES, 28  
     IMSim, 36  
     SpaceFOM, 42

to\_printable\_string  
     TrickHLA::StringUtilities, 723

to\_string  
     DIS::PausePointList, 637  
     IMSim::PausePointList, 632  
     TrickHLA::StringUtilities, 723–725  
     TrickHLA::SyncPnt, 737  
     TrickHLA::SyncPntListBase, 756  
     TrickHLA::TimedSyncPnt, 763

to\_wstring  
     TrickHLA::StringUtilities, 726, 727

tokenize  
     TrickHLA::StringUtilities, 727, 728

tol  
     TrickHLAModel::SineData, 690

tol\_attr  
     TrickHLAModel::SineLagCompensation, 701  
     TrickHLAModel::SinePacking, 713

toMicroseconds  
     TrickHLA::Int64Interval, 418

torque  
     SpaceFOM::DynamicalEntity, 102

toString  
     TrickHLA::Int64Interval, 418  
     TrickHLA::Int64Time, 433

trans\_state\_encoder  
     SpaceFOM::SpaceTimeCoordinateEncoder, 720

TRANSPORT\_FIRST\_VALUE  
     TrickHLA, 52

TRANSPORT\_LAST\_VALUE  
     TrickHLA, 52

TRANSPORT\_RECEIVE\_ORDER  
     TrickHLA, 52

TRANSPORT\_SPECIFIED\_IN\_FOM  
     TrickHLA, 52

TRANSPORT\_TIMESTAMP\_ORDER  
     TrickHLA, 52

TransportationEnum  
     TrickHLA, 52

trick\_hla\_obj  
     TrickHLA::DivestThreadArgs, 98

trick\_MM  
     PhysicalEntityBase.cpp, 902  
     RefFrameBase.cpp, 909

trick\_name  
     TrickHLA::Attribute, 85  
     TrickHLA::Parameter, 622

TrickHLA, 43  
     AttributeMap, 47  
     AttributeOwnershipMap, 47  
     CONFIG\_CYCLIC, 48  
     CONFIG\_INITIALIZE, 48  
     CONFIG\_INTERMITTENT, 48  
     CONFIG\_MAX\_VALUE, 48  
     CONFIG\_NONE, 48  
     DataUpdateEnum, 48  
     DEBUG\_LEVEL\_0\_TRACE, 48  
     DEBUG\_LEVEL\_10\_TRACE, 49  
     DEBUG\_LEVEL\_11\_TRACE, 49  
     DEBUG\_LEVEL\_1\_TRACE, 48  
     DEBUG\_LEVEL\_2\_TRACE, 48  
     DEBUG\_LEVEL\_3\_TRACE, 48  
     DEBUG\_LEVEL\_4\_TRACE, 48  
     DEBUG\_LEVEL\_5\_TRACE, 48  
     DEBUG\_LEVEL\_6\_TRACE, 48  
     DEBUG\_LEVEL\_7\_TRACE, 48  
     DEBUG\_LEVEL\_8\_TRACE, 48  
     DEBUG\_LEVEL\_9\_TRACE, 48  
     DEBUG\_LEVEL\_FULL\_TRACE, 49  
     DEBUG\_LEVEL\_NO\_TRACE, 48  
     DEBUG\_SOURCE\_ALL\_MODULES, 49  
     DEBUG\_SOURCE\_ATTRIBUTE, 49  
     DEBUG\_SOURCE\_FED\_AMB, 49  
     DEBUG\_SOURCE\_FEDERATE, 49  
     DEBUG\_SOURCE\_INTERACTION, 49  
     DEBUG\_SOURCE\_LAG\_COMPENSATION, 49  
     DEBUG\_SOURCE\_MANAGER, 49  
     DEBUG\_SOURCE\_OBJECT, 49  
     DEBUG\_SOURCE\_OWNERSHIP, 49  
     DEBUG\_SOURCE\_PACKING, 49  
     DEBUG\_SOURCE\_PARAMETER, 49  
     DEBUG\_SOURCE\_SYNCPOINT, 49  
     DebugLevelEnum, 48  
     DebugSourceEnum, 49  
     def\_scenario\_timeline, 56  
     def\_sim\_timeline, 56  
     ENCODING\_ASCII\_STRING, 49  
     ENCODING\_BIG\_ENDIAN, 49  
     ENCODING\_BOOLEAN, 50  
     ENCODING\_C\_STRING, 49  
     ENCODING\_FIRST\_VALUE, 49  
     ENCODING\_LAST\_VALUE, 50  
     ENCODING\_LITTLE\_ENDIAN, 49  
     ENCODING\_LOGICAL\_TIME, 49  
     ENCODING\_NO\_ENCODING, 50  
     ENCODING\_OPAQUE\_DATA, 49  
     ENCODING\_UNICODE\_STRING, 49  
     ENCODING\_UNKNOWN, 49  
     EncodingEnum, 49  
     execution\_control\_enum\_to\_int16, 53  
     execution\_control\_enum\_to\_string, 53

EXECUTION\_CONTROL\_FIRST\_VALUE, 50  
EXECUTION\_CONTROL\_FREEZE, 50  
EXECUTION\_CONTROL\_INITIALIZING, 50  
execution\_control\_int16\_to\_enum, 53  
EXECUTION\_CONTROL\_LAST\_VALUE, 50  
EXECUTION\_CONTROL\_RECONFIG, 50  
EXECUTION\_CONTROL\_RESTART, 50  
EXECUTION\_CONTROL\_RUNNING, 50  
EXECUTION\_CONTROL\_SHUTDOWN, 50  
EXECUTION\_CONTROL\_UNINITIALIZED, 50  
ExecutionControlEnum, 50  
FEDERATE\_JOIN\_EARLY, 50  
FEDERATE\_JOIN\_FIRST\_VALUE, 50  
FEDERATE\_JOIN\_LAST\_VALUE, 50  
FEDERATE\_JOIN\_LATE, 50  
FEDERATE\_JOIN NOMINAL, 50  
FEDERATE\_JOIN\_RESTORING, 50  
FEDERATE\_JOIN\_UNKNOWN, 50  
FederateJoinEnum, 50  
HLAAttributeMapQueue, 47  
Initiate\_Restore, 52  
LAG\_COMPENSATION\_FIRST\_VALUE, 51  
LAG\_COMPENSATION\_LAST\_VALUE, 51  
LAG\_COMPENSATION\_NONE, 51  
LAG\_COMPENSATION\_RECEIVE\_SIDE, 51  
LAG\_COMPENSATION\_SEND\_SIDE, 51  
LagCompensationEnum, 50  
ManagerTypeOfInteractionEnum, 51  
MAX\_LOGICAL\_TIME\_SECONDS, 56  
MAX\_VALUE\_IN\_MICROS, 56  
MICROS\_MULTIPLIER, 57  
mode\_transition\_enum\_to\_int16, 54  
mode\_transition\_enum\_to\_string, 54  
MODE\_TRANSITION\_FIRST\_VALUE, 51  
MODE\_TRANSITION\_GOTO\_FREEZE, 51  
MODE\_TRANSITION\_GOTO\_RECONFIG, 51  
MODE\_TRANSITION\_GOTO\_RESTART, 51  
MODE\_TRANSITION\_GOTO\_RUN, 51  
MODE\_TRANSITION\_GOTO\_SHUTDOWN, 51  
MODE\_TRANSITION\_INITIALIZING, 51  
mode\_transition\_int16\_to\_enum, 54  
MODE\_TRANSITION\_LAST\_VALUE, 51  
MODE\_TRANSITION\_UNINITIALIZED, 51  
ModeTransitionEnum, 51  
No\_Restore, 52  
ObjectInstanceMap, 47  
Restore\_Complete, 52  
Restore\_Failed, 52  
Restore\_In\_Progress, 52  
Restore\_Request\_Failed, 52  
Restore\_Request\_Succeeded, 52  
SYNC\_PNT\_STATE\_ACHIEVED, 52  
SYNC\_PNT\_STATE\_ANNOUNCED, 52  
sync\_pnt\_state\_enum\_to\_int16, 55  
sync\_pnt\_state\_enum\_to\_string, 55  
SYNC\_PNT\_STATE\_ERROR, 52  
SYNC\_PNT\_STATE\_EXISTS, 52  
SYNC\_PNT\_STATE\_FIRST\_VALUE, 52  
sync\_pnt\_state\_int16\_to\_enum, 55  
SYNC\_PNT\_STATE\_LAST\_VALUE, 52  
SYNC\_PNT\_STATE\_REGISTERED, 52  
SYNC\_PNT\_STATE\_SYNCHRONIZED, 52  
SYNC\_PNT\_STATE\_UNKNOWN, 52  
SyncPntStateEnum, 51  
THLAAttributeMap, 47  
THLASaveRestoreProcEnum, 52  
TRANSPORT\_FIRST\_VALUE, 52  
TRANSPORT\_LAST\_VALUE, 52  
TRANSPORT\_RECEIVE\_ORDER, 52  
TRANSPORT\_SPECIFIED\_IN\_FOM, 52  
TRANSPORT\_TIMESTAMP\_ORDER, 52  
TransportationEnum, 52  
TRICKHLA\_MANAGER\_BUILTIN\_FREEZE\_INTERACTION, 51  
TRICKHLA\_MANAGER\_BUILTIN\_MTR\_INTERACTION, 51  
TRICKHLA\_MANAGER\_USER\_DEFINED\_INTERACTION, 51  
TrickHLAObjInstancenameMap, 47  
TrickRTIAmbPtr, 47  
VectorOfStrings, 47  
VectorOfWstrings, 48  
TrickHLA/ExecutionConfiguration.cpp  
attrTrickHLA\_\_ExecutionConfiguration, 807  
TrickHLA/ExecutionControl.cpp  
attrTrickHLA\_\_MTRInteractionHandler, 825  
TrickHLA::Attribute, 59  
~Attribute, 64  
attr\_handle, 80  
Attribute, 64, 65  
buffer, 81  
buffer\_capacity, 81  
byteswap, 81  
byteswap\_buffer\_copy, 65  
calculate\_number\_of\_items, 65  
calculate\_size\_and\_number\_of\_items, 66  
calculate\_static\_number\_of\_items, 66  
check\_data\_cycle\_ready, 66  
conditional, 81  
config, 81  
cycle\_cnt, 81  
cycle\_ratio, 82  
cycle\_time, 82  
debug\_handler, 82  
decode\_boolean\_from\_buffer, 66  
decode\_logical\_time, 66  
decode\_opaque\_data\_from\_buffer, 67  
decode\_string\_from\_buffer, 67

determine\_cycle\_ratio, 67  
 divest\_requested, 82  
 encode\_boolean\_to\_buffer, 67  
 encode\_logical\_time, 68  
 encode\_opaque\_data\_to\_buffer, 68  
 encode\_string\_to\_buffer, 68  
 ensure\_buffer\_capacity, 68  
 extract\_data, 68  
 FOM\_name, 82  
 get\_attribute\_handle, 69  
 get\_attribute\_size, 69  
 get\_attribute\_value, 69  
 get\_conditional, 69  
 get\_configuration, 70  
 get\_FOM\_name, 70  
 get\_preferred\_order, 70  
 get\_ref2\_attributes, 70  
 get\_rti\_encoding, 71  
 get\_sim\_variable\_address, 71  
 get\_trick\_name, 71  
 has\_conditional, 71  
 HLAttrue, 83  
 init\_attrTrickHLA\_Attribute, 80  
 initialize, 71  
 initialized, 83  
 InputProcessor, 80  
 is\_byteswap, 72  
 is\_changed, 72  
 is\_data\_cycle\_ready, 72  
 is\_divest\_requested, 73  
 is\_locally\_owned, 73  
 is\_publish, 73  
 is\_pull\_requested, 73  
 is\_push\_requested, 74  
 is\_received, 74  
 is\_remotely\_owned, 74  
 is\_static\_in\_size, 74  
 is\_subscribe, 75  
 is\_supported\_attribute\_type, 75  
 is\_update\_requested, 75  
 locally\_owned, 83  
 mark\_changed, 75  
 mark\_locally\_owned, 76  
 mark\_remotely\_owned, 76  
 mark\_unchanged, 76  
 num\_items, 83  
 operator=, 76  
 pack\_attribute\_buffer, 76  
 preferred\_order, 83  
 print\_buffer, 77  
 publish, 83  
 pull\_requested, 84  
 push\_requested, 84  
 ref2, 84  
 rti\_encoding, 84  
 set\_attribute\_handle, 77  
 set\_configuration, 77  
 set\_debug\_level, 77  
 set\_divest\_requested, 77  
 set\_encoding, 78  
 set\_preferred\_order, 78  
 set\_publish, 78  
 set\_pull\_requested, 78  
 set\_push\_requested, 79  
 set\_subscribe, 79  
 set\_update\_requested, 79  
 size, 84  
 size\_is\_static, 85  
 subscribe, 85  
 trick\_name, 85  
 unmark\_locally\_owned, 80  
 unpack\_attribute\_buffer, 80  
 update\_requested, 85  
 value\_changed, 85  
 TrickHLA::BasicClock, 86  
 ~BasicClock, 86  
 BasicClock, 86  
 first\_pass, 87  
 get\_time, 87  
 init\_attrTrickHLA\_BasicClock, 87  
 InputProcessor, 87  
 TrickHLA::Conditional, 87  
 ~Conditional, 88  
 Conditional, 88  
 init\_attrTrickHLA\_Conditional, 89  
 InputProcessor, 89  
 should\_send, 89  
 TrickHLA::CTETimelineBase, 89  
 ~CTETimelineBase, 91  
 clk\_id, 93  
 clock\_init, 91  
 clock\_stop, 92  
 CTETimelineBase, 91  
 get\_clock\_ID, 92  
 get\_time, 92  
 init\_attrTrickHLA\_CTETimelineBase, 93  
 InputProcessor, 93  
 operator=, 92  
 set\_clock\_ID, 93  
 wall\_clock\_time, 93  
 TrickHLA::DebugHandler, 94  
 ~DebugHandler, 94  
 code\_section, 97  
 debug\_level, 97  
 DebugHandler, 94  
 get\_code\_section, 95  
 get\_debug\_level, 95  
 get\_debug\_level\_as\_int, 95

init\_attrTrickHLA\_DebugHandler, 97  
InputProcessor, 97  
set, 95  
should\_print, 96  
TrickHLA::DivestThreadArgs, 98  
handle\_set, 98  
trick\_hla\_obj, 98  
TrickHLA::ExecutionConfiguration, 148  
~ExecutionConfiguration, 150  
configure, 150  
configure\_attributes, 151  
ExecutionConfiguration, 150  
init\_attrTrickHLA\_ExecutionConfiguration, 152  
InputProcessor, 152  
num\_federates, 152  
operator=, 151  
owner, 152  
pack, 151  
print\_execution\_configuration, 151  
required\_federates, 152  
run\_duration, 153  
run\_duration\_microsec, 153  
setup\_ref\_attributes, 151  
unpack, 152  
TrickHLA::ExecutionConfigurationBase, 153  
~ExecutionConfigurationBase, 156  
clear\_update\_pending, 156  
configure, 156  
configure\_attributes, 157  
execution\_control, 162  
ExecutionConfigurationBase, 155, 156  
get\_execution\_control, 157  
get\_S\_define\_name, 157  
init\_attrTrickHLA\_ExecutionConfigurationBase, 162  
InputProcessor, 162  
operator=, 157  
pack, 157  
pending\_update, 163  
print\_execution\_configuration, 157  
reset\_ownership\_states, 158  
reset\_preferred\_order, 158  
S\_define\_name, 163  
set\_execution\_control, 158  
set\_master, 158  
set\_S\_define\_name, 159  
setup, 159  
setup\_ref\_attributes, 159  
unpack, 161  
update\_pending, 161  
wait\_on\_registration, 161  
wait\_on\_update, 161  
TrickHLA::ExecutionControl, 244  
~ExecutionControl, 247  
add\_initialization\_sync\_points, 247  
add\_multiphase\_init\_sync\_points, 247  
announce\_sync\_point, 248  
clear\_multiphase\_init\_sync\_points, 248  
ExecutionControl, 246, 247  
freeze\_mode\_announce, 248  
freeze\_mode\_transition, 248  
get\_execution\_configuration, 249  
get\_type, 249  
init\_attrTrickHLA\_ExecutionControl, 255  
initialize, 249  
InputProcessor, 255  
join\_federation\_process, 249  
operator=, 249  
post\_multi\_phase\_init\_processes, 250  
pre\_multi\_phase\_init\_processes, 250  
process\_execution\_control\_updates, 250  
process\_mode\_interaction, 250  
publish, 251  
receive\_interaction, 251  
run\_mode\_transition, 251  
send\_mode\_transition\_interaction, 251  
set\_least\_common\_time\_step, 252  
set\_next\_execution\_control\_mode, 252  
set\_time\_padding, 252  
setup\_interaction\_ref\_attributes, 253  
setup\_interaction\_RTI\_handles, 253  
setup\_object\_ref\_attributes, 253  
setup\_object\_RTI\_handles, 253  
shutdown, 253  
shutdown\_mode\_announce, 253  
shutdown\_mode\_transition, 254  
start\_federation\_save\_at\_scenario\_time, 254  
subscribe, 254  
type, 255  
unpublish, 254  
unsubscribe, 254  
wait\_on\_init\_data, 255  
wait\_on\_init\_sync\_point, 255  
TrickHLA::ExecutionControlBase, 256  
~ExecutionControlBase, 264  
achieve\_all\_multiphase\_init\_sync\_pnts, 264  
add\_initialization\_sync\_points, 264  
add\_multiphase\_init\_sync\_points, 264  
add\_object\_to\_map, 265  
check\_for\_shutdown, 265  
check\_for\_shutdown\_with\_termination, 265  
check\_freeze\_exit, 265  
check\_pause, 266  
check\_pause\_at\_init, 266  
clear\_mode\_transition\_requested, 266  
clear\_mode\_values, 267  
clear\_multiphase\_init\_sync\_points, 267  
convert\_loggable\_sync\_pts, 267  
convert\_scenario\_time\_to\_sim\_time, 267

convert\_sim\_time\_to\_scenario\_time, 268  
 cte\_timeline, 294  
 current\_execution\_control\_mode, 295  
 does\_cte\_timeline\_exist, 268  
 does\_init\_complete\_sync\_point\_exist, 268  
 does\_scenario\_timeline\_exist, 269  
 does\_sim\_timeline\_exist, 269  
 enter\_freeze, 269  
 execution\_configuration, 295  
 ExecutionControlBase, 263, 264  
 exit\_freeze, 269  
 federate, 296  
 freeze\_init, 269  
 freeze\_mode\_announce, 270  
 freeze\_mode\_transition, 270  
 get\_cte\_time, 270  
 get\_current\_execution\_control\_mode, 270  
 get\_execution\_configuration, 271  
 get\_federate, 271  
 get\_least\_common\_time\_step, 271  
 get\_manager, 271  
 get\_requested\_execution\_control\_mode, 272  
 get\_scenario\_freeze\_time, 272  
 get\_scenario\_time, 272  
 get\_sim\_time, 273  
 get\_simulation\_freeze\_time, 273  
 get\_time\_padding, 273  
 get\_trickhla\_object, 273  
 get\_type, 274  
 get\_unregistered\_object, 274  
 get\_unregistered\_remote\_object, 274  
 init\_attrTrickHLA\_\_ExecutionControlBase, 294  
 init\_complete\_sp\_exists, 296  
 initialize, 275  
 InputProcessor, 294  
 is\_execution\_configuration\_used, 275  
 is\_in\_freeze, 275  
 is\_in\_reconfig, 275  
 is\_in\_restart, 276  
 is\_initializing, 276  
 is\_lateJoiner, 276  
 is\_lateJoiner\_determined, 276  
 is\_master, 277  
 is\_master\_preset, 278  
 is\_mode\_transition\_requested, 278  
 is\_running, 278  
 is\_save\_and\_restore\_supported, 279  
 is\_save\_initiated, 279  
 is\_shutdown, 279  
 join\_federation\_process, 279  
 lateJoiner, 297  
 lateJoiner\_determined, 297  
 least\_common\_time\_step, 297  
 loggable\_sync\_pts, 297  
 logged\_sync\_pts\_count, 298  
 Manager, 294  
 manager, 298  
 mark\_object\_as\_deleted\_from\_federation, 279  
 master, 298  
 mode\_transition\_requested, 298  
 multiphase\_init\_sync\_pnt\_list, 298  
 multiphase\_init\_sync\_points, 299  
 next\_mode\_cte\_time, 299  
 next\_mode\_scenario\_time, 299  
 object\_instance\_name\_reservation\_failed, 280  
 object\_instance\_name\_reservation\_succeeded, 280  
 operator=, 281  
 perform\_save, 281  
 post\_multi\_phase\_init\_processes, 281  
 pre\_multi\_phase\_init\_processes, 281  
 process\_deleted\_objects, 281  
 process\_execution\_control\_updates, 281  
 process\_mode\_interaction, 282  
 provide\_attribute\_update, 282  
 publish, 282  
 receive\_cyclic\_data, 282  
 receive\_execution\_configuration, 283  
 receive\_interaction, 283  
 register\_interactions\_with\_RTI, 283  
 register\_objects\_with\_RTI, 284  
 reinstate\_logged\_sync\_pts, 284  
 remove\_execution\_configuration, 284  
 requested\_execution\_control\_mode, 299  
 run\_mode\_transition, 284  
 scenario\_freeze\_time, 299  
 scenario\_timeline, 300  
 send\_execution\_configuration, 284  
 send\_mode\_transition\_interaction, 285  
 send\_requested\_data, 285  
 set\_current\_execution\_control\_mode, 285  
 set\_execution\_configuration, 285  
 set\_federate, 286  
 set\_least\_common\_time\_step, 286  
 set\_master, 286  
 set\_mode\_transition\_requested, 287  
 set\_next\_execution\_control\_mode, 287  
 set\_requested\_execution\_control\_mode, 287  
 set\_scenario\_freeze\_time, 288  
 set\_scenario\_timeline, 288  
 set\_simulation\_freeze\_time, 288  
 set\_time\_padding, 288  
 set\_wait\_sleep, 289  
 set\_wait\_timeout, 289  
 setup, 289, 290  
 setup\_checkpoint, 290  
 setup\_interaction\_ref\_attributes, 290  
 setup\_interaction\_RTI\_handles, 290  
 setup\_object\_ref\_attributes, 291

setup\_object\_RTI\_handles, 291  
should\_print, 291  
shutdown, 292  
shutdown\_mode\_announce, 292  
shutdown\_mode\_transition, 292  
sim\_timeline, 300  
simulation\_freeze\_time, 300  
start\_federation\_save\_at\_scenario\_time, 292  
subscribe, 292  
time\_padding, 301  
un\_freeze, 292  
unpublish, 293  
unsubscribe, 293  
use\_preset\_master, 301  
wait\_for\_all\_multiphase\_init\_sync\_pnts, 293  
wait\_for\_sync\_point\_announce, 293  
wait\_on\_init\_data, 293  
wait\_on\_init\_sync\_point, 294  
wait\_sleep, 301  
wait\_timeout, 301  
TrickHLA::FedAmb, 302  
~FedAmb, 307  
announceSynchronizationPoint, 307  
attributelsNotOwned, 307  
attributelsOwnedByRTI, 307  
attributeOwnershipAcquisitionNotification, 308  
attributeOwnershipUnavailable, 308  
attributesInScope, 308  
attributesOutOfScope, 308  
confirmAttributeOwnershipAcquisitionCancellation, 308  
confirmAttributeTransportationTypeChange, 309  
confirmInteractionTransportationTypeChange, 309  
connectionLost, 309  
discoverObjectInstance, 309  
FedAmb, 306, 307  
federate, 321  
federation\_restore\_status\_response\_context\_switch, 321  
federation\_restored\_rebuild\_federate\_handle\_set, 321  
federationNotRestored, 310  
federationNotSaved, 310  
federationRestoreBegin, 310  
federationRestored, 310  
federationRestoreStatusResponse, 310  
federationSaved, 310  
federationSaveStatusResponse, 310  
federationSynchronized, 311  
get\_manager, 311  
informAttributeOwnership, 311  
init\_attrTrickHLA\_FedAmb, 321  
initialize, 311  
initiateFederateRestore, 311  
initiateFederateSave, 312  
InputProcessor, 321  
manager, 322  
multipleObjectInstanceNameReservationFailed, 312  
multipleObjectInstanceNameReservationSucceeded, 312  
objectInstanceId, 312  
objectInstanceId, 312  
objectInstanceId, 312  
operator=, 313  
provideAttributeValueUpdate, 313  
receiveInteraction, 313, 314  
reflectAttributeValues, 314, 315  
removeObjectInstance, 315  
reportAttributeTransportationType, 316  
reportFederationExecutions, 316  
reportInteractionTransportationType, 316  
requestAttributeOwnershipAssumption, 316  
requestAttributeOwnershipRelease, 316  
requestDivestitureConfirmation, 317  
requestFederationRestoreFailed, 317  
requestFederationRestoreSucceeded, 317  
requestRetraction, 317  
reset\_federation\_restored\_rebuild\_federate\_handle\_set, 317  
set\_federation\_restore\_status\_response\_to\_echo, 318  
set\_federation\_restore\_status\_response\_to\_process, 318  
set\_federation\_restored\_rebuild\_federate\_handle\_set, 318  
setup, 318  
should\_print, 318  
startRegistrationForObjectClass, 319  
stopRegistrationForObjectClass, 319  
synchronizationPointRegistrationFailed, 319  
synchronizationPointRegistrationSucceeded, 319  
timeAdvanceGrant, 319  
timeConstrainedEnabled, 320  
timeRegulationEnabled, 320  
turnInteractionsOff, 320  
turnInteractionsOn, 320  
turnUpdatesOffForObjectInstance, 320  
turnUpdatesOnForObjectInstance, 320, 321  
TrickHLA::Federate, 322  
~Federate, 337  
achieve\_and\_wait\_for\_synchronization, 337  
achieve\_synchronization\_point, 338  
add\_a\_single\_entry\_into\_running\_feds, 338  
add\_federate\_instance\_id, 338  
add\_MOM\_HLAfederate\_instance\_id, 338  
add\_MOM\_HLAfederation\_instance\_id, 339  
all\_federates\_joined, 389  
announce\_freeze, 389  
announce\_restore, 389

announce\_save, 389  
 announce\_sync\_point, 389  
 ask\_MOM\_for\_auto\_provide\_setting, 339  
 ask\_MOM\_for\_federate\_names, 339  
 auto\_provide\_setting, 390  
 backup\_auto\_provide\_setting\_from\_MOM\_then\_disable, 340  
 can\_rejoin\_federation, 390  
 check\_for\_shutdown, 340  
 check\_for\_shutdown\_with\_termination, 340  
 check\_freeze, 341  
 check\_HLA\_save\_directory, 341  
 checkpoint\_file\_name, 390  
 checkpoint\_rt\_itimer, 390  
 clear\_running\_feds, 341  
 complete\_restore, 341  
 convert\_sync\_pts, 341  
 copy\_running\_feds\_into\_known\_feds, 342  
 create\_and\_join\_federation, 342  
 create\_federation, 342  
 create\_RTI\_ambassador\_and\_connect, 342  
 cstr\_restore\_label, 390  
 cstr\_save\_label, 390  
 destroy, 343  
 destroy\_orphaned\_federation, 343  
 determine\_federate\_MOM\_object\_instance\_names, 343  
 enable\_async\_delivery, 343  
 enable\_known\_feds, 391  
 enable\_MOM\_auto\_provide\_setting, 344  
 enter\_freeze, 344  
 execution\_control, 391  
 execution\_has\_begun, 391  
 exit\_freeze, 344  
 Federate, 336, 337  
 federate\_ambassador, 391  
 federate\_can\_rejoin\_federation, 344  
 federate\_has\_been\_restarted, 391  
 federate\_id, 392  
 federation\_created\_by\_federate, 392  
 federation\_exists, 392  
 federation\_joined, 392  
 federation\_name, 392  
 federation\_restore\_failed\_callback\_complete, 392  
 federation\_restored, 345  
 federation\_saved, 345  
 federation\_synchronized, 345  
 fix\_FPU\_control\_word, 345  
 FOM\_modules, 393  
 freeze\_delay\_frames, 393  
 freeze\_init, 346  
 freeze\_the\_federation, 393  
 get\_execution\_control, 346  
 get\_fed\_ambassador, 346  
 get\_federate\_name, 346  
 get\_federate\_type, 346  
 get\_federation\_name, 347  
 get\_freeze\_announced, 347  
 get\_freeze\_pending, 347  
 get\_granted\_fed\_time, 347  
 get\_granted\_time, 348  
 get\_joined\_federate\_handles, 348  
 get\_loookahead, 348  
 get\_loookahead\_time, 349  
 get\_manager, 349  
 get\_MOM\_HLAfederate\_class\_handle, 349  
 get\_requested\_fed\_time, 349  
 get\_requested\_time, 350  
 get\_restart, 350  
 get\_restart\_cfg, 350  
 get\_RTI\_ambassador, 350  
 get\_running\_feds\_count, 351  
 get\_stale\_data\_counter, 351  
 got\_startup\_sp, 393  
 granted\_time, 393  
 has\_restore\_been\_announced, 351  
 has\_restore\_process\_restore\_request\_failed, 352  
 has\_restore\_process\_restore\_request\_succeeded, 352  
 has\_restore\_request\_failed, 352  
 has\_restore\_request\_succeeded, 352  
 HLA\_save\_directory, 393  
 HLA\_time, 394  
 in\_time\_regulating\_state, 352  
 inform\_RTI\_of\_restore\_completion, 353  
 init\_attrTrickHLA\_Federate, 389  
 initialize, 353  
 initialize\_MOM\_handles, 353  
 initiate\_restore\_announce, 354  
 initiate\_restore\_flag, 394  
 initiate\_save\_announce, 354  
 initiate\_save\_flag, 394  
 InputProcessor, 389  
 is\_a\_required\_startup\_federate, 354  
 is\_execution\_member, 355  
 is\_federate\_executing, 355  
 is\_federate\_instance\_id, 355  
 is\_federation\_created\_by\_federate, 356  
 is\_HLA\_save\_and\_restore\_supported, 356  
 is\_joined\_federate, 356, 357  
 is\_MOM\_HLAfederate\_class, 357  
 is\_MOM\_HLAfederation\_class, 357  
 is\_MOM\_HLAfederation\_instance\_id, 358  
 is\_required\_federate, 358  
 is\_start\_to\_restore, 358  
 is\_time\_advance\_granted, 359  
 is\_time\_management\_enabled, 359  
 is\_zero\_loookahead\_time, 359

join\_federation, 359  
joined\_federate\_handles, 394  
joined\_federate\_name\_map, 394  
joined\_federate\_names, 395  
known\_feds, 395  
known\_feds\_count, 395  
load\_and\_print\_running\_federate\_names, 361  
local\_settings, 395  
lookahead, 395  
lookahead\_time, 396  
make\_copy\_of\_run\_directory, 396  
manager, 396  
MIM\_module, 396  
MOM\_HLAautoProvide\_handle, 396  
MOM\_HLAautoProvide\_param\_handle, 396  
MOM\_HLAfederate\_class\_handle, 397  
MOM\_HLAfederate\_handle, 397  
mom\_HLAfederate\_inst\_name\_map, 397  
MOM\_HLAfederateName\_handle, 397  
MOM\_HLAfederatesInFederation\_handle, 397  
MOM\_HLAfederateType\_handle, 398  
MOM\_HLAfederation\_class\_handle, 398  
mom\_HLAfederation\_instance\_name\_map, 398  
MOM\_HLAssetSwitches\_class\_handle, 398  
name, 398  
operator=, 361  
orig\_auto\_provide\_setting, 399  
perform\_checkpoint, 361  
perform\_restore, 361  
perform\_time\_advance\_request, 362  
post\_checkpoint, 362  
post\_multiphase\_initialization, 362  
post\_restore, 362  
pre\_multiphase\_initialization, 363  
prev\_restore\_process, 399  
print\_requested\_federation\_restore\_status, 363  
print\_restore\_failure\_reason, 363  
print\_save\_failure\_reason, 364  
print\_version, 364  
process\_requested\_federation\_restore\_status, 364  
process\_requested\_federation\_save\_status, 364  
publish\_data, 399  
publish\_interaction\_class, 365  
read\_running\_feds\_file, 365  
rebuild\_federate\_handles, 365  
register\_generic\_sync\_point, 366  
reinstate\_logged\_sync\_pts, 366  
remove\_federate\_instance\_id, 366  
remove\_MOM\_HLAfederate\_instance\_id, 366  
remove\_MOM\_HLAfederation\_instance\_id, 367  
request\_attribute\_update, 367  
request\_federation\_restore\_status, 367  
request\_federation\_save, 368  
request\_federation\_save\_status, 368  
requested\_federation\_restore\_status, 368  
requested\_time, 399  
resign, 368  
resign\_so\_we\_can\_rejoin, 369  
restart\_cfg\_flag, 399  
restart\_checkpoint, 369  
restart\_flag, 399  
restart\_initialization, 369  
restore\_began, 400  
restore\_checkpoint, 370  
restore\_completed, 400  
restore\_failed, 400  
restore\_federate\_handles\_from\_MOM, 370  
restore\_in\_progress, 400  
restore\_is\_imminent, 400  
restore\_label\_generated, 400  
restore\_name, 401  
restore\_orig\_MOM\_auto\_provide\_setting, 370  
restore\_process, 401  
restore\_request\_complete, 401  
RTI\_ambassador, 401  
running\_feds, 402  
running\_feds\_count, 402  
running\_feds\_count\_at\_time\_of\_restore, 402  
save\_completed, 402  
save\_label\_generated, 402  
save\_name, 402  
save\_request\_complete, 403  
send\_interaction, 370  
set\_all\_federate\_MOM\_instance\_handles\_by\_name, 371  
set\_announce\_save, 371  
set\_checkpoint\_file\_name, 371  
set\_federate\_has\_begun\_execution, 371  
set\_federation\_name, 371  
set\_freeze\_announced, 372  
set\_granted\_time, 372  
set\_lookahead, 373  
set\_MOM\_HLAfederate\_instance\_attributes, 373  
set\_MOM\_HLAfederation\_instance\_attributes, 373  
set\_requested\_time, 374  
set\_restart, 374  
set\_restart\_cfg, 374  
set\_restore\_began, 375  
set\_restore\_completed, 375  
set\_restore\_failed, 375  
set\_restore\_is\_imminent, 375  
set\_restore\_name, 375  
set\_restore\_request\_failed, 376  
set\_restore\_request\_succeeded, 376  
set\_save\_completed, 376  
set\_save\_name, 376  
set\_start\_to\_restore, 376  
set\_start\_to\_save, 377

set\_startup, 377  
 set\_time\_advance\_grant, 377  
 set\_time\_constrained\_state, 378  
 set\_time\_regulation\_state, 378  
 setup, 378  
 setup\_checkpoint, 379  
 setup\_restore, 379  
 setup\_time\_constrained, 379  
 setup\_time\_management, 380  
 setup\_time\_regulation, 380  
 should\_print, 380  
 should\_publish\_data, 381  
 shutdown, 381  
 shutdown\_called, 403  
 shutdown\_time\_constrained, 382  
 shutdown\_time\_management, 382  
 shutdown\_time\_regulating, 382  
 stale\_data\_counter, 403  
 start\_to\_restore, 403  
 start\_to\_save, 403  
 str\_restore\_label, 404  
 str\_save\_label, 404  
 subscribe\_attributes, 382  
 sync\_point\_registration\_failed, 383  
 sync\_point\_registration\_succeeded, 383  
 time\_adv\_grant, 404  
 time\_advance\_request, 383  
 time\_advance\_request\_to\_GALT, 384  
 time\_advance\_request\_to\_GALT\_LCTS\_multiple,  
     384  
 time\_constrained, 404  
 time\_constrained\_state, 404  
 time\_management, 404  
 time\_regulating, 405  
 time\_regulating\_state, 405  
 type, 405  
 un\_freeze, 384  
 unfreeze, 384  
 unfreeze\_after\_save, 405  
 unpublish\_interaction\_class, 384  
 unsubscribe\_all\_HLAfederate\_class\_attributes\_from\_MOM,  
     385  
 unsubscribe\_all\_HLAfederation\_class\_attributes\_from\_MOM  
     operator/, 428  
     385  
 unsubscribe\_attributes, 385  
 update\_running\_feds, 386  
 wait\_for\_federation\_restore\_begun, 386  
 wait\_for\_federation\_restore\_failed\_callback\_to\_complete,  
     386  
 wait\_for\_federation\_restore\_to\_complete, 386  
 wait\_for\_required\_federates\_to\_join, 386  
 wait\_for\_restore\_request\_callback, 387  
 wait\_for\_restore\_status\_to\_complete, 387  
 wait\_for\_save\_status\_to\_complete, 387  
 wait\_for\_time\_advance\_grant, 387, 388  
 wait\_until\_federation\_is\_ready\_to\_restore, 388  
 write\_running\_feds\_file, 388  
 ws\_restore\_label, 405  
 ws\_save\_label, 406  
 TrickHLA::Int64Interval, 411  
     \_HLAinterval, 418  
     ~Int64Interval, 414  
     get, 414  
     getDoubleTime, 414  
     getMicros, 414  
     getSeconds, 414  
     getTimeInMicros, 415  
     init\_attrTrickHLA\_\_Int64Interval, 418  
     InputProcessor, 418  
     Int64Interval, 412, 413  
     operator>, 416  
     operator=, 415, 416  
     setTo, 417  
     toMicroseconds, 418  
     toString, 418  
 TrickHLA::Int64Time, 419  
     \_HLAtime, 433  
     ~Int64Time, 422  
     decode, 422  
     get, 423  
     getDoubleTime, 423  
     getMicros, 423  
     getSeconds, 424  
     getTimeInMicros, 424  
     init\_attrTrickHLA\_\_Int64Time, 433  
     InputProcessor, 433  
     Int64Time, 421, 422  
     operator!=, 424, 425  
     operator<, 428  
     operator<=, 428  
     operator>, 430, 431  
     operator>=, 431  
     operator\*, 425  
     operator+, 426  
     operator+=, 426, 427  
     operator-, 427  
     operator/, 428  
     operator=, 429  
     operator==, 430  
     operator%, 425  
     setTo, 432  
     toString, 433  
 TrickHLA::Interaction, 434  
     ~Interaction, 437  
     changed, 448  
     class\_handle, 449  
     extract\_data, 438  
     FOM\_name, 449

get\_class\_handle, 438  
get\_fed\_lookahead, 438  
get\_federate, 439  
get\_FOM\_name, 439  
get\_granted\_fed\_time, 439  
get\_handler, 439  
get\_manager, 440  
get\_parameter\_count, 440  
get\_parameters, 440  
get\_preferred\_order, 440  
get\_RTI\_ambassador, 441  
handler, 449  
init\_attrTrickHLA\_\_Interaction, 448  
initialize, 441  
InputProcessor, 448  
Interaction, 437  
is\_changed, 441  
is\_publish, 442  
is\_subscribe, 442  
lock, 442  
manager, 449  
mark\_changed, 442  
mark\_unchanged, 442  
mutex, 449  
operator=, 443  
param\_count, 449  
parameters, 450  
preferred\_order, 450  
process\_interaction, 443  
publish, 450  
publish\_interaction, 443  
received\_as\_TSO, 450  
remove, 443  
send, 443, 444  
set\_class\_handle, 444  
set\_FOM\_name, 445  
set\_handler, 445  
set\_parameter\_count, 445  
set\_parameters, 446  
set\_publish, 446  
set\_subscribe, 446  
set\_user\_supplied\_tag, 446  
setup\_preferred\_order\_with\_RTI, 447  
should\_print, 447  
subscribe, 450  
subscribe\_to\_interaction, 447  
time, 450  
unlock, 447  
unpublish\_interaction, 448  
unsubscribe\_from\_interaction, 448  
user\_supplied\_tag, 451  
user\_supplied\_tag\_capacity, 451  
user\_supplied\_tag\_size, 451  
TrickHLA::InteractionHandler, 451  
~InteractionHandler, 453  
get\_cte\_time, 454  
get\_fed\_lookahead, 454  
get\_granted\_fed\_time, 454  
get\_scenario\_time, 455  
get\_sim\_time, 455  
init\_attrTrickHLA\_\_InteractionHandler, 458  
initialize\_callback, 455  
InputProcessor, 458  
interaction, 458  
InteractionHandler, 453, 454  
operator=, 455  
receive\_interaction, 456  
send\_interaction, 456, 457  
should\_print, 457  
TrickHLA::InteractionItem, 458  
~InteractionItem, 462  
checkpoint\_queue, 462  
clear\_parm\_items, 462  
index, 464  
init\_attrTrickHLA\_\_InteractionItem, 464  
initialize, 463  
InputProcessor, 464  
interaction\_type, 464  
InteractionItem, 461, 462  
is\_receive\_order, 463  
is\_timestamp\_order, 463  
operator=, 463  
order\_is\_TSO, 464  
parameter\_queue, 465  
parm\_items, 465  
parm\_items\_count, 465  
restore\_queue, 464  
time, 465  
user\_supplied\_tag, 465  
user\_supplied\_tag\_size, 466  
TrickHLA::Item, 466  
~Item, 467  
init\_attrTrickHLA\_\_Item, 468  
InputProcessor, 468  
Item, 467  
next, 468  
operator=, 467  
TrickHLA::ItemQueue, 468  
~ItemQueue, 470  
count, 474  
dump\_head\_pointers, 471  
empty, 471  
front, 471  
head, 474  
init\_attrTrickHLA\_\_ItemQueue, 473  
InputProcessor, 473  
ItemQueue, 470, 471  
lock, 471

mutex, 474  
 next, 472  
 operator=, 472  
 original\_head, 474  
 pop, 472  
 push, 472  
 rewind, 473  
 size, 473  
 tail, 474  
 unlock, 473  
**TrickHLA::KnownFederate**, 474  
 ~KnownFederate, 475  
 init\_attrTrickHLA\_KnownFederate, 476  
**InputProcessor**, 476  
**KnownFederate**, 475  
 MOM\_instance\_name, 476  
 name, 476  
 operator=, 476  
 required, 477  
**TrickHLA::LagCompensation**, 477  
 ~LagCompensation, 479  
 get\_attribute, 479  
 get\_attribute\_and\_validate, 480  
 get\_cte\_time, 480  
 get\_fed\_lookahead, 480  
 get\_granted\_fed\_time, 480  
 get\_scenario\_time, 481  
 init\_attrTrickHLA\_LagCompensation, 482  
 initialize\_callback, 481  
**InputProcessor**, 482  
**LagCompensation**, 479  
 object, 482  
 operator=, 481  
 receive\_lag\_compensation, 481  
 send\_lag\_compensation, 481  
 should\_print, 482  
**TrickHLA::LoggableSyncPnt**, 483  
 ~LoggableSyncPnt, 484  
 clear, 484  
 init\_attrTrickHLA\_LoggableSyncPnt, 485  
**InputProcessor**, 485  
 label, 485  
**LoggableSyncPnt**, 484  
 operator=, 484  
 state, 485  
**TrickHLA::LoggableTimedSyncPnt**, 485  
 ~LoggableTimedSyncPnt, 487  
 init\_attrTrickHLA\_LoggableTimedSyncPnt, 487  
**InputProcessor**, 487  
**LoggableTimedSyncPnt**, 487  
 operator=, 487  
 time, 488  
**TrickHLA::Manager**, 488  
 ~Manager, 494  
 add\_object\_to\_map, 495  
 check\_interactions, 519  
 check\_interactions\_count, 519  
 clear\_init\_sync\_points, 495  
 clear\_interactions, 495  
 debug\_handler, 519  
 determine\_job\_cycle\_time, 495  
 discover\_object\_instance, 495  
 dump\_interactions, 496  
 execution\_control, 520  
 Federate, 519  
 federate, 520  
 federate\_has\_been\_restored, 520  
 get\_debug\_handler, 496  
 get\_execution\_configuration, 496  
 get\_execution\_control, 497  
 get\_fed\_lookahead, 497  
 get\_federate, 497  
 get\_granted\_fed\_time, 497  
 get\_granted\_time, 498  
 get\_interaction\_count, 498  
 get\_interactions, 498  
 get\_object\_count, 498  
 get\_objects, 498  
 get\_RTI\_ambassador, 499  
 get\_trickhla\_object, 499  
 get\_unregistered\_object, 500  
 get\_unregistered\_remote\_object, 500  
 grant\_pull\_request, 501  
 has\_federate\_been\_restored, 501  
 init\_attrTrickHLA\_Manager, 519  
 initialization\_execution\_control, 501  
 initialize, 501  
 initiate\_federation\_save, 501  
 initiated\_a\_federation\_save, 520  
**InputProcessor**, 519  
 inter\_count, 521  
 interactions, 521  
 interactions\_queue, 521  
 is\_execution\_configuration\_used, 502  
 is\_late\_joining\_federate, 502  
 is\_restore\_federate, 502  
 is\_RTI\_ready, 502  
 is\_shutdown, 503  
 is\_this\_a\_rejoining\_federate, 503  
 job\_cycle\_time, 521  
**Manager**, 494  
 mark\_object\_as\_deleted\_from\_federation, 503  
 mgr\_initialized, 521  
 obj\_count, 521  
 object\_instance\_name\_reservation\_failed, 504  
 object\_instance\_name\_reservation\_succeeded, 504  
 object\_map, 522  
 objects, 522

operator=, 504  
process\_deleted\_objects, 504  
process\_interactions, 505  
process\_ownership, 505  
provide\_attribute\_update, 505  
publish, 505  
publish\_and\_subscribe, 505  
pull\_ownership, 506  
pull\_ownership\_upon\_rejoin, 506  
push\_ownership, 506  
receive\_cyclic\_data, 506  
receive\_init\_data, 507  
receive\_interaction, 507  
register\_objects\_with\_RTI, 508  
rejoining\_federate, 522  
release\_ownership, 508  
request\_data\_update, 508, 509  
reserve\_object\_names\_with\_RTI, 509  
reset\_mgr\_initialized, 509  
restart\_initialization, 509  
restore\_determined, 522  
restore\_federate, 523  
restore\_federation, 523  
restore\_file\_name, 523  
restore\_interactions, 510  
send\_cyclic\_data, 510  
send\_init\_data, 510  
send\_requested\_data, 511  
set\_all\_object\_instance\_handles\_by\_name, 511  
set\_execution\_configuration, 511  
set\_object\_instance\_handles\_by\_name, 511  
setup, 512  
setup\_all\_ref\_attributes, 512  
setup\_all\_RTI\_handles, 513  
setup\_checkpoint, 513  
setup\_checkpoint\_interactions, 513  
setup\_interaction\_ref\_attributes, 513  
setup\_interaction\_RTI\_handles, 514  
setup\_object\_ref\_attributes, 514  
setup\_object\_RTI\_handles, 515  
setup\_preferred\_order\_with\_RTI, 515  
should\_print, 515  
shutdown, 516  
shutdown\_called, 523  
start\_federation\_save, 516  
start\_federation\_save\_at\_scenario\_time, 516  
start\_federation\_save\_at\_sim\_time, 517  
subscribe, 517  
unpublish, 517  
unsubscribe, 517  
wait\_for\_init\_sync\_point, 517  
wait\_on\_discovery\_of\_objects, 518  
wait\_on\_registration\_of\_required\_objects, 518  
wait\_on\_reservation\_of\_object\_names, 518

TrickHLA::Object, 530  
~Object, 538  
any\_attribute\_FOM\_specified\_order, 567  
any\_attribute\_published, 538  
any\_attribute\_subscribed, 538  
any\_attribute\_timestamp\_order, 568  
any\_locally\_owned\_attribute, 539  
any\_locally\_owned\_published\_attribute, 539  
any\_locally\_owned\_published\_cyclic\_data\_ready\_attribute, 540  
any\_locally\_owned\_published\_init\_attribute, 540  
any\_locally\_owned\_published\_requested\_attribute, 540  
any\_remotely\_owned\_subscribed\_attribute, 540  
any\_remotely\_owned\_subscribed\_cyclic\_attribute, 541  
any\_remotely\_owned\_subscribed\_init\_attribute, 541  
attr\_count, 568  
attr\_update\_requested, 568  
attribute\_FOM\_names, 568  
attribute\_values\_map, 569  
attributes, 569  
blocking\_cyclic\_read, 569  
build\_attribute\_map, 541  
changed, 569  
class\_handle, 569  
clock, 570  
create\_attribute\_set, 541  
create\_HLA\_instance, 570  
create\_requested\_attribute\_set, 542  
data\_changed, 570  
deleted, 570  
divest\_requested, 570  
extract\_data, 542  
FOM\_name, 570  
get\_attribute, 542, 543  
get\_attribute\_count, 543  
get\_attribute\_FOM\_names, 544  
get\_attributes, 544  
get\_class\_handle, 544  
get\_fed\_lookahead, 544  
get\_federate, 545  
get\_FOM\_name, 545  
get\_granted\_fed\_time, 545  
get\_granted\_time, 546  
get\_instance\_handle, 546  
get\_lag\_compensation\_type, 546  
get\_last\_update\_time, 546  
get\_name, 547  
get\_name\_string, 547  
get\_RTI\_ambassador, 548  
get\_update\_time\_plus\_lookahead, 548  
grant\_pull\_request, 548  
grant\_push\_request, 548

grant\_push\_request\_pthread, 549  
 init\_attrTrickHLA\_\_Object, 567  
 initialize, 549  
 InputProcessor, 567  
 instance\_handle, 571  
 is\_attribute\_update\_requested, 549  
 is\_changed, 550  
 is\_create\_HLA\_instance, 550  
 is\_instance\_handle\_valid, 550  
 is\_name\_registered, 551  
 is\_name\_required, 551  
 is\_object\_deleted\_from\_RTI, 551  
 is\_required, 551  
 lag\_comp, 571  
 lag\_comp\_type, 571  
 last\_update\_time, 571  
 lock, 551  
 manager, 571  
 mark\_all\_attributes\_as\_nonlocal, 552  
 mark\_changed, 552  
 mark\_required, 552  
 mark\_unchanged, 552  
 mutex, 572  
 name, 572  
 name\_registered, 572  
 name\_required, 572  
 negotiated\_attribute\_ownership\_divestiture, 552  
 notify\_attribute\_ownership\_changed, 553  
 Object, 538  
 object\_deleted\_from\_RTI, 572  
 operator=, 553  
 ownership, 573  
 ownership\_lock, 553  
 ownership\_mutex, 573  
 ownership\_unlock, 553  
 pack\_attribute\_buffers, 553  
 pack\_cyclic\_attribute\_buffers, 554  
 pack\_init\_attribute\_buffers, 554  
 pack\_requested\_attribute\_buffers, 554  
 packing, 573  
 process\_deleted\_object, 554  
 provide\_attribute\_update, 554  
 publish\_object\_attributes, 555  
 pull\_ownership, 555  
 pull\_ownership\_upon\_rejoin, 555  
 pull\_requested, 573  
 push\_ownership, 556  
 receive\_cyclic\_data, 556  
 receive\_init\_data, 556  
 register\_object\_with\_RTI, 557  
 release\_ownership, 557  
 remove, 557  
 remove\_object\_instance, 558  
 removed\_instance, 573  
 request\_attribute\_value\_update, 558  
 required, 574  
 reserve\_object\_name\_with\_RTI, 558  
 restore\_ownership\_transfer\_checkpointed\_data, 558  
 rti\_ambassador, 574  
 send\_cyclic\_data, 559  
 send\_init\_data, 559  
 send\_requested\_data, 560  
 set\_class\_handle, 560  
 set\_core\_job\_cycle\_time, 561  
 set\_create\_HLA\_instance, 561  
 set\_divest\_requested, 561  
 set\_instance\_handle, 561  
 set\_instance\_handle\_and\_name, 562  
 set\_lag\_compensation\_type, 562  
 set\_last\_update\_time, 562  
 set\_name, 563  
 set\_name\_and\_mark\_changed, 563  
 set\_name\_registered, 563  
 set\_name\_unregistered, 563  
 set\_pull\_requested, 563  
 set\_to\_unblocking\_cyclic\_reads, 564  
 setup\_ownership\_transfer\_checkpointed\_data, 564  
 setup\_preferred\_order\_with\_RTI, 564  
 should\_print, 564  
 stop\_publishing\_attributes, 565  
 stop\_subscribing\_attributes, 565  
 subscribe\_to\_object\_attributes, 565  
 thla\_attribute\_map, 574  
 thla\_reflected\_attributes\_queue, 574  
 time\_plus\_lookahead, 574  
 unlock, 565  
 unpack\_attribute\_buffers, 566  
 unpack\_cyclic\_attribute\_buffers, 566  
 unpack\_init\_attribute\_buffers, 566  
 unpublish\_all\_object\_attributes, 566  
 unsubscribe\_all\_object\_attributes, 566  
 wait\_on\_object\_name\_reservation, 567  
 wait\_on\_object\_registration, 567  
 TrickHLA::ObjectDeleted, 575  
 ~ObjectDeleted, 575  
 deleted, 576  
 init\_attrTrickHLA\_\_ObjectDeleted, 576  
 InputProcessor, 576  
 ObjectDeleted, 575  
 TrickHLA::OpaqueBuffer, 576  
 ~OpaqueBuffer, 578  
 alignment, 583  
 buffer, 583  
 byteswap\_buffer\_copy, 578  
 capacity, 583  
 ensure\_buffer\_capacity, 579  
 get\_byte\_alignment, 579  
 get\_capacity, 580

init\_attrTrickHLA\_OpaqueBuffer, 583  
InputProcessor, 583  
OpaqueBuffer, 578  
pull\_from\_buffer, 580  
pull\_pad\_from\_buffer, 581  
pull\_pos, 583  
push\_pad\_to\_buffer, 581  
push\_pos, 584  
push\_to\_buffer, 581  
reset\_buffer\_positions, 582  
reset\_pull\_position, 582  
reset\_push\_position, 582  
set\_byte\_alignment, 582  
TrickHLA::OwnershipHandler, 584  
    ~OwnershipHandler, 587  
    clear\_checkpoint, 587  
    get\_attribute, 588  
    get\_attribute\_count, 588  
    get\_attribute\_FOM\_names, 588  
    get\_cte\_time, 588  
    get\_fed\_lookahead, 589  
    get\_granted\_fed\_time, 589  
    get\_object\_FOM\_name, 589  
    get\_object\_name, 589  
    get\_scenario\_time, 589  
    init\_attrTrickHLA\_OwnershipHandler, 594  
    initialize\_callback, 590  
    InputProcessor, 594  
    is\_locally\_owned, 590  
    is\_published, 590  
    is\_remotely\_owned, 591  
    is\_subscribed, 591  
    Object, 594  
    object, 595  
    operator=, 591  
    OwnershipHandler, 587  
    pull\_items, 595  
    pull\_items\_cnt, 595  
    pull\_ownership, 591, 592  
    pull\_requests, 595  
    push\_items, 595  
    push\_items\_cnt, 595  
    push\_ownership, 592, 593  
    push\_requests, 596  
    restore\_requests, 593  
    setup\_checkpoint\_requests, 594  
    should\_print, 594  
TrickHLA::OwnershipItem, 596  
    ~OwnershipItem, 598  
    clear, 598  
    FOM\_name, 599  
    init\_attrTrickHLA\_OwnershipItem, 598  
    InputProcessor, 598  
    operator=, 598  
    OwnershipItem, 598  
    time, 599  
TrickHLA::Packing, 599  
    ~Packing, 601  
    get\_attribute, 601  
    get\_attribute\_and\_validate, 602  
    get\_cte\_time, 602  
    get\_scenario\_time, 602  
    init\_attrTrickHLA\_Packing, 605  
    initialize\_callback, 602  
    InputProcessor, 605  
    object, 605  
    operator=, 604  
    pack, 604  
    Packing, 601  
    should\_print, 604  
    unpack, 604  
TrickHLA::Parameter, 605  
    ~Parameter, 609  
    address, 619  
    attr, 620  
    buffer, 620  
    buffer\_capacity, 620  
    byteswap, 620  
    byteswap\_buffer\_copy, 609  
    calculate\_number\_of\_items, 610  
    calculate\_size\_and\_number\_of\_items, 610  
    calculate\_static\_number\_of\_items, 611  
    complete\_initialization, 611  
    debug\_handler, 620  
    decode\_boolean\_from\_buffer, 611  
    decode\_logical\_time, 611  
    decode\_opaque\_data\_from\_buffer, 611  
    decode\_string\_from\_buffer, 612  
    encode\_boolean\_to\_buffer, 612  
    encode\_logical\_time, 612  
    encode\_opaque\_data\_to\_buffer, 612  
    encode\_string\_to\_buffer, 612  
    ensure\_buffer\_capacity, 613  
    extract\_data, 613  
    FOM\_name, 621  
    get\_encoded\_parameter\_value, 613  
    get\_FOM\_name, 614  
    get\_parameter\_handle, 614  
    get\_parameter\_size, 614  
    get\_ref2\_attributes, 614  
    get\_rti\_encoding, 615  
    get\_trick\_name, 615  
    HLAtrue, 621  
    init\_attrTrickHLA\_Parameter, 619  
    initialize, 615  
    InputProcessor, 619  
    interaction\_FOM\_name, 621  
    is\_byteswap, 616

is\_changed, 616  
 is\_static\_in\_size, 616  
 is\_supported\_parameter\_type, 616  
 mark\_changed, 617  
 mark\_unchanged, 617  
 num\_items, 621  
 operator=, 617  
 pack\_parameter\_buffer, 617  
 param\_handle, 621  
 Parameter, 609  
 print\_buffer, 618  
 rti\_encoding, 622  
 set\_debug\_level, 618  
 set\_encoding, 618  
 set\_FOM\_name, 618  
 set\_parameter\_handle, 619  
 size, 622  
 size\_is\_static, 622  
 trick\_name, 622  
 unpack\_parameter\_buffer, 619  
 value\_changed, 622  
**TrickHLA::ParameterItem**, 623  
 ~ParameterItem, 625  
 clear, 625  
 data, 626  
 index, 626  
 init\_attrTrickHLA\_ParameterItem, 625  
 InputProcessor, 625  
 operator=, 625  
 ParameterItem, 624, 625  
 size, 626  
**TrickHLA::ReflectedAttributesQueue**, 659  
 ~ReflectedAttributesQueue, 661  
 attribute\_map\_queue, 663  
 clear, 661  
 empty, 661  
 front, 662  
 init\_attrTrickHLA\_ReflectedAttributesQueue, 662  
 InputProcessor, 663  
 operator=, 662  
 pop, 662  
 push, 662  
 queue\_mutex, 663  
 ReflectedAttributesQueue, 661  
**TrickHLA::ScenarioTimeline**, 663  
 ~ScenarioTimeline, 665  
 compute\_HLT, 666  
 compute\_simulation\_time, 666  
 get\_HLT\_offset, 666  
 get\_sim\_offset, 666  
 get\_time, 667  
 hlt\_offset, 669  
 init\_attrTrickHLA\_ScenarioTimeline, 669  
 InputProcessor, 669  
 operator=, 667  
 ScenarioTimeline, 665  
 set\_HLT\_offset, 667  
 set\_sim\_offset, 668  
 sim\_offset, 669  
 sim\_timeline, 669  
 time\_from\_HLT, 668  
 time\_from\_simulation\_time, 668  
**TrickHLA::SimTimeline**, 674  
 ~SimTimeline, 675  
 get\_time, 675  
 init\_attrTrickHLA\_SimTimeline, 676  
 InputProcessor, 676  
 operator=, 676  
 SimTimeline, 675  
**TrickHLA::StringUtilities**, 721  
 ~StringUtilities, 722  
 init\_attrTrickHLA\_StringUtilities, 729  
 InputProcessor, 729  
 ip\_strdup\_wstring, 722  
 operator=, 723  
 StringUtilities, 722  
 to\_printable\_string, 723  
 to\_string, 723–725  
 to\_wstring, 726, 727  
 tokenize, 727, 728  
 trim\_whitespace, 728  
**TrickHLA::SyncPnt**, 729  
 ~SyncPnt, 732  
 achieve\_sync\_point, 732  
 convert, 733  
 exists, 733  
 get\_label, 733  
 get\_state, 733  
 init\_attrTrickHLA\_SyncPnt, 738  
 InputProcessor, 738  
 is\_achieved, 734  
 is\_announced, 734  
 is\_error, 734  
 is\_registered, 735  
 is\_synchronized, 735  
 is\_valid, 735  
 label, 739  
 register\_sync\_point, 735, 736  
 set\_label, 736  
 set\_state, 736  
 set\_wait\_sleep, 737  
 set\_wait\_timeout, 737  
 state, 739  
 SyncPnt, 732  
 to\_string, 737  
 wait\_for\_announce, 737  
 wait\_for\_synchronization, 738  
 wait\_sleep, 739

wait\_timeout, 739  
TrickHLA::SyncPntList, 739  
  ~SyncPntList, 741  
  init\_attrTrickHLA\_\_SyncPntList, 742  
  InputProcessor, 742  
  operator=, 741  
  SyncPntList, 741  
TrickHLA::SyncPntListBase, 742  
  ~SyncPntListBase, 746  
  achieve\_all\_sync\_pnts, 746  
  achieve\_and\_wait\_for\_synchronization, 747  
  achieve\_sync\_pnt, 747  
  add\_sync\_pnt, 748  
  announce\_sync\_point, 748  
  clear\_sync\_pnt, 749  
  contains, 749  
  convert\_sync\_pts, 749  
  debug\_handler, 758  
  get\_reconfig\_name, 750  
  get\_size, 750  
  get\_sync\_pnt, 750  
  get\_sync\_pnt\_state, 750  
  init\_attrTrickHLA\_\_SyncPntListBase, 758  
  InputProcessor, 758  
  is\_sync\_pnt\_announced, 751  
  lock\_read\_only, 751  
  lock\_read\_write, 751  
  mark\_announced, 752  
  mark\_registered, 752  
  mark\_synchronized, 752  
  operator=, 753  
  print\_sync\_pnts, 753  
  read\_locks, 758  
  reconfig\_name, 759  
  register\_all\_sync\_pnts, 753  
  register\_sync\_pnt, 754  
  reset, 755  
  set\_debug\_level, 755  
  sync\_point\_list, 759  
  sync\_point\_registration\_failed, 755  
  sync\_point\_registration\_succeeded, 756  
  SyncPntListBase, 746  
  to\_string, 756  
  unlock\_read\_only, 756  
  unlock\_read\_write, 756  
  wait\_for\_all\_announcements, 757  
  wait\_for\_announcement, 757  
  wait\_for\_list\_synchronization, 757  
  write\_locks, 759  
TrickHLA::TimedSyncPnt, 759  
  ~TimedSyncPnt, 762  
  convert, 762  
  get\_time, 762  
  init\_attrTrickHLA\_\_TimedSyncPnt, 763  
          InputProcessor, 763  
          set\_time, 763  
          time, 763  
          TimedSyncPnt, 761  
          to\_string, 763  
TrickHLA::TimedSyncPntList, 764  
  ~TimedSyncPntList, 766  
  achieve\_all\_sync\_pnts, 767  
  add\_sync\_pnt, 767, 768  
  check\_sync\_pnts, 768  
  convert\_sync\_pts, 768  
  init\_attrTrickHLA\_\_TimedSyncPntList, 769  
  InputProcessor, 769  
  operator=, 769  
  print\_sync\_pnts, 769  
  TimedSyncPntList, 766  
TrickHLA::Timeline, 769  
  ~Timeline, 771  
  epoch, 773  
  get\_elapsed\_time, 771  
  get\_epoch, 771  
  get\_time, 772  
  init\_attrTrickHLA\_\_Timeline, 773  
  InputProcessor, 773  
  operator=, 772  
  set\_epoch, 772  
  Timeline, 771  
TrickHLA::TimeOfDayTimeline, 773  
  ~TimeOfDayTimeline, 775  
  get\_time, 775  
  init\_attrTrickHLA\_\_TimeOfDayTimeline, 775  
  InputProcessor, 775  
  operator=, 775  
  TimeOfDayTimeline, 774, 775  
TrickHLA::Utilities, 776  
  ~Utilities, 777  
  byteswap\_double, 777  
  byteswap\_float, 778  
  byteswap\_int, 778  
  byteswap\_long, 778  
  byteswap\_long\_long, 779  
  byteswap\_short, 779  
  byteswap\_unsigned\_int, 779  
  byteswap\_unsigned\_long, 780  
  byteswap\_unsigned\_long\_long, 780  
  byteswap\_unsigned\_short, 780  
  get\_endianness, 781  
  get\_release\_date, 781  
  get\_version, 781  
  init\_attrTrickHLA\_\_Utilities, 782  
  InputProcessor, 782  
  is\_transmission\_byteswap, 781  
  operator=, 782  
  Utilities, 777

TRICKHLA\_INIT\_FPU\_CONTROL\_WORD  
 Utilities.hh, 982

TRICKHLA\_MANAGER\_BUILTIN\_FREEZE\_INTERACTION  
 TrickHLA, 51

TRICKHLA\_MANAGER\_BUILTIN\_MTR\_INTERACTION  
 TrickHLA, 51

TRICKHLA\_MANAGER\_USER\_DEFINED\_INTERACTION  
 TrickHLA, 51

TRICKHLA\_RELEASE\_DATE  
 Version.hh, 984

TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD  
 Utilities.hh, 982

TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD  
 Utilities.hh, 982

TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD  
 Utilities.hh, 982

TRICKHLA\_VERSION  
 Version.hh, 984

TrickHLAModel, 57

TrickHLAModel::SimpleSimConfig, 669  
 ~SimpleSimConfig, 671  
 init\_attrTrickHLAModel\_\_SimpleSimConfig, 673  
 initialize, 672  
 InputProcessor, 673  
 num\_federates, 673  
 operator=, 672  
 owner, 673  
 pack, 672  
 required\_federates, 673  
 run\_duration, 673  
 run\_duration\_microsec, 673  
 SimpleSimConfig, 671  
 unpack, 672

TrickHLAModel::SineConditional, 676  
 ~SineConditional, 678  
 attr\_pos, 680  
 convert\_FOM\_name\_to\_pos, 678  
 init\_attrTrickHLAModel\_\_SineConditional, 679  
 initialize, 679  
 InputProcessor, 680  
 prev\_sim\_data, 680  
 should\_send, 679  
 sim\_data, 680  
 SineConditional, 678

TrickHLAModel::SineData, 680  
 ~SineData, 683  
 adjust\_phase, 683  
 amp, 689  
 compute\_derivative, 683, 684  
 compute\_value, 684  
 copy\_data, 684  
 dvdt, 689  
 freq, 690  
 get\_amplitude, 685  
 get\_derivative, 685  
 get\_frequency, 685  
 get\_name, 685  
 get\_phase, 686  
 get\_time, 686  
 get\_tolerance, 686  
 get\_value, 686  
 init\_attrTrickHLAModel\_\_SineData, 689  
 InputProcessor, 689  
 integration, 687  
 name, 690  
 phase, 690  
 set\_amplitude, 687  
 set\_derivative, 687  
 set\_frequency, 687  
 set\_name, 688  
 set\_phase, 688  
 set\_time, 688  
 set\_tolerance, 689  
 set\_value, 689  
 SineData, 682  
 time, 690  
 tol, 690  
 value, 690

TrickHLAModel::SineInteractionHandler, 691  
 ~SineInteractionHandler, 693  
 init\_attrTrickHLAModel\_\_SineInteractionHandler, 694  
 InputProcessor, 694  
 message, 694  
 name, 694  
 operator=, 693  
 receive\_cnt, 694  
 receive\_interaction, 693  
 send\_cnt, 695  
 send\_sine\_interaction, 693  
 SineInteractionHandler, 692, 693  
 time, 695  
 year, 695

TrickHLAModel::SineLagCompensation, 695  
 ~SineLagCompensation, 697  
 amp\_attr, 699  
 dvdt\_attr, 700  
 freq\_attr, 700  
 init\_attrTrickHLAModel\_\_SineLagCompensation, 699  
 initialize, 698  
 initialize\_callback, 698  
 InputProcessor, 699  
 lag\_comp\_data, 700  
 name\_attr, 700  
 operator=, 698  
 phase\_attr, 700  
 receive\_lag\_compensation, 698  
 send\_lag\_compensation, 699  
 sim\_data, 700

SineLagCompensation, 697  
time\_attr, 701  
tol\_attr, 701  
value\_attr, 701  
TrickHLAModel::SineObjectDeleted, 701  
  ~SineObjectDeleted, 702  
  deleted, 703  
  init\_attrTrickHLAModel\_\_SineObjectDeleted, 703  
  InputProcessor, 703  
  operator=, 703  
  SineObjectDeleted, 702, 703  
TrickHLAModel::SineOwnershipHandler, 704  
  ~SineOwnershipHandler, 705  
  init\_attrTrickHLAModel\_\_SineOwnershipHandler, 706  
  initialize\_callback, 705  
  InputProcessor, 706  
  operator=, 706  
  SineOwnershipHandler, 705  
TrickHLAModel::SinePacking, 706  
  ~SinePacking, 709  
  amp\_attr, 711  
  buff, 711  
  buff\_size, 711  
  dvdt\_attr, 711  
  freq\_attr, 711  
  init\_attrTrickHLAModel\_\_SinePacking, 710  
  initialize, 709  
  initialize\_callback, 709  
  initialized, 711  
  InputProcessor, 711  
  name\_attr, 712  
  operator=, 710  
  pack, 710  
  pack\_count, 712  
  phase\_attr, 712  
  phase\_deg, 712  
  sim\_data, 712  
  SinePacking, 708, 709  
  time\_attr, 712  
  tol\_attr, 713  
  unpack, 710  
  value\_attr, 713  
TrickHLAObjInstanceNameMap  
  TrickHLA, 47  
TrickRTIAmbPtr  
  TrickHLA, 47  
trigger\_freeze\_interaction  
  IMSim::ExecutionControl, 218  
trim\_whitespace  
  TrickHLA::StringUtilities, 728  
turnInteractionsOff  
  TrickHLA::FedAmb, 320  
turnInteractionsOn

  TrickHLA::FedAmb, 320  
turnUpdatesOffForObjectInstance  
  TrickHLA::FedAmb, 320  
turnUpdatesOnForObjectInstance  
  TrickHLA::FedAmb, 320, 321  
type  
  DIS::ExecutionControl, 181  
  DSES::ExecutionControl, 196  
  IMSim::ExecutionControl, 220  
  SpaceFOM::ExecutionControl, 243  
  SpaceFOM::PhysicalEntityBase, 645  
  TrickHLA::ExecutionControl, 255  
  TrickHLA::Federate, 405  
Types.cpp, 963–966  
Types.hh, 967, 970, 972, 974, 976

un\_freeze  
  IMSim::ExecutionControl, 218  
  TrickHLA::ExecutionControlBase, 292  
  TrickHLA::Federate, 384

unfreeze  
  TrickHLA::Federate, 384

unfreeze\_after\_save  
  TrickHLA::Federate, 405

unlock  
  TrickHLA::Interaction, 447  
  TrickHLA::ItemQueue, 473  
  TrickHLA::Object, 565

unlock\_read\_only  
  TrickHLA::SyncPntListBase, 756

unlock\_read\_write  
  TrickHLA::SyncPntListBase, 756

unmark\_locally\_owned  
  TrickHLA::Attribute, 80

unpack  
  DIS::ExecutionConfiguration, 110  
  DSES::ExecutionConfiguration, 121  
  IMSim::ExecutionConfiguration, 133  
  SpaceFOM::DynamicalEntity, 101  
  SpaceFOM::ExecutionConfiguration, 145  
  SpaceFOM::PhysicalEntityBase, 643  
  SpaceFOM::RefFrameBase, 656  
  TrickHLA::ExecutionConfiguration, 152  
  TrickHLA::ExecutionConfigurationBase, 161  
  TrickHLA::Packing, 604  
  TrickHLAModel::SimpleSimConfig, 672  
  TrickHLAModel::SinePacking, 710

unpack\_attribute\_buffer  
  TrickHLA::Attribute, 80

unpack\_attribute\_buffers  
  TrickHLA::Object, 566

unpack\_cyclic\_attribute\_buffers  
  TrickHLA::Object, 566

unpack\_init\_attribute\_buffers

TrickHLA::Object, 566  
 unpack\_parameter\_buffer  
     TrickHLA::Parameter, 619  
 unpublish  
     DIS::ExecutionControl, 180  
     DSES::ExecutionControl, 195  
     IMSim::ExecutionControl, 218  
     SpaceFOM::ExecutionControl, 241  
     TrickHLA::ExecutionControl, 254  
     TrickHLA::ExecutionControlBase, 293  
     TrickHLA::Manager, 517  
 unpublish\_all\_object\_attributes  
     TrickHLA::Object, 566  
 unpublish\_interaction  
     TrickHLA::Interaction, 448  
 unpublish\_interaction\_class  
     TrickHLA::Federate, 384  
 unsubscribe  
     DIS::ExecutionControl, 180  
     DSES::ExecutionControl, 195  
     IMSim::ExecutionControl, 218  
     SpaceFOM::ExecutionControl, 242  
     TrickHLA::ExecutionControl, 254  
     TrickHLA::ExecutionControlBase, 293  
     TrickHLA::Manager, 517  
 unsubscribe\_all\_HLAfederate\_class\_attributes\_from\_MOM  
     TrickHLA::Federate, 385  
 unsubscribe\_all\_HLAfederation\_class\_attributes\_from\_MOM  
     TrickHLA::Federate, 385  
 unsubscribe\_all\_object\_attributes  
     TrickHLA::Object, 566  
 unsubscribe\_attributes  
     TrickHLA::Federate, 385  
 unsubscribe\_from\_interaction  
     TrickHLA::Interaction, 448  
 update\_pending  
     TrickHLA::ExecutionConfigurationBase, 161  
 update\_requested  
     TrickHLA::Attribute, 85  
 update\_running\_feds  
     TrickHLA::Federate, 386  
 use\_preset\_master  
     TrickHLA::ExecutionControlBase, 301  
 user\_supplied\_tag  
     TrickHLA::Interaction, 451  
     TrickHLA::InteractionItem, 465  
 user\_supplied\_tag\_capacity  
     TrickHLA::Interaction, 451  
 user\_supplied\_tag\_size  
     TrickHLA::Interaction, 451  
     TrickHLA::InteractionItem, 466  
 Utilities  
     TrickHLA::Utilities, 777  
 Utilities.cpp, 979

Utilities.hh, 980  
     \_FPU\_PC\_DOUBLE, 982  
     \_FPU\_PC\_EXTENDED, 982  
     \_FPU\_PC\_MASK, 982  
     \_FPU\_PC\_SINGLE, 982  
     \_FPU\_PC\_UNDEFINED, 982  
     TRICKHLA\_INIT\_FPU\_CONTROL\_WORD, 982  
     TRICKHLA\_RESTORE\_FPU\_CONTROL\_WORD,  
         982  
     TRICKHLA\_SAVE\_FPU\_CONTROL\_WORD, 982  
     TRICKHLA\_VALIDATE\_FPU\_CONTROL\_WORD,  
         982

value  
     TrickHLAModel::SineData, 690  
 value\_attr  
     TrickHLAModel::SineLagCompensation, 701  
     TrickHLAModel::SinePacking, 713  
 value\_changed  
     TrickHLA::Attribute, 85  
     TrickHLA::Parameter, 622  
 vector  
     QuaternionData, 646  
     SpaceFOM::QuaternionEncoder, 650  
 vector\_encoder  
     SpaceFOM::QuaternionEncoder, 650  
 VectorOfStrings  
     TrickHLA, 47  
 VectorOfWstrings  
     TrickHLA, 48  
 vel  
     SpaceTimeCoordinateData, 714  
 velocity  
     SpaceFOM::SpaceTimeCoordinateEncoder, 720  
 velocity\_encoder  
     SpaceFOM::SpaceTimeCoordinateEncoder, 720  
 Version.hh, 983  
     TRICKHLA\_RELEASE\_DATE, 984  
     TRICKHLA\_VERSION, 984

wait\_for\_all\_announcements  
     TrickHLA::SyncPntListBase, 757  
 wait\_for\_all\_multiphase\_init\_sync\_pnts  
     DSES::ExecutionControl, 195  
     IMSim::ExecutionControl, 219  
     TrickHLA::ExecutionControlBase, 293  
 wait\_for\_announce  
     TrickHLA::SyncPnt, 737  
 wait\_for\_announcement  
     TrickHLA::SyncPntListBase, 757  
 wait\_for\_federation\_restore\_begun  
     TrickHLA::Federate, 386  
 wait\_for\_federation\_restore\_failed\_callback\_to\_complete  
     TrickHLA::Federate, 386  
 wait\_for\_federation\_restore\_to\_complete

TrickHLA::Federate, 386  
wait\_for\_init\_sync\_point  
    TrickHLA::Manager, 517  
wait\_for\_list\_synchronization  
    TrickHLA::SyncPntListBase, 757  
wait\_for\_required\_federates\_to\_join  
    TrickHLA::Federate, 386  
wait\_for\_restore\_request\_callback  
    TrickHLA::Federate, 387  
wait\_for\_restore\_status\_to\_complete  
    TrickHLA::Federate, 387  
wait\_for\_save\_status\_to\_complete  
    TrickHLA::Federate, 387  
wait\_for\_sync\_point\_announce  
    TrickHLA::ExecutionControlBase, 293  
wait\_for\_synchronization  
    TrickHLA::SyncPnt, 738  
wait\_for\_time\_advance\_grant  
    TrickHLA::Federate, 387, 388  
wait\_on\_discovery\_of\_objects  
    TrickHLA::Manager, 518  
wait\_on\_init\_data  
    TrickHLA::ExecutionControl, 255  
    TrickHLA::ExecutionControlBase, 293  
wait\_on\_init\_sync\_point  
    DIS::ExecutionControl, 180  
    TrickHLA::ExecutionControl, 255  
    TrickHLA::ExecutionControlBase, 294  
wait\_on\_object\_name\_reservation  
    TrickHLA::Object, 567  
wait\_on\_object\_registration  
    TrickHLA::Object, 567  
wait\_on\_registration  
    TrickHLA::ExecutionConfigurationBase, 161  
wait\_on\_registration\_of\_required\_objects  
    TrickHLA::Manager, 518  
wait\_on\_reservation\_of\_object\_names  
    TrickHLA::Manager, 518  
wait\_on\_root\_frame\_discovered\_synchronization  
    SpaceFOM::ExecutionControl, 242  
wait\_on\_update  
    DIS::ExecutionConfiguration, 111  
    DSES::ExecutionConfiguration, 121  
    IMSim::ExecutionConfiguration, 133  
    SpaceFOM::ExecutionConfiguration, 145  
    TrickHLA::ExecutionConfigurationBase, 161  
wait\_sleep  
    TrickHLA::ExecutionControlBase, 301  
    TrickHLA::SyncPnt, 739  
wait\_timeout  
    TrickHLA::ExecutionControlBase, 301  
    TrickHLA::SyncPnt, 739  
wait\_until\_federation\_is\_ready\_to\_restore  
    TrickHLA::Federate, 388

wall\_clock\_time  
    TrickHLA::CTETimelineBase, 93  
WHITESPACE\_CHARS  
    StringUtilities.hh, 946  
write\_locks  
    TrickHLA::SyncPntListBase, 759  
write\_running\_feds\_file  
    TrickHLA::Federate, 388  
ws\_restore\_label  
    TrickHLA::Federate, 405  
ws\_save\_label  
    TrickHLA::Federate, 406

year  
    TrickHLAModel::SineInteractionHandler, 695