

# TimeModel

5.0

Generated by Doxygen 1.8.5

Wed Jun 1 2022 12:09:18



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>13</b>
6.1	Models . . . . .	13
6.1.1	Detailed Description . . . . .	13
6.2	Environment . . . . .	14
6.2.1	Detailed Description . . . . .	14
6.3	Time . . . . .	15
6.3.1	Detailed Description . . . . .	17
6.3.2	Macro Definition Documentation . . . . .	17
6.3.2.1	PATH . . . . .	17
<b>7</b>	<b>Namespace Documentation</b>	<b>19</b>
7.1	jeod Namespace Reference . . . . .	19
7.1.1	Detailed Description . . . . .	20
7.1.2	Function Documentation . . . . .	20
7.1.2.1	operator  . . . . .	20
<b>8</b>	<b>Data Structure Documentation</b>	<b>21</b>
8.1	jeod::JeodBaseTime Class Reference . . . . .	21
8.1.1	Detailed Description . . . . .	23

8.1.2	Constructor & Destructor Documentation . . . . .	23
8.1.2.1	JeodBaseTime . . . . .	23
8.1.2.2	~JeodBaseTime . . . . .	23
8.1.2.3	JeodBaseTime . . . . .	24
8.1.3	Member Function Documentation . . . . .	24
8.1.3.1	add_parent . . . . .	24
8.1.3.2	add_type_initialize . . . . .	24
8.1.3.3	add_type_update . . . . .	24
8.1.3.4	get_index . . . . .	25
8.1.3.5	initialize_from_parent . . . . .	25
8.1.3.6	initialize_initializer_time . . . . .	25
8.1.3.7	is_initialized . . . . .	25
8.1.3.8	must_be_singleton . . . . .	26
8.1.3.9	operator= . . . . .	26
8.1.3.10	override_initialized . . . . .	26
8.1.3.11	set_index . . . . .	26
8.1.3.12	set_name . . . . .	26
8.1.3.13	set_time_by_days . . . . .	26
8.1.3.14	set_time_by_seconds . . . . .	27
8.1.3.15	update . . . . .	27
8.1.4	Friends And Related Function Documentation . . . . .	27
8.1.4.1	init_attrjeod__JeodBaseTime . . . . .	27
8.1.4.2	InputProcessor . . . . .	27
8.1.4.3	TimeConverter . . . . .	28
8.1.4.4	TimeManagerInit . . . . .	28
8.1.5	Field Documentation . . . . .	28
8.1.5.1	clock_resolution . . . . .	28
8.1.5.2	days . . . . .	28
8.1.5.3	index . . . . .	28
8.1.5.4	initial_value . . . . .	28
8.1.5.5	initialize_from_name . . . . .	29
8.1.5.6	initialized . . . . .	29
8.1.5.7	initializing_value . . . . .	29
8.1.5.8	links . . . . .	29
8.1.5.9	name . . . . .	29
8.1.5.10	seconds . . . . .	30
8.1.5.11	time_manager . . . . .	30
8.1.5.12	update_converter_direction . . . . .	30
8.1.5.13	update_converter_ptr . . . . .	30
8.1.5.14	update_from_name . . . . .	31

8.2	<a href="#">jeod::TimeConverter Class Reference</a>	31
8.2.1	<a href="#">Detailed Description</a>	33
8.2.2	<a href="#">Member Enumeration Documentation</a>	33
8.2.2.1	<a href="#">Direction</a>	33
8.2.3	<a href="#">Constructor &amp; Destructor Documentation</a>	33
8.2.3.1	<a href="#">~TimeConverter</a>	33
8.2.3.2	<a href="#">TimeConverter</a>	33
8.2.3.3	<a href="#">TimeConverter</a>	34
8.2.4	<a href="#">Member Function Documentation</a>	34
8.2.4.1	<a href="#">can_convert</a>	34
8.2.4.2	<a href="#">convert_a_to_b</a>	34
8.2.4.3	<a href="#">convert_b_to_a</a>	34
8.2.4.4	<a href="#">get_a_to_b_offset</a>	34
8.2.4.5	<a href="#">initialize</a>	35
8.2.4.6	<a href="#">is_initialized</a>	35
8.2.4.7	<a href="#">operator=</a>	35
8.2.4.8	<a href="#">override_initialized</a>	35
8.2.4.9	<a href="#">reset_a_to_b_offset</a>	35
8.2.4.10	<a href="#">verify_setup</a>	36
8.2.4.11	<a href="#">verify_table_lookup_ends</a>	36
8.2.5	<a href="#">Friends And Related Function Documentation</a>	36
8.2.5.1	<a href="#">init_attrjeod__TimeConverter</a>	36
8.2.5.2	<a href="#">InputProcessor</a>	36
8.2.5.3	<a href="#">JeodBaseTime</a>	36
8.2.6	<a href="#">Field Documentation</a>	36
8.2.6.1	<a href="#">a_name</a>	36
8.2.6.2	<a href="#">a_to_b_offset</a>	37
8.2.6.3	<a href="#">b_name</a>	37
8.2.6.4	<a href="#">initialized</a>	37
8.2.6.5	<a href="#">valid_directions</a>	38
8.3	<a href="#">jeod::TimeConverter_Dyn_TAI Class Reference</a>	38
8.3.1	<a href="#">Detailed Description</a>	39
8.3.2	<a href="#">Constructor &amp; Destructor Documentation</a>	39
8.3.2.1	<a href="#">TimeConverter_Dyn_TAI</a>	39
8.3.2.2	<a href="#">~TimeConverter_Dyn_TAI</a>	39
8.3.2.3	<a href="#">TimeConverter_Dyn_TAI</a>	39
8.3.3	<a href="#">Member Function Documentation</a>	39
8.3.3.1	<a href="#">convert_a_to_b</a>	39
8.3.3.2	<a href="#">initialize</a>	39
8.3.3.3	<a href="#">operator=</a>	40

8.3.4	Friends And Related Function Documentation . . . . .	40
8.3.4.1	init_attrjeod__TimeConverter_Dyn_TAI . . . . .	40
8.3.4.2	InputProcessor . . . . .	40
8.3.5	Field Documentation . . . . .	40
8.3.5.1	dyn_ptr . . . . .	40
8.3.5.2	tai_ptr . . . . .	40
8.4	jeod::TimeConverter_Dyn_TDB Class Reference . . . . .	40
8.4.1	Detailed Description . . . . .	41
8.4.2	Constructor & Destructor Documentation . . . . .	41
8.4.2.1	TimeConverter_Dyn_TDB . . . . .	41
8.4.2.2	~TimeConverter_Dyn_TDB . . . . .	42
8.4.2.3	TimeConverter_Dyn_TDB . . . . .	42
8.4.3	Member Function Documentation . . . . .	42
8.4.3.1	convert_a_to_b . . . . .	42
8.4.3.2	initialize . . . . .	42
8.4.3.3	operator= . . . . .	42
8.4.4	Friends And Related Function Documentation . . . . .	42
8.4.4.1	init_attrjeod__TimeConverter_Dyn_TDB . . . . .	42
8.4.4.2	InputProcessor . . . . .	42
8.4.5	Field Documentation . . . . .	42
8.4.5.1	dyn_ptr . . . . .	42
8.4.5.2	tdb_ptr . . . . .	43
8.5	jeod::TimeConverter_Dyn_UDE Class Reference . . . . .	43
8.5.1	Detailed Description . . . . .	44
8.5.2	Constructor & Destructor Documentation . . . . .	44
8.5.2.1	TimeConverter_Dyn_UDE . . . . .	44
8.5.2.2	~TimeConverter_Dyn_UDE . . . . .	44
8.5.2.3	TimeConverter_Dyn_UDE . . . . .	44
8.5.3	Member Function Documentation . . . . .	44
8.5.3.1	convert_a_to_b . . . . .	44
8.5.3.2	initialize . . . . .	45
8.5.3.3	operator= . . . . .	45
8.5.3.4	reset_a_to_b_offset . . . . .	45
8.5.4	Friends And Related Function Documentation . . . . .	45
8.5.4.1	init_attrjeod__TimeConverter_Dyn_UDE . . . . .	45
8.5.4.2	InputProcessor . . . . .	45
8.5.5	Field Documentation . . . . .	45
8.5.5.1	dyn_ptr . . . . .	45
8.5.5.2	ude_ptr . . . . .	45
8.6	jeod::TimeConverter_STD_UDE Class Reference . . . . .	46

8.6.1	Detailed Description	47
8.6.2	Constructor & Destructor Documentation	47
8.6.2.1	TimeConverter_STD_UDE	47
8.6.2.2	~TimeConverter_STD_UDE	47
8.6.2.3	TimeConverter_STD_UDE	47
8.6.3	Member Function Documentation	47
8.6.3.1	convert_a_to_b	47
8.6.3.2	convert_b_to_a	47
8.6.3.3	initialize	48
8.6.3.4	operator=	48
8.6.3.5	reset_a_to_b_offset	48
8.6.4	Friends And Related Function Documentation	48
8.6.4.1	init_attrjeod__TimeConverter_STD_UDE	48
8.6.4.2	InputProcessor	48
8.6.5	Field Documentation	48
8.6.5.1	failed_null_test	48
8.6.5.2	std_ptr	49
8.6.5.3	ude_ptr	49
8.7	jeod::TimeConverter_TAI_GPS Class Reference	49
8.7.1	Detailed Description	50
8.7.2	Constructor & Destructor Documentation	50
8.7.2.1	TimeConverter_TAI_GPS	50
8.7.2.2	~TimeConverter_TAI_GPS	50
8.7.2.3	TimeConverter_TAI_GPS	50
8.7.3	Member Function Documentation	50
8.7.3.1	convert_a_to_b	50
8.7.3.2	convert_b_to_a	51
8.7.3.3	initialize	51
8.7.3.4	operator=	51
8.7.4	Friends And Related Function Documentation	51
8.7.4.1	init_attrjeod__TimeConverter_TAI_GPS	51
8.7.4.2	InputProcessor	51
8.7.5	Field Documentation	51
8.7.5.1	gps_ptr	51
8.7.5.2	tai_ptr	51
8.8	jeod::TimeConverter_TAI_TDB Class Reference	52
8.8.1	Detailed Description	53
8.8.2	Constructor & Destructor Documentation	53
8.8.2.1	TimeConverter_TAI_TDB	53
8.8.2.2	~TimeConverter_TAI_TDB	53

8.8.2.3	TimeConverter_TAI_TDB	53
8.8.3	Member Function Documentation	53
8.8.3.1	convert_a_to_b	53
8.8.3.2	convert_b_to_a	54
8.8.3.3	initialize	54
8.8.3.4	operator=	54
8.8.3.5	set_a_to_b_offset	54
8.8.4	Friends And Related Function Documentation	54
8.8.4.1	init_attrjeod__TimeConverter_TAI_TDB	54
8.8.4.2	InputProcessor	54
8.8.5	Field Documentation	54
8.8.5.1	a_to_b_offset	54
8.8.5.2	a_to_b_offset_epoch	55
8.8.5.3	nIter	55
8.8.5.4	nSteps	55
8.8.5.5	prev_tai_seconds	55
8.8.5.6	prev_tdb_seconds	55
8.8.5.7	tai_ptr	55
8.8.5.8	TAI_to_TT_offset	56
8.8.5.9	tdb_ptr	56
8.9	jeod::TimeConverter_TAI_TT Class Reference	56
8.9.1	Detailed Description	57
8.9.2	Constructor & Destructor Documentation	57
8.9.2.1	TimeConverter_TAI_TT	57
8.9.2.2	~TimeConverter_TAI_TT	57
8.9.2.3	TimeConverter_TAI_TT	57
8.9.3	Member Function Documentation	57
8.9.3.1	convert_a_to_b	57
8.9.3.2	convert_b_to_a	57
8.9.3.3	initialize	58
8.9.3.4	operator=	58
8.9.4	Friends And Related Function Documentation	58
8.9.4.1	init_attrjeod__TimeConverter_TAI_TT	58
8.9.4.2	InputProcessor	58
8.9.5	Field Documentation	58
8.9.5.1	tai_ptr	58
8.9.5.2	tt_ptr	58
8.10	jeod::TimeConverter_TAI_UT1 Class Reference	59
8.10.1	Detailed Description	60
8.10.2	Constructor & Destructor Documentation	60



8.10.2.1	<a href="#">TimeConverter_TAI_UT1</a>	60
8.10.2.2	<a href="#">~TimeConverter_TAI_UT1</a>	60
8.10.2.3	<a href="#">TimeConverter_TAI_UT1</a>	61
8.10.3	<a href="#">Member Function Documentation</a>	61
8.10.3.1	<a href="#">convert_a_to_b</a>	61
8.10.3.2	<a href="#">convert_b_to_a</a>	61
8.10.3.3	<a href="#">initialize</a>	61
8.10.3.4	<a href="#">initialize_tai_to_ut1</a>	62
8.10.3.5	<a href="#">operator=</a>	62
8.10.3.6	<a href="#">verify_table_lookup_ends</a>	62
8.10.4	<a href="#">Friends And Related Function Documentation</a>	62
8.10.4.1	<a href="#">init_attrjeod__TimeConverter_TAI_UT1</a>	62
8.10.4.2	<a href="#">InputProcessor</a>	62
8.10.5	<a href="#">Field Documentation</a>	62
8.10.5.1	<a href="#">gradient</a>	62
8.10.5.2	<a href="#">index</a>	63
8.10.5.3	<a href="#">last_index</a>	63
8.10.5.4	<a href="#">next_value</a>	63
8.10.5.5	<a href="#">next_when</a>	63
8.10.5.6	<a href="#">off_table_end</a>	63
8.10.5.7	<a href="#">override_data_table</a>	63
8.10.5.8	<a href="#">prev_value</a>	64
8.10.5.9	<a href="#">prev_when</a>	64
8.10.5.10	<a href="#">tai_ptr</a>	64
8.10.5.11	<a href="#">tai_to_ut1_override_val</a>	64
8.10.5.12	<a href="#">ut1_ptr</a>	64
8.10.5.13	<a href="#">val_vec</a>	64
8.10.5.14	<a href="#">when_vec</a>	65
8.11	<a href="#">jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data Class Reference</a>	65
8.11.1	<a href="#">Detailed Description</a>	65
8.11.2	<a href="#">Member Function Documentation</a>	65
8.11.2.1	<a href="#">initialize</a>	65
8.12	<a href="#">jeod::TimeConverter_TAI_UTC Class Reference</a>	65
8.12.1	<a href="#">Detailed Description</a>	67
8.12.2	<a href="#">Constructor &amp; Destructor Documentation</a>	67
8.12.2.1	<a href="#">TimeConverter_TAI_UTC</a>	67
8.12.2.2	<a href="#">~TimeConverter_TAI_UTC</a>	67
8.12.2.3	<a href="#">TimeConverter_TAI_UTC</a>	67
8.12.3	<a href="#">Member Function Documentation</a>	67
8.12.3.1	<a href="#">convert_a_to_b</a>	67

8.12.3.2	<a href="#">convert_b_to_a</a>	68
8.12.3.3	<a href="#">initialize</a>	68
8.12.3.4	<a href="#">initialize_leap_second</a>	68
8.12.3.5	<a href="#">operator=</a>	68
8.12.3.6	<a href="#">verify_table_lookup_ends</a>	69
8.12.4	<a href="#">Friends And Related Function Documentation</a>	69
8.12.4.1	<a href="#">init_attrjeod__TimeConverter_TAI.UTC</a>	69
8.12.4.2	<a href="#">InputProcessor</a>	69
8.12.5	<a href="#">Field Documentation</a>	69
8.12.5.1	<a href="#">index</a>	69
8.12.5.2	<a href="#">last_index</a>	69
8.12.5.3	<a href="#">leap_sec_override_val</a>	69
8.12.5.4	<a href="#">next_when</a>	69
8.12.5.5	<a href="#">off_table_end</a>	70
8.12.5.6	<a href="#">override_data_table</a>	70
8.12.5.7	<a href="#">prev_when</a>	70
8.12.5.8	<a href="#">tai_ptr</a>	70
8.12.5.9	<a href="#">utc_ptr</a>	70
8.12.5.10	<a href="#">val_vec</a>	71
8.12.5.11	<a href="#">when_vec</a>	71
8.13	<a href="#">jeod::TimeConverter_TAI.UTC_tai_to_utc_default_data Class Reference</a>	71
8.13.1	<a href="#">Detailed Description</a>	71
8.13.2	<a href="#">Member Function Documentation</a>	71
8.13.2.1	<a href="#">initialize</a>	71
8.14	<a href="#">jeod::TimeConverter_UT1_GMST Class Reference</a>	72
8.14.1	<a href="#">Detailed Description</a>	72
8.14.2	<a href="#">Constructor &amp; Destructor Documentation</a>	73
8.14.2.1	<a href="#">TimeConverter_UT1_GMST</a>	73
8.14.2.2	<a href="#">~TimeConverter_UT1_GMST</a>	73
8.14.2.3	<a href="#">TimeConverter_UT1_GMST</a>	73
8.14.3	<a href="#">Member Function Documentation</a>	73
8.14.3.1	<a href="#">convert_a_to_b</a>	73
8.14.3.2	<a href="#">initialize</a>	73
8.14.3.3	<a href="#">operator=</a>	74
8.14.4	<a href="#">Friends And Related Function Documentation</a>	74
8.14.4.1	<a href="#">init_attrjeod__TimeConverter_UT1_GMST</a>	74
8.14.4.2	<a href="#">InputProcessor</a>	74
8.14.5	<a href="#">Field Documentation</a>	74
8.14.5.1	<a href="#">gmst_ptr</a>	74
8.14.5.2	<a href="#">ut1_ptr</a>	74

8.15	<a href="#">jeod::TimeDyn Class Reference</a>	74
8.15.1	<a href="#">Detailed Description</a>	75
8.15.2	<a href="#">Constructor &amp; Destructor Documentation</a>	75
8.15.2.1	<a href="#">TimeDyn</a>	75
8.15.2.2	<a href="#">~TimeDyn</a>	75
8.15.2.3	<a href="#">TimeDyn</a>	76
8.15.3	<a href="#">Member Function Documentation</a>	76
8.15.3.1	<a href="#">initialize_initializer_time</a>	76
8.15.3.2	<a href="#">operator=</a>	76
8.15.3.3	<a href="#">update</a>	76
8.15.3.4	<a href="#">update_offset</a>	76
8.15.4	<a href="#">Friends And Related Function Documentation</a>	77
8.15.4.1	<a href="#">init_attrjeod__TimeDyn</a>	77
8.15.4.2	<a href="#">InputProcessor</a>	77
8.15.5	<a href="#">Field Documentation</a>	77
8.15.5.1	<a href="#">offset</a>	77
8.15.5.2	<a href="#">ref_scale</a>	77
8.15.5.3	<a href="#">scale_factor</a>	77
8.16	<a href="#">jeod::TimeEnum Class Reference</a>	77
8.16.1	<a href="#">Detailed Description</a>	78
8.16.2	<a href="#">Member Enumeration Documentation</a>	78
8.16.2.1	<a href="#">TimeFormat</a>	78
8.17	<a href="#">jeod::TimeGMST Class Reference</a>	78
8.17.1	<a href="#">Detailed Description</a>	79
8.17.2	<a href="#">Constructor &amp; Destructor Documentation</a>	79
8.17.2.1	<a href="#">TimeGMST</a>	79
8.17.2.2	<a href="#">~TimeGMST</a>	79
8.17.2.3	<a href="#">TimeGMST</a>	79
8.17.3	<a href="#">Member Function Documentation</a>	79
8.17.3.1	<a href="#">calculate_calendar_values</a>	79
8.17.3.2	<a href="#">operator=</a>	80
8.17.3.3	<a href="#">set_epoch</a>	80
8.17.3.4	<a href="#">set_time_by_trunc_julian</a>	80
8.17.4	<a href="#">Friends And Related Function Documentation</a>	80
8.17.4.1	<a href="#">init_attrjeod__TimeGMST</a>	80
8.17.4.2	<a href="#">InputProcessor</a>	80
8.18	<a href="#">jeod::TimeGPS Class Reference</a>	80
8.18.1	<a href="#">Detailed Description</a>	82
8.18.2	<a href="#">Constructor &amp; Destructor Documentation</a>	82
8.18.2.1	<a href="#">TimeGPS</a>	82

8.18.2.2	<a href="#">~TimeGPS</a>	82
8.18.2.3	<a href="#">TimeGPS</a>	82
8.18.3	<a href="#">Member Function Documentation</a>	82
8.18.3.1	<a href="#">calculate_calendar_values</a>	82
8.18.3.2	<a href="#">convert_from_calendar</a>	82
8.18.3.3	<a href="#">operator=</a>	83
8.18.3.4	<a href="#">set_epoch</a>	83
8.18.3.5	<a href="#">set_time_by_days</a>	83
8.18.3.6	<a href="#">set_time_by_seconds</a>	83
8.18.3.7	<a href="#">set_time_by_trunc_julian</a>	84
8.18.4	<a href="#">Friends And Related Function Documentation</a>	84
8.18.4.1	<a href="#">init_attrjeod__TimeGPS</a>	84
8.18.4.2	<a href="#">InputProcessor</a>	84
8.18.5	<a href="#">Field Documentation</a>	84
8.18.5.1	<a href="#">day_of_week</a>	84
8.18.5.2	<a href="#">rollover_count</a>	84
8.18.5.3	<a href="#">rollover_count_13_bit</a>	84
8.18.5.4	<a href="#">seconds_of_day</a>	85
8.18.5.5	<a href="#">seconds_of_week</a>	85
8.18.5.6	<a href="#">week</a>	85
8.18.5.7	<a href="#">week_13_bit</a>	85
8.19	<a href="#">jeod::TimeLinks Class Reference</a>	85
8.19.1	<a href="#">Detailed Description</a>	86
8.19.2	<a href="#">Constructor &amp; Destructor Documentation</a>	86
8.19.2.1	<a href="#">TimeLinks</a>	86
8.19.2.2	<a href="#">TimeLinks</a>	86
8.19.2.3	<a href="#">TimeLinks</a>	86
8.19.2.4	<a href="#">~TimeLinks</a>	86
8.19.3	<a href="#">Member Function Documentation</a>	86
8.19.3.1	<a href="#">operator=</a>	86
8.19.4	<a href="#">Friends And Related Function Documentation</a>	86
8.19.4.1	<a href="#">init_attrjeod__TimeLinks</a>	86
8.19.4.2	<a href="#">InputProcessor</a>	86
8.19.5	<a href="#">Field Documentation</a>	86
8.19.5.1	<a href="#">default_path_size</a>	86
8.20	<a href="#">jeod::TimeManager Class Reference</a>	87
8.20.1	<a href="#">Detailed Description</a>	88
8.20.2	<a href="#">Constructor &amp; Destructor Documentation</a>	88
8.20.2.1	<a href="#">TimeManager</a>	88
8.20.2.2	<a href="#">~TimeManager</a>	89

8.20.2.3	TimeManager	89
8.20.3	Member Function Documentation	89
8.20.3.1	get_converter_ptr	89
8.20.3.2	get_jeod_integration_time	89
8.20.3.3	get_time_change_flag	89
8.20.3.4	get_time_ptr	89
8.20.3.5	get_time_ptr	90
8.20.3.6	get_time_scale_factor	90
8.20.3.7	get_timestamp_time	90
8.20.3.8	initialize	90
8.20.3.9	operator=	91
8.20.3.10	register_converter	91
8.20.3.11	register_time	91
8.20.3.12	register_time_named	91
8.20.3.13	time_lookup	92
8.20.3.14	time_standards_exist	92
8.20.3.15	update	92
8.20.3.16	update_time	93
8.20.3.17	verify_table_lookup_ends	93
8.20.4	Friends And Related Function Documentation	94
8.20.4.1	init_attrjeod__TimeManager	94
8.20.4.2	InputProcessor	94
8.20.4.3	TimeManagerInit	94
8.20.5	Field Documentation	94
8.20.5.1	converter_vector	94
8.20.5.2	dyn_time	94
8.20.5.3	num_types	94
8.20.5.4	simtime	94
8.20.5.5	time_change_flag	95
8.20.5.6	time_vector	95
8.21	jeod::TimeManagerInit Class Reference	95
8.21.1	Detailed Description	97
8.21.2	Constructor & Destructor Documentation	97
8.21.2.1	TimeManagerInit	97
8.21.2.2	~TimeManagerInit	97
8.21.2.3	TimeManagerInit	97
8.21.3	Member Function Documentation	97
8.21.3.1	create_init_tree	97
8.21.3.2	create_update_tree	98
8.21.3.3	get_conv_dir_init	98

8.21.3.4	<a href="#">get_conv_dir_upd</a>	98
8.21.3.5	<a href="#">get_conv_ptr_index</a>	99
8.21.3.6	<a href="#">get_status</a>	99
8.21.3.7	<a href="#">increment_status</a>	99
8.21.3.8	<a href="#">initialize</a>	100
8.21.3.9	<a href="#">initialize_manager</a>	100
8.21.3.10	<a href="#">initialize_time_types</a>	100
8.21.3.11	<a href="#">operator=</a>	101
8.21.3.12	<a href="#">organize_update_list</a>	101
8.21.3.13	<a href="#">populate_converter_registry</a>	101
8.21.3.14	<a href="#">set_status</a>	101
8.21.3.15	<a href="#">verify_converter_setup</a>	101
8.21.3.16	<a href="#">verify_times_setup</a>	102
8.21.4	<a href="#">Friends And Related Function Documentation</a>	102
8.21.4.1	<a href="#">init_attrjeod__TimeManagerInit</a>	102
8.21.4.2	<a href="#">InputProcessor</a>	102
8.21.5	<a href="#">Field Documentation</a>	102
8.21.5.1	<a href="#">converter_ptrs_index</a>	102
8.21.5.2	<a href="#">dyn_time_index</a>	102
8.21.5.3	<a href="#">init_converter_dir_table</a>	103
8.21.5.4	<a href="#">initializer</a>	103
8.21.5.5	<a href="#">initializer_index</a>	103
8.21.5.6	<a href="#">num_added_pass</a>	103
8.21.5.7	<a href="#">num_added_total</a>	103
8.21.5.8	<a href="#">sim_start_format</a>	103
8.21.5.9	<a href="#">status</a>	104
8.21.5.10	<a href="#">time_manager</a>	104
8.21.5.11	<a href="#">update_converter_dir_table</a>	104
8.22	<a href="#">jeod::TimeMessages Class Reference</a>	104
8.22.1	<a href="#">Detailed Description</a>	105
8.22.2	<a href="#">Constructor &amp; Destructor Documentation</a>	105
8.22.2.1	<a href="#">TimeMessages</a>	105
8.22.2.2	<a href="#">TimeMessages</a>	105
8.22.3	<a href="#">Member Function Documentation</a>	105
8.22.3.1	<a href="#">operator=</a>	105
8.22.4	<a href="#">Friends And Related Function Documentation</a>	105
8.22.4.1	<a href="#">init_attrjeod__TimeMessages</a>	105
8.22.4.2	<a href="#">InputProcessor</a>	105
8.22.5	<a href="#">Field Documentation</a>	105
8.22.5.1	<a href="#">duplicate_methods</a>	105

8.22.5.2	<a href="#">extension_error</a>	106
8.22.5.3	<a href="#">incomplete_setup_error</a>	106
8.22.5.4	<a href="#">initialization_error</a>	106
8.22.5.5	<a href="#">invalid_data_error</a>	107
8.22.5.6	<a href="#">invalid_node</a>	107
8.22.5.7	<a href="#">invalid_setup_error</a>	107
8.22.5.8	<a href="#">memory_error</a>	108
8.22.5.9	<a href="#">redundancy_error</a>	108
8.23	<a href="#">jeod::TimeMET Class Reference</a>	108
8.23.1	<a href="#">Detailed Description</a>	109
8.23.2	<a href="#">Constructor &amp; Destructor Documentation</a>	109
8.23.2.1	<a href="#">TimeMET</a>	109
8.23.2.2	<a href="#">~TimeMET</a>	109
8.23.2.3	<a href="#">TimeMET</a>	110
8.23.3	<a href="#">Member Function Documentation</a>	110
8.23.3.1	<a href="#">operator=</a>	110
8.23.3.2	<a href="#">update</a>	110
8.23.4	<a href="#">Friends And Related Function Documentation</a>	110
8.23.4.1	<a href="#">init_attrjeod__TimeMET</a>	110
8.23.4.2	<a href="#">InputProcessor</a>	110
8.23.5	<a href="#">Field Documentation</a>	110
8.23.5.1	<a href="#">hold</a>	110
8.23.5.2	<a href="#">previous_hold</a>	110
8.24	<a href="#">jeod::TimeStandard Class Reference</a>	110
8.24.1	<a href="#">Detailed Description</a>	112
8.24.2	<a href="#">Constructor &amp; Destructor Documentation</a>	112
8.24.2.1	<a href="#">TimeStandard</a>	112
8.24.2.2	<a href="#">~TimeStandard</a>	113
8.24.2.3	<a href="#">TimeStandard</a>	113
8.24.3	<a href="#">Member Function Documentation</a>	113
8.24.3.1	<a href="#">add_type_initialize</a>	113
8.24.3.2	<a href="#">calculate_calendar_values</a>	113
8.24.3.3	<a href="#">calendar_update</a>	114
8.24.3.4	<a href="#">convert_from_calendar</a>	114
8.24.3.5	<a href="#">initialize_from_parent</a>	114
8.24.3.6	<a href="#">initialize_initializer_time</a>	115
8.24.3.7	<a href="#">julian_date_at_epoch</a>	115
8.24.3.8	<a href="#">operator=</a>	115
8.24.3.9	<a href="#">seconds_of_year</a>	115
8.24.3.10	<a href="#">set_epoch</a>	116

8.24.3.11	<a href="#">set_time_by_days</a>	116
8.24.3.12	<a href="#">set_time_by_seconds</a>	116
8.24.3.13	<a href="#">set_time_by_trunc_julian</a>	117
8.24.4	<a href="#">Friends And Related Function Documentation</a>	117
8.24.4.1	<a href="#">init_attrjeod__TimeStandard</a>	117
8.24.4.2	<a href="#">InputProcessor</a>	117
8.24.4.3	<a href="#">TimeUDE</a>	117
8.24.5	<a href="#">Field Documentation</a>	117
8.24.5.1	<a href="#">calendar_day</a>	117
8.24.5.2	<a href="#">calendar_hour</a>	117
8.24.5.3	<a href="#">calendar_minute</a>	118
8.24.5.4	<a href="#">calendar_month</a>	118
8.24.5.5	<a href="#">calendar_second</a>	118
8.24.5.6	<a href="#">calendar_year</a>	118
8.24.5.7	<a href="#">julian_date</a>	118
8.24.5.8	<a href="#">last_calendar_update</a>	119
8.24.5.9	<a href="#">prev_julian_day</a>	119
8.24.5.10	<a href="#">seconds_at_year_start</a>	119
8.24.5.11	<a href="#">send_warning_pre_1968</a>	119
8.24.5.12	<a href="#">tjt_at_epoch</a>	119
8.24.5.13	<a href="#">tjt_jd_offset</a>	119
8.24.5.14	<a href="#">tjt_mjt_offset</a>	120
8.24.5.15	<a href="#">trunc_julian_time</a>	120
8.24.5.16	<a href="#">year_of_last_soy</a>	120
8.25	<a href="#">jeod::TimeTAI Class Reference</a>	120
8.25.1	<a href="#">Detailed Description</a>	121
8.25.2	<a href="#">Constructor &amp; Destructor Documentation</a>	121
8.25.2.1	<a href="#">TimeTAI</a>	121
8.25.2.2	<a href="#">~TimeTAI</a>	121
8.25.2.3	<a href="#">TimeTAI</a>	121
8.25.3	<a href="#">Member Function Documentation</a>	121
8.25.3.1	<a href="#">operator=</a>	121
8.25.3.2	<a href="#">set_epoch</a>	121
8.25.4	<a href="#">Friends And Related Function Documentation</a>	122
8.25.4.1	<a href="#">init_attrjeod__TimeTAI</a>	122
8.25.4.2	<a href="#">InputProcessor</a>	122
8.26	<a href="#">jeod::TimeTDB Class Reference</a>	122
8.26.1	<a href="#">Detailed Description</a>	123
8.26.2	<a href="#">Constructor &amp; Destructor Documentation</a>	123
8.26.2.1	<a href="#">TimeTDB</a>	123



8.26.2.2	<a href="#">~TimeTDB</a>	123
8.26.2.3	<a href="#">TimeTDB</a>	123
8.26.3	<a href="#">Member Function Documentation</a>	123
8.26.3.1	<a href="#">operator=</a>	123
8.26.3.2	<a href="#">set_epoch</a>	123
8.26.4	<a href="#">Friends And Related Function Documentation</a>	123
8.26.4.1	<a href="#">init_attrjeod__TimeTDB</a>	123
8.26.4.2	<a href="#">InputProcessor</a>	123
8.27	<a href="#">jeod::TimeTT Class Reference</a>	124
8.27.1	<a href="#">Detailed Description</a>	124
8.27.2	<a href="#">Constructor &amp; Destructor Documentation</a>	124
8.27.2.1	<a href="#">TimeTT</a>	124
8.27.2.2	<a href="#">~TimeTT</a>	125
8.27.2.3	<a href="#">TimeTT</a>	125
8.27.3	<a href="#">Member Function Documentation</a>	125
8.27.3.1	<a href="#">operator=</a>	125
8.27.3.2	<a href="#">set_epoch</a>	125
8.27.4	<a href="#">Friends And Related Function Documentation</a>	125
8.27.4.1	<a href="#">init_attrjeod__TimeTT</a>	125
8.27.4.2	<a href="#">InputProcessor</a>	125
8.28	<a href="#">jeod::TimeUDE Class Reference</a>	125
8.28.1	<a href="#">Detailed Description</a>	128
8.28.2	<a href="#">Constructor &amp; Destructor Documentation</a>	128
8.28.2.1	<a href="#">TimeUDE</a>	128
8.28.2.2	<a href="#">~TimeUDE</a>	128
8.28.2.3	<a href="#">TimeUDE</a>	128
8.28.3	<a href="#">Member Function Documentation</a>	128
8.28.3.1	<a href="#">add_type_initialize</a>	128
8.28.3.2	<a href="#">clock_update</a>	129
8.28.3.3	<a href="#">convert_epoch_to_update</a>	129
8.28.3.4	<a href="#">initialize_from_parent</a>	129
8.28.3.5	<a href="#">initialize_initializer_time</a>	130
8.28.3.6	<a href="#">must_be_singleton</a>	130
8.28.3.7	<a href="#">operator=</a>	131
8.28.3.8	<a href="#">set_epoch_dyn</a>	131
8.28.3.9	<a href="#">set_epoch_initializing_value</a>	131
8.28.3.10	<a href="#">set_epoch_std</a>	131
8.28.3.11	<a href="#">set_epoch_times</a>	132
8.28.3.12	<a href="#">set_epoch_ude</a>	132
8.28.3.13	<a href="#">set_initial_times</a>	132

8.28.3.14	set_time_by_clock	133
8.28.3.15	set_time_by_days	133
8.28.3.16	set_time_by_seconds	133
8.28.3.17	verify_epoch	133
8.28.3.18	verify_init	134
8.28.3.19	verify_update	134
8.28.4	Friends And Related Function Documentation	134
8.28.4.1	init_attrjeod__TimeUDE	134
8.28.4.2	InputProcessor	134
8.28.5	Field Documentation	134
8.28.5.1	clock_day	134
8.28.5.2	clock_hour	134
8.28.5.3	clock_minute	135
8.28.5.4	clock_second	135
8.28.5.5	epoch_data_present	135
8.28.5.6	epoch_day	135
8.28.5.7	epoch_defined_in_name	135
8.28.5.8	epoch_format	135
8.28.5.9	epoch_hour	136
8.28.5.10	epoch_index	136
8.28.5.11	epoch_initializing_value	136
8.28.5.12	epoch_minute	136
8.28.5.13	epoch_month	136
8.28.5.14	epoch_second	136
8.28.5.15	epoch_value_is_set_calendar	137
8.28.5.16	epoch_value_is_set_clock	137
8.28.5.17	epoch_value_is_set_number	137
8.28.5.18	epoch_year	137
8.28.5.19	initial_value_format	137
8.28.5.20	initializing_data_present	137
8.28.5.21	last_clock_update	137
8.28.5.22	update_index	138
8.29	jeod::TimeUT1 Class Reference	138
8.29.1	Detailed Description	139
8.29.2	Constructor & Destructor Documentation	139
8.29.2.1	TimeUT1	139
8.29.2.2	~TimeUT1	139
8.29.2.3	TimeUT1	139
8.29.3	Member Function Documentation	139
8.29.3.1	get_days	139

8.29.3.2	operator=	139
8.29.3.3	set_epoch	139
8.29.4	Friends And Related Function Documentation	140
8.29.4.1	init_attrjeod__TimeUT1	140
8.29.4.2	InputProcessor	140
8.29.5	Field Documentation	140
8.29.5.1	true_ut1	140
8.30	jeod::TimeUTC Class Reference	140
8.30.1	Detailed Description	141
8.30.2	Constructor & Destructor Documentation	141
8.30.2.1	TimeUTC	141
8.30.2.2	~TimeUTC	141
8.30.2.3	TimeUTC	141
8.30.3	Member Function Documentation	141
8.30.3.1	operator=	141
8.30.3.2	set_epoch	141
8.30.4	Friends And Related Function Documentation	142
8.30.4.1	init_attrjeod__TimeUTC	142
8.30.4.2	InputProcessor	142
8.30.5	Field Documentation	142
8.30.5.1	true_utc	142
<b>9</b>	<b>File Documentation</b>	<b>143</b>
9.1	class_declarations.hh File Reference	143
9.1.1	Detailed Description	143
9.2	tai_to_ut1.cc File Reference	143
9.2.1	Macro Definition Documentation	143
9.2.1.1	JEOD_FRIEND_CLASS	143
9.3	tai_to_ut1.hh File Reference	144
9.4	tai_to_utc.cc File Reference	144
9.4.1	Macro Definition Documentation	144
9.4.1.1	JEOD_FRIEND_CLASS	144
9.5	tai_to_utc.hh File Reference	144
9.6	time.cc File Reference	145
9.6.1	Detailed Description	145
9.7	time.hh File Reference	145
9.7.1	Detailed Description	145
9.8	time__add_type_update.cc File Reference	146
9.8.1	Detailed Description	146
9.9	time_converter.cc File Reference	146

9.9.1 Detailed Description . . . . .	146
9.10 time_converter.hh File Reference . . . . .	147
9.10.1 Detailed Description . . . . .	147
9.11 time_converter_dyn_tai.cc File Reference . . . . .	147
9.11.1 Detailed Description . . . . .	148
9.12 time_converter_dyn_tai.hh File Reference . . . . .	148
9.12.1 Detailed Description . . . . .	148
9.13 time_converter_dyn_tdb.cc File Reference . . . . .	148
9.13.1 Detailed Description . . . . .	149
9.14 time_converter_dyn_tdb.hh File Reference . . . . .	149
9.14.1 Detailed Description . . . . .	149
9.15 time_converter_dyn_ude.cc File Reference . . . . .	149
9.15.1 Detailed Description . . . . .	150
9.16 time_converter_dyn_ude.hh File Reference . . . . .	150
9.16.1 Detailed Description . . . . .	150
9.17 time_converter_std_ude.cc File Reference . . . . .	150
9.17.1 Detailed Description . . . . .	151
9.18 time_converter_std_ude.hh File Reference . . . . .	151
9.18.1 Detailed Description . . . . .	151
9.19 time_converter_tai_gps.cc File Reference . . . . .	151
9.19.1 Detailed Description . . . . .	152
9.20 time_converter_tai_gps.hh File Reference . . . . .	152
9.20.1 Detailed Description . . . . .	152
9.21 time_converter_tai_tdb.cc File Reference . . . . .	152
9.21.1 Detailed Description . . . . .	153
9.22 time_converter_tai_tdb.hh File Reference . . . . .	153
9.22.1 Detailed Description . . . . .	153
9.23 time_converter_tai_tt.cc File Reference . . . . .	153
9.23.1 Detailed Description . . . . .	154
9.24 time_converter_tai_tt.hh File Reference . . . . .	154
9.24.1 Detailed Description . . . . .	154
9.25 time_converter_tai_ut1.cc File Reference . . . . .	154
9.25.1 Detailed Description . . . . .	155
9.26 time_converter_tai_ut1.hh File Reference . . . . .	155
9.26.1 Detailed Description . . . . .	155
9.27 time_converter_tai_utc.cc File Reference . . . . .	155
9.27.1 Detailed Description . . . . .	156
9.28 time_converter_tai_utc.hh File Reference . . . . .	156
9.28.1 Detailed Description . . . . .	156
9.29 time_converter_ut1_gmst.cc File Reference . . . . .	156

9.29.1 Detailed Description . . . . .	157
9.30 time_converter_ut1_gmst.hh File Reference . . . . .	157
9.30.1 Detailed Description . . . . .	157
9.31 time_dyn.cc File Reference . . . . .	157
9.31.1 Detailed Description . . . . .	158
9.32 time_dyn.hh File Reference . . . . .	158
9.32.1 Detailed Description . . . . .	158
9.33 time_enum.hh File Reference . . . . .	158
9.33.1 Detailed Description . . . . .	158
9.34 time_gmst.cc File Reference . . . . .	159
9.34.1 Detailed Description . . . . .	159
9.35 time_gmst.hh File Reference . . . . .	159
9.35.1 Detailed Description . . . . .	159
9.36 time_gps.cc File Reference . . . . .	159
9.36.1 Detailed Description . . . . .	160
9.37 time_gps.hh File Reference . . . . .	160
9.37.1 Detailed Description . . . . .	160
9.38 time_links.hh File Reference . . . . .	160
9.38.1 Detailed Description . . . . .	161
9.39 time_manager.cc File Reference . . . . .	161
9.39.1 Detailed Description . . . . .	161
9.40 time_manager.hh File Reference . . . . .	161
9.40.1 Detailed Description . . . . .	162
9.41 time_manager__initialize.cc File Reference . . . . .	162
9.41.1 Detailed Description . . . . .	162
9.42 time_manager_init.cc File Reference . . . . .	162
9.42.1 Detailed Description . . . . .	163
9.43 time_manager_init.hh File Reference . . . . .	163
9.43.1 Detailed Description . . . . .	163
9.44 time_messages.cc File Reference . . . . .	164
9.44.1 Detailed Description . . . . .	164
9.45 time_messages.hh File Reference . . . . .	164
9.45.1 Detailed Description . . . . .	164
9.46 time_met.cc File Reference . . . . .	164
9.46.1 Detailed Description . . . . .	165
9.47 time_met.hh File Reference . . . . .	165
9.47.1 Detailed Description . . . . .	165
9.48 time_standard.cc File Reference . . . . .	165
9.48.1 Detailed Description . . . . .	166
9.49 time_standard.hh File Reference . . . . .	166

9.49.1 Detailed Description . . . . .	166
9.50 time_tai.cc File Reference . . . . .	166
9.50.1 Detailed Description . . . . .	167
9.51 time_tai.hh File Reference . . . . .	167
9.51.1 Detailed Description . . . . .	167
9.52 time_tdb.cc File Reference . . . . .	167
9.52.1 Detailed Description . . . . .	168
9.53 time_tdb.hh File Reference . . . . .	168
9.53.1 Detailed Description . . . . .	168
9.54 time_tt.cc File Reference . . . . .	168
9.54.1 Detailed Description . . . . .	169
9.55 time_tt.hh File Reference . . . . .	169
9.55.1 Detailed Description . . . . .	169
9.56 time_ude.cc File Reference . . . . .	169
9.56.1 Detailed Description . . . . .	170
9.57 time_ude.hh File Reference . . . . .	170
9.57.1 Detailed Description . . . . .	170
9.58 time_ut1.cc File Reference . . . . .	170
9.58.1 Detailed Description . . . . .	171
9.59 time_ut1.hh File Reference . . . . .	171
9.59.1 Detailed Description . . . . .	171
9.60 time_utc.cc File Reference . . . . .	171
9.60.1 Detailed Description . . . . .	171
9.61 time_utc.hh File Reference . . . . .	172
9.61.1 Detailed Description . . . . .	172

# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Models . . . . .	13
Environment . . . . .	14
Time . . . . .	15





## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">jeod</a>	Namespace jeod . . . . .	19
----------------------	--------------------------	----



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

jeod::JeodBaseTime . . . . .	21
jeod::TimeDyn . . . . .	74
jeod::TimeStandard . . . . .	110
jeod::TimeGMST . . . . .	78
jeod::TimeGPS . . . . .	80
jeod::TimeTAI . . . . .	120
jeod::TimeTDB . . . . .	122
jeod::TimeTT . . . . .	124
jeod::TimeUT1 . . . . .	138
jeod::TimeUTC . . . . .	140
jeod::TimeUDE . . . . .	125
jeod::TimeMET . . . . .	108
JeodIntegrationTime	
jeod::TimeManager . . . . .	87
jeod::TimeConverter . . . . .	31
jeod::TimeConverter_Dyn_TAI . . . . .	38
jeod::TimeConverter_Dyn_TDB . . . . .	40
jeod::TimeConverter_Dyn_UDE . . . . .	43
jeod::TimeConverter_STD_UDE . . . . .	46
jeod::TimeConverter_TAI_GPS . . . . .	49
jeod::TimeConverter_TAI_TDB . . . . .	52
jeod::TimeConverter_TAI_TT . . . . .	56
jeod::TimeConverter_TAI_UT1 . . . . .	59
jeod::TimeConverter_TAI_UTC . . . . .	65
jeod::TimeConverter_UT1_GMST . . . . .	72
jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data . . . . .	65
jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data . . . . .	71
jeod::TimeEnum . . . . .	77
jeod::TimeManagerInit . . . . .	95
jeod::TimeMessages . . . . .	104
TreeLinks	
jeod::TimeLinks . . . . .	85



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::JeodBaseTime</a>	
<a href="#">JeodBaseTime</a> is an abstract class, containing the basic structure of all clocks that run in JEOD	21
<a href="#">jeod::TimeConverter</a>	
The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types	31
<a href="#">jeod::TimeConverter_Dyn_TAI</a>	
Define class <a href="#">TimeConverter_Dyn_TAI</a> , which converts from simulation dynamic time to International Atomic Time	38
<a href="#">jeod::TimeConverter_Dyn_TDB</a>	
Define class <a href="#">TimeConverter_Dyn_TDB</a> , which converts from simulation dynamic time to Barycentric Dynamic Time	40
<a href="#">jeod::TimeConverter_Dyn_UDE</a>	
Define class <a href="#">TimeConverter_Dyn_UDE</a> , which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time	43
<a href="#">jeod::TimeConverter_STD_UDE</a>	
Define class <a href="#">TimeConverter_STD_UDE</a> , which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time	46
<a href="#">jeod::TimeConverter_TAI_GPS</a>	
Define class <a href="#">TimeConverter_TAI_GPS</a> , which converts between International Atomic Time and the clock associated with the Global Positioning System	49
<a href="#">jeod::TimeConverter_TAI_TDB</a>	
Define class <a href="#">TimeConverter_TAI_TDB</a> , which converts from International Atomic Time to Barycentric Dynamic Time	52
<a href="#">jeod::TimeConverter_TAI_TT</a>	
Converts between International Atomic Time and Terrestrial Time	56
<a href="#">jeod::TimeConverter_TAI_UT1</a>	
Define class <a href="#">TimeConverter_TAI_UT1</a> , which converts between International Atomic Time and Universal Time	59
<a href="#">jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data</a>	65
<a href="#">jeod::TimeConverter_TAI_UTC</a>	
Converts between International Atomic Time and Coordinated Universal Time	65
<a href="#">jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data</a>	71
<a href="#">jeod::TimeConverter_UT1_GMST</a>	
Converts between Universal Time and Greenwich Mean Sidereal Time	72
<a href="#">jeod::TimeDyn</a>	
Represents the Dynamic Time in the simulation	74

<a href="#">jeod::TimeEnum</a>	Contains an enumeration of the formats in which time can be represented . . . . .	77
<a href="#">jeod::TimeGMST</a>	To represent the clock known as Greenwich Mean Sidereal Time . . . . .	78
<a href="#">jeod::TimeGPS</a>	To represent the time associated with the Global Positioning System . . . . .	80
<a href="#">jeod::TimeLinks</a>	. . . . .	85
<a href="#">jeod::TimeManager</a>	To manage the various time representations and the converters between them throughout the simulation . . . . .	87
<a href="#">jeod::TimeManagerInit</a>	To initialize the Time Manager . . . . .	95
<a href="#">jeod::TimeMessages</a>	Specify the message IDs used in the Time model . . . . .	104
<a href="#">jeod::TimeMET</a>	A type of UDE time that allows for deliberate holds, or pauses . . . . .	108
<a href="#">jeod::TimeStandard</a>	A class that serves as the base for all time representations that are well defined outside the simulation . . . . .	110
<a href="#">jeod::TimeTAI</a>	Represents International Atomic Time . . . . .	120
<a href="#">jeod::TimeTDB</a>	Represents Barycentric Dynamic Time . . . . .	122
<a href="#">jeod::TimeTT</a>	Represents Terrestrial Time . . . . .	124
<a href="#">jeod::TimeUDE</a>	Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition . . . . .	125
<a href="#">jeod::TimeUT1</a>	Represents Universal Time . . . . .	138
<a href="#">jeod::TimeUTC</a>	Represents Coordinated Universal Time . . . . .	140

## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">class_declarations.hh</a>	
Forward declaration of classes defined in <a href="#">time.hh</a>	143
<a href="#">tai_to_ut1.cc</a>	143
<a href="#">tai_to_ut1.hh</a>	144
<a href="#">tai_to_utc.cc</a>	144
<a href="#">tai_to_utc.hh</a>	144
<a href="#">time.cc</a>	
JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD	145
<a href="#">time.hh</a>	
JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD	145
<a href="#">time__add_type_update.cc</a>	
Define JeodBaseTime::add_type_update	146
<a href="#">time_converter.cc</a>	
An abstract class that defines the basic structure of all the methods used by the converter objects	146
<a href="#">time_converter.hh</a>	
The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types	147
<a href="#">time_converter_dyn_tai.cc</a>	
Converts between International Atomic Time and Dynamic Time	147
<a href="#">time_converter_dyn_tai.hh</a>	
Define class TimeConverter_Dyn_TAI, which converts from simulation dynamic time to International Atomic Time	148
<a href="#">time_converter_dyn_tdb.cc</a>	
Converts between Dynamic Time and Barycentric Dynamic Time	148
<a href="#">time_converter_dyn_tdb.hh</a>	
Define class TimeConverter_Dyn_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time	149
<a href="#">time_converter_dyn_ude.cc</a>	
Converts between Dynamic Time and a time with User-Defined-Epoch	149
<a href="#">time_converter_dyn_ude.hh</a>	
Define class TimeConverter_Dyn_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time	150
<a href="#">time_converter_std_ude.cc</a>	
Define member functions for class TimeConverter_STD_UDE	150
<a href="#">time_converter_std_ude.hh</a>	
Define class TimeConverter_STD_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time	151

<a href="#">time_converter_tai_gps.cc</a>	Converts between International Atomic Time and the clock associated with the Global Positioning System . . . . .	151
<a href="#">time_converter_tai_gps.hh</a>	Define class TimeConverter_TAI_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System . . . . .	152
<a href="#">time_converter_tai_tdb.cc</a>	Converts from International Atomic Time to Barycentric Dynamic Time . . . . .	152
<a href="#">time_converter_tai_tdb.hh</a>	Define class TimeConverter_TAI_TDB, which converts from International Atomic Time to Barycentric Dynamic Time . . . . .	153
<a href="#">time_converter_tai_tt.cc</a>	Converts between International Atomic Time and Terrestrial Time . . . . .	153
<a href="#">time_converter_tai_tt.hh</a>	Converts between International Atomic Time and Terrestrial Time . . . . .	154
<a href="#">time_converter_tai_ut1.cc</a>	Converts between International Atomic Time and Universal Time . . . . .	154
<a href="#">time_converter_tai_ut1.hh</a>	Define class TimeConverter_TAI_UT1, which converts between International Atomic Time and Universal Time . . . . .	155
<a href="#">time_converter_tai_utc.cc</a>	Converts between International Atomic Time and Coordinated Universal Time . . . . .	155
<a href="#">time_converter_tai_utc.hh</a>	Converts between International Atomic Time and Coordinated Universal Time . . . . .	156
<a href="#">time_converter_ut1_gmst.cc</a>	Define member functions for class TimeConverter_UT1_GMST . . . . .	156
<a href="#">time_converter_ut1_gmst.hh</a>	Converts between Universal Time and Greenwich Mean Sidereal Time . . . . .	157
<a href="#">time_dyn.cc</a>	Define member functions for Dynamic Time . . . . .	157
<a href="#">time_dyn.hh</a>	Represents the Dynamic Time in the simulation . . . . .	158
<a href="#">time_enum.hh</a>	Contains an enumeration of the formats in which time can be represented . . . . .	158
<a href="#">time_gmst.cc</a>	Define member functions for Greenwich Mean Sidereal Time . . . . .	159
<a href="#">time_gmst.hh</a>	To represent the clock known as Greenwich Mean Sidereal Time . . . . .	159
<a href="#">time_gps.cc</a>	Define member functions for the clock associated with the Global Positioning System . . . . .	159
<a href="#">time_gps.hh</a>	To represent the time associated with the Global Positioning System . . . . .	160
<a href="#">time_links.hh</a>	Define the class TimeLinks, which defines the hierarchy of JEOD time conversions . . . . .	160
<a href="#">time_manager.cc</a>	Define member functions for class TimeManager . . . . .	161
<a href="#">time_manager.hh</a>	To manage the various time representations and the converters between them throughout the simulation . . . . .	161
<a href="#">time_manager_initialize.cc</a>	Define TimeManager::initialize . . . . .	162
<a href="#">time_manager_init.cc</a>	Define member functions for the Time Manager Initialization . . . . .	162
<a href="#">time_manager_init.hh</a>	To initialize the Time Manager . . . . .	163
<a href="#">time_messages.cc</a>	Implement the class TimeMessages . . . . .	164



<a href="#">time_messages.hh</a>	Define the class TimeMessages, the class that specifies the message IDs used in the Time model . . . . .	164
<a href="#">time_met.cc</a>	Define member functions for Mission Elapsed Time . . . . .	164
<a href="#">time_met.hh</a>	A type of UDE time that allows for deliberate holds, or pauses . . . . .	165
<a href="#">time_standard.cc</a>	An abstract class, this defines the basic structure of member functions for all Standard Times . . . . .	165
<a href="#">time_standard.hh</a>	A class that serves as the base for all time representations that are well defined outside the simulation . . . . .	166
<a href="#">time_tai.cc</a>	Define member functions for International Atomic Time . . . . .	166
<a href="#">time_tai.hh</a>	Represents International Atomic Time . . . . .	167
<a href="#">time_tdb.cc</a>	Define member functions Barycentric Dynamic Time . . . . .	167
<a href="#">time_tdb.hh</a>	Represents Barycentric Dynamic Time . . . . .	168
<a href="#">time_tt.cc</a>	Define member functions for Terrestrial Time . . . . .	168
<a href="#">time_tt.hh</a>	Represents Terrestrial Time . . . . .	169
<a href="#">time_ude.cc</a>	Define member functions for those times with a User-Defined-Epoch . . . . .	169
<a href="#">time_ude.hh</a>	Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition . . . . .	170
<a href="#">time_ut1.cc</a>	Define member functions for Universal Time . . . . .	170
<a href="#">time_ut1.hh</a>	Represents Universal Time . . . . .	171
<a href="#">time_utc.cc</a>	Define member functions for Coordinated Universal Time . . . . .	171
<a href="#">time_utc.hh</a>	Represents Coordinated Universal Time . . . . .	172



## Chapter 6

# Module Documentation

### 6.1 Models

#### Modules

- [Environment](#)

#### 6.1.1 Detailed Description

## 6.2 Environment

### Modules

- [Time](#)

### 6.2.1 Detailed Description

## 6.3 Time

### Files

- file [class\\_declarations.hh](#)  
*Forward declaration of classes defined in [time.hh](#).*
- file [time.hh](#)  
*JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.*
- file [time\\_converter.hh](#)  
*The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.*
- file [time\\_converter\\_dyn\\_tai.hh](#)  
*Define class TimeConverter\_Dyn\_TAI, which converts from simulation dynamic time to International Atomic Time.*
- file [time\\_converter\\_dyn\\_tdb.hh](#)  
*Define class TimeConverter\_Dyn\_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.*
- file [time\\_converter\\_dyn\\_ude.hh](#)  
*Define class TimeConverter\_Dyn\_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.*
- file [time\\_converter\\_std\\_ude.hh](#)  
*Define class TimeConverter\_STD\_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.*
- file [time\\_converter\\_tai\\_gps.hh](#)  
*Define class TimeConverter\_TAI\_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.*
- file [time\\_converter\\_tai\\_tdb.hh](#)  
*Define class TimeConverter\_TAI\_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.*
- file [time\\_converter\\_tai\\_tt.hh](#)  
*Converts between International Atomic Time and Terrestrial Time.*
- file [time\\_converter\\_tai\\_ut1.hh](#)  
*Define class TimeConverter\_TAI\_UT1, which converts between International Atomic Time and Universal Time.*
- file [time\\_converter\\_tai\\_utc.hh](#)  
*Converts between International Atomic Time and Coordinated Universal Time.*
- file [time\\_converter\\_ut1\\_gmst.hh](#)  
*Converts between Universal Time and Greenwich Mean Sidereal Time.*
- file [time\\_dyn.hh](#)  
*Represents the Dynamic Time in the simulation.*
- file [time\\_enum.hh](#)  
*Contains an enumeration of the formats in which time can be represented.*
- file [time\\_gmst.hh](#)  
*To represent the clock known as Greenwich Mean Sidereal Time.*
- file [time\\_gps.hh](#)  
*To represent the time associated with the Global Positioning System.*
- file [time\\_links.hh](#)  
*Define the class TimeLinks, which defines the hierarchy of JEOD time conversions.*
- file [time\\_manager.hh](#)  
*To manage the various time representations and the converters between them throughout the simulation.*
- file [time\\_manager\\_init.hh](#)  
*To initialize the Time Manager.*
- file [time\\_messages.hh](#)  
*Define the class TimeMessages, the class that specifies the message IDs used in the Time model.*
- file [time\\_met.hh](#)

- A type of UDE time that allows for deliberate holds, or pauses.*

  - file [time\\_standard.hh](#)
- A class that serves as the base for all time representations that are well defined outside the simulation.*

  - file [time\\_tai.hh](#)
- Represents International Atomic Time.*

  - file [time\\_tdb.hh](#)
- Represents Barycentric Dynamic Time.*

  - file [time\\_tt.hh](#)
- Represents Terrestrial Time.*

  - file [time\\_ude.hh](#)
- Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.*

  - file [time\\_ut1.hh](#)
- Represents Universal Time.*

  - file [time\\_utc.hh](#)
- Represents Coordinated Universal Time.*

  - file [time.cc](#)
- JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.*

  - file [time\\_\\_add\\_type\\_update.cc](#)
- Define JeodBaseTime::add\_type\_update.*

  - file [time\\_converter.cc](#)
- An abstract class that defines the basic structure of all the methods used by the converter objects.*

  - file [time\\_converter\\_dyn\\_tai.cc](#)
- Converts between International Atomic Time and Dynamic Time.*

  - file [time\\_converter\\_dyn\\_tdb.cc](#)
- Converts between Dynamic Time and Barycentric Dynamic Time.*

  - file [time\\_converter\\_dyn\\_ude.cc](#)
- Converts between Dynamic Time and a time with User-Defined-Epoch.*

  - file [time\\_converter\\_std\\_ude.cc](#)
- Define member functions for class TimeConverter\_STD\_UDE.*

  - file [time\\_converter\\_tai\\_gps.cc](#)
- Converts between International Atomic Time and the clock associated with the Global Positioning System.*

  - file [time\\_converter\\_tai\\_tdb.cc](#)
- Converts from International Atomic Time to Barycentric Dynamic Time.*

  - file [time\\_converter\\_tai\\_tt.cc](#)
- Converts between International Atomic Time and Terrestrial Time.*

  - file [time\\_converter\\_tai\\_ut1.cc](#)
- Converts between International Atomic Time and Universal Time.*

  - file [time\\_converter\\_tai\\_utc.cc](#)
- Converts between International Atomic Time and Coordinated Universal Time.*

  - file [time\\_converter\\_ut1\\_gmst.cc](#)
- Define member functions for class TimeConverter\_UT1\_GMST.*

  - file [time\\_dyn.cc](#)
- Define member functions for Dynamic Time.*

  - file [time\\_gmst.cc](#)
- Define member functions for Greenwich Mean Sidereal Time.*

  - file [time\\_gps.cc](#)
- Define member functions for the clock associated with the Global Positioning System.*

  - file [time\\_manager.cc](#)
- Define member functions for class TimeManager.*

  - file [time\\_manager\\_\\_initialize.cc](#)

- *Define TimeManager::initialize.*
- file [time\\_manager\\_init.cc](#)  
*Define member functions for the Time Manager Initialization.*
- file [time\\_messages.cc](#)  
*Implement the class TimeMessages.*
- file [time\\_met.cc](#)  
*Define member functions for Mission Elapsed Time.*
- file [time\\_standard.cc](#)  
*An abstract class, this defines the basic structure of member functions for all Standard Times.*
- file [time\\_tai.cc](#)  
*Define member functions for International Atomic Time.*
- file [time\\_tdb.cc](#)  
*Define member functions Barycentric Dynamic Time.*
- file [time\\_tt.cc](#)  
*Define member functions for Terrestrial Time.*
- file [time\\_ude.cc](#)  
*Define member functions for those times with a User-Defined-Epoch.*
- file [time\\_ut1.cc](#)  
*Define member functions for Universal Time.*
- file [time\\_utc.cc](#)  
*Define member functions for Coordinated Universal Time.*

## Namespaces

- [jeod](#)  
*Namespace jeod.*

## Macros

- `#define PATH "environment/time/"`

### 6.3.1 Detailed Description

### 6.3.2 Macro Definition Documentation

#### 6.3.2.1 `#define PATH "environment/time/"`

Definition at line 38 of file `time_messages.cc`.





## Chapter 7

# Namespace Documentation

### 7.1 jeod Namespace Reference

Namespace jeod.

#### Data Structures

- class [TimeConverter\\_TAI\\_UT1\\_tai\\_to\\_ut1\\_default\\_data](#)
- class [TimeConverter\\_TAI\\_UTC\\_tai\\_to\\_utc\\_default\\_data](#)
- class [JeodBaseTime](#)  
*JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.*
- class [TimeConverter](#)  
*The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.*
- class [TimeConverter\\_Dyn\\_TAI](#)  
*Define class TimeConverter\_Dyn\_TAI, which converts from simulation dynamic time to International Atomic Time.*
- class [TimeConverter\\_Dyn\\_TDB](#)  
*Define class TimeConverter\_Dyn\_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.*
- class [TimeConverter\\_Dyn\\_UDE](#)  
*Define class TimeConverter\_Dyn\_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.*
- class [TimeConverter\\_STD\\_UDE](#)  
*Define class TimeConverter\_STD\_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.*
- class [TimeConverter\\_TAI\\_GPS](#)  
*Define class TimeConverter\_TAI\_GPS, which converts between International Atomic Time and the clock associated with the Global Positioning System.*
- class [TimeConverter\\_TAI\\_TDB](#)  
*Define class TimeConverter\_TAI\_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.*
- class [TimeConverter\\_TAI\\_TT](#)  
*Converts between International Atomic Time and Terrestrial Time.*
- class [TimeConverter\\_TAI\\_UT1](#)  
*Define class TimeConverter\_TAI\_UT1, which converts between International Atomic Time and Universal Time.*
- class [TimeConverter\\_TAI\\_UTC](#)  
*Converts between International Atomic Time and Coordinated Universal Time.*
- class [TimeConverter\\_UT1\\_GMST](#)  
*Converts between Universal Time and Greenwich Mean Sidereal Time.*
- class [TimeDyn](#)

- Represents the Dynamic Time in the simulation.*

  - class [TimeEnum](#)

*Contains an enumeration of the formats in which time can be represented.*
  - class [TimeGMST](#)

*To represent the clock known as Greenwich Mean Sidereal Time.*
  - class [TimeGPS](#)

*To represent the time associated with the Global Positioning System.*
  - class [TimeLinks](#)
  - class [TimeManager](#)

*To manage the various time representations and the converters between them throughout the simulation.*
  - class [TimeManagerInit](#)

*To initialize the Time Manager.*
  - class [TimeMessages](#)

*Specify the message IDs used in the Time model.*
  - class [TimeMET](#)

*A type of UDE time that allows for deliberate holds, or pauses.*
  - class [TimeStandard](#)

*A class that serves as the base for all time representations that are well defined outside the simulation.*
  - class [TimeTAI](#)

*Represents International Atomic Time.*
  - class [TimeTDB](#)

*Represents Barycentric Dynamic Time.*
  - class [TimeTT](#)

*Represents Terrestrial Time.*
  - class [TimeUDE](#)

*Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.*
  - class [TimeUT1](#)

*Represents Universal Time.*
  - class [TimeUTC](#)

*Represents Coordinated Universal Time.*

## Functions

- [TimeConverter::Direction operator|](#) ([TimeConverter::Direction](#) a, [TimeConverter::Direction](#) b)

*Bitwise or operator for combining multiple converter direction flags.*

### 7.1.1 Detailed Description

Namespace jeod. Construct a Time\_MET.

Namespace jeod

### 7.1.2 Function Documentation

#### 7.1.2.1 [TimeConverter::Direction jeod::operator|](#) ( [TimeConverter::Direction](#) a, [TimeConverter::Direction](#) b ) [inline]

Bitwise or operator for combining multiple converter direction flags.

Definition at line 175 of file time\_converter.hh.

## Chapter 8

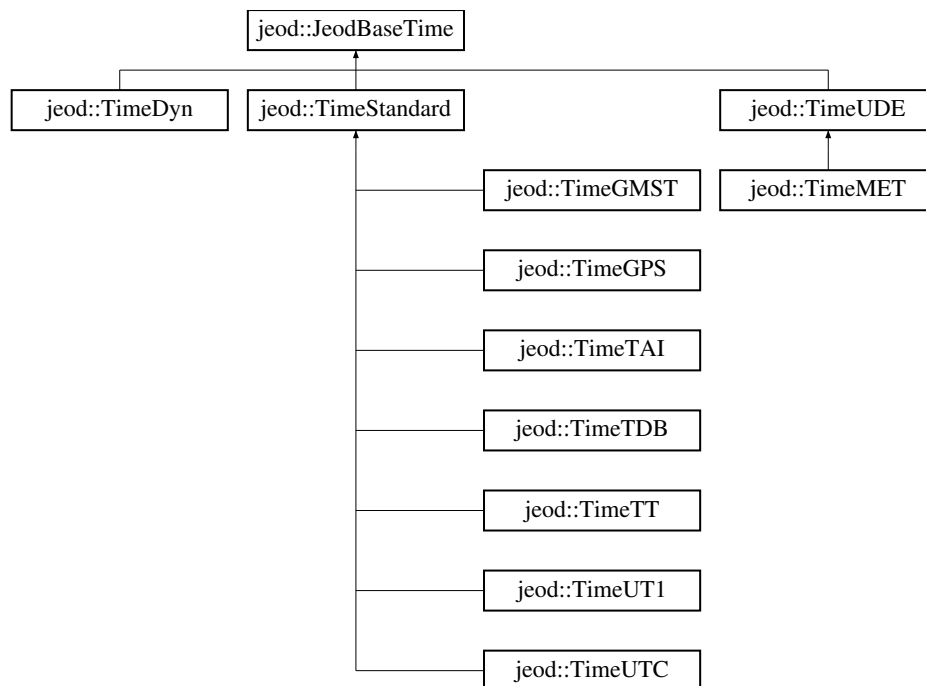
# Data Structure Documentation

### 8.1 jeod::JeodBaseTime Class Reference

[JeodBaseTime](#) is an abstract class, containing the basic structure of all clocks that run in JEOD.

```
#include <time.hh>
```

Inheritance diagram for jeod::JeodBaseTime:



#### Public Member Functions

- [JeodBaseTime](#) ()  
*Construct a [JeodBaseTime](#).*
- virtual [~JeodBaseTime](#) ()  
*Destroy a [JeodBaseTime](#).*
- virtual bool [must\\_be\\_singleton](#) (void)  
*Virtual function that indicates if class must be a singleton Defaults to yes.*
- virtual void [set\\_time\\_by\\_seconds](#) (const double new\_seconds)

- Given a value of seconds, propagate to days.*

  - virtual void `set_time_by_days` (const double new\_days)
- Given a value of days, propagate to seconds.*

  - void `add_type_update` (const int seeking\_status, `TimeManagerInit` \*tm\_init)

*Recursively adds elements to the update tree.*
- void `set_name` (std::string name\_in)

*Setter for the name.*
- void `set_index` (int idx)

*Setter for the index (force user to be carefule)*
- int `get_index` ()

*Getter for the index.*
- void `override_initialized` (bool init)

*Force reset the initialization status.*
- bool `is_initialized` ()

*Read the initialization status.*
- virtual void `initialize_initializer_time` (`TimeManagerInit` \*tm\_init)=0

*Initialize the time class that is used for initialization of the simulation.*
- virtual void `add_type_initialize` (const int seeking\_status, `TimeManagerInit` \*tm\_init)

*Default attempt to add a time-type to the initialization tree.*
- virtual void `initialize_from_parent` (`TimeManagerInit` \*tm\_init)

*Default attempt to initialize a time-type from its parent.*
- virtual void `update` (void)

*Updates each of the derived times from its parent time.*

## Data Fields

- double `initializing_value`

*Value used to define sim start time.*
- int `update_converter_direction`

*Determines which converter function (a\_to\_b (+1) or b\_to\_a (-1)) to use.*
- double `seconds`

*Elapsed time from epoch.*
- NamedItem `name`

*Name of time-type.*
- std::string `initialize_from_name`

*Name of time-type from which initial value is derived.*
- std::string `update_from_name`

*Name of time-type from which update values are derived.*
- `TimeManager` \* `time_manager`

*Pointer to the `TimeManager`.*
- `TimeConverter` \* `update_converter_ptr`

*Pointer to the converter class needed to update the time.*

## Protected Member Functions

- void `add_parent` (`JeodBaseTime` &parent)

*Link the argument time as the update source for this time.*

## Protected Attributes

- int [index](#)  
*Index-value of time-type in the registry.*
- bool [initialized](#)  
*Whether time has been initialized to a real time.*
- double [days](#)  
*Elapsed time from epoch.*
- double [initial\\_value](#)  
*Value of "seconds" at the start of the sim.*
- double [clock\\_resolution](#)  
*The resolution limit when generating clock and calendar-clock values.*
- [TimeLinks](#) [links](#)  
*Linkage to the hierarchy of time conversions.*

## Private Member Functions

- [JeodBaseTime](#) (const [JeodBaseTime](#) &)
- [JeodBaseTime](#) & [operator=](#) (const [JeodBaseTime](#) &)

## Friends

- class [InputProcessor](#)
- class [TimeConverter](#)
- class [TimeManagerInit](#)
- void [init\\_attrjeod\\_\\_JeodBaseTime](#) ()

### 8.1.1 Detailed Description

[JeodBaseTime](#) is an abstract class, containing the basic structure of all clocks that run in JEOD.

Definition at line 65 of file time.hh.

### 8.1.2 Constructor & Destructor Documentation

#### 8.1.2.1 `jeod::JeodBaseTime::JeodBaseTime ( void )`

Construct a [JeodBaseTime](#).

Definition at line 66 of file time.cc.

References [clock\\_resolution](#), [days](#), [index](#), [initial\\_value](#), [initialize\\_from\\_name](#), [initialized](#), [initializing\\_value](#), [seconds](#), [time\\_manager](#), [update\\_converter\\_direction](#), [update\\_converter\\_ptr](#), and [update\\_from\\_name](#).

#### 8.1.2.2 `jeod::JeodBaseTime::~~JeodBaseTime ( void )` [virtual]

Destroy a [JeodBaseTime](#).

Definition at line 221 of file time.cc.

References [links](#).

8.1.2.3 `jeod::JeodBaseTime::JeodBaseTime ( const JeodBaseTime & )` [private]

### 8.1.3 Member Function Documentation

8.1.3.1 `void jeod::JeodBaseTime::add_parent ( JeodBaseTime & parent )` [protected]

Link the argument time as the update source for this time.

#### Assumptions and Limitations

- The linkage tree is currently implemented as a runtime inspection tool, and does not augment time update functionality.

#### Parameters

in	<i>parent</i>	the time responsible for updating this time.
----	---------------	--

Definition at line 120 of file time.cc.

References links.

Referenced by `add_type_update()`.

8.1.3.2 `void jeod::JeodBaseTime::add_type_initialize ( const int seeking_status, TimeManagerInit * time_manager_init )` [virtual]

Default attempt to add a time-type to the initialization tree.

#### Assumptions and Limitations

- Fails for [TimeDyn](#), and has to be overwritten for others.

#### Parameters

in	<i>seeking_status</i>	status-value for auto-seek
in	<i>time_manager_init</i>	TM initializer

Reimplemented in [jeod::TimeUDE](#), and [jeod::TimeStandard](#).

Definition at line 99 of file time.cc.

References `jeod::TimeMessages::invalid_setup_error`.

8.1.3.3 `void jeod::JeodBaseTime::add_type_update ( const int seeking_status, TimeManagerInit * time_manager_init )`

Recursively adds elements to the update tree.

If the "parent" to a time-type is defined, adds the "parent" then returns to adding the "child" type. If the "parent" is not defined it searches for a suitable "parent" from the types already in the tree. If that search is successful, it adds the "child" to the tree, otherwise it returns without change.

#### Assumptions and Limitations

- None

## Parameters

in	<i>seeking_status</i>	status-value for auto-seek.
in	<i>time_manager_init</i>	The TM initializer.

Definition at line 78 of file `time__add_type_update.cc`.

References `add_parent()`, `add_type_update()`, `jeod::TimeManagerInit::get_conv_dir_upd()`, `jeod::TimeManagerInit::get_conv_ptr_index()`, `jeod::TimeManager::get_converter_ptr()`, `jeod::TimeManagerInit::get_status()`, `jeod::TimeManager::get_time_ptr()`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::TimeManagerInit::increment_status()`, `index`, `jeod::TimeConverter::initialize()`, `jeod::TimeMessages::invalid_node`, `jeod::TimeMessages::invalid_setup_error`, `jeod::TimeConverter::is_initialized()`, `links`, `jeod::TimeMessages::memory_error`, `name`, `jeod::TimeManagerInit::num_added_total`, `jeod::TimeManager::num_types`, `jeod::TimeManagerInit::set_status()`, `jeod::TimeManager::time_lookup()`, `time_manager`, `update_converter_direction`, `update_converter_ptr`, and `update_from_name`.

Referenced by `add_type_update()`.

### 8.1.3.4 int jeod::JeodBaseTime::get\_index ( ) [inline]

Getter for the index.

Definition at line 180 of file `time.hh`.

References `index`.

### 8.1.3.5 void jeod::JeodBaseTime::initialize\_from\_parent ( TimeManagerInit \* time\_manager\_init ) [virtual]

Default attempt to initialize a time-type from its parent.

#### Assumptions and Limitations

- Fails for [TimeDyn](#), and has to be overwritten for others.

## Parameters

in	<i>time_manager_init</i>	TM initializer
----	--------------------------	----------------

Reimplemented in [jeod::TimeUDE](#), and [jeod::TimeStandard](#).

Definition at line 133 of file `time.cc`.

References `jeod::TimeMessages::invalid_setup_error`.

Referenced by `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, and `jeod::TimeManagerInit::initialize_time_types()`.

### 8.1.3.6 virtual void jeod::JeodBaseTime::initialize\_initializer\_time ( TimeManagerInit \* tm\_init ) [pure virtual]

Initialize the time class that is used for initialization of the simulation.

## Parameters

<i>tm_init</i>	Time initializer.
----------------	-------------------

Implemented in [jeod::TimeUDE](#), [jeod::TimeStandard](#), and [jeod::TimeDyn](#).

### 8.1.3.7 bool jeod::JeodBaseTime::is\_initialized ( ) [inline]

Read the initialization status.

Definition at line 196 of file time.hh.

References initialized.

Referenced by `jeod::TimeConverter_TAI_UTC::initialize()`, `jeod::TimeStandard::initialize_from_parent()`, and `jeod::TimeUDE::initialize_from_parent()`.

**8.1.3.8** `bool jeod::JeodBaseTime::must_be_singleton ( void ) [virtual]`

Virtual function that indicates if class must be a singleton Defaults to yes.

Returns

Boolean value

Reimplemented in [jeod::TimeUDE](#).

Definition at line 151 of file time.cc.

**8.1.3.9** `JeodBaseTime& jeod::JeodBaseTime::operator= ( const JeodBaseTime & ) [private]`

**8.1.3.10** `void jeod::JeodBaseTime::override_initialized ( bool init ) [inline]`

Force reset the initialization status.

Definition at line 188 of file time.hh.

References initialized.

Referenced by `jeod::TimeUDE::convert_epoch_to_update()`, `jeod::TimeUDE::initialize_from_parent()`, and `jeod::TimeUDE::initialize_initializer_time()`.

**8.1.3.11** `void jeod::JeodBaseTime::set_index ( int idx ) [inline]`

Setter for the index (force user to be carefule)

Definition at line 172 of file time.hh.

References index.

Referenced by `jeod::TimeManager::register_time()`.

**8.1.3.12** `void jeod::JeodBaseTime::set_name ( std::string name_in ) [inline]`

Setter for the name.

Definition at line 164 of file time.hh.

**8.1.3.13** `void jeod::JeodBaseTime::set_time_by_days ( const double new_days ) [virtual]`

Given a value of days, propagate to seconds.

#### Assumptions and Limitations

- 86400 seconds = 1 day



## Parameters

in	<i>new_days</i>	new value for days Units: day
----	-----------------	----------------------------------

Reimplemented in [jeod::TimeUDE](#), [jeod::TimeStandard](#), and [jeod::TimeGPS](#).

Definition at line 210 of file time.cc.

References days, and seconds.

Referenced by [jeod::TimeUDE::set\\_epoch\\_dyn\(\)](#), [jeod::TimeStandard::set\\_time\\_by\\_days\(\)](#), and [jeod::TimeUDE::set\\_time\\_by\\_days\(\)](#).

#### 8.1.3.14 void jeod::JeodBaseTime::set\_time\_by\_seconds ( const double *new\_seconds* ) [virtual]

Given a value of seconds, propagate to days.

## Assumptions and Limitations

- 86400 seconds = 1 day

## Parameters

in	<i>new_seconds</i>	new value for seconds Units: s
----	--------------------	-----------------------------------

Reimplemented in [jeod::TimeUDE](#), [jeod::TimeStandard](#), and [jeod::TimeGPS](#).

Definition at line 194 of file time.cc.

References days, and seconds.

Referenced by [jeod::TimeUDE::initialize\\_from\\_parent\(\)](#), [jeod::TimeUDE::set\\_epoch\\_dyn\(\)](#), [jeod::TimeStandard::set\\_time\\_by\\_seconds\(\)](#), and [jeod::TimeUDE::set\\_time\\_by\\_seconds\(\)](#).

#### 8.1.3.15 void jeod::JeodBaseTime::update ( void ) [virtual]

Updates each of the derived times from its parent time.

## Assumptions and Limitations

- All but [TimeDyn](#) must have a parent; this should be defined by the user, or if not, already determined when the `update_tree` was built

Reimplemented in [jeod::TimeDyn](#), and [jeod::TimeMET](#).

Definition at line 166 of file time.cc.

References [jeod::TimeConverter::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter::convert\\_b\\_to\\_a\(\)](#), [jeod::TimeMessages::memory\\_error](#), `name`, `update_converter_direction`, and `update_converter_ptr`.

Referenced by [jeod::TimeMET::update\(\)](#).

## 8.1.4 Friends And Related Function Documentation

### 8.1.4.1 void init\_attrjeod\_\_JeodBaseTime ( ) [friend]

### 8.1.4.2 friend class InputProcessor [friend]

Definition at line 67 of file time.hh.

#### 8.1.4.3 friend class TimeConverter [friend]

Definition at line 69 of file time.hh.

#### 8.1.4.4 friend class TimeManagerInit [friend]

Definition at line 70 of file time.hh.

### 8.1.5 Field Documentation

#### 8.1.5.1 double jeod::JeodBaseTime::clock\_resolution [protected]

The resolution limit when generating clock and calendar-clock values.

Used for forcing a "tick-over" to the next minute if seconds comes within this value of 60.trick\_units(s)

Definition at line 138 of file time.hh.

Referenced by jeod::TimeStandard::calculate\_calendar\_values(), jeod::TimeUDE::clock\_update(), and JeodBaseTime().

#### 8.1.5.2 double jeod::JeodBaseTime::days [protected]

Elapsed time from epoch.

trick\_units(day)

Definition at line 126 of file time.hh.

Referenced by jeod::TimeStandard::convert\_from\_calendar(), jeod::TimeUT1::get\_days(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeUDE::initialize\_initializer\_time(), JeodBaseTime(), jeod::TimeStandard::seconds\_of\_year(), jeod::TimeUDE::set\_initial\_times(), jeod::TimeUDE::set\_time\_by\_clock(), set\_time\_by\_days(), jeod::TimeStandard::set\_time\_by\_days(), jeod::TimeGPS::set\_time\_by\_seconds(), set\_time\_by\_seconds(), jeod::TimeStandard::set\_time\_by\_seconds(), and jeod::TimeStandard::set\_time\_by\_trunc\_julian().

#### 8.1.5.3 int jeod::JeodBaseTime::index [protected]

Index-value of time-type in the registry.

trick\_units(-)

Definition at line 118 of file time.hh.

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeUDE::add\_type\_initialize(), add\_type\_update(), get\_index(), jeod::TimeManagerInit::initialize(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeUDE::initialize\_initializer\_time(), JeodBaseTime(), set\_index(), and jeod::TimeUDE::verify\_epoch().

#### 8.1.5.4 double jeod::JeodBaseTime::initial\_value [protected]

Value of "seconds" at the start of the sim.

trick\_units(s)

Definition at line 131 of file time.hh.

Referenced by jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeStandard::initialize\_initializer\_time(), and JeodBaseTime().

**8.1.5.5** `std::string jeod::JeodBaseTime::initialize_from_name`

Name of time-type from which initial value is derived.

`trick_units(-)`

Definition at line 98 of file `time.hh`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeStandard::initialize_initializer_time()`, `JeodBaseTime()`, and `jeod::TimeUDE::verify_init()`.

**8.1.5.6** `bool jeod::JeodBaseTime::initialized` `[protected]`

Whether time has been initialized to a real time.

`trick_units(-)`

Definition at line 122 of file `time.hh`.

Referenced by `jeod::TimeManagerInit::initialize()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, `jeod::TimeDyn::initialize_initializer_time()`, `jeod::TimeStandard::initialize_initializer_time()`, `jeod::TimeUDE::initialize_initializer_time()`, `jeod::TimeManagerInit::initialize_time_types()`, `is_initialized()`, `JeodBaseTime()`, `override_initialized()`, and `jeod::TimeConverter::verify_setup()`.

**8.1.5.7** `double jeod::JeodBaseTime::initializing_value`

Value used to define sim start time.

`trick_units(-)`

Definition at line 78 of file `time.hh`.

Referenced by `jeod::TimeStandard::initialize_initializer_time()`, `JeodBaseTime()`, and `jeod::TimeUDE::set_initial_times()`.

**8.1.5.8** `TimeLinks jeod::JeodBaseTime::links` `[protected]`

Linkage to the hierarchy of time conversions.

Provides accessors to parent, siblings and children `trick_units(-)`

Definition at line 144 of file `time.hh`.

Referenced by `add_parent()`, `add_type_update()`, `jeod::TimeDyn::TimeDyn()`, and `~JeodBaseTime()`.

**8.1.5.9** `NamedItem jeod::JeodBaseTime::name`

Name of time-type.

`trick_units(-)`

Definition at line 93 of file `time.hh`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, `add_type_update()`, `jeod::TimeUDE::convert_epoch_to_update()`, `jeod::TimeConverter_Dyn_TAI::initialize()`, `jeod::TimeConverter_Dyn_UDE::initialize()`, `jeod::TimeConverter_STD_UDE::initialize()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, `jeod::TimeStandard::initialize_initializer_time()`, `jeod::TimeUDE::initialize_initializer_time()`, `jeod::TimeManager::register_time()`, `jeod::TimeManager::register_time_named()`, `jeod::TimeUDE::set_epoch_dyn()`, `jeod::TimeUDE::set_epoch_initializing_value()`, `jeod::TimeUDE::set_epoch_std()`, `jeod::TimeUDE::set_epoch_ude()`, `jeod::TimeUDE::set_initial_times()`, `jeod::TimeDyn::TimeDyn()`, `jeod::TimeGMST::TimeGMST()`, `jeod::TimeGPS::TimeGPS()`, `jeod::TimeMET::TimeMET()`, `jeod::TimeTAI::TimeTAI()`, `jeod::TimeTDB::TimeTDB()`, `jeod::TimeTT::TimeTT()`, `jeod::TimeUT1::TimeUT1()`, `jeod::TimeUTC::TimeUTC()`, `update()`, `jeod::TimeUDE::verify_epoch()`, `jeod::TimeUDE::verify_init()`, and `jeod::TimeConverter::verify_setup()`.

#### 8.1.5.10 double jeod::JeodBaseTime::seconds

Elapsed time from epoch.

trick\_units(s)

Definition at line 88 of file time.hh.

Referenced by jeod::TimeUDE::clock\_update(), jeod::TimeConverter\_TAI\_TT::convert\_a\_to\_b(), jeod::TimeConverter\_Dyn\_TDB::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_GPS::convert\_a\_to\_b(), jeod::TimeConverter\_Dyn\_TAI::convert\_a\_to\_b(), jeod::TimeConverter\_Dyn\_UDE::convert\_a\_to\_b(), jeod::TimeConverter\_STD\_UDE::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_TDB::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_TT::convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_GPS::convert\_b\_to\_a(), jeod::TimeConverter\_STD\_UDE::convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_TDB::convert\_b\_to\_a(), jeod::TimeStandard::convert\_from\_calendar(), jeod::TimeManager::get\_timestamp\_time(), jeod::TimeConverter\_Dyn\_TDB::initialize(), jeod::TimeConverter\_Dyn\_TAI::initialize(), jeod::TimeConverter\_Dyn\_UDE::initialize(), jeod::TimeConverter\_STD\_UDE::initialize(), jeod::TimeManagerInit::initialize(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeDyn::initialize\_initializer\_time(), jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeUDE::initialize\_initializer\_time(), JeodBaseTime(), jeod::TimeConverter\_Dyn\_UDE::reset\_a\_to\_b\_offset(), jeod::TimeConverter\_STD\_UDE::reset\_a\_to\_b\_offset(), jeod::TimeStandard::seconds\_of\_year(), jeod::TimeUDE::set\_initial\_times(), jeod::TimeUDE::set\_time\_by\_clock(), set\_time\_by\_days(), set\_time\_by\_seconds(), jeod::TimeGPS::set\_time\_by\_trunc\_julian(), jeod::TimeStandard::set\_time\_by\_trunc\_julian(), jeod::TimeDyn::update(), and jeod::TimeDyn::update\_offset().

#### 8.1.5.11 TimeManager\* jeod::JeodBaseTime::time\_manager

Pointer to the [TimeManager](#).

trick\_units(-)

Definition at line 108 of file time.hh.

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeUDE::add\_type\_initialize(), add\_type\_update(), jeod::TimeStandard::calendar\_update(), jeod::TimeConverter\_TAI\_UTC::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_UTC::convert\_b\_to\_a(), jeod::TimeUDE::convert\_epoch\_to\_update(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeDyn::initialize\_initializer\_time(), jeod::TimeUDE::initialize\_initializer\_time(), jeod::TimeConverter\_TAI\_UTC::initialize\_leap\_second(), jeod::TimeConverter\_TAI\_UT1::initialize\_tai\_to\_ut1(), JeodBaseTime(), jeod::TimeManager::register\_time(), jeod::TimeStandard::seconds\_of\_year(), jeod::TimeDyn::update(), jeod::TimeDyn::update\_offset(), jeod::TimeUDE::verify\_epoch(), jeod::TimeUDE::verify\_init(), jeod::TimeConverter\_TAI\_UTC::verify\_table\_lookup\_ends(), jeod::TimeConverter\_TAI\_UT1::verify\_table\_lookup\_ends(), and jeod::TimeUDE::verify\_update().

#### 8.1.5.12 int jeod::JeodBaseTime::update\_converter\_direction

Determines which converter function (a\_to\_b (+1) or b\_to\_a (-1)) to use.

trick\_units(-)

Definition at line 83 of file time.hh.

Referenced by add\_type\_update(), JeodBaseTime(), and update().

#### 8.1.5.13 TimeConverter\* jeod::JeodBaseTime::update\_converter\_ptr

Pointer to the converter class needed to update the time.

trick\_units(-)

Definition at line 112 of file time.hh.

Referenced by add\_type\_update(), JeodBaseTime(), jeod::TimeMET::update(), and update().

## 8.1.5.14 std::string jeod::JeodBaseTime::update\_from\_name

Name of time-type from which update values are derived.

trick\_units(-)

Definition at line 103 of file time.hh.

Referenced by jeod::TimeUDE::add\_type\_initialize(), add\_type\_update(), jeod::TimeUDE::convert\_epoch\_to\_update(), jeod::TimeManagerInit::create\_init\_tree(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeUDE::initialize\_initializer\_time(), JeodBaseTime(), and jeod::TimeUDE::verify\_update().

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

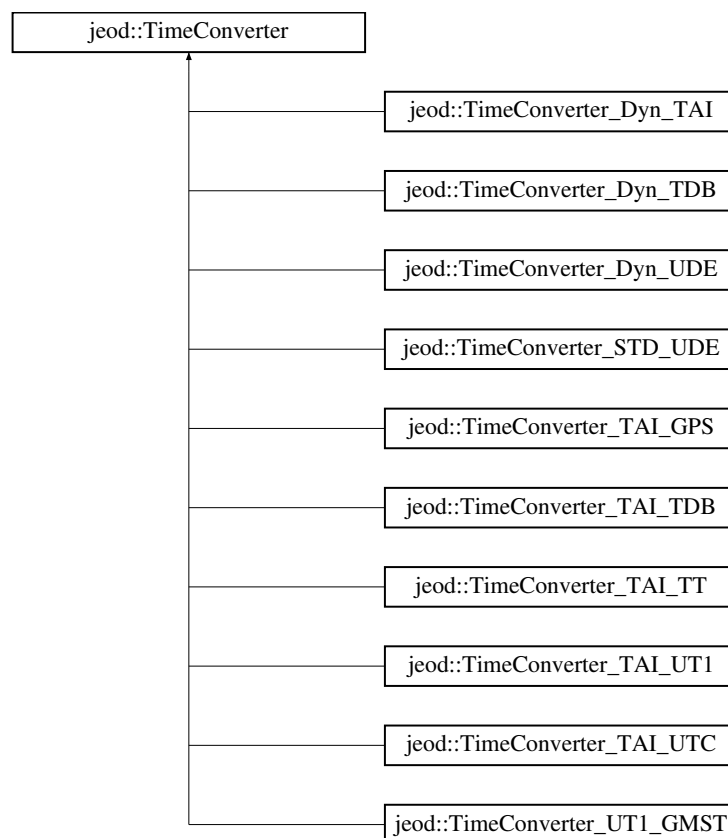
- [time.hh](#)
- [time.cc](#)
- [time\\_\\_add\\_type\\_update.cc](#)

## 8.2 jeod::TimeConverter Class Reference

The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.

```
#include <time_converter.hh>
```

Inheritance diagram for jeod::TimeConverter:



### Public Types

- enum [Direction](#) {  
 NO\_DIRECTION = 0x0000, A\_TO\_B\_INIT = 0x0001, B\_TO\_A\_INIT = 0x0010, A\_TO\_B\_UPDATE = 0x0100,

`B_TO_A_UPDATE = 0x1000, A_TO_B = 0x0101, B_TO_A = 0x1010, ANY_DIRECTION = 0x1111 }`

*Possible conversion directions.*

## Public Member Functions

- virtual `~TimeConverter` ()  
*Destroy a `TimeConverter`.*
- virtual void `initialize` (`JeodBaseTime` \*parent, `JeodBaseTime` \*child, const int direction)=0  
*Initialize the converter.*
- virtual bool `is_initialized` (void)  
*Return internal initialized status bool.*
- void `override_initialized` (bool init)
- bool `can_convert` (`Direction` query)  
*Check whether this converter is able to handle the requested conversion(s).*
- virtual void `convert_a_to_b` (void)  
*Default converter from time 'a' to time 'b'.*
- virtual void `convert_b_to_a` (void)  
*Default converter from time 'b' to time 'a'.*
- virtual void `reset_a_to_b_offset` (void)  
*Resets the offset between type a and type b mid-sim.*
- virtual void `verify_table_lookup_ends` (void)  
*This function does absolutely nothing.*
- double `get_a_to_b_offset` (void)  
*Return the offset from the parent time object to this object.*

## Data Fields

- std::string `a_name`  
*name of time-type "a".*
- std::string `b_name`  
*name of time-type "b".*

## Protected Member Functions

- `TimeConverter` ()  
*Construct a `TimeConverter`.*
- void `verify_setup` (const `JeodBaseTime` \*parent, const `JeodBaseTime` \*child, const int direction)  
*Verify the setup.*

## Protected Attributes

- bool `initialized`  
*whether converter has been initialized.*
- double `a_to_b_offset`  
*Difference between the two time-types.*
- `Direction` `valid_directions`  
*Bit packed flag specifying whether how a converter can be used.*

## Private Member Functions

- [TimeConverter](#) (const [TimeConverter](#) &)
- [TimeConverter](#) & [operator=](#) (const [TimeConverter](#) &)

## Friends

- class [InputProcessor](#)
- class [JeodBaseTime](#)
- void [init\\_attrjeod\\_\\_TimeConverter](#) ()

## 8.2.1 Detailed Description

The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.

Definition at line 58 of file `time_converter.hh`.

## 8.2.2 Member Enumeration Documentation

### 8.2.2.1 enum `jeod::TimeConverter::Direction`

Possible conversion directions.

Enumerator

```
NO_DIRECTION
A_TO_B_INIT
B_TO_A_INIT
A_TO_B_UPDATE
B_TO_A_UPDATE
A_TO_B
B_TO_A
ANY_DIRECTION
```

Definition at line 68 of file `time_converter.hh`.

## 8.2.3 Constructor & Destructor Documentation

### 8.2.3.1 `jeod::TimeConverter::~~TimeConverter ( void )` `[virtual]`

Destroy a [TimeConverter](#).

Definition at line 213 of file `time_converter.cc`.

### 8.2.3.2 `jeod::TimeConverter::TimeConverter ( void )` `[protected]`

Construct a [TimeConverter](#).

Definition at line 57 of file `time_converter.cc`.

References `a_name`, `a_to_b_offset`, `b_name`, `initialized`, `NO_DIRECTION`, and `valid_directions`.

**8.2.3.3** `jeod::TimeConverter::TimeConverter ( const TimeConverter & )` `[private]`

## 8.2.4 Member Function Documentation

**8.2.4.1** `bool jeod::TimeConverter::can_convert ( Direction query )`

Check whether this converter is able to handle the requested conversion(s).

If query is compound (e.g. CONV\_ALL, CONV\_A\_TO\_B\_UPDATE|CONV\_B\_TO\_A\_UPDATE) then return true only if capable of all conversions

### Returns

whether this converter can do all the conversions

### Parameters

<i>in</i>	<i>query</i>	converter directions to check
-----------	--------------	-------------------------------

Definition at line 128 of file `time_converter.cc`.

References `NO_DIRECTION`, and `valid_directions`.

**8.2.4.2** `void jeod::TimeConverter::convert_a_to_b ( void )` `[virtual]`

Default converter from time 'a' to time 'b'.

This default converter simply terminates the program. A subclass must override this default.

Reimplemented in [jeod::TimeConverter\\_TAI\\_UT1](#), [jeod::TimeConverter\\_TAI\\_UTC](#), [jeod::TimeConverter\\_TAI\\_TDB](#), [jeod::TimeConverter\\_STD\\_UDE](#), [jeod::TimeConverter\\_Dyn\\_UDE](#), [jeod::TimeConverter\\_Dyn\\_TAI](#), [jeod::TimeConverter\\_TAI\\_GPS](#), [jeod::TimeConverter\\_Dyn\\_TDB](#), [jeod::TimeConverter\\_TAI\\_TT](#), and [jeod::TimeConverter\\_UT1\\_GMST](#).

Definition at line 154 of file `time_converter.cc`.

References `jeod::TimeMessages::invalid_setup_error`.

Referenced by `jeod::TimeUDE::convert_epoch_to_update()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, `jeod::TimeUDE::initialize_initializer_time()`, and `jeod::JeodBaseTime::update()`.

**8.2.4.3** `void jeod::TimeConverter::convert_b_to_a ( void )` `[virtual]`

Default converter from time 'b' to time 'a'.

This default converter simply terminates the program. A subclass must override this default.

Reimplemented in [jeod::TimeConverter\\_TAI\\_UT1](#), [jeod::TimeConverter\\_TAI\\_UTC](#), [jeod::TimeConverter\\_TAI\\_TDB](#), [jeod::TimeConverter\\_STD\\_UDE](#), [jeod::TimeConverter\\_TAI\\_GPS](#), and [jeod::TimeConverter\\_TAI\\_TT](#).

Definition at line 170 of file `time_converter.cc`.

References `jeod::TimeMessages::invalid_setup_error`.

Referenced by `jeod::TimeUDE::convert_epoch_to_update()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, `jeod::TimeUDE::initialize_initializer_time()`, and `jeod::JeodBaseTime::update()`.

**8.2.4.4** `double jeod::TimeConverter::get_a_to_b_offset ( void )` `[inline]`

Return the offset from the parent time object to this object.



## Returns

a\_to\_b\_offset member.

Definition at line 152 of file time\_converter.hh.

References a\_to\_b\_offset.

**8.2.4.5** `virtual void jeod::TimeConverter::initialize ( JeodBaseTime * parent, JeodBaseTime * child, const int direction )`  
[pure virtual]

Initialize the converter.

## Parameters

in	<i>parent</i>	parent-type
in	<i>child</i>	child-type
in	<i>direction</i>	L-R, or R-L

Implemented in [jeod::TimeConverter\\_TAI\\_UT1](#), [jeod::TimeConverter\\_TAI\\_UTC](#), [jeod::TimeConverter\\_TAI\\_TDB](#), [jeod::TimeConverter\\_STD\\_UDE](#), [jeod::TimeConverter\\_Dyn\\_UDE](#), [jeod::TimeConverter\\_Dyn\\_TAI](#), [jeod::TimeConverter\\_TAI\\_GPS](#), [jeod::TimeConverter\\_Dyn\\_TDB](#), [jeod::TimeConverter\\_TAI\\_TT](#), and [jeod::TimeConverter\\_UT1\\_GMST](#).

Referenced by [jeod::JeodBaseTime::add\\_type\\_update\(\)](#), [jeod::TimeUDE::convert\\_epoch\\_to\\_update\(\)](#), [jeod::TimeStandard::initialize\\_from\\_parent\(\)](#), [jeod::TimeUDE::initialize\\_from\\_parent\(\)](#), and [jeod::TimeUDE::initialize\\_initializer\\_time\(\)](#).

**8.2.4.6** `bool jeod::TimeConverter::is_initialized ( void )` [virtual]

Return internal initialized status bool.

Definition at line 71 of file time\_converter.cc.

References initialized.

Referenced by [jeod::JeodBaseTime::add\\_type\\_update\(\)](#), [jeod::TimeStandard::initialize\\_from\\_parent\(\)](#), and [jeod::TimeUDE::initialize\\_from\\_parent\(\)](#).

**8.2.4.7** `TimeConverter& jeod::TimeConverter::operator= ( const TimeConverter & )` [private]

**8.2.4.8** `void jeod::TimeConverter::override_initialized ( bool init )` [inline]

Definition at line 128 of file time\_converter.hh.

References initialized.

Referenced by [jeod::TimeUDE::convert\\_epoch\\_to\\_update\(\)](#).

**8.2.4.9** `void jeod::TimeConverter::reset_a_to_b_offset ( void )` [virtual]

Resets the offset between type a and type b mid-sim.

Reimplemented in [jeod::TimeConverter\\_STD\\_UDE](#), and [jeod::TimeConverter\\_Dyn\\_UDE](#).

Definition at line 184 of file time\_converter.cc.

Referenced by [jeod::TimeMET::update\(\)](#).

**8.2.4.10** `void jeod::TimeConverter::verify_setup ( const JeodBaseTime * master_ptr, const JeodBaseTime * sub_ptr, const int int_dir )` [protected]

Verify the setup.

#### Assumptions and Limitations

- None

#### Parameters

in	<i>master_ptr</i>	Time used to initialize the converter
in	<i>sub_ptr</i>	Other time-type associated with the converter
in	<i>int_dir</i>	+1 a=parent; -1 b=parent; 0 error

Definition at line 86 of file `time_converter.cc`.

References `jeod::TimeMessages::initialization_error`, `jeod::JeodBaseTime::initialized`, `jeod::TimeMessages::invalid_setup_error`, and `jeod::JeodBaseTime::name`.

Referenced by `jeod::TimeConverter_TAI_TT::initialize()`, `jeod::TimeConverter_UT1_GMST::initialize()`, `jeod::TimeConverter_Dyn_TDB::initialize()`, `jeod::TimeConverter_TAI_GPS::initialize()`, `jeod::TimeConverter_Dyn_TAI::initialize()`, `jeod::TimeConverter_Dyn_UDE::initialize()`, `jeod::TimeConverter_STD_UDE::initialize()`, `jeod::TimeConverter_TAI_TDB::initialize()`, `jeod::TimeConverter_TAI_UTC::initialize()`, and `jeod::TimeConverter_TAI_UT1::initialize()`.

**8.2.4.11** `void jeod::TimeConverter::verify_table_lookup_ends ( void )` [virtual]

This function does absolutely nothing.

It is called when the simulation reverses direction (in time). If the converter uses a table lookup, this function should be replaced in that class. If the converter uses an analytic conversion, no action is needed and this (non)-function should be inherited.

#### Assumptions and Limitations

- None

Reimplemented in [jeod::TimeConverter\\_TAI\\_UT1](#), and [jeod::TimeConverter\\_TAI\\_UTC](#).

Definition at line 204 of file `time_converter.cc`.

## 8.2.5 Friends And Related Function Documentation

**8.2.5.1** `void init_attrjeod__TimeConverter ( )` [friend]

**8.2.5.2** `friend class InputProcessor` [friend]

Definition at line 60 of file `time_converter.hh`.

**8.2.5.3** `friend class JeodBaseTime` [friend]

Definition at line 62 of file `time_converter.hh`.

## 8.2.6 Field Documentation

**8.2.6.1** `std::string jeod::TimeConverter::a_name`

name of time-type "a".

trick\_units(-)

Definition at line 85 of file time\_converter.hh.

Referenced by jeod::TimeManager::register\_converter(), TimeConverter(), jeod::TimeConverter\_Dyn\_TAI::TimeConverter\_Dyn\_TAI(), jeod::TimeConverter\_Dyn\_TDB::TimeConverter\_Dyn\_TDB(), jeod::TimeConverter\_Dyn\_UDE::TimeConverter\_Dyn\_UDE(), jeod::TimeConverter\_STD\_UDE::TimeConverter\_STD\_UDE(), jeod::TimeConverter\_TAI\_GPS::TimeConverter\_TAI\_GPS(), jeod::TimeConverter\_TAI\_TDB::TimeConverter\_TAI\_TDB(), jeod::TimeConverter\_TAI\_TT::TimeConverter\_TAI\_TT(), jeod::TimeConverter\_TAI\_UT1::TimeConverter\_TAI\_UT1(), jeod::TimeConverter\_TAI\_UTC::TimeConverter\_TAI\_UTC(), and jeod::TimeConverter\_UT1\_GMST::TimeConverter\_UT1\_GMST().

#### 8.2.6.2 double jeod::TimeConverter::a\_to\_b\_offset [protected]

Difference between the two time-types.

trick\_units(-)

Definition at line 100 of file time\_converter.hh.

Referenced by jeod::TimeConverter\_Dyn\_TDB::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_GPS::convert\_a\_to\_b(), jeod::TimeConverter\_Dyn\_TAI::convert\_a\_to\_b(), jeod::TimeConverter\_Dyn\_UDE::convert\_a\_to\_b(), jeod::TimeConverter\_STD\_UDE::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_UTC::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_UT1::convert\_a\_to\_b(), jeod::TimeConverter\_TAI\_GPS::convert\_b\_to\_a(), jeod::TimeConverter\_STD\_UDE::convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UTC::convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UT1::convert\_b\_to\_a(), get\_a\_to\_b\_offset(), jeod::TimeConverter\_TAI\_TT::initialize(), jeod::TimeConverter\_Dyn\_TDB::initialize(), jeod::TimeConverter\_TAI\_GPS::initialize(), jeod::TimeConverter\_Dyn\_TAI::initialize(), jeod::TimeConverter\_Dyn\_UDE::initialize(), jeod::TimeConverter\_STD\_UDE::initialize(), jeod::TimeConverter\_TAI\_UTC::initialize(), jeod::TimeConverter\_TAI\_UT1::initialize(), jeod::TimeConverter\_TAI\_UTC::initialize\_leap\_second(), jeod::TimeConverter\_TAI\_UT1::initialize\_tai\_to\_ut1(), jeod::TimeConverter\_Dyn\_UDE::reset\_a\_to\_b\_offset(), jeod::TimeConverter\_STD\_UDE::reset\_a\_to\_b\_offset(), and TimeConverter().

#### 8.2.6.3 std::string jeod::TimeConverter::b\_name

name of time-type "b".

trick\_units(-)

Definition at line 89 of file time\_converter.hh.

Referenced by jeod::TimeManager::register\_converter(), TimeConverter(), jeod::TimeConverter\_Dyn\_TAI::TimeConverter\_Dyn\_TAI(), jeod::TimeConverter\_Dyn\_TDB::TimeConverter\_Dyn\_TDB(), jeod::TimeConverter\_Dyn\_UDE::TimeConverter\_Dyn\_UDE(), jeod::TimeConverter\_STD\_UDE::TimeConverter\_STD\_UDE(), jeod::TimeConverter\_TAI\_GPS::TimeConverter\_TAI\_GPS(), jeod::TimeConverter\_TAI\_TDB::TimeConverter\_TAI\_TDB(), jeod::TimeConverter\_TAI\_TT::TimeConverter\_TAI\_TT(), jeod::TimeConverter\_TAI\_UT1::TimeConverter\_TAI\_UT1(), jeod::TimeConverter\_TAI\_UTC::TimeConverter\_TAI\_UTC(), and jeod::TimeConverter\_UT1\_GMST::TimeConverter\_UT1\_GMST().

#### 8.2.6.4 bool jeod::TimeConverter::initialized [protected]

whether converter has been initialized.

trick\_units(-)

Definition at line 96 of file time\_converter.hh.

Referenced by jeod::TimeConverter\_TAI\_TT::initialize(), jeod::TimeConverter\_UT1\_GMST::initialize(), jeod::TimeConverter\_Dyn\_TDB::initialize(), jeod::TimeConverter\_TAI\_GPS::initialize(), jeod::TimeConverter\_Dyn\_TAI::initialize(), jeod::TimeConverter\_Dyn\_UDE::initialize(), jeod::TimeConverter\_STD\_UDE::initialize(), jeod::TimeConverter\_TAI\_TDB::initialize(), jeod::TimeConverter\_TAI\_UTC::initialize(), jeod::TimeConverter\_TAI\_UT1::initialize(), is\_initialized(), override\_initialized(), and TimeConverter().

### 8.2.6.5 Direction jeod::TimeConverter::valid\_directions [protected]

Bit packed flag specifying whether how a converter can be used.

Definition at line 104 of file time\_converter.hh.

Referenced by can\_convert(), TimeConverter(), jeod::TimeConverter\_Dyn\_TAI::TimeConverter\_Dyn\_TAI(), jeod::TimeConverter\_Dyn\_TDB::TimeConverter\_Dyn\_TDB(), jeod::TimeConverter\_Dyn\_UDE::TimeConverter\_Dyn\_UDE(), jeod::TimeConverter\_STD\_UDE::TimeConverter\_STD\_UDE(), jeod::TimeConverter\_TAI\_GPS::TimeConverter\_TAI\_GPS(), jeod::TimeConverter\_TAI\_TDB::TimeConverter\_TAI\_TDB(), jeod::TimeConverter\_TAI\_TT::TimeConverter\_TAI\_TT(), jeod::TimeConverter\_TAI\_UT1::TimeConverter\_TAI\_UT1(), jeod::TimeConverter\_TAI\_UTC::TimeConverter\_TAI\_UTC(), and jeod::TimeConverter\_UT1\_GMST::TimeConverter\_UT1\_GMST().

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

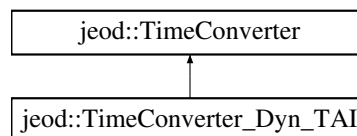
- [time\\_converter.hh](#)
- [time\\_converter.cc](#)

## 8.3 jeod::TimeConverter\_Dyn\_TAI Class Reference

Define class [TimeConverter\\_Dyn\\_TAI](#), which converts from simulation dynamic time to International Atomic Time.

```
#include <time_converter_dyn_tai.hh>
```

Inheritance diagram for jeod::TimeConverter\_Dyn\_TAI:



### Public Member Functions

- [TimeConverter\\_Dyn\\_TAI](#) ()  
*Construct a [TimeConverter\\_Dyn\\_TAI](#).*
- [~TimeConverter\\_Dyn\\_TAI](#) ()  
*Destroy a [TimeConverter\\_Dyn\\_TAI](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeDyn](#) to [TimeTAI](#).*

### Private Member Functions

- [TimeConverter\\_Dyn\\_TAI](#) (const [TimeConverter\\_Dyn\\_TAI](#) &)
- [TimeConverter\\_Dyn\\_TAI](#) & operator= (const [TimeConverter\\_Dyn\\_TAI](#) &)

### Private Attributes

- [TimeDyn](#) \* dyn\_ptr  
*Converter parent time, always a [TimeDyn](#) for this converter.*
- [TimeTAI](#) \* tai\_ptr  
*Converter child time, always a [TimeTAI](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_Dyn\\_TAI](#) ()

## Additional Inherited Members

### 8.3.1 Detailed Description

Define class [TimeConverter\\_Dyn\\_TAI](#), which converts from simulation dynamic time to International Atomic Time.

Definition at line 57 of file `time_converter_dyn_tai.hh`.

### 8.3.2 Constructor & Destructor Documentation

#### 8.3.2.1 `jeod::TimeConverter_Dyn_TAI::TimeConverter_Dyn_TAI ( void )`

Construct a [TimeConverter\\_Dyn\\_TAI](#).

Definition at line 59 of file `time_converter_dyn_tai.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::A_TO_B_UPDATE`, `jeod::TimeConverter::b_name`, `dyn_ptr`, `tai_ptr`, and `jeod::TimeConverter::valid_directions`.

#### 8.3.2.2 `jeod::TimeConverter_Dyn_TAI::~~TimeConverter_Dyn_TAI ( void )`

Destroy a [TimeConverter\\_Dyn\\_TAI](#).

Definition at line 181 of file `time_converter_dyn_tai.cc`.

#### 8.3.2.3 `jeod::TimeConverter_Dyn_TAI::TimeConverter_Dyn_TAI ( const TimeConverter_Dyn_TAI & ) [private]`

### 8.3.3 Member Function Documentation

#### 8.3.3.1 `void jeod::TimeConverter_Dyn_TAI::convert_a_to_b ( void ) [virtual]`

Convert from [TimeDyn](#) to [TimeTAI](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 165 of file `time_converter_dyn_tai.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `dyn_ptr`, `jeod::JeodBaseTime::seconds`, `jeod::TimeStandard::set_time_by_seconds()`, and `tai_ptr`.

#### 8.3.3.2 `void jeod::TimeConverter_Dyn_TAI::initialize ( JeodBaseTime * parent_ptr, JeodBaseTime * child_ptr, const int int_dir ) [virtual]`

Initialize the converter.

Parameters

<code>in</code>	<code>parent_ptr</code>	Time used to initialize the converter
<code>in</code>	<code>child_ptr</code>	Other Time used to initialize the converter

<code>in</code>	<code>int_dir</code>	Conversion direction: +1 a=parent; -1 b=parent; 0 error
-----------------	----------------------	---

Implements [jeod::TimeConverter](#).

Definition at line 77 of file `time_converter_dyn_tai.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `dyn_ptr`, `jeod::TimeMessages::initialization_error`, `jeod::TimeConverter::initialized`, `jeod::TimeMessages::invalid_setup_error`, `jeod::JeodBaseTime::name`, `jeod::JeodBaseTime::seconds`, `tai_ptr`, and `jeod::TimeConverter::verify_setup()`.

**8.3.3.3 TimeConverter\_Dyn\_TAI& jeod::TimeConverter\_Dyn\_TAI::operator= ( const TimeConverter\_Dyn\_TAI & )**  
[private]

## 8.3.4 Friends And Related Function Documentation

**8.3.4.1 void init\_attrjeod\_\_TimeConverter\_Dyn\_TAI ( )** [friend]

**8.3.4.2 friend class InputProcessor** [friend]

Definition at line 60 of file `time_converter_dyn_tai.hh`.

## 8.3.5 Field Documentation

**8.3.5.1 TimeDyn\* jeod::TimeConverter\_Dyn\_TAI::dyn\_ptr** [private]

Converter parent time, always a [TimeDyn](#) for this converter.

`trick_units(-)`

Definition at line 66 of file `time_converter_dyn_tai.hh`.

Referenced by `convert_a_to_b()`, `initialize()`, and `TimeConverter_Dyn_TAI()`.

**8.3.5.2 TimeTAI\* jeod::TimeConverter\_Dyn\_TAI::tai\_ptr** [private]

Converter child time, always a [TimeTAI](#) for this converter.

`trick_units(-)`

Definition at line 71 of file `time_converter_dyn_tai.hh`.

Referenced by `convert_a_to_b()`, `initialize()`, and `TimeConverter_Dyn_TAI()`.

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

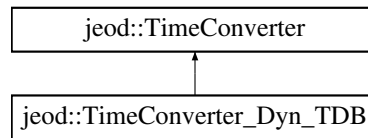
- [time\\_converter\\_dyn\\_tai.hh](#)
- [time\\_converter\\_dyn\\_tai.cc](#)

## 8.4 jeod::TimeConverter\_Dyn\_TDB Class Reference

Define class [TimeConverter\\_Dyn\\_TDB](#), which converts from simulation dynamic time to Barycentric Dynamic Time.

```
#include <time_converter_dyn_tdb.hh>
```

Inheritance diagram for `jeod::TimeConverter_Dyn_TDB`:



## Public Member Functions

- [TimeConverter\\_Dyn\\_TDB](#) ()  
*Construct a [TimeConverter\\_Dyn\\_TDB](#).*
- [~TimeConverter\\_Dyn\\_TDB](#) ()  
*Destroy a [TimeConverter\\_Dyn\\_TDB](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeDyn](#) to [TimeTDB](#).*

## Private Member Functions

- [TimeConverter\\_Dyn\\_TDB](#) (const [TimeConverter\\_Dyn\\_TDB](#) &)
- [TimeConverter\\_Dyn\\_TDB](#) & operator= (const [TimeConverter\\_Dyn\\_TDB](#) &)

## Private Attributes

- [TimeDyn](#) \* dyn\_ptr  
*Converter parent time, always a [TimeDyn](#) for this converter.*
- [TimeTDB](#) \* tdb\_ptr  
*Converter child time, always a [TimeTDB](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_Dyn\\_TDB](#) ()

## Additional Inherited Members

### 8.4.1 Detailed Description

Define class [TimeConverter\\_Dyn\\_TDB](#), which converts from simulation dynamic time to Barycentric Dynamic Time.  
Definition at line 56 of file `time_converter_dyn_tdb.hh`.

### 8.4.2 Constructor & Destructor Documentation

#### 8.4.2.1 jeod::TimeConverter\_Dyn\_TDB::TimeConverter\_Dyn\_TDB ( void )

Construct a [TimeConverter\\_Dyn\\_TDB](#).

Definition at line 59 of file `time_converter_dyn_tdb.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::A_TO_B`, `jeod::TimeConverter::b_name`, `dyn_ptr`, `tdb_ptr`, and `jeod::TimeConverter::valid_directions`.

#### 8.4.2.2 jeod::TimeConverter\_Dyn\_TDB::~~TimeConverter\_Dyn\_TDB ( void )

Destroy a [TimeConverter\\_Dyn\\_TDB](#).

Definition at line 151 of file `time_converter_dyn_tdb.cc`.

#### 8.4.2.3 jeod::TimeConverter\_Dyn\_TDB::TimeConverter\_Dyn\_TDB ( const TimeConverter\_Dyn\_TDB & ) [private]

### 8.4.3 Member Function Documentation

#### 8.4.3.1 void jeod::TimeConverter\_Dyn\_TDB::convert\_a\_to\_b ( void ) [virtual]

Convert from [TimeDyn](#) to [TimeTDB](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 139 of file `time_converter_dyn_tdb.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [dyn\\_ptr](#), [jeod::JeodBaseTime::seconds](#), [jeod::TimeStandard::set\\_time\\_by\\_seconds\(\)](#), and [tdb\\_ptr](#).

#### 8.4.3.2 void jeod::TimeConverter\_Dyn\_TDB::initialize ( JeodBaseTime \* parent\_ptr, JeodBaseTime \* child\_ptr, const int int\_dir ) [virtual]

Initialize the converter.

Parameters

in	<i>parent_ptr</i>	Time used to initialize the converter
in	<i>child_ptr</i>	Other Time used to initialize the converter
in	<i>int_dir</i>	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements [jeod::TimeConverter](#).

Definition at line 77 of file `time_converter_dyn_tdb.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [dyn\\_ptr](#), [jeod::TimeMessages::initialization\\_error](#), [jeod::TimeConverter::initialized](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [jeod::JeodBaseTime::seconds](#), [tdb\\_ptr](#), and [jeod::TimeConverter::verify\\_setup\(\)](#).

#### 8.4.3.3 TimeConverter\_Dyn\_TDB& jeod::TimeConverter\_Dyn\_TDB::operator= ( const TimeConverter\_Dyn\_TDB & ) [private]

### 8.4.4 Friends And Related Function Documentation

#### 8.4.4.1 void init\_attrjeod\_\_TimeConverter\_Dyn\_TDB ( ) [friend]

#### 8.4.4.2 friend class InputProcessor [friend]

Definition at line 58 of file `time_converter_dyn_tdb.hh`.

### 8.4.5 Field Documentation

#### 8.4.5.1 TimeDyn\* jeod::TimeConverter\_Dyn\_TDB::dyn\_ptr [private]

Converter parent time, always a [TimeDyn](#) for this converter.

`trick_units(-)`

Definition at line 64 of file `time_converter_dyn_tdb.hh`.



Referenced by `convert_a_to_b()`, `initialize()`, and `TimeConverter_Dyn_TDB()`.

#### 8.4.5.2 TimeTDB\* jeod::TimeConverter\_Dyn\_TDB::tdb\_ptr [private]

Converter child time, always a [TimeTDB](#) for this converter.

`trick_units(-)`

Definition at line 69 of file `time_converter_dyn_tdb.hh`.

Referenced by `convert_a_to_b()`, `initialize()`, and `TimeConverter_Dyn_TDB()`.

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

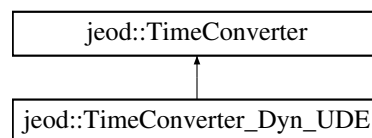
- [time\\_converter\\_dyn\\_tdb.hh](#)
- [time\\_converter\\_dyn\\_tdb.cc](#)

## 8.5 jeod::TimeConverter\_Dyn\_UDE Class Reference

Define class [TimeConverter\\_Dyn\\_UDE](#), which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

```
#include <time_converter_dyn_ude.hh>
```

Inheritance diagram for `jeod::TimeConverter_Dyn_UDE`:



### Public Member Functions

- [TimeConverter\\_Dyn\\_UDE](#) ()  
*Construct a [TimeConverter\\_Dyn\\_UDE](#).*
- [~TimeConverter\\_Dyn\\_UDE](#) ()  
*Destroy a [TimeConverter\\_Dyn\\_UDE](#).*
- void [reset\\_a\\_to\\_b\\_offset](#) (void)  
*Resets the value of `a_to_b_offset`.*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeDyn](#) to [TimeUDE](#).*

### Private Member Functions

- [TimeConverter\\_Dyn\\_UDE](#) (const [TimeConverter\\_Dyn\\_UDE](#) &)
- [TimeConverter\\_Dyn\\_UDE](#) & operator= (const [TimeConverter\\_Dyn\\_UDE](#) &)

## Private Attributes

- [TimeDyn](#) \* [dyn\\_ptr](#)  
*Converter parent time, always a [TimeDyn](#) for this converter.*
- [TimeUDE](#) \* [ude\\_ptr](#)  
*Converter child time, always a [TimeUDE](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_Dyn\\_UDE](#) ()

## Additional Inherited Members

### 8.5.1 Detailed Description

Define class [TimeConverter\\_Dyn\\_UDE](#), which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

Definition at line 58 of file [time\\_converter\\_dyn\\_ude.hh](#).

### 8.5.2 Constructor & Destructor Documentation

#### 8.5.2.1 [jeod::TimeConverter\\_Dyn\\_UDE::TimeConverter\\_Dyn\\_UDE \( void \)](#)

Construct a [TimeConverter\\_Dyn\\_UDE](#).

Definition at line 60 of file [time\\_converter\\_dyn\\_ude.cc](#).

References [jeod::TimeConverter::a\\_name](#), [jeod::TimeConverter::A\\_TO\\_B](#), [jeod::TimeConverter::b\\_name](#), [dyn\\_ptr](#), [ude\\_ptr](#), and [jeod::TimeConverter::valid\\_directions](#).

#### 8.5.2.2 [jeod::TimeConverter\\_Dyn\\_UDE::~~TimeConverter\\_Dyn\\_UDE \( void \)](#)

Destroy a [TimeConverter\\_Dyn\\_UDE](#).

Definition at line 166 of file [time\\_converter\\_dyn\\_ude.cc](#).

#### 8.5.2.3 [jeod::TimeConverter\\_Dyn\\_UDE::TimeConverter\\_Dyn\\_UDE \( const \[TimeConverter\\\_Dyn\\\_UDE\]\(#\) & \) \[private\]](#)

### 8.5.3 Member Function Documentation

#### 8.5.3.1 [void jeod::TimeConverter\\_Dyn\\_UDE::convert\\_a\\_to\\_b \( void \) \[virtual\]](#)

Convert from [TimeDyn](#) to [TimeUDE](#).

## Assumptions and Limitations

- Time class UDE is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from [jeod::TimeConverter](#).

Definition at line 142 of file [time\\_converter\\_dyn\\_ude.cc](#).

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [dyn\\_ptr](#), [jeod::JeodBaseTime::seconds](#), [jeod::TimeUDE::set\\_time\\_by\\_seconds\(\)](#), and [ude\\_ptr](#).

**8.5.3.2** `void jeod::TimeConverter_Dyn_UDE::initialize ( JeodBaseTime * parent_ptr, JeodBaseTime * child_ptr, const int int_dir )` `[virtual]`

Initialize the converter.

#### Assumptions and Limitations

- This class converts from [TimeDyn](#) to [TimeUDE](#) *only*

#### Parameters

in	<i>parent_ptr</i>	Time used to initialize the converter
in	<i>child_ptr</i>	Other Time used to initialize the converter
in	<i>int_dir</i>	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements [jeod::TimeConverter](#).

Definition at line 81 of file `time_converter_dyn_ude.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `dyn_ptr`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::TimeConverter::initialized`, `jeod::TimeMessages::invalid_setup_error`, `jeod::JeodBaseTime::name`, `jeod::JeodBaseTime::seconds`, `ude_ptr`, and `jeod::TimeConverter::verify_setup()`.

**8.5.3.3** `TimeConverter_Dyn_UDE& jeod::TimeConverter_Dyn_UDE::operator= ( const TimeConverter_Dyn_UDE & )` `[private]`

**8.5.3.4** `void jeod::TimeConverter_Dyn_UDE::reset_a_to_b_offset ( void )` `[virtual]`

Resets the value of `a_to_b_offset`.

Reimplemented from [jeod::TimeConverter](#).

Definition at line 156 of file `time_converter_dyn_ude.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `dyn_ptr`, `jeod::JeodBaseTime::seconds`, and `ude_ptr`.

## 8.5.4 Friends And Related Function Documentation

**8.5.4.1** `void init_attrjeod__TimeConverter_Dyn_UDE ( )` `[friend]`

**8.5.4.2** `friend class InputProcessor` `[friend]`

Definition at line 60 of file `time_converter_dyn_ude.hh`.

## 8.5.5 Field Documentation

**8.5.5.1** `TimeDyn* jeod::TimeConverter_Dyn_UDE::dyn_ptr` `[private]`

Converter parent time, always a [TimeDyn](#) for this converter.

`trick_units(-)`

Definition at line 67 of file `time_converter_dyn_ude.hh`.

Referenced by `convert_a_to_b()`, `initialize()`, `reset_a_to_b_offset()`, and `TimeConverter_Dyn_UDE()`.

**8.5.5.2** `TimeUDE* jeod::TimeConverter_Dyn_UDE::ude_ptr` `[private]`

Converter child time, always a [TimeUDE](#) for this converter.

trick\_units(-)

Definition at line 72 of file time\_converter\_dyn\_ude.hh.

Referenced by convert\_a\_to\_b(), initialize(), reset\_a\_to\_b\_offset(), and TimeConverter\_Dyn\_UDE().

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

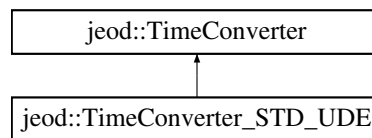
- [time\\_converter\\_dyn\\_ude.hh](#)
- [time\\_converter\\_dyn\\_ude.cc](#)

## 8.6 jeod::TimeConverter\_STD\_UDE Class Reference

Define class [TimeConverter\\_STD\\_UDE](#), which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

```
#include <time_converter_std_ude.hh>
```

Inheritance diagram for jeod::TimeConverter\_STD\_UDE:



### Public Member Functions

- [TimeConverter\\_STD\\_UDE](#) ()  
*Construct a [TimeConverter\\_STD\\_UDE](#).*
- [~TimeConverter\\_STD\\_UDE](#) ()  
*Destroy a [TimeConverter\\_STD\\_UDE](#).*
- void [reset\\_a\\_to\\_b\\_offset](#) (void)  
*Resets the value of a\_to\_b\_offset.*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from TimeSTD to [TimeUDE](#).*
- void [convert\\_b\\_to\\_a](#) (void)  
*Convert from [TimeUDE](#) to TimeSTD.*

### Private Member Functions

- [TimeConverter\\_STD\\_UDE](#) (const [TimeConverter\\_STD\\_UDE](#) &)
- [TimeConverter\\_STD\\_UDE](#) & operator= (const [TimeConverter\\_STD\\_UDE](#) &)

### Private Attributes

- bool [failed\\_null\\_test](#)  
*Initializing converter could be done in one of two ways.*
- [TimeStandard](#) \* [std\\_ptr](#)  
*Converter parent time, always a TimeSTD for this converter.*
- [TimeUDE](#) \* [ude\\_ptr](#)  
*Converter parent time, always a [TimeUDE](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_STD\\_UDE](#) ()

## Additional Inherited Members

### 8.6.1 Detailed Description

Define class [TimeConverter\\_STD\\_UDE](#), which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

Definition at line 59 of file `time_converter_std_ude.hh`.

### 8.6.2 Constructor & Destructor Documentation

#### 8.6.2.1 jeod::TimeConverter\_STD\_UDE::TimeConverter\_STD\_UDE ( void )

Construct a [TimeConverter\\_STD\\_UDE](#).

Definition at line 59 of file `time_converter_std_ude.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::ANY_DIRECTION`, `jeod::TimeConverter::b_name`, `failed_null_test`, `std_ptr`, `ude_ptr`, and `jeod::TimeConverter::valid_directions`.

#### 8.6.2.2 jeod::TimeConverter\_STD\_UDE::~~TimeConverter\_STD\_UDE ( void )

Destroy a [TimeConverter\\_STD\\_UDE](#).

Definition at line 191 of file `time_converter_std_ude.cc`.

#### 8.6.2.3 jeod::TimeConverter\_STD\_UDE::TimeConverter\_STD\_UDE ( const TimeConverter\_STD\_UDE & ) [private]

### 8.6.3 Member Function Documentation

#### 8.6.3.1 void jeod::TimeConverter\_STD\_UDE::convert\_a\_to\_b ( void ) [virtual]

Convert from TimeSTD to [TimeUDE](#).

#### Assumptions and Limitations

- Time class UDE is based on time class STD, and counts the elapsed STD time only

Reimplemented from [jeod::TimeConverter](#).

Definition at line 148 of file `time_converter_std_ude.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `jeod::JeodBaseTime::seconds`, `jeod::TimeUDE::set_time_by_seconds()`, `std_ptr`, and `ude_ptr`.

#### 8.6.3.2 void jeod::TimeConverter\_STD\_UDE::convert\_b\_to\_a ( void ) [virtual]

Convert from [TimeUDE](#) to TimeSTD.

#### Assumptions and Limitations

- Time class UDE is based on time class STD, and counts the elapsed STD time only

Reimplemented from [jeod::TimeConverter](#).

Definition at line 165 of file `time_converter_std_ude.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `jeod::JeodBaseTime::seconds`, `jeod::TimeStandard::set_time_by_seconds()`, `std_ptr`, and `ude_ptr`.

**8.6.3.3** `void jeod::TimeConverter_STD_UDE::initialize ( JeodBaseTime * parent_ptr, JeodBaseTime * child_ptr, const int int_dir ) [virtual]`

Initialize the converter.

#### Assumptions and Limitations

- This class converts from [TimeDyn](#) to [TimeUDE](#) *only*

#### Parameters

in	<i>parent_ptr</i>	Time used to initialize the converter
in	<i>child_ptr</i>	Other Time used to initialize the converter
in	<i>int_dir</i>	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements [jeod::TimeConverter](#).

Definition at line 81 of file `time_converter_std_ude.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `failed_null_test`, `jeod::TimeConverter::initialized`, `jeod::TimeMessages::invalid_setup_error`, `jeod::JeodBaseTime::name`, `jeod::JeodBaseTime::seconds`, `std_ptr`, `ude_ptr`, and `jeod::TimeConverter::verify_setup()`.

**8.6.3.4** `TimeConverter_STD_UDE& jeod::TimeConverter_STD_UDE::operator= ( const TimeConverter_STD_UDE & ) [private]`

**8.6.3.5** `void jeod::TimeConverter_STD_UDE::reset_a_to_b_offset ( void ) [virtual]`

Resets the value of `a_to_b_offset`.

Reimplemented from [jeod::TimeConverter](#).

Definition at line 178 of file `time_converter_std_ude.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `jeod::JeodBaseTime::seconds`, `std_ptr`, and `ude_ptr`.

## 8.6.4 Friends And Related Function Documentation

**8.6.4.1** `void init_attrjeod_TimeConverter_STD_UDE ( ) [friend]`

**8.6.4.2** `friend class InputProcessor [friend]`

Definition at line 61 of file `time_converter_std_ude.hh`.

## 8.6.5 Field Documentation

**8.6.5.1** `bool jeod::TimeConverter_STD_UDE::failed_null_test [private]`

Initializing converter could be done in one of two ways.

If it fails the first time, this flag is set. If it fails a second time, it terminates.`trick_units(-)`

Definition at line 69 of file `time_converter_std_ude.hh`.

Referenced by `initialize()`, and `TimeConverter_STD_UDE()`.

#### 8.6.5.2 TimeStandard\* jeod::TimeConverter\_STD\_UDE::std\_ptr [private]

Converter parent time, always a `TimeSTD` for this converter.

`trick_units(-)`

Definition at line 73 of file `time_converter_std_ude.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, `reset_a_to_b_offset()`, and `TimeConverter_STD_UDE()`.

#### 8.6.5.3 TimeUDE\* jeod::TimeConverter\_STD\_UDE::ude\_ptr [private]

Converter parent time, always a `TimeUDE` for this converter.

`trick_units(-)`

Definition at line 78 of file `time_converter_std_ude.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, `reset_a_to_b_offset()`, and `TimeConverter_STD_UDE()`.

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

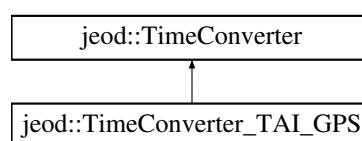
- [time\\_converter\\_std\\_ude.hh](#)
- [time\\_converter\\_std\\_ude.cc](#)

## 8.7 jeod::TimeConverter\_TAI\_GPS Class Reference

Define class `TimeConverter_TAI_GPS`, which converts between International Atomic Time and the clock associated with the Global Positioning System.

```
#include <time_converter_tai_gps.hh>
```

Inheritance diagram for `jeod::TimeConverter_TAI_GPS`:



### Public Member Functions

- `TimeConverter_TAI_GPS ()`  
*Construct a `TimeConverter_TAI_GPS`.*
- `~TimeConverter_TAI_GPS ()`  
*Destroy a `TimeConverter_TAI_GPS`.*
- `void initialize (JeodBaseTime *parent, JeodBaseTime *child, const int direction)`  
*Initialize the converter.*
- `void convert_a_to_b (void)`  
*Convert from `TimeTAI` to `TimeGPS`.*
- `void convert_b_to_a (void)`  
*Convert from `TimeGPS` to `TimeTAI`.*

## Private Member Functions

- [TimeConverter\\_TAI\\_GPS](#) (const [TimeConverter\\_TAI\\_GPS](#) &)
- [TimeConverter\\_TAI\\_GPS](#) & operator= (const [TimeConverter\\_TAI\\_GPS](#) &)

## Private Attributes

- [TimeTAI](#) \* tai\_ptr  
*Converter parent time, always a [TimeTAI](#) for this converter.*
- [TimeGPS](#) \* gps\_ptr  
*Converter parent time, always a [TimeGPS](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI\\_GPS](#) ()

## Additional Inherited Members

### 8.7.1 Detailed Description

Define class [TimeConverter\\_TAI\\_GPS](#), which converts between International Atomic Time and the clock associated with the Global Positioning System.

Definition at line 56 of file `time_converter_tai_gps.hh`.

### 8.7.2 Constructor & Destructor Documentation

#### 8.7.2.1 `jeod::TimeConverter_TAI_GPS::TimeConverter_TAI_GPS ( void )`

Construct a [TimeConverter\\_TAI\\_GPS](#).

Definition at line 60 of file `time_converter_tai_gps.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::ANY_DIRECTION`, `jeod::TimeConverter::b_name`, `gps_ptr`, `tai_ptr`, and `jeod::TimeConverter::valid_directions`.

#### 8.7.2.2 `jeod::TimeConverter_TAI_GPS::~~TimeConverter_TAI_GPS ( void )`

Destroy a [TimeConverter\\_TAI\\_GPS](#).

Definition at line 138 of file `time_converter_tai_gps.cc`.

#### 8.7.2.3 `jeod::TimeConverter_TAI_GPS::TimeConverter_TAI_GPS ( const TimeConverter_TAI_GPS & ) [private]`

### 8.7.3 Member Function Documentation

#### 8.7.3.1 `void jeod::TimeConverter_TAI_GPS::convert_a_to_b ( void ) [virtual]`

Convert from [TimeTAI](#) to [TimeGPS](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 114 of file `time_converter_tai_gps.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `gps_ptr`, `jeod::JeodBaseTime::seconds`, `jeod::TimeGPS::set_time_by_seconds()`, and `tai_ptr`.



8.7.3.2 `void jeod::TimeConverter_TAI_GPS::convert_b_to_a ( void ) [virtual]`

Convert from [TimeGPS](#) to [TimeTAI](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 126 of file `time_converter_tai_gps.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [gps\\_ptr](#), [jeod::JeodBaseTime::seconds](#), [jeod::TimeStandard::set\\_time\\_by\\_seconds\(\)](#), and [tai\\_ptr](#).

8.7.3.3 `void jeod::TimeConverter_TAI_GPS::initialize ( JeodBaseTime * parent_ptr, JeodBaseTime * child_ptr, const int int_dir ) [virtual]`

Initialize the converter.

#### Assumptions and Limitations

- None

#### Parameters

<code>in</code>	<code>parent_ptr</code>	Time used to initialize the converter
<code>in</code>	<code>child_ptr</code>	Other Time used to initialize the converter
<code>in</code>	<code>int_dir</code>	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements [jeod::TimeConverter](#).

Definition at line 81 of file `time_converter_tai_gps.cc`.

References [jeod::TimeConverter::a\\_to\\_b\\_offset](#), [gps\\_ptr](#), [jeod::TimeConverter::initialized](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [tai\\_ptr](#), [jeod::TimeStandard::tjt\\_at\\_epoch](#), and [jeod::TimeConverter::verify\\_setup\(\)](#).

8.7.3.4 `TimeConverter_TAI_GPS& jeod::TimeConverter_TAI_GPS::operator= ( const TimeConverter_TAI_GPS & ) [private]`

## 8.7.4 Friends And Related Function Documentation

8.7.4.1 `void init_attrjeod_TimeConverter_TAI_GPS ( ) [friend]`

8.7.4.2 `friend class InputProcessor [friend]`

Definition at line 58 of file `time_converter_tai_gps.hh`.

## 8.7.5 Field Documentation

8.7.5.1 `TimeGPS* jeod::TimeConverter_TAI_GPS::gps_ptr [private]`

Converter parent time, always a [TimeGPS](#) for this converter.

`trick_units(-)`

Definition at line 70 of file `time_converter_tai_gps.hh`.

Referenced by [convert\\_a\\_to\\_b\(\)](#), [convert\\_b\\_to\\_a\(\)](#), [initialize\(\)](#), and [TimeConverter\\_TAI\\_GPS\(\)](#).

8.7.5.2 `TimeTAI* jeod::TimeConverter_TAI_GPS::tai_ptr [private]`

Converter parent time, always a [TimeTAI](#) for this converter.

trick\_units(-)

Definition at line 65 of file time\_converter\_tai\_gps.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), and TimeConverter\_TAI\_GPS().

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

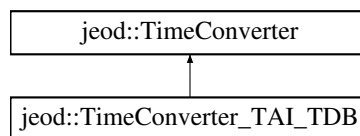
- [time\\_converter\\_tai\\_gps.hh](#)
- [time\\_converter\\_tai\\_gps.cc](#)

## 8.8 jeod::TimeConverter\_TAI\_TDB Class Reference

Define class [TimeConverter\\_TAI\\_TDB](#), which converts from International Atomic Time to Barycentric Dynamic Time.

```
#include <time_converter_tai_tdb.hh>
```

Inheritance diagram for jeod::TimeConverter\_TAI\_TDB:



### Public Member Functions

- [TimeConverter\\_TAI\\_TDB](#) ()
- [~TimeConverter\\_TAI\\_TDB](#) ()
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [set\\_a\\_to\\_b\\_offset](#) (void)
- void [convert\\_a\\_to\\_b](#) (void)  
*Default converter from time 'a' to time 'b'.*
- void [convert\\_b\\_to\\_a](#) (void)  
*Default converter from time 'b' to time 'a'.*

### Private Member Functions

- [TimeConverter\\_TAI\\_TDB](#) (const [TimeConverter\\_TAI\\_TDB](#) &)
- [TimeConverter\\_TAI\\_TDB](#) & [operator=](#) (const [TimeConverter\\_TAI\\_TDB](#) &)

### Private Attributes

- double [TAI\\_to\\_TT\\_offset](#)  
*The offset from TAI to TT.*
- double [a\\_to\\_b\\_offset\\_epoch](#)  
*The epoch value of a\_to\_b\_offset.*
- double [a\\_to\\_b\\_offset](#)  
*Calculated value of a\_to\_b\_offset.*
- double [prev\\_tai\\_seconds](#)  
*TAI seconds from previous loop iteration.*

- double [prev\\_tdb\\_seconds](#)  
*TDB seconds from previous loop iteration.*
- int [nSteps](#)  
*Counter for number of steps in iteration.*
- int [nlter](#)  
*Counter for number of iterations.*
- [TimeTAI](#) \* [tai\\_ptr](#)  
*Converter parent time, always a [TimeTAI](#) for this converter.*
- [TimeTDB](#) \* [tdb\\_ptr](#)  
*Converter parent time, always a [TimeTDB](#) for this converter.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI\\_TDB](#) ()

## Additional Inherited Members

### 8.8.1 Detailed Description

Define class [TimeConverter\\_TAI\\_TDB](#), which converts from International Atomic Time to Barycentric Dynamic Time.

Definition at line 58 of file `time_converter_tai_tdb.hh`.

### 8.8.2 Constructor & Destructor Documentation

#### 8.8.2.1 `jeod::TimeConverter_TAI_TDB::TimeConverter_TAI_TDB ( void )`

Definition at line 70 of file `time_converter_tai_tdb.cc`.

References `jeod::TimeConverter::a_name`, `a_to_b_offset`, `a_to_b_offset_epoch`, `jeod::TimeConverter::ANY_DIRECTION`, `jeod::TimeConverter::b_name`, `nlter`, `nSteps`, `prev_tai_seconds`, `prev_tdb_seconds`, `tai_ptr`, `TAI_to_TT_offset`, `tdb_ptr`, and `jeod::TimeConverter::valid_directions`.

#### 8.8.2.2 `jeod::TimeConverter_TAI_TDB::~~TimeConverter_TAI_TDB ( void )`

Definition at line 196 of file `time_converter_tai_tdb.cc`.

#### 8.8.2.3 `jeod::TimeConverter_TAI_TDB::TimeConverter_TAI_TDB ( const TimeConverter_TAI_TDB & ) [private]`

### 8.8.3 Member Function Documentation

#### 8.8.3.1 `void jeod::TimeConverter_TAI_TDB::convert_a_to_b ( void ) [virtual]`

Default converter from time 'a' to time 'b'.

This default converter simply terminates the program. A subclass must override this default.

Reimplemented from [jeod::TimeConverter](#).

Definition at line 154 of file `time_converter_tai_tdb.cc`.

References `a_to_b_offset`, `a_to_b_offset_epoch`, `jeod::JeodBaseTime::seconds`, `set_a_to_b_offset()`, `jeod::TimeStandard::set_time_by_seconds()`, `tai_ptr`, and `tdb_ptr`.

**8.8.3.2** `void jeod::TimeConverter_TAI_TDB::convert_b_to_a ( void ) [virtual]`

Default converter from time 'b' to time 'a'.

This default converter simply terminates the program. A subclass must override this default.

Reimplemented from [jeod::TimeConverter](#).

Definition at line 170 of file `time_converter_tai_tdb.cc`.

References `a_to_b_offset`, `a_to_b_offset_epoch`, `nlter`, `nSteps`, `prev_tai_seconds`, `prev_tdb_seconds`, `jeod::JeodBaseTime::seconds`, `set_a_to_b_offset()`, `jeod::TimeStandard::set_time_by_seconds()`, `tai_ptr`, and `tdb_ptr`.

**8.8.3.3** `void jeod::TimeConverter_TAI_TDB::initialize ( JeodBaseTime * parent, JeodBaseTime * child, const int direction ) [virtual]`

Initialize the converter.

Parameters

in	<i>parent</i>	parent-type
in	<i>child</i>	child-type
in	<i>direction</i>	L-R, or R-L

Implements [jeod::TimeConverter](#).

Definition at line 99 of file `time_converter_tai_tdb.cc`.

References `a_to_b_offset_epoch`, `jeod::TimeConverter::initialized`, `jeod::TimeMessages::invalid_setup_error`, `set_a_to_b_offset()`, `tai_ptr`, `TAI_to_TT_offset`, `tdb_ptr`, `jeod::TimeStandard::tjt_at_epoch`, and `jeod::TimeConverter::verify_setup()`.

**8.8.3.4** `TimeConverter_TAI_TDB& jeod::TimeConverter_TAI_TDB::operator= ( const TimeConverter_TAI_TDB & ) [private]`

**8.8.3.5** `void jeod::TimeConverter_TAI_TDB::set_a_to_b_offset ( void )`

Definition at line 135 of file `time_converter_tai_tdb.cc`.

References `a_to_b_offset`, `tai_ptr`, `jeod::TimeStandard::tjt_at_epoch`, and `jeod::TimeStandard::trunc_julian_time`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, and `initialize()`.

## 8.8.4 Friends And Related Function Documentation

**8.8.4.1** `void init_attrjeod_TimeConverter_TAI_TDB ( ) [friend]`

**8.8.4.2** `friend class InputProcessor [friend]`

Definition at line 60 of file `time_converter_tai_tdb.hh`.

## 8.8.5 Field Documentation

**8.8.5.1** `double jeod::TimeConverter_TAI_TDB::a_to_b_offset [private]`

Calculated value of `a_to_b_offset`.

`trick_units(-)`

Definition at line 75 of file `time_converter_tai_tdb.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `set_a_to_b_offset()`, and `TimeConverter_TAI_TDB()`.

**8.8.5.2** `double jeod::TimeConverter_TAI_TDB::a_to_b_offset_epoch` `[private]`

The epoch value of `a_to_b_offset`.

`trick_units(s)`

Definition at line 71 of file `time_converter_tai_tdb.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, and `TimeConverter_TAI_TDB()`.

**8.8.5.3** `int jeod::TimeConverter_TAI_TDB::nlter` `[private]`

Counter for number of iterations.

`trick_units(-)`

Definition at line 91 of file `time_converter_tai_tdb.hh`.

Referenced by `convert_b_to_a()`, and `TimeConverter_TAI_TDB()`.

**8.8.5.4** `int jeod::TimeConverter_TAI_TDB::nSteps` `[private]`

Counter for number of steps in iteration.

`trick_units(-)`

Definition at line 87 of file `time_converter_tai_tdb.hh`.

Referenced by `convert_b_to_a()`, and `TimeConverter_TAI_TDB()`.

**8.8.5.5** `double jeod::TimeConverter_TAI_TDB::prev_tai_seconds` `[private]`

TAI seconds from previous loop iteration.

`trick_units(s)`

Definition at line 79 of file `time_converter_tai_tdb.hh`.

Referenced by `convert_b_to_a()`, and `TimeConverter_TAI_TDB()`.

**8.8.5.6** `double jeod::TimeConverter_TAI_TDB::prev_tdb_seconds` `[private]`

TDB seconds from previous loop iteration.

`trick_units(s)`

Definition at line 83 of file `time_converter_tai_tdb.hh`.

Referenced by `convert_b_to_a()`, and `TimeConverter_TAI_TDB()`.

**8.8.5.7** `TimeTAI* jeod::TimeConverter_TAI_TDB::tai_ptr` `[private]`

Converter parent time, always a [TimeTAI](#) for this converter.

`trick_units(-)`

Definition at line 95 of file `time_converter_tai_tdb.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, `set_a_to_b_offset()`, and `TimeConverter_TAI_TDB()`.

#### 8.8.5.8 `double jeod::TimeConverter_TAI_TDB::TAI_to_TT_offset` [private]

The offset from TAI to TT.

`trick_units(s)`

Definition at line 67 of file `time_converter_tai_tdb.hh`.

Referenced by `initialize()`, and `TimeConverter_TAI_TDB()`.

#### 8.8.5.9 `TimeTDB* jeod::TimeConverter_TAI_TDB::tdb_ptr` [private]

Converter parent time, always a [TimeTDB](#) for this converter.

`trick_units(-)`

Definition at line 99 of file `time_converter_tai_tdb.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, and `TimeConverter_TAI_TDB()`.

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

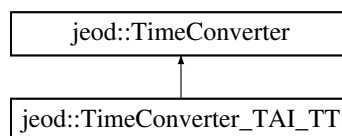
- [time\\_converter\\_tai\\_tdb.hh](#)
- [time\\_converter\\_tai\\_tdb.cc](#)

## 8.9 `jeod::TimeConverter_TAI_TT` Class Reference

Converts between International Atomic Time and Terrestrial Time.

```
#include <time_converter_tai_tt.hh>
```

Inheritance diagram for `jeod::TimeConverter_TAI_TT`:



### Public Member Functions

- [TimeConverter\\_TAI\\_TT](#) ()  
*Construct a [TimeConverter\\_TAI\\_TT](#).*
- [~TimeConverter\\_TAI\\_TT](#) ()  
*Destroy a [TimeConverter\\_TAI\\_TT](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeTAI](#) to [TimeTT](#).*
- void [convert\\_b\\_to\\_a](#) (void)  
*Convert from [TimeTT](#) to [TimeTAI](#).*

### Private Member Functions

- [TimeConverter\\_TAI\\_TT](#) (const [TimeConverter\\_TAI\\_TT](#) &)
- [TimeConverter\\_TAI\\_TT](#) & [operator=](#) (const [TimeConverter\\_TAI\\_TT](#) &)

## Private Attributes

- [TimeTAI](#) \* tai\_ptr  
Converter parent time, always a [TimeTAI](#) for this converter.
- [TimeTT](#) \* tt\_ptr  
Converter parent time, always a [TimeTT](#) for this converter.

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI\\_TT](#) ()

## Additional Inherited Members

### 8.9.1 Detailed Description

Converts between International Atomic Time and Terrestrial Time.

Definition at line 54 of file `time_converter_tai_tt.hh`.

### 8.9.2 Constructor & Destructor Documentation

#### 8.9.2.1 `jeod::TimeConverter_TAI_TT::TimeConverter_TAI_TT ( void )`

Construct a [TimeConverter\\_TAI\\_TT](#).

Definition at line 59 of file `time_converter_tai_tt.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::ANY_DIRECTION`, `jeod::TimeConverter::b_name`, `tai_ptr`, `tt_ptr`, and `jeod::TimeConverter::valid_directions`.

#### 8.9.2.2 `jeod::TimeConverter_TAI_TT::~~TimeConverter_TAI_TT ( void )`

Destroy a [TimeConverter\\_TAI\\_TT](#).

Definition at line 138 of file `time_converter_tai_tt.cc`.

#### 8.9.2.3 `jeod::TimeConverter_TAI_TT::TimeConverter_TAI_TT ( const TimeConverter_TAI_TT & ) [private]`

### 8.9.3 Member Function Documentation

#### 8.9.3.1 `void jeod::TimeConverter_TAI_TT::convert_a_to_b ( void ) [virtual]`

Convert from [TimeTAI](#) to [TimeTT](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 109 of file `time_converter_tai_tt.cc`.

References `jeod::JeodBaseTime::seconds`, `jeod::TimeStandard::set_time_by_seconds()`, `tai_ptr`, and `tt_ptr`.

#### 8.9.3.2 `void jeod::TimeConverter_TAI_TT::convert_b_to_a ( void ) [virtual]`

Convert from [TimeTT](#) to [TimeTAI](#).

## Assumptions and Limitations

- Time class MET is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from [jeod::TimeConverter](#).

Definition at line 126 of file `time_converter_tai_tt.cc`.

References `jeod::JeodBaseTime::seconds`, `jeod::TimeStandard::set_time_by_seconds()`, `tai_ptr`, and `tt_ptr`.

**8.9.3.3** `void jeod::TimeConverter_TAI_TT::initialize ( JeodBaseTime * parent_ptr, JeodBaseTime * child_ptr, const int int_dir ) [virtual]`

Initialize the converter.

### Parameters

in	<i>parent_ptr</i>	Time used to initialize the converter
in	<i>child_ptr</i>	Other Time used to initialize the converter
in	<i>int_dir</i>	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements [jeod::TimeConverter](#).

Definition at line 77 of file `time_converter_tai_tt.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `jeod::TimeConverter::initialized`, `jeod::TimeMessages::invalid_setup_error`, `tai_ptr`, `tt_ptr`, and `jeod::TimeConverter::verify_setup()`.

**8.9.3.4** `TimeConverter_TAI_TT& jeod::TimeConverter_TAI_TT::operator= ( const TimeConverter_TAI_TT & ) [private]`

## 8.9.4 Friends And Related Function Documentation

**8.9.4.1** `void init_attrjeod__TimeConverter_TAI_TT ( ) [friend]`

**8.9.4.2** `friend class InputProcessor [friend]`

Definition at line 56 of file `time_converter_tai_tt.hh`.

## 8.9.5 Field Documentation

**8.9.5.1** `TimeTAI* jeod::TimeConverter_TAI_TT::tai_ptr [private]`

Converter parent time, always a [TimeTAI](#) for this converter.

`trick_units(-)`

Definition at line 63 of file `time_converter_tai_tt.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, and `TimeConverter_TAI_TT()`.

**8.9.5.2** `TimeTT* jeod::TimeConverter_TAI_TT::tt_ptr [private]`

Converter parent time, always a [TimeTT](#) for this converter.

`trick_units(-)`

Definition at line 68 of file `time_converter_tai_tt.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, and `TimeConverter_TAI_TT()`.

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



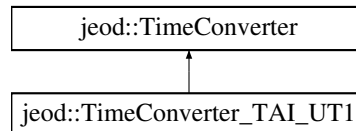
- [time\\_converter\\_tai\\_tt.hh](#)
- [time\\_converter\\_tai\\_tt.cc](#)

## 8.10 jeod::TimeConverter\_TAI\_UT1 Class Reference

Define class [TimeConverter\\_TAI\\_UT1](#), which converts between International Atomic Time and Universal Time.

```
#include <time_converter_tai_ut1.hh>
```

Inheritance diagram for jeod::TimeConverter\_TAI\_UT1:



### Public Member Functions

- [TimeConverter\\_TAI\\_UT1](#) ()  
*Construct a [TimeConverter\\_TAI\\_UT1](#).*
- [~TimeConverter\\_TAI\\_UT1](#) ()  
*Destroy a [TimeConverter\\_TAI\\_UT1](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeTAI](#) to [TimeUT1](#).*
- void [convert\\_b\\_to\\_a](#) (void)  
*Convert from [TimeUT1](#) to [TimeTAI](#).*

### Data Fields

- bool [override\\_data\\_table](#)  
*"True" to enter user-specified tai-ut1 offset*
- double [tai\\_to\\_ut1\\_override\\_val](#)  
*User specified value (UT1 - TAI)*
- int [last\\_index](#)  
*Index of last datum in table.*
- int [index](#)  
*Current location in table.*
- double \* [val\\_vec](#)  
*Vector of values of difference between TAI-UT1.*
- double \* [when\\_vec](#)  
*Vector of corresponding times.*

### Private Member Functions

- void [initialize\\_tai\\_to\\_ut1](#) (void)  
*The conversion from Atomic Time (TAI) to Universal Time (UT1) involves the addition of value that is a continuous function of TAI.*
- void [verify\\_table\\_lookup\\_ends](#) (void)

*Used when time reverses direction.*

- [TimeConverter\\_TAI\\_UT1](#) (const [TimeConverter\\_TAI\\_UT1](#) &)
- [TimeConverter\\_TAI\\_UT1](#) & operator= (const [TimeConverter\\_TAI\\_UT1](#) &)

## Private Attributes

- [TimeTAI](#) \* [tai\\_ptr](#)  
*Converter parent time, always a [TimeTAI](#) for this converter.*
- [TimeUT1](#) \* [ut1\\_ptr](#)  
*Converter parent time, always a [TimeUT1](#) for this converter.*
- double [prev\\_when](#)  
*Time of previous calibrated datum.*
- double [prev\\_value](#)  
*Offset value of previous datum.*
- double [next\\_when](#)  
*Time of next calibrated datum.*
- double [next\\_value](#)  
*Offset value of next datum.*
- double [gradient](#)  
*Rate at which "value" changes wrt "when".*
- bool [off\\_table\\_end](#)  
*Gone past the end of the table.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI\\_UT1](#) ()

## Additional Inherited Members

### 8.10.1 Detailed Description

Define class [TimeConverter\\_TAI\\_UT1](#), which converts between International Atomic Time and Universal Time.

Definition at line 56 of file [time\\_converter\\_tai\\_ut1.hh](#).

### 8.10.2 Constructor & Destructor Documentation

#### 8.10.2.1 [jeod::TimeConverter\\_TAI\\_UT1::TimeConverter\\_TAI\\_UT1](#) ( void )

Construct a [TimeConverter\\_TAI\\_UT1](#).

Definition at line 63 of file [time\\_converter\\_tai\\_ut1.cc](#).

References [jeod::TimeConverter::a\\_name](#), [jeod::TimeConverter::ANY\\_DIRECTION](#), [jeod::TimeConverter::b\\_name](#), [gradient](#), [index](#), [last\\_index](#), [next\\_value](#), [next\\_when](#), [off\\_table\\_end](#), [override\\_data\\_table](#), [prev\\_value](#), [prev\\_when](#), [tai\\_ptr](#), [tai\\_to\\_ut1\\_override\\_val](#), [ut1\\_ptr](#), [val\\_vec](#), [jeod::TimeConverter::valid\\_directions](#), and [when\\_vec](#).

#### 8.10.2.2 [jeod::TimeConverter\\_TAI\\_UT1::~~TimeConverter\\_TAI\\_UT1](#) ( void )

Destroy a [TimeConverter\\_TAI\\_UT1](#).

Definition at line 494 of file [time\\_converter\\_tai\\_ut1.cc](#).

References [val\\_vec](#), and [when\\_vec](#).

8.10.2.3 `jeod::TimeConverter_TAI_UT1::TimeConverter_TAI_UT1 ( const TimeConverter_TAI_UT1 & ) [private]`

### 8.10.3 Member Function Documentation

8.10.3.1 `void jeod::TimeConverter_TAI_UT1::convert_a_to_b ( void ) [virtual]`

Convert from [TimeTAI](#) to [TimeUT1](#).

#### Assumptions and Limitations

- Time class MET is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from [jeod::TimeConverter](#).

Definition at line 261 of file `time_converter_tai_ut1.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `gradient`, `index`, `jeod::TimeMessages::invalid_data_error`, `last_index`, `next_value`, `next_when`, `off_table_end`, `prev_value`, `prev_when`, `jeod::TimeStandard::set_time_by_trunc_julian()`, `tai_ptr`, `jeod::TimeUT1::true_ut1`, `jeod::TimeStandard::trunc_julian_time`, `ut1_ptr`, `val_vec`, and `when_vec`.

8.10.3.2 `void jeod::TimeConverter_TAI_UT1::convert_b_to_a ( void ) [virtual]`

Convert from [TimeUT1](#) to [TimeTAI](#).

#### Assumptions and Limitations

- Time class MET is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from [jeod::TimeConverter](#).

Definition at line 356 of file `time_converter_tai_ut1.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `gradient`, `index`, `jeod::TimeMessages::invalid_data_error`, `last_index`, `next_value`, `next_when`, `off_table_end`, `prev_value`, `prev_when`, `jeod::TimeStandard::set_time_by_trunc_julian()`, `tai_ptr`, `jeod::TimeUT1::true_ut1`, `jeod::TimeStandard::trunc_julian_time`, `ut1_ptr`, `val_vec`, and `when_vec`.

8.10.3.3 `void jeod::TimeConverter_TAI_UT1::initialize ( JeodBaseTime * parent_ptr, JeodBaseTime * child_ptr, const int int_dir ) [virtual]`

Initialize the converter.

#### Assumptions and Limitations

- None

#### Parameters

in	<i>parent_ptr</i>	Time used to initialize the converter
in	<i>child_ptr</i>	Other Time used to initialize the converter
in	<i>int_dir</i>	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements [jeod::TimeConverter](#).

Definition at line 98 of file `time_converter_tai_ut1.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `jeod::TimeMessages::initialization_error`, `initialize_tai_to_ut1()`, `jeod::TimeConverter::initialized`, `next_when`, `prev_when`, `tai_ptr`, `jeod::TimeStandard::trunc_julian_time`, `ut1_ptr`, and `jeod::TimeConverter::verify_setup()`.

#### 8.10.3.4 void jeod::TimeConverter\_TAI\_UT1::initialize\_tai\_to\_ut1 ( void ) [private]

The conversion from Atomic Time (TAI) to Universal Time (UT1) involves the addition of value that is a continuous function of TAI.

That value is tabulated at regular points of TAI. This function initializes that table and sets the preliminary values.

##### Assumptions and Limitations

- The table does not go into the future.

Definition at line 148 of file time\_converter\_tai\_ut1.cc.

References jeod::TimeConverter::a\_to\_b\_offset, jeod::TimeManager::dyn\_time, gradient, index, jeod::TimeMessages::invalid\_data\_error, last\_index, next\_value, next\_when, off\_table\_end, override\_data\_table, prev\_value, prev\_when, jeod::TimeDyn::scale\_factor, tai\_ptr, tai\_to\_ut1\_override\_val, jeod::JeodBaseTime::time\_manager, jeod::TimeUT1::true\_ut1, jeod::TimeStandard::trunc\_julian\_time, ut1\_ptr, val\_vec, and when\_vec.

Referenced by initialize().

#### 8.10.3.5 TimeConverter\_TAI\_UT1& jeod::TimeConverter\_TAI\_UT1::operator= ( const TimeConverter\_TAI\_UT1 & ) [private]

#### 8.10.3.6 void jeod::TimeConverter\_TAI\_UT1::verify\_table\_lookup\_ends ( void ) [private],[virtual]

Used when time reverses direction.

Checks whether the table lookup function is using input values that are outside the scope of the table, and sets the flags appropriately

##### Assumptions and Limitations

- None

Reimplemented from [jeod::TimeConverter](#).

Definition at line 452 of file time\_converter\_tai\_ut1.cc.

References jeod::TimeManager::dyn\_time, index, last\_index, next\_when, off\_table\_end, prev\_when, jeod::TimeDyn::scale\_factor, tai\_ptr, jeod::JeodBaseTime::time\_manager, jeod::TimeUT1::true\_ut1, jeod::TimeStandard::trunc\_julian\_time, ut1\_ptr, and when\_vec.

### 8.10.4 Friends And Related Function Documentation

#### 8.10.4.1 void init\_attrjeod\_\_TimeConverter\_TAI\_UT1 ( ) [friend]

#### 8.10.4.2 friend class InputProcessor [friend]

Definition at line 58 of file time\_converter\_tai\_ut1.hh.

### 8.10.5 Field Documentation

#### 8.10.5.1 double jeod::TimeConverter\_TAI\_UT1::gradient [private]

Rate at which "value" changes wrt "when".

trick\_units(-)

Definition at line 122 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_tai\_to\_ut1(), and TimeConverter\_TAI\_UT1().

**8.10.5.2 int jeod::TimeConverter\_TAI\_UT1::index**

Current location in table.

trick\_units(-)

Definition at line 92 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_tai\_to\_ut1(), TimeConverter\_TAI\_UT1(), and verify\_table\_lookup\_ends().

**8.10.5.3 int jeod::TimeConverter\_TAI\_UT1::last\_index**

Index of last datum in table.

trick\_units(-)

Definition at line 88 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1\_default\_data::initialize(), initialize\_tai\_to\_ut1(), TimeConverter\_TAI\_UT1(), and verify\_table\_lookup\_ends().

**8.10.5.4 double jeod::TimeConverter\_TAI\_UT1::next\_value [private]**

Offset value of next datum.

trick\_units(s)

Definition at line 118 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_tai\_to\_ut1(), and TimeConverter\_TAI\_UT1().

**8.10.5.5 double jeod::TimeConverter\_TAI\_UT1::next\_when [private]**

Time of next calibrated datum.

trick\_units(day)

Definition at line 114 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), initialize\_tai\_to\_ut1(), TimeConverter\_TAI\_UT1(), and verify\_table\_lookup\_ends().

**8.10.5.6 bool jeod::TimeConverter\_TAI\_UT1::off\_table\_end [private]**

Gone past the end of the table.

trick\_units(-)

Definition at line 126 of file time\_converter\_tai\_ut1.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_tai\_to\_ut1(), TimeConverter\_TAI\_UT1(), and verify\_table\_lookup\_ends().

**8.10.5.7 bool jeod::TimeConverter\_TAI\_UT1::override\_data\_table**

"True" to enter user-specified tai-ut1 offset

trick\_units(-)

Definition at line 65 of file time\_converter\_tai\_ut1.hh.

Referenced by jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1\_default\_data::initialize(), initialize\_tai\_to\_ut1(), TimeConverter\_TAI\_UT1(), and jeod::TimeManagerInit::verify\_converter\_setup().

**8.10.5.8** `double jeod::TimeConverter_TAI_UT1::prev_value` `[private]`

Offset value of previous datum.

trick\_units(s)

Definition at line 110 of file `time_converter_tai_ut1.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize_tai_to_ut1()`, and `TimeConverter_TAI_UT1()`.

**8.10.5.9** `double jeod::TimeConverter_TAI_UT1::prev_when` `[private]`

Time of previous calibrated datum.

trick\_units(day)

Definition at line 106 of file `time_converter_tai_ut1.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, `initialize_tai_to_ut1()`, `TimeConverter_TAI_UT1()`, and `verify_table_lookup_ends()`.

**8.10.5.10** `TimeTAI* jeod::TimeConverter_TAI_UT1::tai_ptr` `[private]`

Converter parent time, always a [TimeTAI](#) for this converter.

trick\_units(-)

Definition at line 72 of file `time_converter_tai_ut1.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, `initialize_tai_to_ut1()`, `TimeConverter_TAI_UT1()`, and `verify_table_lookup_ends()`.

**8.10.5.11** `double jeod::TimeConverter_TAI_UT1::tai_to_ut1_override_val`

User specified value (UT1 - TAI)

trick\_units(s)

Definition at line 83 of file `time_converter_tai_ut1.hh`.

Referenced by `initialize_tai_to_ut1()`, and `TimeConverter_TAI_UT1()`.

**8.10.5.12** `TimeUT1* jeod::TimeConverter_TAI_UT1::ut1_ptr` `[private]`

Converter parent time, always a [TimeUT1](#) for this converter.

trick\_units(-)

Definition at line 77 of file `time_converter_tai_ut1.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, `initialize_tai_to_ut1()`, `TimeConverter_TAI_UT1()`, and `verify_table_lookup_ends()`.

**8.10.5.13** `double* jeod::TimeConverter_TAI_UT1::val_vec`

Vector of values of difference between TAI-UT1.

trick\_units(s)

Definition at line 96 of file `time_converter_tai_ut1.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data::initialize()`, `initialize_tai_to_ut1()`, `TimeConverter_TAI_UT1()`, and `~TimeConverter_TAI_UT1()`.

## 8.10.5.14 double\* jeod::TimeConverter\_TAI\_UT1::when\_vec

Vector of corresponding times.

trick\_units(day)

Definition at line 100 of file time\_converter\_tai\_ut1.hh.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `jeod::TimeConverter_TAI_UT1_tai_to_ut1_default_data::initialize()`, `initialize_tai_to_ut1()`, `TimeConverter_TAI_UT1()`, `verify_table_lookup_ends()`, and `~TimeConverter_TAI_UT1()`.

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

- [time\\_converter\\_tai\\_ut1.hh](#)
- [time\\_converter\\_tai\\_ut1.cc](#)

## 8.11 jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1\_default\_data Class Reference

```
#include <tai_to_ut1.hh>
```

### Public Member Functions

- void [initialize](#) ([TimeConverter\\_TAI\\_UT1 \\*](#))

### 8.11.1 Detailed Description

Definition at line 18 of file tai\_to\_ut1.hh.

### 8.11.2 Member Function Documentation

#### 8.11.2.1 void jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1\_default\_data::initialize ( [TimeConverter\\_TAI\\_UT1 \\*](#) [TimeConverter\\_TAI\\_UT1\\_ptr](#) )

Definition at line 45 of file tai\_to\_ut1.cc.

References `jeod::TimeConverter_TAI_UT1::last_index`, `jeod::TimeConverter_TAI_UT1::override_data_table`, `jeod::TimeConverter_TAI_UT1::val_vec`, and `jeod::TimeConverter_TAI_UT1::when_vec`.

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

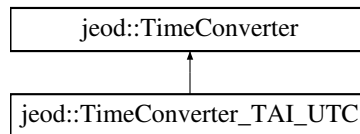
- [tai\\_to\\_ut1.hh](#)
- [tai\\_to\\_ut1.cc](#)

## 8.12 jeod::TimeConverter\_TAI\_UTC Class Reference

Converts between International Atomic Time and Coordinated Universal Time.

```
#include <time_converter_tai_utc.hh>
```

Inheritance diagram for `jeod::TimeConverter_TAI_UTC`:



## Public Member Functions

- [TimeConverter\\_TAI\\_UTC](#) ()  
Construct a *TimeConverter\_TAI\_UTC*.
- [~TimeConverter\\_TAI\\_UTC](#) ()  
Destroy a *TimeConverter\_TAI\_UTC*.
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
Initialize the converter.
- void [convert\\_a\\_to\\_b](#) (void)  
Convert from *TimeTAI* to *TimeUTC*.
- void [convert\\_b\\_to\\_a](#) (void)  
Convert from *TimeUTC* to *TimeTAI*.

## Data Fields

- bool [override\\_data\\_table](#)  
"True" to enter user-specified tai-utc offset
- double [leap\\_sec\\_override\\_val](#)  
User specified value (TAI - UTC)
- int [last\\_index](#)  
Maximum index in the leap tables.
- int [index](#)  
Current index in the leap tables.
- int \* [val\\_vec](#)  
Tabulated values of leap\_value.
- double \* [when\\_vec](#)  
Tabulated values of Julian time corresponding to changes in leap\_value.

## Private Member Functions

- void [initialize\\_leap\\_second](#) (void)  
The conversion from Atomic Time (TAI) to Universal Time (UTC) involves the addition of leap seconds.
- void [verify\\_table\\_lookup\\_ends](#) (void)  
Used when time reverses direction.
- [TimeConverter\\_TAI\\_UTC](#) (const [TimeConverter\\_TAI\\_UTC](#) &)
- [TimeConverter\\_TAI\\_UTC](#) & [operator=](#) (const [TimeConverter\\_TAI\\_UTC](#) &)

## Private Attributes

- [TimeTAI](#) \* [tai\\_ptr](#)  
Converter parent time, always a *TimeTAI* for this converter.
- [TimeUTC](#) \* [utc\\_ptr](#)  
Converter parent time, always a *TimeUTC* for this converter.
- double [next\\_when](#)



*The next (future) UTC time of a leap second instance.*

- double [prev\\_when](#)

*The most recent (past) UTC time of a leap second instance.*

- bool [off\\_table\\_end](#)

*Flag to indicate that the current time is not covered by the leap-second tables.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_TAI\\_UTC](#) ()

## Additional Inherited Members

### 8.12.1 Detailed Description

Converts between International Atomic Time and Coordinated Universal Time.

Definition at line 57 of file `time_converter_tai_utc.hh`.

### 8.12.2 Constructor & Destructor Documentation

#### 8.12.2.1 jeod::TimeConverter\_TAI\_UTC::TimeConverter\_TAI\_UTC ( void )

Construct a [TimeConverter\\_TAI\\_UTC](#).

Definition at line 62 of file `time_converter_tai_utc.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::A_TO_B`, `jeod::TimeConverter::b_name`, `jeod::TimeConverter::B_TO_A_INIT`, `index`, `last_index`, `leap_sec_override_val`, `next_when`, `off_table_end`, `override_data_table`, `prev_when`, `tai_ptr`, `utc_ptr`, `val_vec`, `jeod::TimeConverter::valid_directions`, and `when_vec`.

#### 8.12.2.2 jeod::TimeConverter\_TAI\_UTC::~~TimeConverter\_TAI\_UTC ( void )

Destroy a [TimeConverter\\_TAI\\_UTC](#).

Definition at line 462 of file `time_converter_tai_utc.cc`.

References `val_vec`, and `when_vec`.

#### 8.12.2.3 jeod::TimeConverter\_TAI\_UTC::TimeConverter\_TAI\_UTC ( const TimeConverter\_TAI\_UTC & ) [private]

### 8.12.3 Member Function Documentation

#### 8.12.3.1 void jeod::TimeConverter\_TAI\_UTC::convert\_a\_to\_b ( void ) [virtual]

Convert from [TimeTAI](#) to [TimeUTC](#).

## Assumptions and Limitations

- Time class MET is based on time class TAI, and counts the elapsed TAI time only

Reimplemented from [jeod::TimeConverter](#).

Definition at line 267 of file `time_converter_tai_utc.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `jeod::TimeManager::dyn_time`, `index`, `jeod::TimeMessages::invalid_data_error`, `last_index`, `next_when`, `off_table_end`, `prev_when`, `jeod::TimeDyn::scale_factor`, `jeod::TimeStandard::set_time_by_trunc_julian()`, `tai_ptr`, `jeod::JeodBaseTime::time_manager`, `jeod::TimeUTC::true_utc`, `jeod::TimeStandard::trunc_julian_time`, `utc_ptr`, `val_vec`, and `when_vec`.

#### 8.12.3.2 `void jeod::TimeConverter_TAI_UTC::convert_b_to_a ( void ) [virtual]`

Convert from [TimeUTC](#) to [TimeTAI](#).

Reimplemented from [jeod::TimeConverter](#).

Definition at line 353 of file `time_converter_tai_utc.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `jeod::TimeManager::dyn_time`, `index`, `jeod::TimeMessages::invalid_data_error`, `last_index`, `next_when`, `off_table_end`, `prev_when`, `jeod::TimeDyn::scale_factor`, `jeod::TimeStandard::set_time_by_trunc_julian()`, `tai_ptr`, `jeod::JeodBaseTime::time_manager`, `jeod::TimeUTC::true_utc`, `jeod::TimeStandard::trunc_julian_time`, `utc_ptr`, `val_vec`, and `when_vec`.

#### 8.12.3.3 `void jeod::TimeConverter_TAI_UTC::initialize ( JeodBaseTime * parent_ptr, JeodBaseTime * child_ptr, const int int_dir ) [virtual]`

Initialize the converter.

##### Parameters

<code>in</code>	<code>parent_ptr</code>	Time used to initialize the converter
<code>in</code>	<code>child_ptr</code>	Other Time used to initialize the converter
<code>in</code>	<code>int_dir</code>	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements [jeod::TimeConverter](#).

Definition at line 91 of file `time_converter_tai_utc.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `index`, `jeod::TimeMessages::initialization_error`, `initialize_leap_second()`, `jeod::TimeConverter::initialized`, `jeod::JeodBaseTime::is_initialized()`, `tai_ptr`, `jeod::TimeStandard::trunc_julian_time`, `utc_ptr`, `val_vec`, `jeod::TimeConverter::verify_setup()`, and `when_vec`.

#### 8.12.3.4 `void jeod::TimeConverter_TAI_UTC::initialize_leap_second ( void ) [private]`

The conversion from Atomic Time (TAI) to Universal Time (UTC) involves the addition of leap seconds.

The number of leap seconds at any given (historical) time is provided in a table. This function initializes that table and sets the preliminary values.

##### Assumptions and Limitations

- The table does not go into the future.

Definition at line 151 of file `time_converter_tai_utc.cc`.

References `jeod::TimeConverter::a_to_b_offset`, `jeod::TimeManager::dyn_time`, `index`, `jeod::TimeMessages::invalid_data_error`, `jeod::TimeMessages::invalid_setup_error`, `last_index`, `leap_sec_override_val`, `next_when`, `off_table_end`, `override_data_table`, `prev_when`, `jeod::TimeDyn::scale_factor`, `tai_ptr`, `jeod::JeodBaseTime::time_manager`, `jeod::TimeUTC::true_utc`, `jeod::TimeStandard::trunc_julian_time`, `utc_ptr`, `val_vec`, and `when_vec`.

Referenced by `initialize()`.

#### 8.12.3.5 `TimeConverter_TAI_UTC& jeod::TimeConverter_TAI_UTC::operator= ( const TimeConverter_TAI_UTC & ) [private]`

**8.12.3.6** `void jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends ( void )` `[private]`, `[virtual]`

Used when time reverses direction.

Checks whether the table lookup function is using input values that are outside the scope of the table, and sets the flags appropriately

Reimplemented from [jeod::TimeConverter](#).

Definition at line 421 of file `time_converter_tai_utc.cc`.

References `jeod::TimeManager::dyn_time`, `index`, `last_index`, `next_when`, `off_table_end`, `prev_when`, `jeod::TimeDyn::scale_factor`, `tai_ptr`, `jeod::JeodBaseTime::time_manager`, `jeod::TimeUTC::true_utc`, `jeod::TimeStandard::trunc_julian_time`, `utc_ptr`, and `when_vec`.

## 8.12.4 Friends And Related Function Documentation

**8.12.4.1** `void init_attrjeod__TimeConverter_TAI_UTC ( )` `[friend]`

**8.12.4.2** `friend class InputProcessor` `[friend]`

Definition at line 59 of file `time_converter_tai_utc.hh`.

## 8.12.5 Field Documentation

**8.12.5.1** `int jeod::TimeConverter_TAI_UTC::index`

Current index in the leap tables.

`trick_units(-)`

Definition at line 91 of file `time_converter_tai_utc.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `initialize()`, `initialize_leap_second()`, `TimeConverter_TAI_UTC()`, and `verify_table_lookup_ends()`.

**8.12.5.2** `int jeod::TimeConverter_TAI_UTC::last_index`

Maximum index in the leap tables.

`trick_units(-)`

Definition at line 87 of file `time_converter_tai_utc.hh`.

Referenced by `convert_a_to_b()`, `convert_b_to_a()`, `jeod::TimeConverter_TAI_UTC_tai_to_utc_default_data::initialize()`, `initialize_leap_second()`, `TimeConverter_TAI_UTC()`, and `verify_table_lookup_ends()`.

**8.12.5.3** `double jeod::TimeConverter_TAI_UTC::leap_sec_override_val`

User specified value (TAI - UTC)

`trick_units(s)`

Definition at line 83 of file `time_converter_tai_utc.hh`.

Referenced by `initialize_leap_second()`, and `TimeConverter_TAI_UTC()`.

**8.12.5.4** `double jeod::TimeConverter_TAI_UTC::next_when` `[private]`

The next (future) UTC time of a leap second instance.

trick\_units(-)

Definition at line 105 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and verify\_table\_lookup\_ends().

#### 8.12.5.5 bool jeod::TimeConverter\_TAI\_UTC::off\_table\_end [private]

Flag to indicate that the current time is not covered by the leap-second tables.

trick\_units(-)

Definition at line 115 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and verify\_table\_lookup\_ends().

#### 8.12.5.6 bool jeod::TimeConverter\_TAI\_UTC::override\_data\_table

"True" to enter user-specified tai-utc offset

trick\_units(-)

Definition at line 66 of file time\_converter\_tai\_utc.hh.

Referenced by jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data::initialize(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and jeod::TimeManagerInit::verify\_converter\_setup().

#### 8.12.5.7 double jeod::TimeConverter\_TAI\_UTC::prev\_when [private]

The most recent (past) UTC time of a leap second instance.

trick\_units(-)

Definition at line 110 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and verify\_table\_lookup\_ends().

#### 8.12.5.8 TimeTAI\* jeod::TimeConverter\_TAI\_UTC::tai\_ptr [private]

Converter parent time, always a [TimeTAI](#) for this converter.

trick\_units(-)

Definition at line 72 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and verify\_table\_lookup\_ends().

#### 8.12.5.9 TimeUTC\* jeod::TimeConverter\_TAI\_UTC::utc\_ptr [private]

Converter parent time, always a [TimeUTC](#) for this converter.

trick\_units(-)

Definition at line 77 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), initialize(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and verify\_table\_lookup\_ends().

## 8.12.5.10 int\* jeod::TimeConverter\_TAI\_UTC::val\_vec

Tabulated values of leap\_value.

trick\_units(s)

Definition at line 95 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data::initialize(), initialize(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), and ~TimeConverter\_TAI\_UTC().

## 8.12.5.11 double\* jeod::TimeConverter\_TAI\_UTC::when\_vec

Tabulated values of Julian time corresponding to changes in leap\_value.

trick\_units(day)

Definition at line 100 of file time\_converter\_tai\_utc.hh.

Referenced by convert\_a\_to\_b(), convert\_b\_to\_a(), jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data::initialize(), initialize(), initialize\_leap\_second(), TimeConverter\_TAI\_UTC(), verify\_table\_lookup\_ends(), and ~TimeConverter\_TAI\_UTC().

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

- [time\\_converter\\_tai\\_utc.hh](#)
- [time\\_converter\\_tai\\_utc.cc](#)

## 8.13 jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data Class Reference

```
#include <tai_to_utc.hh>
```

## Public Member Functions

- void [initialize](#) (TimeConverter\_TAI\_UTC \*)

## 8.13.1 Detailed Description

Definition at line 18 of file tai\_to\_utc.hh.

## 8.13.2 Member Function Documentation

## 8.13.2.1 void jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data::initialize ( TimeConverter\_TAI\_UTC \* TimeConverter\_TAI\_UTC\_ptr )

Definition at line 40 of file tai\_to\_utc.cc.

References jeod::TimeConverter\_TAI\_UTC::last\_index, jeod::TimeConverter\_TAI\_UTC::override\_data\_table, jeod::TimeConverter\_TAI\_UTC::val\_vec, and jeod::TimeConverter\_TAI\_UTC::when\_vec.

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

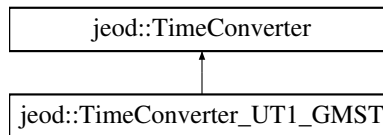
- [tai\\_to\\_utc.hh](#)
- [tai\\_to\\_utc.cc](#)

## 8.14 jeod::TimeConverter\_UT1\_GMST Class Reference

Converts between Universal Time and Greenwich Mean Sidereal Time.

```
#include <time_converter_ut1_gmst.hh>
```

Inheritance diagram for jeod::TimeConverter\_UT1\_GMST:



### Public Member Functions

- [TimeConverter\\_UT1\\_GMST](#) ()  
*Construct a [TimeConverter\\_UT1\\_GMST](#).*
- [~TimeConverter\\_UT1\\_GMST](#) ()  
*Destroy a [TimeConverter\\_UT1\\_GMST](#).*
- void [initialize](#) ([JeodBaseTime](#) \*parent, [JeodBaseTime](#) \*child, const int direction)  
*Initialize the converter.*
- void [convert\\_a\\_to\\_b](#) (void)  
*Convert from [TimeUT1](#) to [TimeGMST](#).*

### Private Member Functions

- [TimeConverter\\_UT1\\_GMST](#) (const [TimeConverter\\_UT1\\_GMST](#) &)
- [TimeConverter\\_UT1\\_GMST](#) & operator= (const [TimeConverter\\_UT1\\_GMST](#) &)

### Private Attributes

- [TimeUT1](#) \* [ut1\\_ptr](#)  
*Converter parent time, always a [TimeUT1](#) for this converter.*
- [TimeGMST](#) \* [gmst\\_ptr](#)  
*Converter parent time, always a [TimeGMST](#) for this converter.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeConverter\\_UT1\\_GMST](#) ()

### Additional Inherited Members

#### 8.14.1 Detailed Description

Converts between Universal Time and Greenwich Mean Sidereal Time.

Definition at line 54 of file [time\\_converter\\_ut1\\_gmst.hh](#).

## 8.14.2 Constructor & Destructor Documentation

### 8.14.2.1 jeod::TimeConverter\_UT1\_GMST::TimeConverter\_UT1\_GMST ( void )

Construct a [TimeConverter\\_UT1\\_GMST](#).

Definition at line 59 of file `time_converter_ut1_gmst.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::A_TO_B`, `jeod::TimeConverter::b_name`, `gmst_ptr`, `ut1_ptr`, and `jeod::TimeConverter::valid_directions`.

### 8.14.2.2 jeod::TimeConverter\_UT1\_GMST::~~TimeConverter\_UT1\_GMST ( void )

Destroy a [TimeConverter\\_UT1\\_GMST](#).

Definition at line 149 of file `time_converter_ut1_gmst.cc`.

### 8.14.2.3 jeod::TimeConverter\_UT1\_GMST::TimeConverter\_UT1\_GMST ( const TimeConverter\_UT1\_GMST & ) [private]

## 8.14.3 Member Function Documentation

### 8.14.3.1 void jeod::TimeConverter\_UT1\_GMST::convert\_a\_to\_b ( void ) [virtual]

Convert from [TimeUT1](#) to [TimeGMST](#).

#### Assumptions and Limitations

- None

Reimplemented from [jeod::TimeConverter](#).

Definition at line 112 of file `time_converter_ut1_gmst.cc`.

References `jeod::TimeUT1::get_days()`, `gmst_ptr`, `jeod::TimeStandard::set_time_by_days()`, and `ut1_ptr`.

### 8.14.3.2 void jeod::TimeConverter\_UT1\_GMST::initialize ( JeodBaseTime \* parent\_ptr, JeodBaseTime \* child\_ptr, const int int\_dir ) [virtual]

Initialize the converter.

#### Assumptions and Limitations

- None

#### Parameters

in	<i>parent_ptr</i>	Time used to initialize the converter
in	<i>child_ptr</i>	Other Time used to initialize the converter
in	<i>int_dir</i>	Conversion direction: +1 a=parent; -1 b=parent; 0 error

Implements [jeod::TimeConverter](#).

Definition at line 80 of file `time_converter_ut1_gmst.cc`.

References `gmst_ptr`, `jeod::TimeConverter::initialized`, `jeod::TimeMessages::invalid_setup_error`, `ut1_ptr`, and `jeod::TimeConverter::verify_setup()`.

8.14.3.3 `TimeConverter_UT1_GMST& jeod::TimeConverter_UT1_GMST::operator= ( const TimeConverter_UT1_GMST & ) [private]`

#### 8.14.4 Friends And Related Function Documentation

8.14.4.1 `void init_attrjeod__TimeConverter_UT1_GMST ( ) [friend]`

8.14.4.2 `friend class InputProcessor [friend]`

Definition at line 56 of file `time_converter_ut1_gmst.hh`.

#### 8.14.5 Field Documentation

8.14.5.1 `TimeGMST* jeod::TimeConverter_UT1_GMST::gmst_ptr [private]`

Converter parent time, always a [TimeGMST](#) for this converter.

`trick_units(-)`

Definition at line 68 of file `time_converter_ut1_gmst.hh`.

Referenced by `convert_a_to_b()`, `initialize()`, and `TimeConverter_UT1_GMST()`.

8.14.5.2 `TimeUT1* jeod::TimeConverter_UT1_GMST::ut1_ptr [private]`

Converter parent time, always a [TimeUT1](#) for this converter.

`trick_units(-)`

Definition at line 63 of file `time_converter_ut1_gmst.hh`.

Referenced by `convert_a_to_b()`, `initialize()`, and `TimeConverter_UT1_GMST()`.

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

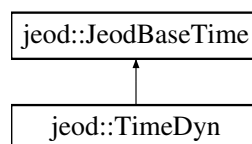
- [time\\_converter\\_ut1\\_gmst.hh](#)
- [time\\_converter\\_ut1\\_gmst.cc](#)

### 8.15 jeod::TimeDyn Class Reference

Represents the Dynamic Time in the simulation.

```
#include <time_dyn.hh>
```

Inheritance diagram for `jeod::TimeDyn`:



#### Public Member Functions

- [TimeDyn](#) ()  
Construct a *Time\_Dyn*.
- [~TimeDyn](#) ()



*Destroy a Time\_Dyn.*

- bool [update\\_offset](#) (void)

*Changeing time direction and/or scale factor.*

## Data Fields

- double [scale\\_factor](#)

*Multiplicative difference between sim-time and dyn-time.*

## Private Member Functions

- void [initialize\\_initializer\\_time](#) (TimeManagerInit \*tm\_init)  
*Each time type is initialized from its parent in the initialization tree, except one.*
- void [update](#) (void)  
*TimeDyn updates directly from simtime, and everything else from TimeDyn.*
- TimeDyn (const TimeDyn &)
- TimeDyn & operator= (const TimeDyn &)

## Private Attributes

- double [ref\\_scale](#)  
*Private copy of scale\_factor.*
- double [offset](#)  
*Extrapolated difference between sim-time and dyn-time at the sim-start (0 if there are no changes to direction or scale)*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeDyn](#) ()

## Additional Inherited Members

### 8.15.1 Detailed Description

Represents the Dynamic Time in the simulation.

Definition at line 53 of file time\_dyn.hh.

### 8.15.2 Constructor & Destructor Documentation

#### 8.15.2.1 jeod::TimeDyn::TimeDyn ( void )

Construct a Time\_Dyn.

Definition at line 58 of file time\_dyn.cc.

References [jeod::JeodBaseTime::links](#), [jeod::JeodBaseTime::name](#), [offset](#), [ref\\_scale](#), and [scale\\_factor](#).

#### 8.15.2.2 jeod::TimeDyn::~~TimeDyn ( void )

Destroy a Time\_Dyn.

Definition at line 149 of file time\_dyn.cc.

8.15.2.3 `jeod::TimeDyn::TimeDyn ( const TimeDyn & ) [private]`

### 8.15.3 Member Function Documentation

8.15.3.1 `void jeod::TimeDyn::initialize_initializer_time ( TimeManagerInit * time_manager_init ) [private], [virtual]`

Each time type is initialized from its parent in the initialization tree, except one.

In order to have an absolute reference time, one of the time types must be defined ahead of time. This is called the initializer time. This function initializes the initializer time.

#### Assumptions and Limitations

- [TimeDyn](#) cannot be used as the initializer time
- Each time representation can have its own initializer function, or can inherit the one in [TimeDerived](#)

#### Parameters

in	<i>time_manager_init</i>	TM initializer
----	--------------------------	----------------

Implements [jeod::JeodBaseTime](#).

Definition at line 82 of file `time_dyn.cc`.

References [jeod::JeodBaseTime::initialized](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [jeod::JeodBaseTime::seconds](#), [jeod::JeodBaseTime::time\\_manager](#), and [jeod::TimeManager::time\\_standards\\_exist\(\)](#).

8.15.3.2 `TimeDyn& jeod::TimeDyn::operator= ( const TimeDyn & ) [private]`

8.15.3.3 `void jeod::TimeDyn::update ( void ) [private], [virtual]`

[TimeDyn](#) updates directly from `simtime`, and everything else from [TimeDyn](#).

This function does that first update from `simtime`

#### Assumptions and Limitations

- `ref_scale` is positive for forward-progressing sims, and negative for reverse-progressing sims.

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 110 of file `time_dyn.cc`.

References `offset`, `ref_scale`, [jeod::JeodBaseTime::seconds](#), [jeod::TimeManager::simtime](#), and [jeod::JeodBaseTime::time\\_manager](#).

8.15.3.4 `bool jeod::TimeDyn::update_offset ( void )`

Changing time direction and/or scale factor.

#### Returns

Void

Definition at line 124 of file `time_dyn.cc`.

References `offset`, `ref_scale`, `scale_factor`, [jeod::JeodBaseTime::seconds](#), [jeod::TimeManager::simtime](#), [jeod::JeodBaseTime::time\\_manager](#), and [jeod::TimeManager::verify\\_table\\_lookup\\_ends\(\)](#).

Referenced by [jeod::TimeManager::update\(\)](#).

## 8.15.4 Friends And Related Function Documentation

8.15.4.1 `void init_attrjeod__TimeDyn ( )` `[friend]`

8.15.4.2 `friend class InputProcessor` `[friend]`

Definition at line 55 of file `time_dyn.hh`.

## 8.15.5 Field Documentation

8.15.5.1 `double jeod::TimeDyn::offset` `[private]`

Extrapolated difference between sim-time and dyn-time at the sim-start (0 if there are no changes to direction or scale)

`trick_units(-)`

Definition at line 76 of file `time_dyn.hh`.

Referenced by `TimeDyn()`, `update()`, and `update_offset()`.

8.15.5.2 `double jeod::TimeDyn::ref_scale` `[private]`

Private copy of `scale_factor`.

This value should not be changed externally; it is used for comparison purposes to identify when "scale\_factor" has changed.`trick_units(-)`

Definition at line 71 of file `time_dyn.hh`.

Referenced by `TimeDyn()`, `update()`, and `update_offset()`.

8.15.5.3 `double jeod::TimeDyn::scale_factor`

Multiplicative difference between sim-time and dyn-time.

This is the value that is changed externally.`trick_units(-)`

Definition at line 63 of file `time_dyn.hh`.

Referenced by `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `jeod::TimeManager::get_time_scale_factor()`, `jeod::TimeConverter_TAI_UTC::initialize_leap_second()`, `jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1()`, `TimeDyn()`, `update_offset()`, `jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends()`, and `jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends()`.

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

- [time\\_dyn.hh](#)
- [time\\_dyn.cc](#)

## 8.16 jeod::TimeEnum Class Reference

Contains an enumeration of the formats in which time can be represented.

```
#include <time_enum.hh>
```

## Public Types

- enum `TimeFormat` {  
`undefined` = -1, `Julian`, `julian`, `modified_julian`,  
`truncated_julian`, `calendar`, `clock`, `days_since_epoch`,  
`seconds_since_epoch` }

*The enumeration of the formats in which time can be represented.*

### 8.16.1 Detailed Description

Contains an enumeration of the formats in which time can be represented.

Definition at line 46 of file `time_enum.hh`.

### 8.16.2 Member Enumeration Documentation

#### 8.16.2.1 enum `jeod::TimeEnum::TimeFormat`

The enumeration of the formats in which time can be represented.

#### Enumerator

***undefined*** Default setting.

***Julian*** Full Julian representation.

***julian*** Full Julian representation.

***modified\_julian*** Modified-Julian representation.

***truncated\_julian*** Truncated-Julian representation.

***calendar*** Calendar (Gregorian) representation.

***clock*** "Calendar" representation for MET.

***days\_since\_epoch*** Days since the type's defined epoch.

***seconds\_since\_epoch*** Seconds since the type's defined epoch.

Definition at line 54 of file `time_enum.hh`.

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

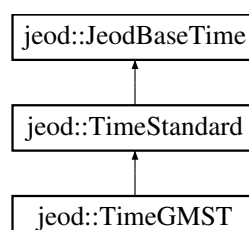
- [time\\_enum.hh](#)

## 8.17 jeod::TimeGMST Class Reference

To represent the clock known as Greenwich Mean Sidereal Time.

```
#include <time_gmst.hh>
```

Inheritance diagram for `jeod::TimeGMST`:



## Public Member Functions

- [TimeGMST](#) ()  
*Construct a Time\_GMST.*
- [~TimeGMST](#) ()  
*Destroy a Time\_GMST.*
- void [set\\_time\\_by\\_trunc\\_julian](#) (const double nonsense)  
*TJT does not function in GMST.*

## Private Member Functions

- void [calculate\\_calendar\\_values](#) (void)  
*Protection against inheriting nonsense function.*
- void [set\\_epoch](#) (void)  
*No action.*
- [TimeGMST](#) (const [TimeGMST](#) &)
- [TimeGMST](#) & [operator=](#) (const [TimeGMST](#) &)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeGMST](#) ()

## Additional Inherited Members

### 8.17.1 Detailed Description

To represent the clock known as Greenwich Mean Sidereal Time.

Definition at line 48 of file `time_gmst.hh`.

### 8.17.2 Constructor & Destructor Documentation

#### 8.17.2.1 `jeod::TimeGMST::TimeGMST ( void )`

Construct a Time\_GMST.

Definition at line 56 of file `time_gmst.cc`.

References `jeod::JeodBaseTime::name`.

#### 8.17.2.2 `jeod::TimeGMST::~~TimeGMST ( void )`

Destroy a Time\_GMST.

Definition at line 103 of file `time_gmst.cc`.

#### 8.17.2.3 `jeod::TimeGMST::TimeGMST ( const TimeGMST & ) [private]`

### 8.17.3 Member Function Documentation

#### 8.17.3.1 `void jeod::TimeGMST::calculate_calendar_values ( void ) [private], [virtual]`

Protection against inheriting nonsense function.

### Assumptions and Limitations

- GMST does not have a conventional calendar

Reimplemented from [jeod::TimeStandard](#).

Definition at line 69 of file `time_gmst.cc`.

References `jeod::TimeMessages::invalid_data_error`.

**8.17.3.2** `TimeGMST& jeod::TimeGMST::operator= ( const TimeGMST & )` `[private]`

**8.17.3.3** `void jeod::TimeGMST::set_epoch ( void )` `[inline],[private],[virtual]`

No action.

Function is required to make this class instantiable.

Implements [jeod::TimeStandard](#).

Definition at line 70 of file `time_gmst.hh`.

**8.17.3.4** `void jeod::TimeGMST::set_time_by_trunc_julian ( const double nonsense )`

TJT does not function in GMST.

### Assumptions and Limitations

- 86400 seconds = 1 day

### Parameters

<code>in</code>	<code><i>nonsense</i></code>	Any old invalid value
-----------------	------------------------------	-----------------------

Definition at line 87 of file `time_gmst.cc`.

References `jeod::TimeMessages::invalid_data_error`.

## 8.17.4 Friends And Related Function Documentation

**8.17.4.1** `void init_attrjeod__TimeGMST ( )` `[friend]`

**8.17.4.2** `friend class InputProcessor` `[friend]`

Definition at line 50 of file `time_gmst.hh`.

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

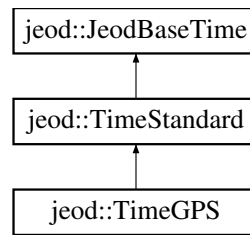
- [time\\_gmst.hh](#)
- [time\\_gmst.cc](#)

## 8.18 jeod::TimeGPS Class Reference

To represent the time associated with the Global Positioning System.

```
#include <time_gps.hh>
```

Inheritance diagram for `jeod::TimeGPS`:



## Public Member Functions

- [TimeGPS](#) ()  
*Construct a Time\_GPS.*
- [~TimeGPS](#) ()  
*Destroy a TimeGPS.*
- void [set\\_time\\_by\\_seconds](#) (const double new\_seconds)  
*Given a value of seconds, propagate to other reps.*
- void [set\\_time\\_by\\_days](#) (const double new\_seconds)  
*Given a value of days, propagate to other values.*
- void [set\\_time\\_by\\_trunc\\_julian](#) (const double new\_tjt)  
*TJT does not function in GPS.*

## Data Fields

- double [seconds\\_of\\_day](#)  
*Seconds elapsed in last (partial) day.*
- double [seconds\\_of\\_week](#)  
*Seconds elapsed in last (partial) week.*
- int [day\\_of\\_week](#)  
*Number of whole days this week.*
- int [rollover\\_count](#)  
*Number of rollovers (1024 week blocks) since epoch.*
- int [week](#)  
*Number of weeks in current 1024-week block.*
- int [rollover\\_count\\_13\\_bit](#)  
*Number of rollovers (8192 week blocks) since epoch.*
- int [week\\_13\\_bit](#)  
*Number of weeks in current 8192-week block.*

## Private Member Functions

- void [calculate\\_calendar\\_values](#) (void)  
*Protection against inheriting nonsense function.*
- void [convert\\_from\\_calendar](#) (void)  
*Protection against inheriting nonsense function.*
- void [set\\_epoch](#) (void)  
*Sets the epoch for GPS time.*
- [TimeGPS](#) (const [TimeGPS](#) &)
- [TimeGPS](#) & [operator=](#) (const [TimeGPS](#) &)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeGPS](#) ()

## Additional Inherited Members

### 8.18.1 Detailed Description

To represent the time associated with the Global Positioning System.

Definition at line 52 of file `time_gps.hh`.

### 8.18.2 Constructor & Destructor Documentation

#### 8.18.2.1 `jeod::TimeGPS::TimeGPS ( void )`

Construct a `Time_GPS`.

Definition at line 56 of file `time_gps.cc`.

References `day_of_week`, `jeod::JeodBaseTime::name`, `rollover_count`, `rollover_count_13_bit`, `seconds_of_day`, `seconds_of_week`, `set_epoch()`, `week`, and `week_13_bit`.

#### 8.18.2.2 `jeod::TimeGPS::~~TimeGPS ( void )`

Destroy a [TimeGPS](#).

Definition at line 194 of file `time_gps.cc`.

#### 8.18.2.3 `jeod::TimeGPS::TimeGPS ( const TimeGPS & )` `[private]`

### 8.18.3 Member Function Documentation

#### 8.18.3.1 `void jeod::TimeGPS::calculate_calendar_values ( void )` `[private]`, `[virtual]`

Protection against inheriting nonsense function.

#### Assumptions and Limitations

- GPS does not have a conventional calendar

Reimplemented from [jeod::TimeStandard](#).

Definition at line 111 of file `time_gps.cc`.

References `jeod::TimeMessages::invalid_data_error`.

#### 8.18.3.2 `void jeod::TimeGPS::convert_from_calendar ( void )` `[private]`, `[virtual]`

Protection against inheriting nonsense function.

#### Assumptions and Limitations

- GPS does not have a conventional calendar



Reimplemented from [jeod::TimeStandard](#).

Definition at line 93 of file `time_gps.cc`.

References `jeod::TimeMessages::invalid_data_error`.

**8.18.3.3** `TimeGPS& jeod::TimeGPS::operator= ( const TimeGPS & )` `[private]`

**8.18.3.4** `void jeod::TimeGPS::set_epoch ( void )` `[private]`, `[virtual]`

Sets the epoch for GPS time.

Implements [jeod::TimeStandard](#).

Definition at line 76 of file `time_gps.cc`.

References `jeod::TimeStandard::tjt_at_epoch`.

Referenced by `TimeGPS()`.

**8.18.3.5** `void jeod::TimeGPS::set_time_by_days ( const double new_days )` `[virtual]`

Given a value of days, propagate to other values.

#### Assumptions and Limitations

- 86400 seconds = 1 day

#### Parameters

<code>in</code>	<code>new_days</code>	new value for days Units: day
-----------------	-----------------------	----------------------------------

Reimplemented from [jeod::TimeStandard](#).

Definition at line 164 of file `time_gps.cc`.

References `set_time_by_seconds()`.

**8.18.3.6** `void jeod::TimeGPS::set_time_by_seconds ( const double new_seconds )` `[virtual]`

Given a value of seconds, propagate to other reps.

#### Assumptions and Limitations

- 86400 seconds = 1 day

#### Parameters

<code>in</code>	<code>new_seconds</code>	new value for seconds Units: s
-----------------	--------------------------	-----------------------------------

Reimplemented from [jeod::TimeStandard](#).

Definition at line 129 of file `time_gps.cc`.

References `day_of_week`, `jeod::JeodBaseTime::days`, `rollover_count`, `rollover_count_13_bit`, `seconds_of_day`, `seconds_of_week`, `jeod::TimeStandard::set_time_by_seconds()`, `week`, and `week_13_bit`.

Referenced by `jeod::TimeConverter_TAI_GPS::convert_a_to_b()`, `set_time_by_days()`, and `set_time_by_trunc_julian()`.

**8.18.3.7** `void jeod::TimeGPS::set_time_by_trunc_julian ( const double new_tjt )`

TJT does not function in GPS.

#### Assumptions and Limitations

- 86400 seconds = 1 day

#### Parameters

<i>in</i>	<i>new_tjt</i>	new value for Truncated Julian Time Units: day
-----------	----------------	---

Definition at line 180 of file `time_gps.cc`.

References `jeod::JeodBaseTime::seconds`, `set_time_by_seconds()`, and `jeod::TimeStandard::set_time_by_trunc_julian()`.

### 8.18.4 Friends And Related Function Documentation

**8.18.4.1** `void init_attrjeod_TimeGPS ( ) [friend]`

**8.18.4.2** `friend class InputProcessor [friend]`

Definition at line 54 of file `time_gps.hh`.

### 8.18.5 Field Documentation

**8.18.5.1** `int jeod::TimeGPS::day_of_week`

Number of whole days this week.

`trick_units(day)`

Definition at line 70 of file `time_gps.hh`.

Referenced by `set_time_by_seconds()`, and `TimeGPS()`.

**8.18.5.2** `int jeod::TimeGPS::rollover_count`

Number of rollovers (1024 week blocks) since epoch.

`trick_units(-)`

Definition at line 74 of file `time_gps.hh`.

Referenced by `set_time_by_seconds()`, and `TimeGPS()`.

**8.18.5.3** `int jeod::TimeGPS::rollover_count_13_bit`

Number of rollovers (8192 week blocks) since epoch.

`trick_units(-)`

Definition at line 82 of file `time_gps.hh`.

Referenced by `set_time_by_seconds()`, and `TimeGPS()`.

**8.18.5.4 double jeod::TimeGPS::seconds\_of\_day**

Seconds elapsed in last (partial) day.

trick\_units(s)

Definition at line 62 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

**8.18.5.5 double jeod::TimeGPS::seconds\_of\_week**

Seconds elapsed in last (partial) week.

trick\_units(s)

Definition at line 66 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

**8.18.5.6 int jeod::TimeGPS::week**

Number of weeks in current 1024-week block.

trick\_units(-)

Definition at line 78 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

**8.18.5.7 int jeod::TimeGPS::week\_13\_bit**

Number of weeks in current 8192-week block.

trick\_units(-)

Definition at line 86 of file time\_gps.hh.

Referenced by set\_time\_by\_seconds(), and TimeGPS().

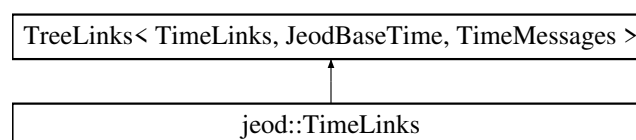
The documentation for this class was generated from the following files:

- [time\\_gps.hh](#)
- [time\\_gps.cc](#)

**8.19 jeod::TimeLinks Class Reference**

```
#include <time_links.hh>
```

Inheritance diagram for jeod::TimeLinks:

**Public Member Functions**

- [TimeLinks](#) ([JeodBaseTime](#) &time\_in)

- `TimeLinks ()=delete`
- `TimeLinks (const TimeLinks &)=delete`
- `void operator= (const TimeLinks &)=delete`
- `virtual ~TimeLinks ()=default`

*Default destructor.*

## Static Private Attributes

- static const unsigned int `default_path_size` = 8

*Default allocated number of entries in linkage container.*

## Friends

- class `InputProcessor`
- void `init_attrjeod__TimeLinks ()`

### 8.19.1 Detailed Description

Definition at line 43 of file `time_links.hh`.

### 8.19.2 Constructor & Destructor Documentation

8.19.2.1 `jeod::TimeLinks::TimeLinks ( JeodBaseTime & time_in ) [inline]`

Definition at line 49 of file `time_links.hh`.

8.19.2.2 `jeod::TimeLinks::TimeLinks ( ) [delete]`

8.19.2.3 `jeod::TimeLinks::TimeLinks ( const TimeLinks & ) [delete]`

8.19.2.4 `virtual jeod::TimeLinks::~~TimeLinks ( ) [virtual],[default]`

Default destructor.

### 8.19.3 Member Function Documentation

8.19.3.1 `void jeod::TimeLinks::operator= ( const TimeLinks & ) [delete]`

### 8.19.4 Friends And Related Function Documentation

8.19.4.1 `void init_attrjeod__TimeLinks ( ) [friend]`

8.19.4.2 `friend class InputProcessor [friend]`

Definition at line 45 of file `time_links.hh`.

### 8.19.5 Field Documentation

8.19.5.1 `const unsigned int jeod::TimeLinks::default_path_size = 8 [static],[private]`

Default allocated number of entries in linkage container.

trick\_units(-)

Definition at line 70 of file time\_links.hh.

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

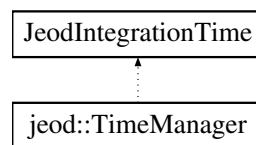
- [time\\_links.hh](#)

## 8.20 jeod::TimeManager Class Reference

To manage the various time representations and the converters between them throughout the simulation.

```
#include <time_manager.hh>
```

Inheritance diagram for jeod::TimeManager:



### Public Member Functions

- [TimeManager](#) ()  
*Construct a [TimeManager](#).*
- [~TimeManager](#) ()  
*Destroy a [TimeManager](#).*
- void [initialize](#) ([TimeManagerInit](#) \*time\_manager\_init)  
*initializes the time manager*
- int [time\\_lookup](#) (const std::string &name) const  
*Uses a string comparison to find where in the [TimeManager](#) record a time type of a particular name is located.*
- bool [get\\_time\\_change\\_flag](#) () const  
*Returns the boolean value [time\\_change\\_flag](#).*
- [JeodBaseTime](#) \* [get\\_time\\_ptr](#) (const std::string &name) const  
*Return a pointer to the Time object with the provided name, or NULL if no such Time object has been registered.*
- [JeodBaseTime](#) \* [get\\_time\\_ptr](#) (const int index) const  
*Return a pointer to the Time object with the provided index, or NULL if no such Time object has been registered.*
- [TimeConverter](#) \* [get\\_converter\\_ptr](#) (const int index) const  
*Return a pointer to the [TimeConverter](#) object with the provided index, or NULL if no such [TimeConverter](#) object has been registered.*
- bool [time\\_standards\\_exist](#) (void)  
*Tests for the existence in the registry of time types that inherit from [TimeStandard](#).*
- virtual void [update](#) (double time)  
*This function manages the time update process.*
- void [verify\\_table\\_lookup\\_ends](#) (void)  
*This function is called when the simulation reverses direction (in time).*
- void [register\\_time](#) ([JeodBaseTime](#) &time\_ref)  
*Registers the time representation with the Time Manager.*
- void [register\\_time\\_named](#) ([JeodBaseTime](#) &time\_ref, const std::string &name)  
*Reassigns the name to the type; this is used when there are multiple instances of a time type such as a MET or UDE.*
- void [register\\_converter](#) ([TimeConverter](#) &converter\_ref, std::string name\_a="", std::string name\_b="")  
*Registers the time converters with the Time Manager.*

- JeodIntegrationTime & [get\\_jeod\\_integration\\_time](#) ()  
*Expose the private inheritance from JeodIntegrationTime.*
- virtual double [get\\_time\\_scale\\_factor](#) () const  
*Returns the scale factor from sim time to dynamic time.*
- virtual double [get\\_timestamp\\_time](#) () const  
*Returns the time used to timestamp objects, currently dynamic time seconds.*

## Data Fields

- double [simtime](#)  
*Simulation time (sys.exec.out.time).*
- [TimeDyn](#) dyn\_time  
*The instance of [TimeDyn](#), the dynamic time that is used as the integration time.*
- int [num\\_types](#)  
*Size of time\_types\_ptrs vector.*

## Private Member Functions

- virtual void [update\\_time](#) (double time)  
*Update each of the representations of time, calling the update functions for each such representation in dependency order.*
- [TimeManager](#) (const [TimeManager](#) &)
- [TimeManager](#) & operator= (const [TimeManager](#) &)

## Private Attributes

- bool [time\\_change\\_flag](#)  
*Indicates that the dynamic scale factor changed.*
- std::vector< [JeodBaseTime](#) \* > [time\\_vector](#)  
*List of pointers to time-types.*
- std::vector< [TimeConverter](#) \* > [converter\\_vector](#)  
*List of pointers to time-converters.*

## Friends

- class [InputProcessor](#)
- class [TimeManagerInit](#)
- void [init\\_attrjeod\\_\\_TimeManager](#) ()

### 8.20.1 Detailed Description

To manage the various time representations and the converters between them throughout the simulation.  
Definition at line 65 of file time\_manager.hh.

### 8.20.2 Constructor & Destructor Documentation

#### 8.20.2.1 jeod::TimeManager::TimeManager ( void )

Construct a [TimeManager](#).

Definition at line 72 of file time\_manager.cc.

**8.20.2.2 jeod::TimeManager::~~TimeManager ( void )**

Destroy a [TimeManager](#).

Definition at line 512 of file `time_manager.cc`.

References `converter_vector`, and `time_vector`.

**8.20.2.3 jeod::TimeManager::TimeManager ( const TimeManager & ) [private]****8.20.3 Member Function Documentation****8.20.3.1 TimeConverter \* jeod::TimeManager::get\_converter\_ptr ( const int index ) const**

Return a pointer to the [TimeConverter](#) object with the provided index, or NULL if no such [TimeConverter](#) object has been registered.

Returns

[TimeConverter](#) object corresponding to index in the vector of such types.

Parameters

<i>in</i>	<i>index</i>	Index of object
-----------	--------------	-----------------

Definition at line 95 of file `time_manager.cc`.

References `converter_vector`.

Referenced by `jeod::JeodBaseTime::add_type_update()`, `jeod::TimeUDE::convert_epoch_to_update()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, and `jeod::TimeUDE::initialize_initializer_time()`.

**8.20.3.2 JeodIntegrationTime & jeod::TimeManager::get\_jeod\_integration\_time ( void )**

Expose the private inheritance from `JeodIntegrationTime`.

Definition at line 115 of file `time_manager.cc`.

**8.20.3.3 bool jeod::TimeManager::get\_time\_change\_flag ( void ) const**

Returns the boolean value `time_change_flag`.

Returns

`time_change_flag`

Definition at line 127 of file `time_manager.cc`.

References `time_change_flag`.

**8.20.3.4 JeodBaseTime \* jeod::TimeManager::get\_time\_ptr ( const std::string & name ) const**

Return a pointer to the Time object with the provided name, or NULL if no such Time object has been registered.

Returns

Time object corresponding to name

**Parameters**

<i>in</i>	<i>name</i>	Name of time object
-----------	-------------	---------------------

Definition at line 169 of file time\_manager.cc.

References time\_lookup().

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeUDE::add\_type\_initialize(), jeod::JeodBaseTime::add\_type\_update(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeUDE::initialize\_initializer\_time(), jeod::TimeManagerInit::initialize\_time\_types(), and jeod::TimeUDE::verify\_update().

### 8.20.3.5 JeodBaseTime \* jeod::TimeManager::get\_time\_ptr ( const int *index* ) const

Return a pointer to the Time object with the provided index, or NULL if no such Time object has been registered.

**Returns**

Time object corresponding to name

**Parameters**

<i>in</i>	<i>index</i>	Name of time object
-----------	--------------	---------------------

Definition at line 184 of file time\_manager.cc.

References time\_vector.

### 8.20.3.6 double jeod::TimeManager::get\_time\_scale\_factor ( void ) const [virtual]

Returns the scale factor from sim time to dynamic time.

**Returns**

dyn\_time.scale\_factor

Definition at line 140 of file time\_manager.cc.

References dyn\_time, and jeod::TimeDyn::scale\_factor.

### 8.20.3.7 double jeod::TimeManager::get\_timestamp\_time ( void ) const [virtual]

Returns the time used to timestamp objects, currently dynamic time seconds.

**Returns**

dyn\_time.seconds

Definition at line 154 of file time\_manager.cc.

References dyn\_time, and jeod::JeodBaseTime::seconds.

### 8.20.3.8 void jeod::TimeManager::initialize ( TimeManagerInit \* *time\_manager\_init* )

initializes the time manager



## Parameters

in	<i>time_manager_init</i>	Initialization parameters
----	--------------------------	---------------------------

Definition at line 70 of file `time_manager_initialize.cc`.

### 8.20.3.9 TimeManager& jeod::TimeManager::operator= ( const TimeManager & ) [private]

### 8.20.3.10 void jeod::TimeManager::register\_converter ( TimeConverter & conv\_ref, std::string name\_a = " ", std::string name\_b = " " )

Registers the time converters with the Time Manager.

## Assumptions and Limitations

- the input values `name_a` and `name_b` will only be used if the converter-type names have not already been set. So registering a Dyn\_UDE converter will ignore `name_a` completely because it is already set.

## Parameters

in, out	<i>conv_ref</i>	ref. to converter being registered
in	<i>name_a</i>	name of type-a in the converter
in	<i>name_b</i>	name of type-b in the converter

Definition at line 273 of file `time_manager.cc`.

References `jeod::TimeConverter::a_name`, `jeod::TimeConverter::b_name`, `converter_vector`, `jeod::TimeMessages::incomplete_setup_error`, and `jeod::TimeMessages::redundancy_error`.

### 8.20.3.11 void jeod::TimeManager::register\_time ( JeodBaseTime & time\_ref )

Registers the time representation with the Time Manager.

Records the frequency at which the representation should be updated.

## Assumptions and Limitations

- None

## Parameters

in, out	<i>time_ref</i>	reference to time-type being registered
---------	-----------------	---

Definition at line 207 of file `time_manager.cc`.

References `jeod::JeodBaseTime::name`, `jeod::TimeMessages::redundancy_error`, `jeod::JeodBaseTime::set_index()`, `jeod::JeodBaseTime::time_manager`, and `time_vector`.

Referenced by `register_time_named()`.

### 8.20.3.12 void jeod::TimeManager::register\_time\_named ( JeodBaseTime & time\_ref, const std::string & name )

Reassigns the name to the type; this is used when there are multiple instances of a time type such as a MET or UDE.

Registers the time representation with the Time Manager. Records the frequency at which the representation should be updated. TODO: check for duplicates

## Assumptions and Limitations

- None

**Parameters**

<i>in, out</i>	<i>time_ref</i>	reference to time-type being registered
<i>in</i>	<i>name</i>	name of the instance being registered.

Definition at line 241 of file `time_manager.cc`.

References `jeod::JeodBaseTime::name`, and `register_time()`.

**8.20.3.13 int jeod::TimeManager::time\_lookup ( const std::string & name ) const**

Uses a string comparison to find where in the [TimeManager](#) record a time type of a particular name is located.

Returns the integer corresponding to the time type's index in the [TimeManager](#).

**Assumptions and Limitations**

- Rarely used. If the time type address is known, it is easier to access its index "time\_type.index" which returns the same result.

**Returns**

index value of time-type

**Parameters**

<i>in</i>	<i>name</i>	name of time-type
-----------	-------------	-------------------

Definition at line 362 of file `time_manager.cc`.

References `jeod::TimeMessages::invalid_setup_error`, and `time_vector`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::JeodBaseTime::add_type_update()`, `jeod::TimeManagerInit::create_init_tree()`, `get_time_ptr()`, `jeod::TimeManagerInit::initialize()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeManagerInit::populate_converter_registry()`, `jeod::TimeManagerInit::verify_converter_setup()`, `jeod::TimeUDE::verify_epoch()`, `jeod::TimeUDE::verify_init()`, and `jeod::TimeUDE::verify_update()`.

**8.20.3.14 bool jeod::TimeManager::time\_standards\_exist ( void )**

Tests for the existence in the registry of time types that inherit from [TimeStandard](#).

**Assumptions and Limitations**

- None

**Returns**

true/false

Definition at line 337 of file `time_manager.cc`.

References `time_vector`.

Referenced by `jeod::TimeDyn::initialize_initializer_time()`, and `jeod::TimeUDE::initialize_initializer_time()`.

**8.20.3.15 void jeod::TimeManager::update ( double current\_simtime ) [virtual]**

This function manages the time update process.

It first updates each of the representations of time, calling the update functions for each time representation in dependency order. After updating the representations of time, the function then updates the dynamic time scale factor. Time change subscribers are notified if the scale factor has changed.

Note that by updating first and then checking for a change in the rate/direction of time means that these changes in rate/direction will first take affect on the next call to `update_time` or `update`.

#### Assumptions and Limitations

- Derived times must have a parent; this should be defined by the user, or if not, already determined when the `update_tree` was built

#### Parameters

<code>in</code>	<code>current_simtime</code>	input time from simulation engine; it always runs forwards and allows for determination of what has and has not already been done. Units: s
-----------------	------------------------------	--

Definition at line 426 of file `time_manager.cc`.

References `dyn_time`, `num_types`, `simtime`, `time_change_flag`, `time_vector`, and `jeod::TimeDyn::update_offset()`.

Referenced by `jeod::TimeStandard::calendar_update()`.

#### 8.20.3.16 void jeod::TimeManager::update\_time ( double *current\_simtime* ) [private],[virtual]

Update each of the representations of time, calling the update functions for each such representation in dependency order.

Note that this function only does the first part of the task performed by [TimeManager::update](#). It does not check for changes in the rate/direction of time.

#### Assumptions and Limitations

- Derived times must have a parent; this should be defined by the user, or if not, already determined when the `update_tree` was built

#### Parameters

<code>in</code>	<code>current_simtime</code>	input time from simulation engine; it always runs forwards and allows for determination of what has and has not already been done. Units: s
-----------------	------------------------------	--

Definition at line 468 of file `time_manager.cc`.

References `num_types`, `simtime`, and `time_vector`.

#### 8.20.3.17 void jeod::TimeManager::verify\_table\_lookup\_ends ( void )

This function is called when the simulation reverses direction (in time).

It calls each time converter that uses a table lookup to check whether the current time is off the end of the table. This is important because once the off-table-end flag is set, the only reason to unset it is when time reverses direction)

#### Assumptions and Limitations

- None

Definition at line 499 of file `time_manager.cc`.

References `converter_vector`.

Referenced by `jeod::TimeDyn::update_offset()`.

## 8.20.4 Friends And Related Function Documentation

8.20.4.1 `void init_attrjeod_TimeManager ( ) [friend]`

8.20.4.2 `friend class InputProcessor [friend]`

Definition at line 67 of file `time_manager.hh`.

8.20.4.3 `friend class TimeManagerInit [friend]`

Definition at line 69 of file `time_manager.hh`.

## 8.20.5 Field Documentation

8.20.5.1 `std::vector<TimeConverter*> jeod::TimeManager::converter_vector [private]`

List of pointers to time-converters.

`trick_io(**)`

Definition at line 107 of file `time_manager.hh`.

Referenced by `get_converter_ptr()`, `jeod::TimeManagerInit::populate_converter_registry()`, `register_converter()`, `verify_table_lookup_ends()`, and `~TimeManager()`.

8.20.5.2 `TimeDyn jeod::TimeManager::dyn_time`

The instance of [TimeDyn](#), the dynamic time that is used as the integration time.

`trick_units(-)`

Definition at line 84 of file `time_manager.hh`.

Referenced by `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `get_time_scale_factor()`, `get_timestamp_time()`, `jeod::TimeManagerInit::initialize()`, `jeod::TimeConverter_TAI_UTC::initialize_leap_second()`, `jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1()`, `update()`, `jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends()`, and `jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends()`.

8.20.5.3 `int jeod::TimeManager::num_types`

Size of `time_types_ptrs` vector.

`trick_units(-)`

Definition at line 89 of file `time_manager.hh`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, `jeod::JeodBaseTime::add_type_update()`, `jeod::TimeUDE::convert_epoch_to_update()`, `jeod::TimeManagerInit::create_init_tree()`, `jeod::TimeManagerInit::create_update_tree()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, `jeod::TimeUDE::initialize_initializer_time()`, `jeod::TimeManagerInit::initialize_time_types()`, `jeod::TimeManagerInit::organize_update_list()`, `jeod::TimeManagerInit::populate_converter_registry()`, `update()`, `update_time()`, `jeod::TimeManagerInit::verify_converter_setup()`, and `jeod::TimeManagerInit::verify_times_setup()`.

8.20.5.4 `double jeod::TimeManager::simtime`

Simulation time (`sys.exec.out.time`).

`trick_units(-)`

Definition at line 78 of file `time_manager.hh`.

Referenced by `jeod::TimeStandard::calendar_update()`, `jeod::TimeStandard::seconds_of_year()`, `jeod::TimeDyn::update()`, `update()`, `jeod::TimeDyn::update_offset()`, and `update_time()`.

#### 8.20.5.5 `bool jeod::TimeManager::time_change_flag` [private]

Indicates that the dynamic scale factor changed.

`trick_units(-)`

Definition at line 97 of file `time_manager.hh`.

Referenced by `get_time_change_flag()`, and `update()`.

#### 8.20.5.6 `std::vector<JeodBaseTime*> jeod::TimeManager::time_vector` [private]

List of pointers to time-types.

`trick_io(**)`

Definition at line 102 of file `time_manager.hh`.

Referenced by `jeod::TimeManagerInit::create_init_tree()`, `jeod::TimeManagerInit::create_update_tree()`, `get_time_ptr()`, `jeod::TimeManagerInit::initialize()`, `jeod::TimeManagerInit::initialize_time_types()`, `jeod::TimeManagerInit::organize_update_list()`, `jeod::TimeManagerInit::populate_converter_registry()`, `register_time()`, `time_lookup()`, `time_standards_exist()`, `update()`, `update_time()`, `jeod::TimeManagerInit::verify_times_setup()`, and `~TimeManager()`.

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

- [time\\_manager.hh](#)
- [time\\_manager.cc](#)
- [time\\_manager\\_\\_initialize.cc](#)

## 8.21 jeod::TimeManagerInit Class Reference

To initialize the Time Manager.

```
#include <time_manager_init.hh>
```

### Public Member Functions

- [TimeManagerInit](#) ()  
*Construct a [TimeManagerInit](#).*
- [~TimeManagerInit](#) ()  
*Destroy a [TimeManagerInit](#).*
- [get\\_conv\\_ptr\\_index](#) (const int conv\_index)  
*Takes a calculated converter index - calculated by combining the two time-type indices - and return the index in the [time\\_manager](#)'s vector of converters that corresponds to those two time-types.*
- [get\\_conv\\_dir\\_init](#) (const int conv\_index)  
*Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".*
- [get\\_conv\\_dir\\_upd](#) (const int conv\_index)  
*Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".*
- [get\\_status](#) (const int index)  
*Returns the status of a time-type.*
- [set\\_status](#) (const int index, const int status\_value)

- *Receives an updated value for the status of a time-type.*
- void `increment_status` (const int slave\_index, const int master\_index)  
*Modifies the status of one time-type to be one higher than that of another type for initialization purposes.*
- void `initialize_manager` (TimeManager \*time\_mgr)  
*The master program behind the initialization of the time types and the time converters.*
- void `organize_update_list` ()  
*Reorganizes the update list according to initialization status.*

## Data Fields

- int `num_added_total`  
*Count of the total number of time-types placed in the update tree or in the initialization tree.*
- TimeEnum::TimeFormat `sim_start_format`  
*Calendar, truncated\_julian, etc.*
- TimeManager \* `time_manager`  
*Pointer to the Time Manager.*
- std::string `initializer`  
*Name of the time-type used for initialization.*

## Protected Attributes

- int `initializer_index`  
*Index-value of the initializer.*
- int `dyn_time_index`  
*Index-value of the type dyn-time.*
- int `num_added_pass`  
*Count of number of time-types placed in the update tree or in the initialization tree in any given pass.*
- int \* `converter_ptrs_index`  
*List of the indices (in the TimeManager->time\_converter\_ptrs vector) of all registered converters, sorted by the indices of the time-types the converters act upon (most pairs of time-types have no converter registered; the value of these indices is -1)*
- int \* `init_converter_dir_table`  
*List of directions available for initialization for each of the converters listed in converter\_class\_ptrs.*
- int \* `update_converter_dir_table`  
*List of directions available for run-time updates for each of the converters listed in converter\_class\_ptrs.*
- int \* `status`  
*A running ledger of properly linked times during update tree and initialization tree construction.*

## Private Member Functions

- void `initialize` (void)  
*The TimeManagerInit determines initialization and update paths for conversions between time-types.*
- void `verify_times_setup` (void)  
*A number of checks that the setup is self-consistent.*
- void `populate_converter_registry` (void)  
*The converter registry accounts for all of the converter functions that provide conversions between time types.*
- void `verify_converter_setup` (void)  
*To verify that there are no incompatibilities between specific converters.*
- void `initialize_time_types` (void)  
*Initialize each time type so that it has a starting value corresponding to dynamic\_time = 0 and such that the starting values are consistent.*

- void [create\\_init\\_tree](#) (void)  
*Build and verify a "tree-like" structure to ensure that all time representations can be initialized from the single "initializer" representation.*
- void [create\\_update\\_tree](#) (void)  
*(To verify that the update procedures have a tree-like structure, and that all time representations can be updated from the dynamic time.) (Contains 3 functions - create\_update\_tree builds the tree, populated recursively by add\_type\_update.*
- [TimeManagerInit](#) (const [TimeManagerInit](#) &)
- [TimeManagerInit](#) & operator= (const [TimeManagerInit](#) &)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeManagerInit](#) ()

### 8.21.1 Detailed Description

To initialize the Time Manager.

Definition at line 55 of file `time_manager_init.hh`.

### 8.21.2 Constructor & Destructor Documentation

#### 8.21.2.1 `jeod::TimeManagerInit::TimeManagerInit ( void )`

Construct a [TimeManagerInit](#).

Definition at line 67 of file `time_manager_init.cc`.

References `converter_ptrs_index`, `dyn_time_index`, `init_converter_dir_table`, `initializer`, `initializer_index`, `num_added_pass`, `num_added_total`, `sim_start_format`, `status`, `jeod::TimeEnum::undefined`, and `update_converter_dir_table`.

#### 8.21.2.2 `jeod::TimeManagerInit::~~TimeManagerInit ( void )`

Destroy a [TimeManagerInit](#).

Definition at line 815 of file `time_manager_init.cc`.

References `converter_ptrs_index`, `init_converter_dir_table`, `status`, and `update_converter_dir_table`.

#### 8.21.2.3 `jeod::TimeManagerInit::TimeManagerInit ( const TimeManagerInit & ) [private]`

### 8.21.3 Member Function Documentation

#### 8.21.3.1 `void jeod::TimeManagerInit::create_init_tree ( void ) [private]`

Build and verify a "tree-like" structure to ensure that all time representations can be initialized from the single "initializer" representation.

`Create_init_tree` builds the tree, using `add_type_initialize` to populate the tree recursively.

## Assumptions and Limitations

- This is vastly improved if the user defines the parent type "initialize\_from" for each time representation, except the top-level initializer type

- Otherwise, the code will build the tree automatically, but it takes longer and may be less than ideal

Definition at line 434 of file `time_manager_init.cc`.

References `dyn_time_index`, `jeod::TimeMessages::initialization_error`, `initializer_index`, `jeod::TimeMessages::invalid_setup_error`, `num_added_pass`, `num_added_total`, `jeod::TimeManager::num_types`, `status`, `jeod::TimeManager::time_lookup()`, `time_manager`, `jeod::TimeManager::time_vector`, and `jeod::JeodBaseTime::update_from_name`.

Referenced by `initialize_manager()`.

#### 8.21.3.2 `void jeod::TimeManagerInit::create_update_tree ( void ) [private]`

(To verify that the update procedures have a tree-like structure, and that all time representations can be updated from the dynamic time.) (Contains 3 functions - `create_update_tree` builds the tree, populated recursively by `add_type_update`.

`record_update` records the update paths to facilitate runtime updates)

##### Assumptions and Limitations

- None

Definition at line 577 of file `time_manager_init.cc`.

References `dyn_time_index`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::TimeMessages::initialization_error`, `num_added_pass`, `num_added_total`, `jeod::TimeManager::num_types`, `organize_update_list()`, `status`, `time_manager`, and `jeod::TimeManager::time_vector`.

Referenced by `initialize_manager()`.

#### 8.21.3.3 `int jeod::TimeManagerInit::get_conv_dir_init ( const int index )`

Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".

##### Assumptions and Limitations

- Returns 0 if no suitable converter available at initialization

##### Returns

Index corresponding to [TimeConverter](#)

##### Parameters

<code>in</code>	<code>index</code>	Index of object
-----------------	--------------------	-----------------

Definition at line 716 of file `time_manager_init.cc`.

References `init_converter_dir_table`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, `jeod::TimeUDE::convert_epoch_to_update()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, and `jeod::TimeUDE::initialize_initializer_time()`.

#### 8.21.3.4 `int jeod::TimeManagerInit::get_conv_dir_upd ( const int index )`

Takes a calculated converter index - calculated by combining the indices of two time-types, a "from" and a "to" - and returns the direction needed to use the appropriate converter to go from "from" to "to".



## Assumptions and Limitations

- Returns 0 if no suitable converter available at update

## Returns

Index corresponding to [TimeConverter](#)

## Parameters

<i>in</i>	<i>index</i>	Index of object
-----------	--------------	-----------------

Definition at line 742 of file `time_manager_init.cc`.

References `update_converter_dir_table`.

Referenced by `jeod::JeodBaseTime::add_type_update()`.

#### 8.21.3.5 int jeod::TimeManagerInit::get\_conv\_ptr\_index ( const int *index\_in* )

Takes a calculated converter index - calculated by combining the two time-type indices - and return the index in the `time_manager`'s vector of converters that corresponds to those two time-types.

## Returns

Index corresponding to [TimeConverter](#)

## Parameters

<i>in</i>	<i>index_in</i>	Index of object
-----------	-----------------	-----------------

Definition at line 690 of file `time_manager_init.cc`.

References `converter_ptrs_index`.

Referenced by `jeod::JeodBaseTime::add_type_update()`, `jeod::TimeUDE::convert_epoch_to_update()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, and `jeod::TimeUDE::initialize_initializer_time()`.

#### 8.21.3.6 int jeod::TimeManagerInit::get\_status ( const int *index* )

Returns the status of a time-type.

## Returns

Integer corresponding to Status

## Parameters

<i>in</i>	<i>index</i>	Index of object
-----------	--------------	-----------------

Definition at line 764 of file `time_manager_init.cc`.

References `status`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, and `jeod::JeodBaseTime::add_type_update()`.

#### 8.21.3.7 void jeod::TimeManagerInit::increment\_status ( const int *index\_slave*, const int *index\_master* )

Modifies the status of one time-type to be one higher than that of another type for initialization purposes.

**Parameters**

in	<i>index_slave</i>	Index of object
in	<i>index_master</i>	Index of object

Definition at line 801 of file `time_manager_init.cc`.

References `num_added_pass`, and `status`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, and `jeod::JeodBaseTime::add_type_update()`.

#### 8.21.3.8 void jeod::TimeManagerInit::initialize ( void ) [private]

The [TimeManagerInit](#) determines initialization and update paths for conversions between time-types.

This function creates and initializes the data structures necessary for these determinations

**Assumptions and Limitations**

- None

Definition at line 142 of file `time_manager_init.cc`.

References `converter_ptrs_index`, `jeod::TimeManager::dyn_time`, `dyn_time_index`, `jeod::JeodBaseTime::index`, `init_converter_dir_table`, `jeod::JeodBaseTime::initialized`, `initializer`, `initializer_index`, `jeod::JeodBaseTime::seconds`, `status`, `jeod::TimeManager::time_lookup()`, `time_manager`, `jeod::TimeManager::time_vector`, `update_converter_dir_table`, and `verify_times_setup()`.

Referenced by `initialize_manager()`.

#### 8.21.3.9 void jeod::TimeManagerInit::initialize\_manager ( TimeManager \* time\_mgr )

The master program behind the initialization of the time types and the time converters.

**Assumptions and Limitations**

- None

**Parameters**

in, out	<i>time_mgr</i>	The time manager
---------	-----------------	------------------

Definition at line 97 of file `time_manager_init.cc`.

References `create_init_tree()`, `create_update_tree()`, `initialize()`, `initialize_time_types()`, `populate_converter_registry()`, `time_manager`, and `verify_converter_setup()`.

#### 8.21.3.10 void jeod::TimeManagerInit::initialize\_time\_types ( void ) [private]

Initialize each time type so that it has a starting value corresponding to `dynamic_time = 0` and such that the starting values are consistent.

`initialize_time_types` repetitively calls `initialize_from_parent` for each time type; `initialize_from_parent` recursively adds types moving up the tree as necessary

**Assumptions and Limitations**

- An initializer time defined by the user

Definition at line 544 of file `time_manager_init.cc`.

References `jeod::TimeManager::get_time_ptr()`, `jeod::JeodBaseTime::initialize_from_parent()`, `jeod::JeodBaseTime::initialized`, `initializer_index`, `jeod::TimeManager::num_types`, `time_manager`, and `jeod::TimeManager::time_vector`.

Referenced by `initialize_manager()`.

**8.21.3.11** `TimeManagerInit& jeod::TimeManagerInit::operator= ( const TimeManagerInit & )` [private]

**8.21.3.12** `void jeod::TimeManagerInit::organize_update_list ( )`

Reorganizes the update list according to initialization status.

Definition at line 640 of file `time_manager_init.cc`.

References `jeod::TimeManager::num_types`, `status`, `time_manager`, and `jeod::TimeManager::time_vector`.

Referenced by `create_update_tree()`.

**8.21.3.13** `void jeod::TimeManagerInit::populate_converter_registry ( void )` [private]

The converter registry accounts for all of the converter functions that provide conversions between time types.

This function populates that registry so that the existence of functional converter functions can be tested efficiently.

#### Assumptions and Limitations

- None

Definition at line 279 of file `time_manager_init.cc`.

References `jeod::TimeConverter::A_TO_B_INIT`, `jeod::TimeConverter::A_TO_B_UPDATE`, `jeod::TimeConverter::B_TO_A_INIT`, `jeod::TimeConverter::B_TO_A_UPDATE`, `converter_ptrs_index`, `jeod::TimeManager::converter_vector`, `init_converter_dir_table`, `jeod::TimeManager::num_types`, `jeod::TimeMessages::redundancy_error`, `jeod::TimeManager::time_lookup()`, `time_manager`, `jeod::TimeManager::time_vector`, and `update_converter_dir_table`.

Referenced by `initialize_manager()`.

**8.21.3.14** `void jeod::TimeManagerInit::set_status ( const int index, const int new_status )`

Receives an updated value for the status of a time-type.

#### Parameters

<code>in</code>	<code>index</code>	Index of object
<code>in</code>	<code>new_status</code>	New status value

Definition at line 785 of file `time_manager_init.cc`.

References `status`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, and `jeod::JeodBaseTime::add_type_update()`.

**8.21.3.15** `void jeod::TimeManagerInit::verify_converter_setup ( void )` [private]

To verify that there are no incompatibilities between specific converters.

#### Assumptions and Limitations

- The instance of [TimeTAI](#), if it exists, has name "TAI"
- The instance of [TimeUTC](#), if it exists, has name "UTC"

- The instance of [TimeUT1](#), if it exists, has name "UT1"

Definition at line 371 of file `time_manager_init.cc`.

References `converter_ptrs_index`, `jeod::TimeMessages::invalid_setup_error`, `jeod::TimeManager::num_types`, `jeod::TimeConverter_TAI_UT1::override_data_table`, `jeod::TimeConverter_TAI_UTC::override_data_table`, `jeod::TimeManager::time_lookup()`, and `time_manager`.

Referenced by `initialize_manager()`.

#### 8.21.3.16 `void jeod::TimeManagerInit::verify_times_setup ( void ) [private]`

A number of checks that the setup is self-consistent.

#### Assumptions and Limitations

- None

Definition at line 208 of file `time_manager_init.cc`.

References `jeod::TimeMessages::incomplete_setup_error`, `initializer`, `initializer_index`, `jeod::TimeMessages::invalid_setup_error`, `jeod::TimeManager::num_types`, `jeod::TimeMessages::redundancy_error`, `time_manager`, and `jeod::TimeManager::time_vector`.

Referenced by `initialize()`.

### 8.21.4 Friends And Related Function Documentation

#### 8.21.4.1 `void init_attrjeod__TimeManagerInit ( ) [friend]`

#### 8.21.4.2 `friend class InputProcessor [friend]`

Definition at line 57 of file `time_manager_init.hh`.

### 8.21.5 Field Documentation

#### 8.21.5.1 `int* jeod::TimeManagerInit::converter_ptrs_index [protected]`

List of the indices (in the `TimeManager->time_converter_ptrs` vector) of all registered converters, sorted by the indices of the time-types the converters act upon (most pairs of time-types have no converter registered; the value of these indices is -1)

`trick_units(-)`

Definition at line 102 of file `time_manager_init.hh`.

Referenced by `get_conv_ptr_index()`, `initialize()`, `populate_converter_registry()`, `TimeManagerInit()`, `verify_converter_setup()`, and `~TimeManagerInit()`.

#### 8.21.5.2 `int jeod::TimeManagerInit::dyn_time_index [protected]`

Index-value of the type dyn-time.

`trick_units(-)`

Definition at line 89 of file `time_manager_init.hh`.

Referenced by `create_init_tree()`, `create_update_tree()`, `initialize()`, and `TimeManagerInit()`.

**8.21.5.3 int\* jeod::TimeManagerInit::init\_converter\_dir\_table [protected]**

List of directions available for initialization for each of the converters listed in converter\_class\_ptrs.

trick\_units(-)

Definition at line 107 of file time\_manager\_init.hh.

Referenced by get\_conv\_dir\_init(), initialize(), populate\_converter\_registry(), TimeManagerInit(), and ~TimeManagerInit().

**8.21.5.4 std::string jeod::TimeManagerInit::initializer**

Name of the time-type used for initialization.

trick\_units(-)

Definition at line 79 of file time\_manager\_init.hh.

Referenced by initialize(), TimeManagerInit(), and verify\_times\_setup().

**8.21.5.5 int jeod::TimeManagerInit::initializer\_index [protected]**

Index-value of the initializer.

trick\_units(-)

Definition at line 84 of file time\_manager\_init.hh.

Referenced by create\_init\_tree(), initialize(), initialize\_time\_types(), TimeManagerInit(), and verify\_times\_setup().

**8.21.5.6 int jeod::TimeManagerInit::num\_added\_pass [protected]**

Count of number of time-types placed in the update tree or in the initialization tree in any given pass.

trick\_units(-)

Definition at line 94 of file time\_manager\_init.hh.

Referenced by create\_init\_tree(), create\_update\_tree(), increment\_status(), and TimeManagerInit().

**8.21.5.7 int jeod::TimeManagerInit::num\_added\_total**

Count of the total number of time-types placed in the update tree or in the initialization tree.

trick\_units(-)

Definition at line 66 of file time\_manager\_init.hh.

Referenced by jeod::JeodBaseTime::add\_type\_update(), create\_init\_tree(), create\_update\_tree(), and TimeManagerInit().

**8.21.5.8 TimeEnum::TimeFormat jeod::TimeManagerInit::sim\_start\_format**

Calendar, truncated\_julian, etc.

trick\_units(-)

Definition at line 70 of file time\_manager\_init.hh.

Referenced by jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeUDE::initialize\_initializer\_time(), and TimeManagerInit().

### 8.21.5.9 `int* jeod::TimeManagerInit::status` [protected]

A running ledger of properly linked times during update tree and initialization tree construction.

Entries correspond to times of shared indexes in `time_vector` ( e.g. `status[2]` : status of `time_manager->time_vector[2]` ) Update tree encoding: -2: undefined. Requires auto-assignment or causes error. -1: definitive error. Process will terminate. 0: uninitialized 1: THE 1st generation (root) time. `dyn_time` for update tree. 2: a 2nd generation time, converted from root time. `n`: a `n`th gen time, converted from `(n-1)`th gen time. `trick_units(-)`

Definition at line 126 of file `time_manager_init.hh`.

Referenced by `create_init_tree()`, `create_update_tree()`, `get_status()`, `increment_status()`, `initialize()`, `organize_update_list()`, `set_status()`, `TimeManagerInit()`, and `~TimeManagerInit()`.

### 8.21.5.10 `TimeManager* jeod::TimeManagerInit::time_manager`

Pointer to the Time Manager.

Automatically linked during init routines. `trick_units(-)`

Definition at line 74 of file `time_manager_init.hh`.

Referenced by `create_init_tree()`, `create_update_tree()`, `initialize()`, `initialize_manager()`, `initialize_time_types()`, `organize_update_list()`, `populate_converter_registry()`, `verify_converter_setup()`, and `verify_times_setup()`.

### 8.21.5.11 `int* jeod::TimeManagerInit::update_converter_dir_table` [protected]

List of directions available for run-time updates for each of the converters listed in `converter_class_ptrs`.

`trick_units(-)`

Definition at line 112 of file `time_manager_init.hh`.

Referenced by `get_conv_dir_upd()`, `initialize()`, `populate_converter_registry()`, `TimeManagerInit()`, and `~TimeManagerInit()`.

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

- [time\\_manager\\_init.hh](#)
- [time\\_manager\\_init.cc](#)

## 8.22 `jeod::TimeMessages` Class Reference

Specify the message IDs used in the Time model.

```
#include <time_messages.hh>
```

### Static Public Attributes

- static char const \* [initialization\\_error](#)  
*Error issued when initialization fails due to some non-obvious cause.*
- static char const \* [memory\\_error](#)  
*Error issued when system fails because something is not where it should be, or has a value other than its assumed value.*
- static char const \* [invalid\\_setup\\_error](#)  
*Error issued when user tries to use something that doesn't exist in the model.*
- static char const \* [invalid\\_data\\_error](#)  
*Error issued when a variable is found with an illegal value.*
- static char const \* [invalid\\_node](#)

Issued when a [TimeLinks](#) node is improperly linked.

- static char const \* [incomplete\\_setup\\_error](#)

Error issued when user tries to use something that doesn't exist in the simulation. This is usually a user error, brought about by not having registered something that is later needed (e.g.

- static char const \* [redundancy\\_error](#)

Error issued when some value is multiply defined, and the code cannot determine which value to use.

- static char const \* [duplicate\\_methods](#)

Informational only.

- static char const \* [extension\\_error](#)

Issued when some functionality relies heavily on the release architecture, and is likely to break with inconsistent extensions.

## Private Member Functions

- [TimeMessages](#) (void)
- [TimeMessages](#) (const [TimeMessages](#) &)
- [TimeMessages](#) & operator= (const [TimeMessages](#) &)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeMessages](#) ()

### 8.22.1 Detailed Description

Specify the message IDs used in the Time model.

Definition at line 52 of file `time_messages.hh`.

### 8.22.2 Constructor & Destructor Documentation

8.22.2.1 `jeod::TimeMessages::TimeMessages ( void )` `[private]`

8.22.2.2 `jeod::TimeMessages::TimeMessages ( const TimeMessages & )` `[private]`

### 8.22.3 Member Function Documentation

8.22.3.1 `TimeMessages& jeod::TimeMessages::operator= ( const TimeMessages & )` `[private]`

### 8.22.4 Friends And Related Function Documentation

8.22.4.1 `void init_attrjeod__TimeMessages ( )` `[friend]`

8.22.4.2 `friend class InputProcessor` `[friend]`

Definition at line 55 of file `time_messages.hh`.

### 8.22.5 Field Documentation

8.22.5.1 `char const * jeod::TimeMessages::duplicate_methods` `[static]`

**Initial value:**

```
=
    "environment/time/" "duplicate_methods"
```

Informational only.

Issued when there are multiple equivalent methods for doing something, and one method is chosen over another. Wherever the code fills in data to accommodate, it sends this informational broadcast.trick\_units(-)

Definition at line 114 of file time\_messages.hh.

Referenced by jeod::TimeUDE::initialize\_initializer\_time().

#### 8.22.5.2 char const \* jeod::TimeMessages::extension\_error [static]

**Initial value:**

```
=
    "environment/time/" "extension_error"
```

Issued when some functionality relies heavily on the release architecture, and is likely to break with inconsistent extensions.

trick\_units(-)

Definition at line 120 of file time\_messages.hh.

#### 8.22.5.3 char const \* jeod::TimeMessages::incomplete\_setup\_error [static]

**Initial value:**

```
=
    "environment/time/" "incomplete_setup_error"
```

Error issued when user tries to use something that doesn't exist. In the simulation This is usually a user error, brought about by not having registered something that is later needed (e.g.

not registering a TAI-UTC converter, but specifying that UTC updates from TAI). Note the distinction between invalid (typically, cannot exist) and incomplete (typically, did not define)trick\_units(-)

Definition at line 100 of file time\_messages.hh.

Referenced by jeod::TimeStandard::add\_type\_initialize(), jeod::TimeUDE::add\_type\_initialize(), jeod::JeodBaseTime::add\_type\_update(), jeod::TimeUDE::convert\_epoch\_to\_update(), jeod::TimeManagerInit::create\_update\_tree(), jeod::TimeConverter\_Dyn\_UDE::initialize(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeUDE::initialize\_initializer\_time(), jeod::TimeManager::register\_converter(), jeod::TimeUDE::set\_epoch\_dyn(), jeod::TimeUDE::set\_epoch\_std(), jeod::TimeUDE::set\_epoch\_ude(), jeod::TimeUDE::set\_initial\_times(), jeod::TimeUDE::verify\_epoch(), jeod::TimeManagerInit::verify\_times\_setup(), and jeod::TimeUDE::verify\_update().

#### 8.22.5.4 char const \* jeod::TimeMessages::initialization\_error [static]

**Initial value:**

```
=
    "environment/time/" "initialization_error"
```

Error issued when initialization fails due to some non-obvious cause.

This error is likely due to an algorithm flaw.trick\_units(-)

Definition at line 65 of file time\_messages.hh.



Referenced by `jeod::TimeManagerInit::create_init_tree()`, `jeod::TimeManagerInit::create_update_tree()`, `jeod::TimeConverter_Dyn_TDB::initialize()`, `jeod::TimeConverter_Dyn_TAI::initialize()`, `jeod::TimeConverter_TAI_UTC::initialize()`, `jeod::TimeConverter_TAI_UT1::initialize()`, `jeod::TimeStandard::initialize_from_parent()`, `jeod::TimeUDE::initialize_from_parent()`, and `jeod::TimeConverter::verify_setup()`.

#### 8.22.5.5 `char const * jeod::TimeMessages::invalid_data_error` `[static]`

##### Initial value:

```
=
    "environment/time/" "invalid_data_error"
```

Error issued when a variable is found with an illegal value.

This is usually a user error, having set some value externally to some unrecognizable value. `trick_units(-)`

Definition at line 85 of file `time_messages.hh`.

Referenced by `jeod::TimeGMST::calculate_calendar_values()`, `jeod::TimeGPS::calculate_calendar_values()`, `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UT1::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `jeod::TimeConverter_TAI_UT1::convert_b_to_a()`, `jeod::TimeGPS::convert_from_calendar()`, `jeod::TimeStandard::initialize_initializer_time()`, `jeod::TimeConverter_TAI_UTC::initialize_leap_second()`, `jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1()`, and `jeod::TimeGMST::set_time_by_trunc_julian()`.

#### 8.22.5.6 `char const * jeod::TimeMessages::invalid_node` `[static]`

##### Initial value:

```
=
    "environment/time/" "invalid_node"
```

Issued when a [TimeLinks](#) node is improperly linked.

`trick_units(-)`

Definition at line 90 of file `time_messages.hh`.

Referenced by `jeod::JeodBaseTime::add_type_update()`.

#### 8.22.5.7 `char const * jeod::TimeMessages::invalid_setup_error` `[static]`

##### Initial value:

```
=
    "environment/time/" "invalid_setup_error"
```

Error issued when user tries to use something that doesn't exist in the model.

This is usually a user error. Note the distinction between invalid and incomplete. `trick_units(-)`

Definition at line 78 of file `time_messages.hh`.

Referenced by `jeod::TimeStandard::add_type_initialize()`, `jeod::TimeUDE::add_type_initialize()`, `jeod::JeodBaseTime::add_type_initialize()`, `jeod::JeodBaseTime::add_type_update()`, `jeod::TimeConverter::convert_a_to_b()`, `jeod::TimeConverter::convert_b_to_a()`, `jeod::TimeManagerInit::create_init_tree()`, `jeod::TimeConverter_TAI_TT::initialize()`, `jeod::TimeConverter_UT1_GMST::initialize()`, `jeod::TimeConverter_Dyn_TDB::initialize()`, `jeod::TimeConverter_TAI_GPS::initialize()`, `jeod::TimeConverter_Dyn_TAI::initialize()`, `jeod::TimeConverter_Dyn_UDE::initialize()`, `jeod::TimeConverter_STD_UDE::initialize()`, `jeod::TimeConverter_TAI_TDB::initialize()`, `jeod::JeodBaseTime::initialize_from_parent()`, `jeod::TimeDyn::initialize_initializer_time()`, `jeod::TimeStandard::initialize_initializer_time()`, `jeod::TimeUDE::initialize_initializer_time()`, and `jeod::TimeConverter_TAI_UTC::initialize_leap_`

second(), jeod::TimeUDE::set\_epoch\_dyn(), jeod::TimeUDE::set\_epoch\_initializing\_value(), jeod::TimeUDE::set\_epoch\_std(), jeod::TimeUDE::set\_epoch\_times(), jeod::TimeUDE::set\_epoch\_ude(), jeod::TimeUDE::set\_initial\_times(), jeod::TimeManager::time\_lookup(), jeod::TimeManagerInit::verify\_converter\_setup(), jeod::TimeUDE::verify\_epoch(), jeod::TimeConverter::verify\_setup(), jeod::TimeManagerInit::verify\_times\_setup(), and jeod::TimeUDE::verify\_update().

#### 8.22.5.8 char const \* jeod::TimeMessages::memory\_error [static]

##### Initial value:

```
=
    "environment/time/" "memory_error"
```

Error issued when system fails because something is not where it should be, or has a value other than its assumed value.

trick\_units(-)

Definition at line 71 of file time\_messages.hh.

Referenced by jeod::JeodBaseTime::add\_type\_update(), jeod::TimeStandard::initialize\_from\_parent(), jeod::TimeUDE::set\_initial\_times(), and jeod::JeodBaseTime::update().

#### 8.22.5.9 char const \* jeod::TimeMessages::redundancy\_error [static]

##### Initial value:

```
=
    "environment/time/" "redundancy_error"
```

Error issued when some value is multiply defined, and the code cannot determine which value to use.

Usually a user-error, from attempting to use too many of the initialization options simultaneously.trick\_units(-)

Definition at line 107 of file time\_messages.hh.

Referenced by jeod::TimeUDE::initialize\_from\_parent(), jeod::TimeStandard::initialize\_initializer\_time(), jeod::TimeManagerInit::populate\_converter\_registry(), jeod::TimeManager::register\_converter(), jeod::TimeManager::register\_time(), jeod::TimeUDE::set\_epoch\_dyn(), jeod::TimeUDE::set\_initial\_times(), jeod::TimeUDE::verify\_epoch(), jeod::TimeUDE::verify\_init(), and jeod::TimeManagerInit::verify\_times\_setup().

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

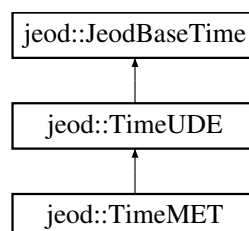
- [time\\_messages.hh](#)
- [time\\_messages.cc](#)

## 8.23 jeod::TimeMET Class Reference

A type of UDE time that allows for deliberate holds, or pauses.

```
#include <time_met.hh>
```

Inheritance diagram for jeod::TimeMET:



## Public Member Functions

- [TimeMET](#) ()
- [~TimeMET](#) ()  
*Destroy a Time\_MET.*
- void [update](#) (void)  
*Updates to current time.*

## Data Fields

- bool [hold](#)  
*Flags whether to hold time at current value.*

## Private Member Functions

- [TimeMET](#) (const [TimeMET](#) &)
- [TimeMET](#) & [operator=](#) (const [TimeMET](#) &)

## Private Attributes

- bool [previous\\_hold](#)  
*Previously known value of hold, used for recalculating converters.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeMET](#) ()

## Additional Inherited Members

### 8.23.1 Detailed Description

A type of UDE time that allows for deliberate holds, or pauses.

Definition at line 50 of file `time_met.hh`.

### 8.23.2 Constructor & Destructor Documentation

#### 8.23.2.1 `jeod::TimeMET::TimeMET ( void )`

Definition at line 69 of file `time_met.cc`.

References `jeod::JeodBaseTime::name`.

#### 8.23.2.2 `jeod::TimeMET::~~TimeMET ( void )`

Destroy a Time\_MET.

Definition at line 111 of file `time_met.cc`.

8.23.2.3 `jeod::TimeMET::TimeMET ( const TimeMET & ) [private]`

### 8.23.3 Member Function Documentation

8.23.3.1 `TimeMET& jeod::TimeMET::operator= ( const TimeMET & ) [private]`

8.23.3.2 `void jeod::TimeMET::update ( void ) [virtual]`

Updates to current time.

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 83 of file `time_met.cc`.

References `hold`, `previous_hold`, `jeod::TimeConverter::reset_a_to_b_offset()`, `jeod::JeodBaseTime::update()`, and `jeod::JeodBaseTime::update_converter_ptr`.

### 8.23.4 Friends And Related Function Documentation

8.23.4.1 `void init_attrjeod__TimeMET ( ) [friend]`

8.23.4.2 `friend class InputProcessor [friend]`

Definition at line 52 of file `time_met.hh`.

### 8.23.5 Field Documentation

8.23.5.1 `bool jeod::TimeMET::hold`

Flags whether to hold time at current value.

`trick_units(-)`

Definition at line 59 of file `time_met.hh`.

Referenced by `update()`.

8.23.5.2 `bool jeod::TimeMET::previous_hold [private]`

Previously known value of `hold`, used for recalculating converters.

`trick_units(-)`

Definition at line 65 of file `time_met.hh`.

Referenced by `update()`.

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

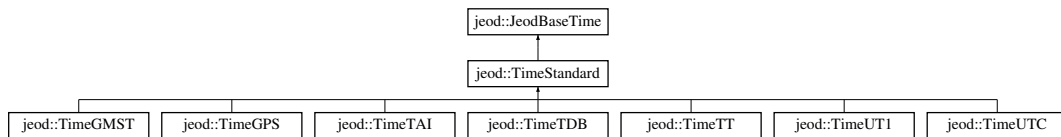
- [time\\_met.hh](#)
- [time\\_met.cc](#)

## 8.24 jeod::TimeStandard Class Reference

A class that serves as the base for all time representations that are well defined outside the simulation.

```
#include <time_standard.hh>
```

Inheritance diagram for `jeod::TimeStandard`:



## Public Member Functions

- [TimeStandard](#) ()  
*Construct a [TimeStandard](#).*
- virtual [~TimeStandard](#) ()  
*Destroy a [TimeStandard](#).*
- void [calendar\\_update](#) (double simtime)  
*Calls the function that converts the Julian-type representation of time (dd.xxxx days) to a calendar representation.*
- void [initialize\\_initializer\\_time](#) (TimeManagerInit \*tm\_init)  
*Each time type is initialized from its parent in the initialization tree, except one.*
- void [add\\_type\\_initialize](#) (const int seeking\_status, TimeManagerInit \*tm\_init)  
*Recursively adds elements to the initialization tree.*
- void [initialize\\_from\\_parent](#) (TimeManagerInit \*tm\_init)  
*Initialize a time type from its parent on the initialization tree.*
- virtual void [set\\_time\\_by\\_seconds](#) (const double new\_seconds)  
*Given a value of seconds, propagate to days and trunc\_julian\_time.*
- virtual void [set\\_time\\_by\\_days](#) (const double new\_days)  
*Given a value of days, propagate to seconds and trunc\_julian\_time.*
- void [set\\_time\\_by\\_trunc\\_julian](#) (const double new\_tjt)  
*Given a value of tjt, propagate to seconds and days.*
- double [julian\\_date\\_at\\_epoch](#) (void)  
*Returns the full Julian date at epoch, rather than the Truncated Julian Time.*
- double [seconds\\_of\\_year](#) (void)  
*Generate the number of seconds elapsed this year.*

## Data Fields

- double [last\\_calendar\\_update](#)  
*The simtime when the calendar update was last run.*
- int [prev\\_julian\\_day](#)  
*Used for determining whether to update the date in the calendar function.*
- double [seconds\\_at\\_year\\_start](#)  
*The value of "seconds" at the start of the year in which the last seconds\_of\_year calculation was made.*
- int [year\\_of\\_last\\_soy](#)  
*The year in which the last seconds\_of\_year calculation was made.*
- bool [send\\_warning\\_pre\\_1968](#)  
*This flag can be turned off by developers wanting to avoid warnings about a simulation being initialized pre-1968.*
- const double [tjt\\_mjt\\_offset](#)  
*Difference between Truncated Julian and Modified Julian.*
- const double [tjt\\_jd\\_offset](#)  
*Difference between Julian and Truncated Julian.*
- double [trunc\\_julian\\_time](#)  
*Truncated Julian time for this time-type.*
- double [julian\\_date](#)

*Conventional Julian Date.*

- double [tjt\\_at\\_epoch](#)

*Truncated Julian Date at epoch.*

- int [calendar\\_day](#)

*Gregorian calendar date day number.*

- int [calendar\\_hour](#)

*24-hour clock hour number.*

- int [calendar\\_minute](#)

*Clock minute number.*

- double [calendar\\_second](#)

*Clock second number.*

- int [calendar\\_year](#)

*Gregorian calendar year.*

- int [calendar\\_month](#)

*Gregorian calendar month.*

## Protected Member Functions

- virtual void [convert\\_from\\_calendar](#) (void)

*Calculate Truncated Julian date/time from Gregorian calendar date and 24-hour clock representation.*

- virtual void [calculate\\_calendar\\_values](#) (void)

*Calculate Gregorian calendar date and 24-hour clock representation from Truncated Julian date/time.*

- virtual void [set\\_epoch](#) (void)=0

*Set the epoch time.*

## Private Member Functions

- [TimeStandard](#) (const [TimeStandard](#) &)
- [TimeStandard](#) & [operator=](#) (const [TimeStandard](#) &)

## Friends

- class [InputProcessor](#)
- class [TimeUDE](#)
- void [init\\_attrjeod\\_\\_TimeStandard](#) ()

## Additional Inherited Members

### 8.24.1 Detailed Description

A class that serves as the base for all time representations that are well defined outside the simulation.

Definition at line 58 of file `time_standard.hh`.

### 8.24.2 Constructor & Destructor Documentation

#### 8.24.2.1 `jeod::TimeStandard::TimeStandard ( void )`

Construct a [TimeStandard](#).

Definition at line 62 of file `time_standard.cc`.

### 8.24.2.2 jeod::TimeStandard::~~TimeStandard ( void ) [virtual]

Destroy a [TimeStandard](#).

Definition at line 767 of file `time_standard.cc`.

### 8.24.2.3 jeod::TimeStandard::TimeStandard ( const TimeStandard & ) [private]

## 8.24.3 Member Function Documentation

### 8.24.3.1 void jeod::TimeStandard::add\_type\_initialize ( const int *seeking\_status*, TimeManagerInit \* *time\_manager\_init* ) [virtual]

Recursively adds elements to the initialization tree.

If the "parent" to a time-type is defined, adds the "parent" then returns to adding the "child" type. If the "parent" is not defined it searches for a suitable "parent" from the types already in the tree. If that search is successful, it adds the "child" to the tree, otherwise it returns without change.

#### Assumptions and Limitations

- This is vastly improved if the user defines the parent type "initialize\_from" for each time representation, except the top-level initializer type.
- Otherwise, the code will build the tree automatically, but it takes longer and may be less than ideal

#### Parameters

in	<i>seeking_status</i>	status-value for auto-seek
in	<i>time_manager_init</i>	The TM initializer.

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 178 of file `time_standard.cc`.

References `jeod::TimeManagerInit::get_conv_dir_init()`, `jeod::TimeManagerInit::get_status()`, `jeod::TimeManager::get_time_ptr()`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::TimeManagerInit::increment_status()`, `jeod::JeodBaseTime::index`, `jeod::JeodBaseTime::initialize_from_name`, `jeod::TimeMessages::invalid_setup_error`, `jeod::JeodBaseTime::name`, `jeod::TimeManager::num_types`, `jeod::TimeManagerInit::set_status()`, `jeod::TimeManager::time_lookup()`, and `jeod::JeodBaseTime::time_manager`.

### 8.24.3.2 void jeod::TimeStandard::calculate\_calendar\_values ( void ) [protected], [virtual]

Calculate Gregorian calendar date and 24-hour clock representation from Truncated Julian date/time.

#### Assumptions and Limitations

- Coverage is from March 1, 1600 onward.
- Produces a time in 24-hour clock format.
- Assumes that the values year, month, day, hour, minute, second, and truncated\_julian\_time are all present in the same class.

Reimplemented in [jeod::TimeGPS](#), and [jeod::TimeGMST](#).

Definition at line 297 of file `time_standard.cc`.

References `calendar_day`, `calendar_hour`, `calendar_minute`, `calendar_month`, `calendar_second`, `calendar_year`, `jeod::JeodBaseTime::clock_resolution`, `prev_julian_day`, and `trunc_julian_time`.

Referenced by `calendar_update()`, and `seconds_of_year()`.

#### 8.24.3.3 void jeod::TimeStandard::calendar\_update ( double *simtime* )

Calls the function that converts the Julian-type representation of time (dd.xxxx days) to a calendar representation. Makes sure that the time type on which it is called is up-to-date before doing so.

##### Assumptions and Limitations

- Derived times must have a parent; this should be defined by the user, or if not, already determined when the update\_tree was built.

##### Parameters

in	<i>simtime</i>	Simulation elapsed time, on the simulation clock Units: s
----	----------------	--

Definition at line 397 of file time\_standard.cc.

References calculate\_calendar\_values(), last\_calendar\_update, jeod::TimeManager::simtime, jeod::JeodBaseTime::time\_manager, and jeod::TimeManager::update().

#### 8.24.3.4 void jeod::TimeStandard::convert\_from\_calendar ( void ) [protected],[virtual]

Calculate Truncated Julian date/time from Gregorian calendar date and 24-hour clock representation.

##### Assumptions and Limitations

- Coverage s from March 1, 1600 onward.
- Assumes that time is in 24-hour clock format; 1:00:00 pm cannot be read correctly, but 13:00:00 can.
- Assumes that the values year, month, day, hour, minute, second, and truncated\_julian\_time are all present in the same class.

Reimplemented in [jeod::TimeGPS](#).

Definition at line 427 of file time\_standard.cc.

References calendar\_day, calendar\_hour, calendar\_minute, calendar\_month, calendar\_second, calendar\_year, jeod::JeodBaseTime::days, jeod::JeodBaseTime::seconds, tjt\_at\_epoch, and trunc\_julian\_time.

Referenced by initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.3.5 void jeod::TimeStandard::initialize\_from\_parent ( TimeManagerInit \* *time\_manager\_init* ) [virtual]

Initialize a time type from its parent on the initialization tree.

##### Assumptions and Limitations

- More than 1 time-type defined, otherwise this is not called.

##### Parameters

in	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 623 of file time\_standard.cc.

References jeod::TimeConverter::convert\_a\_to\_b(), jeod::TimeConverter::convert\_b\_to\_a(), jeod::TimeManagerInit::get\_conv\_dir\_init(), jeod::TimeManagerInit::get\_conv\_ptr\_index(), jeod::TimeManager::get\_converter\_ptr(), jeod::TimeManager::get\_time\_ptr(), jeod::TimeMessages::incomplete\_setup\_error, jeod::JeodBaseTime::index,



jeod::JeodBaseTime::initial\_value, jeod::TimeMessages::initialization\_error, jeod::TimeConverter::initialize(), jeod::JeodBaseTime::initialize\_from\_name, jeod::JeodBaseTime::initialize\_from\_parent(), jeod::JeodBaseTime::initialized, jeod::TimeConverter::is\_initialized(), jeod::JeodBaseTime::is\_initialized(), jeod::TimeMessages::memory\_error, jeod::JeodBaseTime::name, jeod::TimeManager::num\_types, jeod::JeodBaseTime::seconds, jeod::TimeManager::time\_lookup(), and jeod::JeodBaseTime::time\_manager.

#### 8.24.3.6 void jeod::TimeStandard::initialize\_initializer\_time ( TimeManagerInit \* time\_manager\_init ) [virtual]

Each time type is initialized from its parent in the initialization tree, except one.

In order to have an absolute reference time, one of the time types must be defined ahead of time. This is called the initializer time. This function initializes the initializer time.

##### Assumptions and Limitations

- [TimeDyn](#) cannot be used as the initializer time.
- Each time representation can have its own initializer function, or can inherit the one in [TimeStandard](#).

##### Parameters

in	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Implements [jeod::JeodBaseTime](#).

Definition at line 486 of file time\_standard.cc.

References [jeod::TimeEnum::calendar](#), [calendar\\_day](#), [calendar\\_hour](#), [calendar\\_minute](#), [calendar\\_month](#), [calendar\\_second](#), [calendar\\_year](#), [convert\\_from\\_calendar\(\)](#), [jeod::JeodBaseTime::days](#), [jeod::TimeEnum::days\\_since\\_epoch](#), [jeod::TimeMessages::incomplete\\_setup\\_error](#), [jeod::JeodBaseTime::initial\\_value](#), [jeod::JeodBaseTime::initialize\\_from\\_name](#), [jeod::JeodBaseTime::initialized](#), [jeod::JeodBaseTime::initializing\\_value](#), [jeod::TimeMessages::invalid\\_data\\_error](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [jeod::TimeEnum::Julian](#), [jeod::TimeEnum::julian](#), [jeod::TimeEnum::modified\\_julian](#), [jeod::JeodBaseTime::name](#), [jeod::TimeMessages::redundancy\\_error](#), [jeod::JeodBaseTime::seconds](#), [jeod::TimeEnum::seconds\\_since\\_epoch](#), [send\\_warning\\_pre\\_1968](#), [jeod::TimeManagerInit::sim\\_start\\_format](#), [tjt\\_at\\_epoch](#), [trunc\\_julian\\_time](#), [jeod::TimeEnum::truncated\\_julian](#), and [jeod::TimeEnum::undefined](#).

#### 8.24.3.7 double jeod::TimeStandard::julian\_date\_at\_epoch ( void )

Returns the full Julian date at epoch, rather than the Truncated Julian Time.

##### Returns

Truncated Julian Time at the epoch of the time-type.  
Units: day

Definition at line 150 of file time\_standard.cc.

References [tjt\\_at\\_epoch](#), and [tjt\\_jd\\_offset](#).

#### 8.24.3.8 TimeStandard& jeod::TimeStandard::operator= ( const TimeStandard & ) [private]

#### 8.24.3.9 double jeod::TimeStandard::seconds\_of\_year ( void )

Generate the number of seconds elapsed this year.

##### Assumptions and Limitations

- Relies on the accuracy of the JEOD2.0 calendar.

**Returns**

Current second of year.

Definition at line 705 of file `time_standard.cc`.

References `calculate_calendar_values()`, `calendar_day`, `calendar_hour`, `calendar_minute`, `calendar_month`, `calendar_second`, `calendar_year`, `convert_from_calendar()`, `jeod::JeodBaseTime::days`, `last_calendar_update`, `jeod::JeodBaseTime::seconds`, `seconds_at_year_start`, `jeod::TimeManager::simtime`, `jeod::JeodBaseTime::time_manager`, `trunc_julian_time`, and `year_of_last_soy`.

#### 8.24.3.10 `virtual void jeod::TimeStandard::set_epoch ( void )` `[protected]`, `[pure virtual]`

Set the epoch time.

Implemented in [jeod::TimeGPS](#), [jeod::TimeUT1](#), [jeod::TimeUTC](#), [jeod::TimeGMST](#), [jeod::TimeTAI](#), [jeod::TimeTDB](#), and [jeod::TimeTT](#).

#### 8.24.3.11 `void jeod::TimeStandard::set_time_by_days ( const double new_days )` `[virtual]`

Given a value of days, propagate to seconds and `trunc_julian_time`.

**Assumptions and Limitations**

- 86400 seconds = 1 day

**Parameters**

<code>in</code>	<code>new_days</code>	new value for days Units: day
-----------------	-----------------------	----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Reimplemented in [jeod::TimeGPS](#).

Definition at line 114 of file `time_standard.cc`.

References `jeod::JeodBaseTime::days`, `julian_date`, `jeod::JeodBaseTime::set_time_by_days()`, `tjt_at_epoch`, `tjt_jd_offset`, and `trunc_julian_time`.

Referenced by `jeod::TimeConverter_UT1_GMST::convert_a_to_b()`, and `jeod::TimeUDE::set_epoch_std()`.

#### 8.24.3.12 `void jeod::TimeStandard::set_time_by_seconds ( const double new_seconds )` `[virtual]`

Given a value of seconds, propagate to days and `trunc_julian_time`.

**Assumptions and Limitations**

- 86400 seconds = 1 day

**Parameters**

<code>in</code>	<code>new_seconds</code>	new value for seconds Units: s
-----------------	--------------------------	-----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Reimplemented in [jeod::TimeGPS](#).

Definition at line 95 of file `time_standard.cc`.

References `jeod::JeodBaseTime::days`, `julian_date`, `jeod::JeodBaseTime::set_time_by_seconds()`, `tjt_at_epoch`, `tjt_jd_offset`, and `trunc_julian_time`.

Referenced by `jeod::TimeConverter_TAI_TT::convert_a_to_b()`, `jeod::TimeConverter_Dyn_TDB::convert_a_to_b()`, `jeod::TimeConverter_Dyn_TAI::convert_a_to_b()`, `jeod::TimeConverter_TAI_TDB::convert_a_to_b()`, `jeod::TimeConverter_TAI_TT::convert_b_to_a()`, `jeod::TimeConverter_TAI_GPS::convert_b_to_a()`, `jeod::TimeConverter_S-TD_UDE::convert_b_to_a()`, `jeod::TimeConverter_TAI_TDB::convert_b_to_a()`, `jeod::TimeUDE::set_epoch_std()`, and `jeod::TimeGPS::set_time_by_seconds()`.

#### 8.24.3.13 void jeod::TimeStandard::set\_time\_by\_trunc\_julian ( const double *new\_tjt* )

Given a value of `tjt`, propagate to seconds and days.

#### Assumptions and Limitations

- 86400 seconds = 1 day

#### Parameters

<i>in</i>	<i>new_tjt</i>	new value for Truncated Julian Time Units: day
-----------	----------------	---

Definition at line 133 of file `time_standard.cc`.

References `jeod::JeodBaseTime::days`, `julian_date`, `jeod::JeodBaseTime::seconds`, `tjt_at_epoch`, `tjt_jd_offset`, and `trunc_julian_time`.

Referenced by `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UT1::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `jeod::TimeConverter_TAI_UT1::convert_b_to_a()`, `jeod::TimeUDE::set_epoch_std()`, and `jeod::TimeGPS::set_time_by_trunc_julian()`.

### 8.24.4 Friends And Related Function Documentation

#### 8.24.4.1 void init\_attrjeod\_\_TimeStandard ( ) [friend]

#### 8.24.4.2 friend class InputProcessor [friend]

Definition at line 60 of file `time_standard.hh`.

#### 8.24.4.3 friend class TimeUDE [friend]

Definition at line 62 of file `time_standard.hh`.

### 8.24.5 Field Documentation

#### 8.24.5.1 int jeod::TimeStandard::calendar\_day

Gregorian calendar date day number.

`trick_units(day)`

Definition at line 130 of file `time_standard.hh`.

Referenced by `calculate_calendar_values()`, `convert_from_calendar()`, `initialize_initializer_time()`, `seconds_of_year()`, and `jeod::TimeUDE::set_epoch_std()`.

#### 8.24.5.2 int jeod::TimeStandard::calendar\_hour

24-hour clock hour number.

`trick_units(hr)`

Definition at line 135 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.3 int jeod::TimeStandard::calendar\_minute

Clock minute number.

trick\_units(min)

Definition at line 140 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.4 int jeod::TimeStandard::calendar\_month

Gregorian calendar month.

trick\_units(-)

Definition at line 155 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.5 double jeod::TimeStandard::calendar\_second

Clock second number.

trick\_units(s)

Definition at line 145 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.6 int jeod::TimeStandard::calendar\_year

Gregorian calendar year.

trick\_units(-)

Definition at line 150 of file time\_standard.hh.

Referenced by calculate\_calendar\_values(), convert\_from\_calendar(), initialize\_initializer\_time(), seconds\_of\_year(), and jeod::TimeUDE::set\_epoch\_std().

#### 8.24.5.7 double jeod::TimeStandard::julian\_date

Conventional Julian Date.

NOTE - because this value is typically so large, it has very little room for fine-detail precision. It should only ever be used as an output for the likes of terminal displays and for input to legacy code. Never use for newly developed code.trick\_units(day)

Definition at line 120 of file time\_standard.hh.

Referenced by set\_time\_by\_days(), set\_time\_by\_seconds(), and set\_time\_by\_trunc\_julian().

**8.24.5.8 double jeod::TimeStandard::last\_calendar\_update**

The simtime when the calendar update was last run.

trick\_units(-)

Definition at line 70 of file time\_standard.hh.

Referenced by calendar\_update(), and seconds\_of\_year().

**8.24.5.9 int jeod::TimeStandard::prev\_julian\_day**

Used for determining whether to update the date in the calendar function.

trick\_units(day)

Definition at line 76 of file time\_standard.hh.

Referenced by calculate\_calendar\_values().

**8.24.5.10 double jeod::TimeStandard::seconds\_at\_year\_start**

The value of "seconds" at the start of the year in which the last seconds\_of\_year calculation was made.

Used for seconds\_of\_year calculations only.trick\_units(s)

Definition at line 83 of file time\_standard.hh.

Referenced by seconds\_of\_year().

**8.24.5.11 bool jeod::TimeStandard::send\_warning\_pre\_1968**

This flag can be turned off by developers wanting to avoid warnings about a simulation being initialized pre-1968.

The flag defaults to true - warning will be sent.trick\_units(-)

Definition at line 97 of file time\_standard.hh.

Referenced by initialize\_initializer\_time().

**8.24.5.12 double jeod::TimeStandard::tjt\_at\_epoch**

Truncated Julian Date at epoch.

trick\_units(day)

Definition at line 125 of file time\_standard.hh.

Referenced by convert\_from\_calendar(), jeod::TimeConverter\_TAI\_GPS::initialize(), jeod::TimeConverter\_TAI\_TDB::initialize(), initialize\_initializer\_time(), julian\_date\_at\_epoch(), jeod::TimeConverter\_TAI\_TDB::set\_a\_to\_b\_offset(), jeod::TimeTT::set\_epoch(), jeod::TimeTAI::set\_epoch(), jeod::TimeTDB::set\_epoch(), jeod::TimeUTC::set\_epoch(), jeod::TimeUT1::set\_epoch(), jeod::TimeGPS::set\_epoch(), set\_time\_by\_days(), set\_time\_by\_seconds(), and set\_time\_by\_trunc\_julian().

**8.24.5.13 const double jeod::TimeStandard::tjt\_jd\_offset**

Difference between Julian and Truncated Julian.

trick\_units(day)

Definition at line 107 of file time\_standard.hh.

Referenced by julian\_date\_at\_epoch(), set\_time\_by\_days(), set\_time\_by\_seconds(), and set\_time\_by\_trunc\_julian().

#### 8.24.5.14 `const double jeod::TimeStandard::tjt_mjt_offset`

Difference between Truncated Julian and Modified Julian.

`trick_units(day)`

Definition at line 102 of file `time_standard.hh`.

#### 8.24.5.15 `double jeod::TimeStandard::trunc_julian_time`

Truncated Julian time for this time-type.

`trick_units(day)`

Definition at line 112 of file `time_standard.hh`.

Referenced by `calculate_calendar_values()`, `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UT1::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `jeod::TimeConverter_TAI_UT1::convert_b_to_a()`, `convert_from_calendar()`, `jeod::TimeConverter_TAI_UTC::initialize()`, `jeod::TimeConverter_TAI_UT1::initialize()`, `initialize_initializer_time()`, `jeod::TimeConverter_TAI_UTC::initialize_leap_second()`, `jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1()`, `seconds_of_year()`, `jeod::TimeConverter_TAI_TDB::set_a_to_b_offset()`, `set_time_by_days()`, `set_time_by_seconds()`, `set_time_by_trunc_julian()`, `jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends()`, and `jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends()`.

#### 8.24.5.16 `int jeod::TimeStandard::year_of_last_soy`

The year in which the last `seconds_of_year` calculation was made.

At the start of this year, seconds had value `seconds_at_year_start`. Used for `seconds_of_year` calculations only.

`trick_units(-)`

Definition at line 90 of file `time_standard.hh`.

Referenced by `seconds_of_year()`.

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

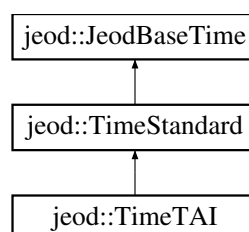
- [time\\_standard.hh](#)
- [time\\_standard.cc](#)

## 8.25 `jeod::TimeTAI` Class Reference

Represents International Atomic Time.

```
#include <time_tai.hh>
```

Inheritance diagram for `jeod::TimeTAI`:



### Public Member Functions

- [TimeTAI\(\)](#)

- *Construct a Time\_TAI.*
- [~TimeTAI](#) ()
- *Destroy a Time\_TAI.*

### Private Member Functions

- [TimeTAI](#) (const [TimeTAI](#) &)
- [TimeTAI](#) & [operator=](#) (const [TimeTAI](#) &)
- void [set\\_epoch](#) (void)
- *Sets the epoch for TAI time.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeTAI](#) ()

### Additional Inherited Members

#### 8.25.1 Detailed Description

Represents International Atomic Time.

Definition at line 49 of file `time_tai.hh`.

#### 8.25.2 Constructor & Destructor Documentation

##### 8.25.2.1 jeod::TimeTAI::TimeTAI ( void )

Construct a Time\_TAI.

Definition at line 51 of file `time_tai.cc`.

References `jeod::JeodBaseTime::name`, and `set_epoch()`.

##### 8.25.2.2 jeod::TimeTAI::~~TimeTAI ( void )

Destroy a Time\_TAI.

Definition at line 75 of file `time_tai.cc`.

##### 8.25.2.3 jeod::TimeTAI::TimeTAI ( const TimeTAI & ) [private]

#### 8.25.3 Member Function Documentation

##### 8.25.3.1 TimeTAI& jeod::TimeTAI::operator= ( const TimeTAI & ) [private]

##### 8.25.3.2 void jeod::TimeTAI::set\_epoch ( void ) [private], [virtual]

Sets the epoch for TAI time.

Implements [jeod::TimeStandard](#).

Definition at line 63 of file `time_tai.cc`.

References `jeod::TimeStandard::tjt_at_epoch`.

Referenced by `TimeTAI()`.

## 8.25.4 Friends And Related Function Documentation

8.25.4.1 `void init_attrjeod__TimeTAI ( )` [[friend](#)]

8.25.4.2 `friend class InputProcessor` [[friend](#)]

Definition at line 52 of file `time_tai.hh`.

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

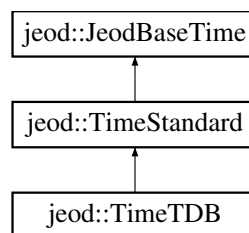
- [time\\_tai.hh](#)
- [time\\_tai.cc](#)

## 8.26 jeod::TimeTDB Class Reference

Represents Barycentric Dynamic Time.

```
#include <time_tdb.hh>
```

Inheritance diagram for `jeod::TimeTDB`:



### Public Member Functions

- [TimeTDB](#) ()  
*Construct a Time\_TDB.*
- [~TimeTDB](#) ()  
*Destroy a Time\_TDB.*

### Private Member Functions

- [TimeTDB](#) (const [TimeTDB](#) &)
- [TimeTDB](#) & [operator=](#) (const [TimeTDB](#) &)
- void [set\\_epoch](#) (void)  
*Sets the epoch for TDB time.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeTDB](#) ()



## Additional Inherited Members

### 8.26.1 Detailed Description

Represents Barycentric Dynamic Time.

Definition at line 49 of file `time_tdb.hh`.

### 8.26.2 Constructor & Destructor Documentation

#### 8.26.2.1 `jeod::TimeTDB::TimeTDB ( void )`

Construct a `Time_TDB`.

Definition at line 52 of file `time_tdb.cc`.

References `jeod::JeodBaseTime::name`, and `set_epoch()`.

#### 8.26.2.2 `jeod::TimeTDB::~~TimeTDB ( void )`

Destroy a `Time_TDB`.

Definition at line 76 of file `time_tdb.cc`.

#### 8.26.2.3 `jeod::TimeTDB::TimeTDB ( const TimeTDB & ) [private]`

### 8.26.3 Member Function Documentation

#### 8.26.3.1 `TimeTDB& jeod::TimeTDB::operator= ( const TimeTDB & ) [private]`

#### 8.26.3.2 `void jeod::TimeTDB::set_epoch ( void ) [private],[virtual]`

Sets the epoch for TDB time.

Implements [jeod::TimeStandard](#).

Definition at line 64 of file `time_tdb.cc`.

References `jeod::TimeStandard::tjt_at_epoch`.

Referenced by `TimeTDB()`.

### 8.26.4 Friends And Related Function Documentation

#### 8.26.4.1 `void init_attrjeod__TimeTDB ( ) [friend]`

#### 8.26.4.2 `friend class InputProcessor [friend]`

Definition at line 51 of file `time_tdb.hh`.

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

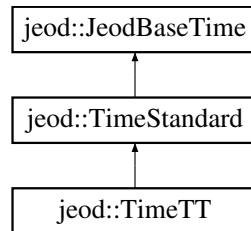
- [time\\_tdb.hh](#)
- [time\\_tdb.cc](#)

## 8.27 jeod::TimeTT Class Reference

Represents Terrestrial Time.

```
#include <time_tt.hh>
```

Inheritance diagram for jeod::TimeTT:



### Public Member Functions

- [TimeTT](#) ()  
*Construct a Time\_TT.*
- [~TimeTT](#) ()  
*Destroy a Time\_TT.*

### Private Member Functions

- [TimeTT](#) (const [TimeTT](#) &)
- [TimeTT](#) & [operator=](#) (const [TimeTT](#) &)
- void [set\\_epoch](#) (void)  
*Sets the epoch for TT time.*

### Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeTT](#) ()

### Additional Inherited Members

#### 8.27.1 Detailed Description

Represents Terrestrial Time.

Definition at line 49 of file time\_tt.hh.

#### 8.27.2 Constructor & Destructor Documentation

##### 8.27.2.1 jeod::TimeTT::TimeTT ( void )

Construct a Time\_TT.

Definition at line 52 of file time\_tt.cc.

References [jeod::JeodBaseTime::name](#), and [set\\_epoch\(\)](#).

## 8.27.2.2 jeod::TimeTT::~~TimeTT ( void )

Destroy a Time\_TT.

Definition at line 76 of file time\_tt.cc.

## 8.27.2.3 jeod::TimeTT::TimeTT ( const TimeTT &amp; ) [private]

## 8.27.3 Member Function Documentation

## 8.27.3.1 TimeTT&amp; jeod::TimeTT::operator= ( const TimeTT &amp; ) [private]

## 8.27.3.2 void jeod::TimeTT::set\_epoch ( void ) [private],[virtual]

Sets the epoch for TT time.

Implements [jeod::TimeStandard](#).

Definition at line 64 of file time\_tt.cc.

References [jeod::TimeStandard::tjt\\_at\\_epoch](#).

Referenced by [TimeTT\(\)](#).

## 8.27.4 Friends And Related Function Documentation

## 8.27.4.1 void init\_attrjeod\_\_TimeTT ( ) [friend]

## 8.27.4.2 friend class InputProcessor [friend]

Definition at line 51 of file time\_tt.hh.

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

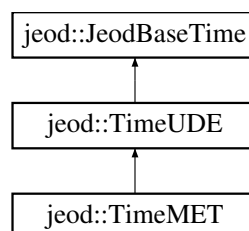
- [time\\_tt.hh](#)
- [time\\_tt.cc](#)

## 8.28 jeod::TimeUDE Class Reference

Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.

```
#include <time_ude.hh>
```

Inheritance diagram for jeod::TimeUDE:



## Public Member Functions

- [TimeUDE \(\)](#)

Constructor for class `TimeUDE`.

- virtual `~TimeUDE ()`  
Destructor for `TimeUDE`.
- void `initialize_initializer_time (TimeManagerInit *tm_init)`  
Each time type is initialized from its parent in the initialization tree, except one.
- void `add_type_initialize (const int seeking_status, TimeManagerInit *tm_init)`  
Adds a UDE type to the initialization tree when it is appropriate to do so.
- void `initialize_from_parent (TimeManagerInit *tm_init)`  
Initializes this time-type.
- void `set_time_by_clock (void)`  
sets the decimal representation of time by the clock
- void `set_time_by_seconds (const double new_seconds)`  
Given a seconds value, sets days and clock values.
- void `set_time_by_days (const double new_days)`  
Given a seconds value, sets days and clock values.
- void `set_epoch_initializing_value (const double simtime, const double epoch_initializing_value)`  
sets the initial epoch value

## Data Fields

- int `epoch_year`  
Gregorian calendar year number at epoch.
- int `epoch_month`  
Gregorian calendar month number at epoch.
- int `epoch_day`  
Gregorian calendar day number at epoch.
- int `epoch_hour`  
24-hour clock hour number at epoch.
- int `epoch_minute`  
Clock minute number at epoch.
- double `epoch_second`  
Clock seconds value at epoch.
- int `clock_day`  
Whole number of days since epoch, in clock format.
- int `clock_hour`  
Whole number of hours since epoch, in clock format.
- int `clock_minute`  
Whole number of minutes since epoch, in clock format.
- double `clock_second`  
Number of seconds since epoch, in clock format.
- double `last_clock_update`  
Simtime at the last time the clock was updated.
- `TimeEnum::TimeFormat` `epoch_format`  
Format for expressing the epoch of this type (calendar, julian, etc)
- `TimeEnum::TimeFormat` `initial_value_format`  
Format for expressing the initial value of this type (calendar, julian, etc )
- `std::string` `epoch_defined_in_name`  
Name of time type in which epoch defined.

## Protected Member Functions

- bool [must\\_be\\_singleton](#) ()  
*Returns false in response to the question "does this time class have to be a singleton".*
- void [convert\\_epoch\\_to\\_update](#) (JeodBaseTime \*epoch\_ptr, JeodBaseTime \*update\_ptr, TimeManagerInit \*tm\_init)  
*Converts the time, as specified in the epoch time-type to the update\_from time-type.*
- void [set\\_epoch\\_dyn](#) (TimeDyn \*epoch\_ptr)  
*Temporarily overwrites the simulation data in time type "epoch" with the epoch value.*
- void [set\\_epoch\\_times](#) (JeodBaseTime \*epoch\_ptr)  
*To set the times in the epoch time type coincident with the zero-point of this time-type.*
- void [set\\_epoch\\_ude](#) (TimeUDE \*epoch\_ptr)  
*Overwrites the data in time type "epoch" with that in this class that specifies the epoch.*
- void [set\\_epoch\\_std](#) (TimeStandard \*epoch\_ptr)  
*Overwrites the data in time type "epoch" with that in this class that specifies the epoch.*
- void [set\\_initial\\_times](#) (void)  
*Sets the initial value of this type from the myriad of initialization options.*
- void [clock\\_update](#) ()  
*converts the decimal seconds value to a clock interface*
- void [verify\\_epoch](#) (void)  
*Verifies that the epoch assignments are legitimate, and tests for the presence and legitimacy of values for defining the epoch.*
- void [verify\\_init](#) (void)  
*Verifies that any assignment to initialize\_from is flagged as inappropriate, and tests for the presence of initializing data.*
- void [verify\\_update](#) (void)  
*Ensures that the time-type identified as "update\_from" is legitimate.*

## Protected Attributes

- double [epoch\\_initializing\\_value](#)  
*Value of epoch in appropriate format.*
- bool [initializing\\_data\\_present](#)  
*Whether initializing data is present.*
- bool [epoch\\_data\\_present](#)  
*Whether epoch data is present.*
- bool [epoch\\_value\\_is\\_set\\_number](#)  
*Whether there is some numerical input that could set epoch.*
- bool [epoch\\_value\\_is\\_set\\_calendar](#)  
*Whether there is some calendar input that could set epoch.*
- bool [epoch\\_value\\_is\\_set\\_clock](#)  
*Whether there is some clock input that could set epoch.*
- int [update\\_index](#)  
*The index of the time-type from which this one is updated.*
- int [epoch\\_index](#)  
*The index of the time-type in which this one's epoch is defined.*

## Private Member Functions

- [TimeUDE](#) (const [TimeUDE](#) &)
- [TimeUDE](#) & [operator=](#) (const [TimeUDE](#) &)

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeUDE](#) ()

### 8.28.1 Detailed Description

Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.

Definition at line 55 of file `time_ude.hh`.

### 8.28.2 Constructor & Destructor Documentation

#### 8.28.2.1 `jeod::TimeUDE::TimeUDE ( void )`

Constructor for class [TimeUDE](#).

#### Assumptions and Limitations

- None

Definition at line 66 of file `time_ude.cc`.

#### 8.28.2.2 `jeod::TimeUDE::~~TimeUDE ( void ) [virtual]`

Destructor for [TimeUDE](#).

Definition at line 1473 of file `time_ude.cc`.

#### 8.28.2.3 `jeod::TimeUDE::TimeUDE ( const TimeUDE & ) [private]`

### 8.28.3 Member Function Documentation

#### 8.28.3.1 `void jeod::TimeUDE::add_type_initialize ( const int seeking_status, TimeManagerInit * time_manager_init ) [virtual]`

Adds a UDE type to the initialization tree when it is appropriate to do so.

#### Assumptions and Limitations

- The time type from which the UDE updates must be in the tree above the UDE.
- If the time type in which the epoch is defined is another UDE, it also must be in the tree above this UDE
- This function is only called when the UDE is NOT being used to initialize the simulation.

#### Parameters

in	<i>seeking_status</i>	An indicator of relative level of progression in the tree.
in	<i>time_manager_init</i>	The TM initializer.

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 127 of file `time_ude.cc`.

References `epoch_defined_in_name`, `epoch_index`, `jeod::TimeManagerInit::get_conv_dir_init()`, `jeod::TimeManagerInit::get_status()`, `jeod::TimeManager::get_time_ptr()`, `jeod::TimeMessages::incomplete_setup_error`, `jeod::TimeManagerInit::increment_status()`, `jeod::JeodBaseTime::index`, `jeod::TimeMessages::invalid_setup_error`,

jeod::JeodBaseTime::name, jeod::TimeManager::num\_types, jeod::TimeManagerInit::set\_status(), jeod::JeodBaseTime::time\_manager, jeod::JeodBaseTime::update\_from\_name, update\_index, verify\_epoch(), and verify\_update().

### 8.28.3.2 void jeod::TimeUDE::clock\_update ( void ) [protected]

converts the decimal seconds value to a clock interface

#### Assumptions and Limitations

- 24 hrs = 1 day; 60 minutes - 1 hour ; 60 seconds = 1 minute

Definition at line 1281 of file time\_ude.cc.

References clock\_day, clock\_hour, clock\_minute, jeod::JeodBaseTime::clock\_resolution, clock\_second, and jeod::JeodBaseTime::seconds.

Referenced by set\_time\_by\_days(), and set\_time\_by\_seconds().

### 8.28.3.3 void jeod::TimeUDE::convert\_epoch\_to\_update ( JeodBaseTime \* epoch\_ptr, JeodBaseTime \* update\_from\_ptr, TimeManagerInit \* time\_manager\_init ) [protected]

Converts the time, as specified in the epoch time-type to the update\_from time-type.

This sets the update\_from time at the epoch of "this", and allows for the initialization of the converter.

#### Assumptions and Limitations

- That there is a converter available to do this in one step
- Future work may include an extension to this routine to cover other cases.

#### Parameters

in	<i>epoch_ptr</i>	pointer to the epoch time-type
in	<i>update_from_ptr</i>	pointer to the time-type from which this time-type will be updated.
in	<i>time_manager_init</i>	The TM initializer.

Definition at line 254 of file time\_ude.cc.

References jeod::TimeConverter::convert\_a\_to\_b(), jeod::TimeConverter::convert\_b\_to\_a(), epoch\_defined\_in\_name, epoch\_index, jeod::TimeManagerInit::get\_conv\_dir\_init(), jeod::TimeManagerInit::get\_conv\_ptr\_index(), jeod::TimeManager::get\_converter\_ptr(), jeod::TimeMessages::incomplete\_setup\_error, jeod::TimeConverter::initialize(), jeod::JeodBaseTime::name, jeod::TimeManager::num\_types, jeod::TimeConverter::override\_initialized(), jeod::JeodBaseTime::override\_initialized(), jeod::JeodBaseTime::time\_manager, jeod::JeodBaseTime::update\_from\_name, and update\_index.

Referenced by initialize\_from\_parent(), and initialize\_initializer\_time().

### 8.28.3.4 void jeod::TimeUDE::initialize\_from\_parent ( TimeManagerInit \* time\_manager\_init ) [virtual]

Initializes this time-type.

#### Assumptions and Limitations

- The subject object has a parent, a time-type with which it ticks. This has already been tested for.

## Parameters

in	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 318 of file time\_ude.cc.

References [jeod::TimeConverter::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter::convert\\_b\\_to\\_a\(\)](#), [convert\\_epoch\\_to\\_update\(\)](#), [jeod::JeodBaseTime::days](#), [epoch\\_data\\_present](#), [epoch\\_index](#), [jeod::TimeManagerInit::get\\_conv\\_dir\\_init\(\)](#), [jeod::TimeManagerInit::get\\_conv\\_ptr\\_index\(\)](#), [jeod::TimeManager::get\\_converter\\_ptr\(\)](#), [jeod::TimeManager::get\\_time\\_ptr\(\)](#), [jeod::TimeMessages::incomplete\\_setup\\_error](#), [jeod::JeodBaseTime::index](#), [jeod::JeodBaseTime::initial\\_value](#), [jeod::TimeMessages::initialization\\_error](#), [jeod::TimeConverter::initialize\(\)](#), [jeod::JeodBaseTime::initialize\\_from\\_parent\(\)](#), [jeod::JeodBaseTime::initialized](#), [initializing\\_data\\_present](#), [jeod::TimeConverter::is\\_initialized\(\)](#), [jeod::JeodBaseTime::is\\_initialized\(\)](#), [jeod::JeodBaseTime::name](#), [jeod::TimeManager::num\\_types](#), [jeod::JeodBaseTime::override\\_initialized\(\)](#), [jeod::TimeMessages::redundancy\\_error](#), [jeod::JeodBaseTime::seconds](#), [set\\_epoch\\_times\(\)](#), [jeod::JeodBaseTime::set\\_time\\_by\\_seconds\(\)](#), [jeod::JeodBaseTime::time\\_manager](#), [jeod::JeodBaseTime::update\\_from\\_name](#), [update\\_index](#), and [verify\\_init\(\)](#).

**8.28.3.5** `void jeod::TimeUDE::initialize_initializer_time ( TimeManagerInit * time_manager_init ) [virtual]`

Each time type is initialized from its parent in the initialization tree, except one.

In order to have an absolute reference time, one of the time types must be defined ahead of time. This is called the initializer time. This function initializes the initializer time.

## Parameters

in	<i>time_manager_init</i>	The TM initializer.
----	--------------------------	---------------------

Implements [jeod::JeodBaseTime](#).

Definition at line 513 of file time\_ude.cc.

References [jeod::TimeConverter::convert\\_a\\_to\\_b\(\)](#), [jeod::TimeConverter::convert\\_b\\_to\\_a\(\)](#), [convert\\_epoch\\_to\\_update\(\)](#), [jeod::JeodBaseTime::days](#), [jeod::TimeMessages::duplicate\\_methods](#), [epoch\\_data\\_present](#), [epoch\\_index](#), [jeod::TimeManagerInit::get\\_conv\\_dir\\_init\(\)](#), [jeod::TimeManagerInit::get\\_conv\\_ptr\\_index\(\)](#), [jeod::TimeManager::get\\_converter\\_ptr\(\)](#), [jeod::TimeManager::get\\_time\\_ptr\(\)](#), [jeod::TimeMessages::incomplete\\_setup\\_error](#), [jeod::JeodBaseTime::index](#), [initial\\_value\\_format](#), [jeod::TimeConverter::initialize\(\)](#), [jeod::JeodBaseTime::initialized](#), [initializing\\_data\\_present](#), [jeod::TimeMessages::invalid\\_setup\\_error](#), [jeod::JeodBaseTime::name](#), [jeod::TimeManager::num\\_types](#), [jeod::JeodBaseTime::override\\_initialized\(\)](#), [jeod::JeodBaseTime::seconds](#), [set\\_epoch\\_times\(\)](#), [jeod::TimeManagerInit::sim\\_start\\_format](#), [jeod::JeodBaseTime::time\\_manager](#), [jeod::TimeManager::time\\_standards\\_exist\(\)](#), [jeod::TimeEnum::undefined](#), [jeod::JeodBaseTime::update\\_from\\_name](#), [update\\_index](#), [verify\\_epoch\(\)](#), [verify\\_init\(\)](#), and [verify\\_update\(\)](#).

**8.28.3.6** `bool jeod::TimeUDE::must_be_singleton ( void ) [protected],[virtual]`

Returns false in response to the question "does this time class have to be a singleton".

## Assumptions and Limitations

- There can be more than one UDE

## Returns

false

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 106 of file time\_ude.cc.



8.28.3.7 **TimeUDE**& jeod::TimeUDE::operator= ( const TimeUDE & ) [private]

8.28.3.8 void jeod::TimeUDE::set\_epoch\_dyn ( TimeDyn \* epoch\_ptr ) [protected]

Temporarily overwrites the simulation data in time type "epoch" with the epoch value.

#### Assumptions and Limitations

- "Epoch" is DynTime

#### Parameters

in	epoch_ptr	pointer to the epoch time-type
----	-----------	--------------------------------

Definition at line 756 of file time\_ude.cc.

References jeod::TimeEnum::calendar, jeod::TimeEnum::clock, jeod::TimeEnum::days\_since\_epoch, epoch\_data\_present, epoch\_day, epoch\_defined\_in\_name, epoch\_format, epoch\_hour, epoch\_initializing\_value, epoch\_minute, epoch\_month, epoch\_value\_is\_set\_number, epoch\_year, jeod::TimeMessages::incomplete\_setup\_error, initializing\_data\_present, jeod::TimeMessages::invalid\_setup\_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeEnum::modified\_julian, jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy\_error, jeod::TimeEnum::seconds\_since\_epoch, jeod::JeodBaseTime::set\_time\_by\_days(), jeod::JeodBaseTime::set\_time\_by\_seconds(), jeod::TimeEnum::truncated\_julian, and jeod::TimeEnum::undefined.

Referenced by set\_epoch\_times().

8.28.3.9 void jeod::TimeUDE::set\_epoch\_initializing\_value ( const double simtime, const double epoch )

sets the initial epoch value

#### Assumptions and Limitations

- Assumes that the number that is passed in is correctly entered with the correct units interpretation.

#### Parameters

in	simtime	Used to verify that this is at initialization
in	epoch	the value to be used.

Definition at line 1257 of file time\_ude.cc.

References epoch\_initializing\_value, jeod::TimeMessages::invalid\_setup\_error, and jeod::JeodBaseTime::name.

8.28.3.10 void jeod::TimeUDE::set\_epoch\_std ( TimeStandard \* epoch\_ptr ) [protected]

Overwrites the data in time type "epoch" with that in this class that specifies the epoch.

#### Assumptions and Limitations

- "Epoch" is Absolute Derived Time

#### Parameters

in	epoch_ptr	pointer to the epoch time-type
----	-----------	--------------------------------

Definition at line 861 of file time\_ude.cc.

References jeod::TimeEnum::calendar, jeod::TimeStandard::calendar\_day, jeod::TimeStandard::calendar\_hour, jeod::TimeStandard::calendar\_minute, jeod::TimeStandard::calendar\_month, jeod::TimeStandard::calendar\_second, jeod::TimeStandard::calendar\_year, jeod::TimeEnum::clock, jeod::TimeStandard::convert\_from\_calendar(), jeod::TimeEnum::days\_since\_epoch, epoch\_day, epoch\_defined\_in\_name, epoch\_format, epoch\_hour,

epoch\_initializing\_value, epoch\_minute, epoch\_month, epoch\_second, epoch\_value\_is\_set\_calendar, epoch\_value\_is\_set\_number, epoch\_year, jeod::TimeMessages::incomplete\_setup\_error, jeod::TimeMessages::invalid\_setup\_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeEnum::modified\_julian, jeod::JeodBaseTime::name, jeod::TimeEnum::seconds\_since\_epoch, jeod::TimeStandard::set\_time\_by\_days(), jeod::TimeStandard::set\_time\_by\_seconds(), jeod::TimeStandard::set\_time\_by\_trunc\_julian(), jeod::TimeEnum::truncated\_julian, and jeod::TimeEnum::undefined.

Referenced by set\_epoch\_times().

#### 8.28.3.11 void jeod::TimeUDE::set\_epoch\_times ( JeodBaseTime \* epoch\_ptr ) [protected]

To set the times in the epoch time type coincident with the zero-point of this time-type.

##### Assumptions and Limitations

- "This" is being defined by epoch.

##### Parameters

in	epoch_ptr	pointer to the epoch time-type
----	-----------	--------------------------------

Definition at line 719 of file time\_ude.cc.

References jeod::TimeMessages::invalid\_setup\_error, set\_epoch\_dyn(), set\_epoch\_std(), and set\_epoch\_ude().

Referenced by initialize\_from\_parent(), and initialize\_initializer\_time().

#### 8.28.3.12 void jeod::TimeUDE::set\_epoch\_ude ( TimeUDE \* epoch\_ptr ) [protected]

Overwrites the data in time type "epoch" with that in this class that specifies the epoch.

##### Assumptions and Limitations

- "Epoch" is a User-Defined-Epoch Time.

##### Parameters

in	epoch_ptr	pointer to the epoch time-type
----	-----------	--------------------------------

Definition at line 995 of file time\_ude.cc.

References jeod::TimeEnum::calendar, jeod::TimeEnum::clock, clock\_day, clock\_hour, clock\_minute, clock\_second, jeod::TimeEnum::days\_since\_epoch, epoch\_day, epoch\_defined\_in\_name, epoch\_format, epoch\_hour, epoch\_initializing\_value, epoch\_minute, epoch\_second, epoch\_value\_is\_set\_clock, epoch\_value\_is\_set\_number, jeod::TimeMessages::incomplete\_setup\_error, jeod::TimeMessages::invalid\_setup\_error, jeod::TimeEnum::Julian, jeod::TimeEnum::julian, jeod::TimeEnum::modified\_julian, jeod::JeodBaseTime::name, jeod::TimeEnum::seconds\_since\_epoch, set\_time\_by\_clock(), set\_time\_by\_seconds(), jeod::TimeEnum::truncated\_julian, and jeod::TimeEnum::undefined.

Referenced by set\_epoch\_times().

#### 8.28.3.13 void jeod::TimeUDE::set\_initial\_times ( void ) [protected]

Sets the initial value of this type from the myriad of initialization options.

##### Assumptions and Limitations

- At least one of the following is non-zero: initializing value, clock\_day, clock\_hour, clock\_minute, clock\_second, seconds, days

Definition at line 1097 of file time\_ude.cc.

References `jeod::TimeEnum::calendar`, `clock_day`, `clock_hour`, `clock_minute`, `clock_second`, `jeod::JeodBaseTime::days`, `jeod::TimeEnum::days_since_epoch`, `jeod::TimeMessages::incomplete_setup_error`, `initial_value_format`, `initializing_data_present`, `jeod::JeodBaseTime::initializing_value`, `jeod::TimeMessages::invalid_setup_error`, `jeod::TimeEnum::Julian`, `jeod::TimeEnum::julian`, `jeod::TimeMessages::memory_error`, `jeod::TimeEnum::modified_julian`, `jeod::JeodBaseTime::name`, `jeod::TimeMessages::redundancy_error`, `jeod::JeodBaseTime::seconds`, `jeod::TimeEnum::seconds_since_epoch`, `jeod::TimeEnum::truncated_julian`, and `jeod::TimeEnum::undefined`.

Referenced by `verify_init()`.

#### 8.28.3.14 void jeod::TimeUDE::set\_time\_by\_clock ( void )

sets the decimal representation of time by the clock

##### Assumptions and Limitations

- 24 hrs = 1 day; 60 minutes - 1 hour ; 60 seconds = 1 minute

Definition at line 1237 of file time\_ude.cc.

References `clock_day`, `clock_hour`, `clock_minute`, `clock_second`, `jeod::JeodBaseTime::days`, and `jeod::JeodBaseTime::seconds`.

Referenced by `set_epoch_ude()`.

#### 8.28.3.15 void jeod::TimeUDE::set\_time\_by\_days ( const double *new\_days* ) [virtual]

Given a seconds value, sets days and clock values.

##### Parameters

in	<i>new_days</i>	new value for days Units: day
----	-----------------	----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 1204 of file time\_ude.cc.

References `clock_update()`, and `jeod::JeodBaseTime::set_time_by_days()`.

#### 8.28.3.16 void jeod::TimeUDE::set\_time\_by\_seconds ( const double *new\_seconds* ) [virtual]

Given a seconds value, sets days and clock values.

##### Parameters

in	<i>new_seconds</i>	new value for seconds Units: s
----	--------------------	-----------------------------------

Reimplemented from [jeod::JeodBaseTime](#).

Definition at line 1218 of file time\_ude.cc.

References `clock_update()`, and `jeod::JeodBaseTime::set_time_by_seconds()`.

Referenced by `jeod::TimeConverter_Dyn_UDE::convert_a_to_b()`, `jeod::TimeConverter_STD_UDE::convert_a_to_b()`, and `set_epoch_ude()`.

#### 8.28.3.17 void jeod::TimeUDE::verify\_epoch ( void ) [protected]

Verifies that the epoch assignments are legitimate, and tests for the presence and legitimacy of values for defining the epoch.

Definition at line 1313 of file time\_ude.cc.

References epoch\_data\_present, epoch\_day, epoch\_defined\_in\_name, epoch\_format, epoch\_hour, epoch\_index, epoch\_initializing\_value, epoch\_minute, epoch\_month, epoch\_second, epoch\_value\_is\_set\_calendar, epoch\_value\_is\_set\_clock, epoch\_value\_is\_set\_number, epoch\_year, jeod::TimeMessages::incomplete\_setup\_error, jeod::JeodBaseTime::index, jeod::TimeMessages::invalid\_setup\_error, jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy\_error, jeod::TimeManager::time\_lookup(), jeod::JeodBaseTime::time\_manager, and jeod::TimeEnum::undefined.

Referenced by add\_type\_initialize(), and initialize\_initializer\_time().

#### 8.28.3.18 void jeod::TimeUDE::verify\_init ( void ) [protected]

Verifies that any assignment to initialize\_from is flagged as inappropriate, and tests for the presence of initializing data.

Definition at line 1407 of file time\_ude.cc.

References jeod::JeodBaseTime::initialize\_from\_name, jeod::JeodBaseTime::name, jeod::TimeMessages::redundancy\_error, set\_initial\_times(), jeod::TimeManager::time\_lookup(), and jeod::JeodBaseTime::time\_manager.

Referenced by initialize\_from\_parent(), and initialize\_initializer\_time().

#### 8.28.3.19 void jeod::TimeUDE::verify\_update ( void ) [protected]

Ensures that the time-type identified as "update\_from" is legitimate.

Definition at line 1441 of file time\_ude.cc.

References jeod::TimeManager::get\_time\_ptr(), jeod::TimeMessages::incomplete\_setup\_error, jeod::TimeMessages::invalid\_setup\_error, jeod::TimeManager::time\_lookup(), jeod::JeodBaseTime::time\_manager, jeod::JeodBaseTime::update\_from\_name, and update\_index.

Referenced by add\_type\_initialize(), and initialize\_initializer\_time().

### 8.28.4 Friends And Related Function Documentation

#### 8.28.4.1 void init\_attrjeod\_TimeUDE ( ) [friend]

#### 8.28.4.2 friend class InputProcessor [friend]

Definition at line 57 of file time\_ude.hh.

### 8.28.5 Field Documentation

#### 8.28.5.1 int jeod::TimeUDE::clock\_day

Whole number of days since epoch, in clock format.

trick\_units(-)

Definition at line 89 of file time\_ude.hh.

Referenced by clock\_update(), set\_epoch\_ude(), set\_initial\_times(), and set\_time\_by\_clock().

#### 8.28.5.2 int jeod::TimeUDE::clock\_hour

Whole number of hours since epoch, in clock format.

trick\_units(-)

Definition at line 94 of file time\_ude.hh.

Referenced by clock\_update(), set\_epoch\_ude(), set\_initial\_times(), and set\_time\_by\_clock().

#### 8.28.5.3 int jeod::TimeUDE::clock\_minute

Whole number of minutes since epoch, in clock format.

trick\_units(-)

Definition at line 99 of file time\_ude.hh.

Referenced by clock\_update(), set\_epoch\_ude(), set\_initial\_times(), and set\_time\_by\_clock().

#### 8.28.5.4 double jeod::TimeUDE::clock\_second

Number of seconds since epoch, in clock format.

trick\_units(s)

Definition at line 104 of file time\_ude.hh.

Referenced by clock\_update(), set\_epoch\_ude(), set\_initial\_times(), and set\_time\_by\_clock().

#### 8.28.5.5 bool jeod::TimeUDE::epoch\_data\_present [protected]

Whether epoch data is present.

trick\_units(-)

Definition at line 140 of file time\_ude.hh.

Referenced by initialize\_from\_parent(), initialize\_initializer\_time(), set\_epoch\_dyn(), and verify\_epoch().

#### 8.28.5.6 int jeod::TimeUDE::epoch\_day

Gregorian calendar day number at epoch.

trick\_units(day)

Definition at line 72 of file time\_ude.hh.

Referenced by set\_epoch\_dyn(), set\_epoch\_std(), set\_epoch\_ude(), and verify\_epoch().

#### 8.28.5.7 std::string jeod::TimeUDE::epoch\_defined\_in\_name

Name of time type in which epoch defined.

trick\_units(-)

Definition at line 124 of file time\_ude.hh.

Referenced by add\_type\_initialize(), convert\_epoch\_to\_update(), set\_epoch\_dyn(), set\_epoch\_std(), set\_epoch\_ude(), and verify\_epoch().

#### 8.28.5.8 TimeEnum::TimeFormat jeod::TimeUDE::epoch\_format

Format for expressing the epoch of this type (calendar, julian, etc)

trick\_units(-)

Definition at line 114 of file time\_ude.hh.

Referenced by `set_epoch_dyn()`, `set_epoch_std()`, `set_epoch_ude()`, and `verify_epoch()`.

#### 8.28.5.9 `int jeod::TimeUDE::epoch_hour`

24-hour clock hour number at epoch.

`trick_units(hr)`

Definition at line 76 of file `time_ude.hh`.

Referenced by `set_epoch_dyn()`, `set_epoch_std()`, `set_epoch_ude()`, and `verify_epoch()`.

#### 8.28.5.10 `int jeod::TimeUDE::epoch_index` `[protected]`

The index of the time-type in which this one's epoch is defined.

`trick_units(-)`

Definition at line 165 of file `time_ude.hh`.

Referenced by `add_type_initialize()`, `convert_epoch_to_update()`, `initialize_from_parent()`, `initialize_initializer_time()`, and `verify_epoch()`.

#### 8.28.5.11 `double jeod::TimeUDE::epoch_initializing_value` `[protected]`

Value of epoch in appropriate format.

`trick_units(-)`

Definition at line 130 of file `time_ude.hh`.

Referenced by `set_epoch_dyn()`, `set_epoch_initializing_value()`, `set_epoch_std()`, `set_epoch_ude()`, and `verify_epoch()`.

#### 8.28.5.12 `int jeod::TimeUDE::epoch_minute`

Clock minute number at epoch.

`trick_units(min)`

Definition at line 80 of file `time_ude.hh`.

Referenced by `set_epoch_dyn()`, `set_epoch_std()`, `set_epoch_ude()`, and `verify_epoch()`.

#### 8.28.5.13 `int jeod::TimeUDE::epoch_month`

Gregorian calendar month number at epoch.

`trick_units(-)`

Definition at line 68 of file `time_ude.hh`.

Referenced by `set_epoch_dyn()`, `set_epoch_std()`, and `verify_epoch()`.

#### 8.28.5.14 `double jeod::TimeUDE::epoch_second`

Clock seconds value at epoch.

`trick_units(s)`

Definition at line 84 of file `time_ude.hh`.

Referenced by `set_epoch_std()`, `set_epoch_ude()`, and `verify_epoch()`.

**8.28.5.15** `bool jeod::TimeUDE::epoch_value_is_set_calendar` `[protected]`

Whether there is some calendar input that could set epoch.

`trick_units(-)`

Definition at line 150 of file `time_ude.hh`.

Referenced by `set_epoch_std()`, and `verify_epoch()`.

**8.28.5.16** `bool jeod::TimeUDE::epoch_value_is_set_clock` `[protected]`

Whether there is some clock input that could set epoch.

`trick_units(-)`

Definition at line 155 of file `time_ude.hh`.

Referenced by `set_epoch_ude()`, and `verify_epoch()`.

**8.28.5.17** `bool jeod::TimeUDE::epoch_value_is_set_number` `[protected]`

Whether there is some numerical input that could set epoch.

`trick_units(-)`

Definition at line 145 of file `time_ude.hh`.

Referenced by `set_epoch_dyn()`, `set_epoch_std()`, `set_epoch_ude()`, and `verify_epoch()`.

**8.28.5.18** `int jeod::TimeUDE::epoch_year`

Gregorian calendar year number at epoch.

`trick_units(-)`

Definition at line 64 of file `time_ude.hh`.

Referenced by `set_epoch_dyn()`, `set_epoch_std()`, and `verify_epoch()`.

**8.28.5.19** `TimeEnum::TimeFormat jeod::TimeUDE::initial_value_format`

Format for expressing the initial value of this type (calendar, julian, etc )

`trick_units(-)`

Definition at line 119 of file `time_ude.hh`.

Referenced by `initialize_initializer_time()`, and `set_initial_times()`.

**8.28.5.20** `bool jeod::TimeUDE::initializing_data_present` `[protected]`

Whether initializing data is present.

`trick_units(-)`

Definition at line 135 of file `time_ude.hh`.

Referenced by `initialize_from_parent()`, `initialize_initializer_time()`, `set_epoch_dyn()`, and `set_initial_times()`.

**8.28.5.21** `double jeod::TimeUDE::last_clock_update`

Simtime at the last time the clock was updated.

trick\_units(s)

Definition at line 109 of file time\_ude.hh.

#### 8.28.5.22 int jeod::TimeUDE::update\_index [protected]

The index of the time-type from which this one is updated.

trick\_units(-)

Definition at line 160 of file time\_ude.hh.

Referenced by add\_type\_initialize(), convert\_epoch\_to\_update(), initialize\_from\_parent(), initialize\_initializer\_time(), and verify\_update().

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

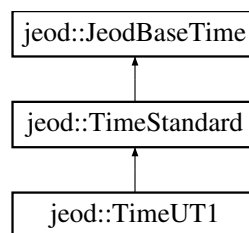
- [time\\_ude.hh](#)
- [time\\_ude.cc](#)

## 8.29 jeod::TimeUT1 Class Reference

Represents Universal Time.

```
#include <time_ut1.hh>
```

Inheritance diagram for jeod::TimeUT1:



### Public Member Functions

- [TimeUT1](#) ()  
*Construct a Time\_UT1.*
- [~TimeUT1](#) ()  
*Destroy a Time\_UT1.*
- double [get\\_days](#) ()  
*Accesses days.*

### Data Fields

- bool [true\\_ut1](#)  
*"False" for comparison with older versions of JEOD*

### Private Member Functions

- [TimeUT1](#) (const [TimeUT1](#) &)
- [TimeUT1](#) & [operator=](#) (const [TimeUT1](#) &)
- void [set\\_epoch](#) (void)  
*Sets the epoch for UT1 time.*



## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeUT1](#) ()

## Additional Inherited Members

### 8.29.1 Detailed Description

Represents Universal Time.

Definition at line 49 of file time\_ut1.hh.

### 8.29.2 Constructor & Destructor Documentation

#### 8.29.2.1 jeod::TimeUT1::TimeUT1 ( void )

Construct a Time\_UT1.

Definition at line 52 of file time\_ut1.cc.

References [jeod::JeodBaseTime::name](#), [set\\_epoch\(\)](#), and [true\\_ut1](#).

#### 8.29.2.2 jeod::TimeUT1::~~TimeUT1 ( void )

Destroy a Time\_UT1.

Definition at line 86 of file time\_ut1.cc.

#### 8.29.2.3 jeod::TimeUT1::TimeUT1 ( const TimeUT1 & ) [private]

### 8.29.3 Member Function Documentation

#### 8.29.3.1 double jeod::TimeUT1::get\_days ( void )

Accesses days.

#### Returns

days value

Units: d

Definition at line 76 of file time\_ut1.cc.

References [jeod::JeodBaseTime::days](#).

Referenced by [jeod::TimeConverter\\_UT1\\_GMST::convert\\_a\\_to\\_b\(\)](#).

#### 8.29.3.2 TimeUT1& jeod::TimeUT1::operator= ( const TimeUT1 & ) [private]

#### 8.29.3.3 void jeod::TimeUT1::set\_epoch ( void ) [private],[virtual]

Sets the epoch for UT1 time.

Implements [jeod::TimeStandard](#).

Definition at line 64 of file time\_ut1.cc.

References `jeod::TimeStandard::tjt_at_epoch`.

Referenced by `TimeUT1()`.

## 8.29.4 Friends And Related Function Documentation

8.29.4.1 `void init_attrjeod__TimeUT1 ( ) [friend]`

8.29.4.2 `friend class InputProcessor [friend]`

Definition at line 51 of file `time_ut1.hh`.

## 8.29.5 Field Documentation

8.29.5.1 `bool jeod::TimeUT1::true_ut1`

"False" for comparison with older versions of JEOD

`trick_units(-)`

Definition at line 58 of file `time_ut1.hh`.

Referenced by `jeod::TimeConverter_TAI_UT1::convert_a_to_b()`, `jeod::TimeConverter_TAI_UT1::convert_b_to_a()`, `jeod::TimeConverter_TAI_UT1::initialize_tai_to_ut1()`, `TimeUT1()`, and `jeod::TimeConverter_TAI_UT1::verify_table_lookup_ends()`.

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

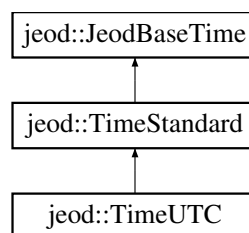
- [time\\_ut1.hh](#)
- [time\\_ut1.cc](#)

## 8.30 jeod::TimeUTC Class Reference

Represents Coordinated Universal Time.

```
#include <time_utc.hh>
```

Inheritance diagram for `jeod::TimeUTC`:



### Public Member Functions

- [TimeUTC \(\)](#)  
*Construct a Time\_UTC.*
- [~TimeUTC \(\)](#)  
*Destroy a Time\_UTC.*

## Data Fields

- bool [true\\_utc](#)  
*"False" for comparison with older versions of JEOD*

## Private Member Functions

- [TimeUTC](#) (const [TimeUTC](#) &)
- [TimeUTC](#) & [operator=](#) (const [TimeUTC](#) &)
- void [set\\_epoch](#) (void)  
*Sets the epoch for UTC time.*

## Friends

- class [InputProcessor](#)
- void [init\\_attrjeod\\_\\_TimeUTC](#) ()

## Additional Inherited Members

### 8.30.1 Detailed Description

Represents Coordinated Universal Time.

Definition at line 50 of file `time_utc.hh`.

### 8.30.2 Constructor & Destructor Documentation

#### 8.30.2.1 `jeod::TimeUTC::TimeUTC ( void )`

Construct a `Time_UTC`.

Definition at line 52 of file `time_utc.cc`.

References `jeod::JeodBaseTime::name`, `set_epoch()`, and `true_utc`.

#### 8.30.2.2 `jeod::TimeUTC::~~TimeUTC ( void )`

Destroy a `Time_UTC`.

Definition at line 76 of file `time_utc.cc`.

#### 8.30.2.3 `jeod::TimeUTC::TimeUTC ( const TimeUTC & )` `[private]`

### 8.30.3 Member Function Documentation

#### 8.30.3.1 `TimeUTC& jeod::TimeUTC::operator= ( const TimeUTC & )` `[private]`

#### 8.30.3.2 `void jeod::TimeUTC::set_epoch ( void )` `[private]`, `[virtual]`

Sets the epoch for UTC time.

Implements [jeod::TimeStandard](#).

Definition at line 65 of file `time_utc.cc`.

References `jeod::TimeStandard::tjt_at_epoch`.

Referenced by TimeUTC().

## 8.30.4 Friends And Related Function Documentation

8.30.4.1 `void init_attrjeod__TimeUTC ( )` [*friend*]

8.30.4.2 `friend class InputProcessor` [*friend*]

Definition at line 52 of file time\_utc.hh.

## 8.30.5 Field Documentation

8.30.5.1 `bool jeod::TimeUTC::true_utc`

"False" for comparison with older versions of JEOD

trick\_units(-)

Definition at line 59 of file time\_utc.hh.

Referenced by `jeod::TimeConverter_TAI_UTC::convert_a_to_b()`, `jeod::TimeConverter_TAI_UTC::convert_b_to_a()`, `jeod::TimeConverter_TAI_UTC::initialize_leap_second()`, `TimeUTC()`, and `jeod::TimeConverter_TAI_UTC::verify_table_lookup_ends()`.

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

- [time\\_utc.hh](#)
- [time\\_utc.cc](#)

## Chapter 9

# File Documentation

### 9.1 `class_declarations.hh` File Reference

Forward declaration of classes defined in [time.hh](#).

#### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.1.1 Detailed Description

Forward declaration of classes defined in [time.hh](#).

Definition in file [class\\_declarations.hh](#).

### 9.2 `tai_to_ut1.cc` File Reference

```
#include "environment/time/include/time_converter_tai_ut1.hh"  
#include "utils/memory/include/jeod_alloc.hh"  
#include "../include/tai_to_ut1.hh"
```

#### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### Macros

- `#define JEOD_FRIEND_CLASS TimeConverter_TAI_UT1_tai_to_ut1_default_data`

#### 9.2.1 Macro Definition Documentation

##### 9.2.1.1 `#define JEOD_FRIEND_CLASS TimeConverter_TAI_UT1_tai_to_ut1_default_data`

Definition at line 29 of file `tai_to_ut1.cc`.

## 9.3 tai\_to\_ut1.hh File Reference

### Data Structures

- class [jeod::TimeConverter\\_TAI\\_UT1\\_tai\\_to\\_ut1\\_default\\_data](#)

### Namespaces

- [jeod](#)

*Namespace jeod.*

## 9.4 tai\_to\_utc.cc File Reference

```
#include "environment/time/include/time_converter_tai_utc.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/tai_to_utc.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### Macros

- #define [JEOD\\_FRIEND\\_CLASS](#) TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data

### 9.4.1 Macro Definition Documentation

#### 9.4.1.1 #define JEOD\_FRIEND\_CLASS TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data

Definition at line 23 of file tai\_to\_utc.cc.

## 9.5 tai\_to\_utc.hh File Reference

### Data Structures

- class [jeod::TimeConverter\\_TAI\\_UTC\\_tai\\_to\\_utc\\_default\\_data](#)

### Namespaces

- [jeod](#)

*Namespace jeod.*

## 9.6 time.cc File Reference

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.6.1 Detailed Description

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

Definition in file [time.cc](#).

## 9.7 time.hh File Reference

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

```
#include <string>
#include <utility>
#include "utils/named_item/include/named_item.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
#include "time_links.hh"
```

### Data Structures

- class [jeod::JeodBaseTime](#)

*[JeodBaseTime](#) is an abstract class, containing the basic structure of all clocks that run in JEOD.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.7.1 Detailed Description

JeodBaseTime is an abstract class, containing the basic structure of all clocks that run in JEOD.

Definition in file [time.hh](#).

## 9.8 time\_\_add\_type\_update.cc File Reference

Define JeodBaseTime::add\_type\_update.

```
#include <cstdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.8.1 Detailed Description

Define JeodBaseTime::add\_type\_update. This is a final method that draws in a lot of the time model functionality. Making this method a separate compilation unit enables models that only need the vtable for class Time can get that from time.o without pulling in the whole of the time model.

Definition in file [time\\_\\_add\\_type\\_update.cc](#).

## 9.9 time\_converter.cc File Reference

An abstract class that defines the basic structure of all the methods used by the converter objects.

```
#include <cstdlib>
#include <stdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter.hh"
#include "../include/time.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.9.1 Detailed Description

An abstract class that defines the basic structure of all the methods used by the converter objects.

Definition in file [time\\_converter.cc](#).



## 9.10 time\_converter.hh File Reference

The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
```

### Data Structures

- class [jeod::TimeConverter](#)

*The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### Functions

- TimeConverter::Direction [jeod::operator|](#) (TimeConverter::Direction a, TimeConverter::Direction b)

*Bitwise or operator for combining multiple converter direction flags.*

#### 9.10.1 Detailed Description

The Time Converter is an abstract class that defines the basic structure of all the methods used by the converter objects; converters are the objects that specify the conversion algorithms between time-types.

Definition in file [time\\_converter.hh](#).

## 9.11 time\_converter\_dyn\_tai.cc File Reference

Converts between International Atomic Time and Dynamic Time.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_dyn_tai.hh"
#include "../include/time_dyn.hh"
#include "../include/time_tai.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.11.1 Detailed Description

Converts between International Atomic Time and Dynamic Time.

Definition in file [time\\_converter\\_dyn\\_tai.cc](#).

## 9.12 time\_converter\_dyn\_tai.hh File Reference

Define class TimeConverter\_Dyn\_TAI, which converts from simulation dynamic time to International Atomic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_Dyn\\_TAI](#)

Define class [TimeConverter\\_Dyn\\_TAI](#), which converts from simulation dynamic time to International Atomic Time.

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.12.1 Detailed Description

Define class TimeConverter\_Dyn\_TAI, which converts from simulation dynamic time to International Atomic Time.

Definition in file [time\\_converter\\_dyn\\_tai.hh](#).

## 9.13 time\_converter\_dyn\_tdb.cc File Reference

Converts between Dynamic Time and Barycentric Dynamic Time.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_dyn_tdb.hh"
#include "../include/time_dyn.hh"
#include "../include/time_tdb.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.13.1 Detailed Description

Converts between Dynamic Time and Barycentric Dynamic Time.

Definition in file [time\\_converter\\_dyn\\_tdb.cc](#).

## 9.14 time\_converter\_dyn\_tdb.hh File Reference

Define class TimeConverter\_Dyn\_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_Dyn\\_TDB](#)

Define class [TimeConverter\\_Dyn\\_TDB](#), which converts from simulation dynamic time to Barycentric Dynamic Time.

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.14.1 Detailed Description

Define class TimeConverter\_Dyn\_TDB, which converts from simulation dynamic time to Barycentric Dynamic Time.

Definition in file [time\\_converter\\_dyn\\_tdb.hh](#).

## 9.15 time\_converter\_dyn\_ude.cc File Reference

Converts between Dynamic Time and a time with User-Defined-Epoch.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_dyn_ude.hh"
#include "../include/time_dyn.hh"
#include "../include/time_ude.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.15.1 Detailed Description

Converts between Dynamic Time and a time with User-Defined-Epoch.

Definition in file [time\\_converter\\_dyn\\_ude.cc](#).

## 9.16 time\_converter\_dyn\_ude.hh File Reference

Define class TimeConverter\_Dyn\_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_Dyn\\_UDE](#)

Define class [TimeConverter\\_Dyn\\_UDE](#), which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time.

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.16.1 Detailed Description

Define class TimeConverter\_Dyn\_UDE, which converts from simulation dynamic time to any specific instance of the generic User-Defined-Epoch Time. There can be multiple instances of this class.

Definition in file [time\\_converter\\_dyn\\_ude.hh](#).

## 9.17 time\_converter\_std\_ude.cc File Reference

Define member functions for class TimeConverter\_STD\_UDE.

```
#include <cmath>
#include <cstdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_std_ude.hh"
#include "../include/time_standard.hh"
#include "../include/time_ude.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.17.1 Detailed Description

Define member functions for class TimeConverter\_STD\_UDE.

Definition in file [time\\_converter\\_std\\_ude.cc](#).

## 9.18 time\_converter\_std\_ude.hh File Reference

Define class TimeConverter\_STD\_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_STD\\_UDE](#)

Define class [TimeConverter\\_STD\\_UDE](#), which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time.

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.18.1 Detailed Description

Define class TimeConverter\_STD\_UDE, which converts from any specific example of the generic Standard Time to any specific example of the generic User-Defined-Epoch Time. There can be multiple such instances of this class.

Definition in file [time\\_converter\\_std\\_ude.hh](#).

## 9.19 time\_converter\_tai\_gps.cc File Reference

Converts between International Atomic Time and the clock associated with the Global Positioning System.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_gps.hh"
#include "../include/time_tai.hh"
#include "../include/time_gps.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.19.1 Detailed Description

Converts between International Atomic Time and the clock associated with the Global Positioning System.

Definition in file [time\\_converter\\_tai\\_gps.cc](#).

## 9.20 time\_converter\_tai\_gps.hh File Reference

Define class `TimeConverter_TAI_GPS`, which converts between International Atomic Time and the clock associated with the Global Positioning System.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_TAI\\_GPS](#)

Define class *TimeConverter\_TAI\_GPS*, which converts between International Atomic Time and the clock associated with the Global Positioning System.

### Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.20.1 Detailed Description

Define class `TimeConverter_TAI_GPS`, which converts between International Atomic Time and the clock associated with the Global Positioning System.

Definition in file [time\\_converter\\_tai\\_gps.hh](#).

## 9.21 time\_converter\_tai\_tdb.cc File Reference

Converts from International Atomic Time to Barycentric Dynamic Time.

```
#include <cmath>
#include <cstdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_tdb.hh"
#include "../include/time_tai.hh"
#include "../include/time_tdb.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.21.1 Detailed Description

Converts from International Atomic Time to Barycentric Dynamic Time.

Definition in file [time\\_converter\\_tai\\_tdb.cc](#).

## 9.22 time\_converter\_tai\_tdb.hh File Reference

Define class TimeConverter\_TAI\_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_TAI\\_TDB](#)

Define class [TimeConverter\\_TAI\\_TDB](#), which converts from International Atomic Time to Barycentric Dynamic Time.

### Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.22.1 Detailed Description

Define class TimeConverter\_TAI\_TDB, which converts from International Atomic Time to Barycentric Dynamic Time.

Definition in file [time\\_converter\\_tai\\_tdb.hh](#).

## 9.23 time\_converter\_tai\_tt.cc File Reference

Converts between International Atomic Time and Terrestrial Time.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_tt.hh"
#include "../include/time_tai.hh"
#include "../include/time_tt.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

Namespace *jeod*.

### 9.23.1 Detailed Description

Converts between International Atomic Time and Terrestrial Time.

Definition in file [time\\_converter\\_tai\\_tt.cc](#).

## 9.24 time\_converter\_tai\_tt.hh File Reference

Converts between International Atomic Time and Terrestrial Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_TAI\\_TT](#)  
*Converts between International Atomic Time and Terrestrial Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.24.1 Detailed Description

Converts between International Atomic Time and Terrestrial Time.

Definition in file [time\\_converter\\_tai\\_tt.hh](#).

## 9.25 time\_converter\_tai\_ut1.cc File Reference

Converts between International Atomic Time and Universal Time.

```
#include <cmath>
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_ut1.hh"
#include "../include/time_tai.hh"
#include "../include/time_ut1.hh"
#include "../include/time_manager.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*



### 9.25.1 Detailed Description

Converts between International Atomic Time and Universal Time.

Definition in file [time\\_converter\\_tai\\_ut1.cc](#).

## 9.26 time\_converter\_tai\_ut1.hh File Reference

Define class TimeConverter\_TAI\_UT1, which converts between International Atomic Time and Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_TAI\\_UT1](#)

Define class [TimeConverter\\_TAI\\_UT1](#), which converts between International Atomic Time and Universal Time.

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.26.1 Detailed Description

Define class TimeConverter\_TAI\_UT1, which converts between International Atomic Time and Universal Time.

Definition in file [time\\_converter\\_tai\\_ut1.hh](#).

## 9.27 time\_converter\_tai\_utc.cc File Reference

Converts between International Atomic Time and Coordinated Universal Time.

```
#include <cmath>
#include <cstddef>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_tai_utc.hh"
#include "../include/time_tai.hh"
#include "../include/time_utc.hh"
#include "../include/time_manager.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

Namespace [jeod](#).

### 9.27.1 Detailed Description

Converts between International Atomic Time and Coordinated Universal Time.

Definition in file [time\\_converter\\_tai\\_utc.cc](#).

## 9.28 time\_converter\_tai\_utc.hh File Reference

Converts between International Atomic Time and Coordinated Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_TAI\\_UTC](#)

*Converts between International Atomic Time and Coordinated Universal Time.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.28.1 Detailed Description

Converts between International Atomic Time and Coordinated Universal Time.

Definition in file [time\\_converter\\_tai\\_utc.hh](#).

## 9.29 time\_converter\_ut1\_gmst.cc File Reference

Define member functions for class TimeConverter\_UT1\_GMST.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_converter_ut1_gmst.hh"
#include "../include/time_ut1.hh"
#include "../include/time_gmst.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.29.1 Detailed Description

Define member functions for class TimeConverter\_UT1\_GMST.

Definition in file [time\\_converter\\_ut1\\_gmst.cc](#).

## 9.30 time\_converter\_ut1\_gmst.hh File Reference

Converts between Universal Time and Greenwich Mean Sidereal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_converter.hh"
```

### Data Structures

- class [jeod::TimeConverter\\_UT1\\_GMST](#)

*Converts between Universal Time and Greenwich Mean Sidereal Time.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.30.1 Detailed Description

Converts between Universal Time and Greenwich Mean Sidereal Time.

Definition in file [time\\_converter\\_ut1\\_gmst.hh](#).

## 9.31 time\_dyn.cc File Reference

Define member functions for Dynamic Time.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_dyn.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_standard.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.31.1 Detailed Description

Define member functions for Dynamic Time.

Definition in file [time\\_dyn.cc](#).

## 9.32 time\_dyn.hh File Reference

Represents the Dynamic Time in the simulation.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time.hh"
```

### Data Structures

- class [jeod::TimeDyn](#)  
*Represents the Dynamic Time in the simulation.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.32.1 Detailed Description

Represents the Dynamic Time in the simulation.

Definition in file [time\\_dyn.hh](#).

## 9.33 time\_enum.hh File Reference

Contains an enumeration of the formats in which time can be represented.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

### Data Structures

- class [jeod::TimeEnum](#)  
*Contains an enumeration of the formats in which time can be represented.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.33.1 Detailed Description

Contains an enumeration of the formats in which time can be represented.

Definition in file [time\\_enum.hh](#).

## 9.34 time\_gmst.cc File Reference

Define member functions for Greenwich Mean Sidereal Time.

```
#include <cstdint>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/time_gmst.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.34.1 Detailed Description

Define member functions for Greenwich Mean Sidereal Time.

Definition in file [time\\_gmst.cc](#).

## 9.35 time\_gmst.hh File Reference

To represent the clock known as Greenwich Mean Sidereal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

### Data Structures

- class [jeod::TimeGMST](#)

*To represent the clock known as Greenwich Mean Sidereal Time.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

#### 9.35.1 Detailed Description

To represent the clock known as Greenwich Mean Sidereal Time.

Definition in file [time\\_gmst.hh](#).

## 9.36 time\_gps.cc File Reference

Define member functions for the clock associated with the Global Positioning System.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_gps.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.36.1 Detailed Description

Define member functions for the clock associated with the Global Positioning System.

Definition in file [time\\_gps.cc](#).

## 9.37 time\_gps.hh File Reference

To represent the time associated with the Global Positioning System.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

## Data Structures

- class [jeod::TimeGPS](#)

*To represent the time associated with the Global Positioning System.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.37.1 Detailed Description

To represent the time associated with the Global Positioning System.

Definition in file [time\\_gps.hh](#).

## 9.38 time\_links.hh File Reference

Define the class TimeLinks, which defines the hierarchy of JEOD time conversions.

```
#include "class_declarations.hh"
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/ref_frames/include/tree_links.hh"
```

## Data Structures

- class [jeod::TimeLinks](#)

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.38.1 Detailed Description

Define the class TimeLinks, which defines the hierarchy of JEOD time conversions.

Definition in file [time\\_links.hh](#).

## 9.39 time\_manager.cc File Reference

Define member functions for class TimeManager.

```
#include <algorithm>
#include <cstdint>
#include <cstring>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_standard.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.39.1 Detailed Description

Define member functions for class TimeManager.

Definition in file [time\\_manager.cc](#).

## 9.40 time\_manager.hh File Reference

To manage the various time representations and the converters between them throughout the simulation.

```
#include <string>
#include <vector>
#include "utils/sim_interface/include/jeod_class.hh"
#include "utils/integration/include/jeod_integration_time.hh"
#include "time_dyn.hh"
```

## Data Structures

- class [jeod::TimeManager](#)

*To manage the various time representations and the converters between them throughout the simulation.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.40.1 Detailed Description

To manage the various time representations and the converters between them throughout the simulation.

Definition in file [time\\_manager.hh](#).

## 9.41 time\_manager\_\_initialize.cc File Reference

Define TimeManager::initialize.

```
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/named_item/include/named_item.hh"
#include "../include/time.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.41.1 Detailed Description

Define TimeManager::initialize. This method allocates resources and invokes TimeManagerInit functionality. This method needs to be defined as a separate compilation unit.

Definition in file [time\\_manager\\_\\_initialize.cc](#).

## 9.42 time\_manager\_init.cc File Reference

Define member functions for the Time Manager Initialization.



```
#include <cstdint>
#include <typeinfo>
#include <algorithm>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_manager.hh"
#include "../include/time_ude.hh"
#include "../include/time_converter_tai_utc.hh"
#include "../include/time_converter_tai_utl.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.42.1 Detailed Description

Define member functions for the Time Manager Initialization.

Definition in file [time\\_manager\\_init.cc](#).

## 9.43 time\_manager\_init.hh File Reference

To initialize the Time Manager.

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_enum.hh"
```

## Data Structures

- class [jeod::TimeManagerInit](#)

*To initialize the Time Manager.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.43.1 Detailed Description

To initialize the Time Manager.

Definition in file [time\\_manager\\_init.hh](#).

## 9.44 time\_messages.cc File Reference

Implement the class TimeMessages.

```
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### Macros

- #define [PATH](#) "environment/time/"

#### 9.44.1 Detailed Description

Implement the class TimeMessages.

Definition in file [time\\_messages.cc](#).

## 9.45 time\_messages.hh File Reference

Define the class TimeMessages, the class that specifies the message IDs used in the Time model.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

### Data Structures

- class [jeod::TimeMessages](#)  
*Specify the message IDs used in the Time model.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.45.1 Detailed Description

Define the class TimeMessages, the class that specifies the message IDs used in the Time model.

Definition in file [time\\_messages.hh](#).

## 9.46 time\_met.cc File Reference

Define member functions for Mission Elapsed Time.

```
#include <cstdint>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_met.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.46.1 Detailed Description

Define member functions for Mission Elapsed Time.

Definition in file [time\\_met.cc](#).

## 9.47 time\_met.hh File Reference

A type of UDE time that allows for deliberate holds, or pauses.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_ude.hh"
```

## Data Structures

- class [jeod::TimeMET](#)

*A type of UDE time that allows for deliberate holds, or pauses.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.47.1 Detailed Description

A type of UDE time that allows for deliberate holds, or pauses.

Definition in file [time\\_met.hh](#).

## 9.48 time\_standard.cc File Reference

An abstract class, this defines the basic structure of member functions for all Standard Times.

```
#include <cmath>
#include "utils/message/include/message_handler.hh"
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_standard.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_messages.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.48.1 Detailed Description

An abstract class, this defines the basic structure of member functions for all Standard Times.

Definition in file [time\\_standard.cc](#).

## 9.49 time\_standard.hh File Reference

A class that serves as the base for all time representations that are well defined outside the simulation.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time.hh"
```

## Data Structures

- class [jeod::TimeStandard](#)

*A class that serves as the base for all time representations that are well defined outside the simulation.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.49.1 Detailed Description

A class that serves as the base for all time representations that are well defined outside the simulation.

Definition in file [time\\_standard.hh](#).

## 9.50 time\_tai.cc File Reference

Define member functions for International Atomic Time.

```
#include <cstdint>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_tai.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.50.1 Detailed Description

Define member functions for International Atomic Time.

Definition in file [time\\_tai.cc](#).

## 9.51 time\_tai.hh File Reference

Represents International Atomic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

## Data Structures

- class [jeod::TimeTAI](#)

*Represents International Atomic Time.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.51.1 Detailed Description

Represents International Atomic Time.

Definition in file [time\\_tai.hh](#).

## 9.52 time\_tdb.cc File Reference

Define member functions Barycentric Dynamic Time.

```
#include <cstdint>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_tdb.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.52.1 Detailed Description

Define member functions Barycentric Dynamic Time.

Definition in file [time\\_tdb.cc](#).

## 9.53 time\_tdb.hh File Reference

Represents Barycentric Dynamic Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

## Data Structures

- class [jeod::TimeTDB](#)

*Represents Barycentric Dynamic Time.*

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.53.1 Detailed Description

Represents Barycentric Dynamic Time.

Definition in file [time\\_tdb.hh](#).

## 9.54 time\_tt.cc File Reference

Define member functions for Terrestrial Time.

```
#include <cstdint>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_tt.hh"
```

## Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.54.1 Detailed Description

Define member functions for Terrestrial Time.

Definition in file [time\\_tt.cc](#).

## 9.55 time\_tt.hh File Reference

Represents Terrestrial Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

### Data Structures

- class [jeod::TimeTT](#)  
*Represents Terrestrial Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.55.1 Detailed Description

Represents Terrestrial Time.

Definition in file [time\\_tt.hh](#).

## 9.56 time\_ude.cc File Reference

Define member functions for those times with a User-Defined-Epoch.

```
#include <cmath>
#include <cstdlib>
#include "utils/message/include/message_handler.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_ude.hh"
#include "../include/time_dyn.hh"
#include "../include/time_converter.hh"
#include "../include/time_manager_init.hh"
#include "../include/time_manager.hh"
#include "../include/time_standard.hh"
#include "../include/time_messages.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.56.1 Detailed Description

Define member functions for those times with a User-Defined-Epoch.

Definition in file [time\\_ude.cc](#).

## 9.57 time\_ude.hh File Reference

Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.

```
#include <string>
#include "utils/sim_interface/include/jeod_class.hh"
#include "time.hh"
#include "time_enum.hh"
```

### Data Structures

- class [jeod::TimeUDE](#)

*Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.*

### Namespaces

- [jeod](#)

*Namespace jeod.*

### 9.57.1 Detailed Description

Represents all instances of times with a user-defined epoch, accepting that Mission Elapsed Time requires some further definition.

Definition in file [time\\_ude.hh](#).

## 9.58 time\_ut1.cc File Reference

Define member functions for Universal Time.

```
#include <cstdlib>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_ut1.hh"
```

### Namespaces

- [jeod](#)

*Namespace jeod.*



### 9.58.1 Detailed Description

Define member functions for Universal Time.

Definition in file [time\\_ut1.cc](#).

## 9.59 time\_ut1.hh File Reference

Represents Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include "time_standard.hh"
```

### Data Structures

- class [jeod::TimeUT1](#)  
*Represents Universal Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.59.1 Detailed Description

Represents Universal Time.

Definition in file [time\\_ut1.hh](#).

## 9.60 time\_utc.cc File Reference

Define member functions for Coordinated Universal Time.

```
#include <cstdint>
#include "utils/named_item/include/named_item.hh"
#include "utils/memory/include/jeod_alloc.hh"
#include "../include/time_utc.hh"
```

### Namespaces

- [jeod](#)  
*Namespace jeod.*

### 9.60.1 Detailed Description

Define member functions for Coordinated Universal Time.

Definition in file [time\\_utc.cc](#).

## 9.61 time\_utc.hh File Reference

Represents Coordinated Universal Time.

```
#include "utils/sim_interface/include/jeod_class.hh"  
#include "time_standard.hh"
```

### Data Structures

- class [jeod::TimeUTC](#)  
*Represents Coordinated Universal Time.*

### Namespaces

- [jeod](#)  
*Namespace jeod.*

#### 9.61.1 Detailed Description

Represents Coordinated Universal Time.

Definition in file [time\\_utc.hh](#).

# Index

- ~JeodBaseTime
  - jeod::JeodBaseTime, [23](#)
- ~TimeConverter
  - jeod::TimeConverter, [33](#)
- ~TimeConverter\_Dyn\_TAI
  - jeod::TimeConverter\_Dyn\_TAI, [39](#)
- ~TimeConverter\_Dyn\_TDB
  - jeod::TimeConverter\_Dyn\_TDB, [41](#)
- ~TimeConverter\_Dyn\_UDE
  - jeod::TimeConverter\_Dyn\_UDE, [44](#)
- ~TimeConverter\_STD\_UDE
  - jeod::TimeConverter\_STD\_UDE, [47](#)
- ~TimeConverter\_TAI\_GPS
  - jeod::TimeConverter\_TAI\_GPS, [50](#)
- ~TimeConverter\_TAI\_TDB
  - jeod::TimeConverter\_TAI\_TDB, [53](#)
- ~TimeConverter\_TAI\_TT
  - jeod::TimeConverter\_TAI\_TT, [57](#)
- ~TimeConverter\_TAI\_UT1
  - jeod::TimeConverter\_TAI\_UT1, [60](#)
- ~TimeConverter\_TAI\_UTC
  - jeod::TimeConverter\_TAI\_UTC, [67](#)
- ~TimeConverter\_UT1\_GMST
  - jeod::TimeConverter\_UT1\_GMST, [73](#)
- ~TimeDyn
  - jeod::TimeDyn, [75](#)
- ~TimeGMST
  - jeod::TimeGMST, [79](#)
- ~TimeGPS
  - jeod::TimeGPS, [82](#)
- ~TimeLinks
  - jeod::TimeLinks, [86](#)
- ~TimeMET
  - jeod::TimeMET, [109](#)
- ~TimeManager
  - jeod::TimeManager, [88](#)
- ~TimeManagerInit
  - jeod::TimeManagerInit, [97](#)
- ~TimeStandard
  - jeod::TimeStandard, [112](#)
- ~TimeTAI
  - jeod::TimeTAI, [121](#)
- ~TimeTDB
  - jeod::TimeTDB, [123](#)
- ~TimeTT
  - jeod::TimeTT, [124](#)
- ~TimeUDE
  - jeod::TimeUDE, [128](#)
- ~TimeUT1
  - jeod::TimeUT1, [139](#)
- ~TimeUTC
  - jeod::TimeUTC, [141](#)
- A\_TO\_B
  - jeod::TimeConverter, [33](#)
- A\_TO\_B\_INIT
  - jeod::TimeConverter, [33](#)
- A\_TO\_B\_UPDATE
  - jeod::TimeConverter, [33](#)
- ANY\_DIRECTION
  - jeod::TimeConverter, [33](#)
- a\_name
  - jeod::TimeConverter, [36](#)
- a\_to\_b\_offset
  - jeod::TimeConverter, [37](#)
  - jeod::TimeConverter\_TAI\_TDB, [54](#)
- a\_to\_b\_offset\_epoch
  - jeod::TimeConverter\_TAI\_TDB, [54](#)
- add\_parent
  - jeod::JeodBaseTime, [24](#)
- add\_type\_initialize
  - jeod::JeodBaseTime, [24](#)
  - jeod::TimeStandard, [113](#)
  - jeod::TimeUDE, [128](#)
- add\_type\_update
  - jeod::JeodBaseTime, [24](#)
- B\_TO\_A
  - jeod::TimeConverter, [33](#)
- B\_TO\_A\_INIT
  - jeod::TimeConverter, [33](#)
- B\_TO\_A\_UPDATE
  - jeod::TimeConverter, [33](#)
- b\_name
  - jeod::TimeConverter, [37](#)
- calculate\_calendar\_values
  - jeod::TimeGMST, [79](#)
  - jeod::TimeGPS, [82](#)
  - jeod::TimeStandard, [113](#)
- calendar
  - jeod::TimeEnum, [78](#)
- calendar\_day
  - jeod::TimeStandard, [117](#)
- calendar\_hour
  - jeod::TimeStandard, [117](#)
- calendar\_minute
  - jeod::TimeStandard, [118](#)
- calendar\_month

- jeod::TimeStandard, 118
- calendar\_second
  - jeod::TimeStandard, 118
- calendar\_update
  - jeod::TimeStandard, 113
- calendar\_year
  - jeod::TimeStandard, 118
- can\_convert
  - jeod::TimeConverter, 34
- class\_declarations.hh, 143
- clock
  - jeod::TimeEnum, 78
- clock\_day
  - jeod::TimeUDE, 134
- clock\_hour
  - jeod::TimeUDE, 134
- clock\_minute
  - jeod::TimeUDE, 135
- clock\_resolution
  - jeod::JeodBaseTime, 28
- clock\_second
  - jeod::TimeUDE, 135
- clock\_update
  - jeod::TimeUDE, 129
- convert\_a\_to\_b
  - jeod::TimeConverter, 34
  - jeod::TimeConverter\_Dyn\_TAI, 39
  - jeod::TimeConverter\_Dyn\_TDB, 42
  - jeod::TimeConverter\_Dyn\_UDE, 44
  - jeod::TimeConverter\_STD\_UDE, 47
  - jeod::TimeConverter\_TAI\_GPS, 50
  - jeod::TimeConverter\_TAI\_TDB, 53
  - jeod::TimeConverter\_TAI\_TT, 57
  - jeod::TimeConverter\_TAI\_UT1, 61
  - jeod::TimeConverter\_TAI\_UTC, 67
  - jeod::TimeConverter\_UT1\_GMST, 73
- convert\_b\_to\_a
  - jeod::TimeConverter, 34
  - jeod::TimeConverter\_STD\_UDE, 47
  - jeod::TimeConverter\_TAI\_GPS, 50
  - jeod::TimeConverter\_TAI\_TDB, 53
  - jeod::TimeConverter\_TAI\_TT, 57
  - jeod::TimeConverter\_TAI\_UT1, 61
  - jeod::TimeConverter\_TAI\_UTC, 68
- convert\_epoch\_to\_update
  - jeod::TimeUDE, 129
- convert\_from\_calendar
  - jeod::TimeGPS, 82
  - jeod::TimeStandard, 114
- converter\_ptrs\_index
  - jeod::TimeManagerInit, 102
- converter\_vector
  - jeod::TimeManager, 94
- create\_init\_tree
  - jeod::TimeManagerInit, 97
- create\_update\_tree
  - jeod::TimeManagerInit, 98
- day\_of\_week
  - jeod::TimeGPS, 84
- days
  - jeod::JeodBaseTime, 28
- days\_since\_epoch
  - jeod::TimeEnum, 78
- default\_path\_size
  - jeod::TimeLinks, 86
- Direction
  - jeod::TimeConverter, 33
- duplicate\_methods
  - jeod::TimeMessages, 105
- dyn\_ptr
  - jeod::TimeConverter\_Dyn\_TAI, 40
  - jeod::TimeConverter\_Dyn\_TDB, 42
  - jeod::TimeConverter\_Dyn\_UDE, 45
- dyn\_time
  - jeod::TimeManager, 94
- dyn\_time\_index
  - jeod::TimeManagerInit, 102
- Environment, 14
- epoch\_data\_present
  - jeod::TimeUDE, 135
- epoch\_day
  - jeod::TimeUDE, 135
- epoch\_defined\_in\_name
  - jeod::TimeUDE, 135
- epoch\_format
  - jeod::TimeUDE, 135
- epoch\_hour
  - jeod::TimeUDE, 136
- epoch\_index
  - jeod::TimeUDE, 136
- epoch\_initializing\_value
  - jeod::TimeUDE, 136
- epoch\_minute
  - jeod::TimeUDE, 136
- epoch\_month
  - jeod::TimeUDE, 136
- epoch\_second
  - jeod::TimeUDE, 136
- epoch\_value\_is\_set\_calendar
  - jeod::TimeUDE, 136
- epoch\_value\_is\_set\_clock
  - jeod::TimeUDE, 137
- epoch\_value\_is\_set\_number
  - jeod::TimeUDE, 137
- epoch\_year
  - jeod::TimeUDE, 137
- extension\_error
  - jeod::TimeMessages, 106
- failed\_null\_test
  - jeod::TimeConverter\_STD\_UDE, 48
- get\_a\_to\_b\_offset
  - jeod::TimeConverter, 34
- get\_conv\_dir\_init
  - jeod::TimeManagerInit, 98

- get\_conv\_dir\_upd
  - jeod::TimeManagerInit, 98
- get\_conv\_ptr\_index
  - jeod::TimeManagerInit, 99
- get\_converter\_ptr
  - jeod::TimeManager, 89
- get\_days
  - jeod::TimeUT1, 139
- get\_index
  - jeod::JeodBaseTime, 25
- get\_jeod\_integration\_time
  - jeod::TimeManager, 89
- get\_status
  - jeod::TimeManagerInit, 99
- get\_time\_change\_flag
  - jeod::TimeManager, 89
- get\_time\_ptr
  - jeod::TimeManager, 89, 90
- get\_time\_scale\_factor
  - jeod::TimeManager, 90
- get\_timestamp\_time
  - jeod::TimeManager, 90
- gmst\_ptr
  - jeod::TimeConverter\_UT1\_GMST, 74
- gps\_ptr
  - jeod::TimeConverter\_TAI\_GPS, 51
- gradient
  - jeod::TimeConverter\_TAI\_UT1, 62
- hold
  - jeod::TimeMET, 110
- incomplete\_setup\_error
  - jeod::TimeMessages, 106
- increment\_status
  - jeod::TimeManagerInit, 99
- index
  - jeod::JeodBaseTime, 28
  - jeod::TimeConverter\_TAI\_UT1, 62
  - jeod::TimeConverter\_TAI\_UTC, 69
- init\_attrjeod\_\_JeodBaseTime
  - jeod::JeodBaseTime, 27
- init\_attrjeod\_\_TimeConverter
  - jeod::TimeConverter, 36
- init\_attrjeod\_\_TimeDyn
  - jeod::TimeDyn, 77
- init\_attrjeod\_\_TimeGMST
  - jeod::TimeGMST, 80
- init\_attrjeod\_\_TimeGPS
  - jeod::TimeGPS, 84
- init\_attrjeod\_\_TimeLinks
  - jeod::TimeLinks, 86
- init\_attrjeod\_\_TimeMET
  - jeod::TimeMET, 110
- init\_attrjeod\_\_TimeManager
  - jeod::TimeManager, 94
- init\_attrjeod\_\_TimeManagerInit
  - jeod::TimeManagerInit, 102
- init\_attrjeod\_\_TimeMessages
  - jeod::TimeMessages, 105
- init\_attrjeod\_\_TimeStandard
  - jeod::TimeStandard, 117
- init\_attrjeod\_\_TimeTAI
  - jeod::TimeTAI, 122
- init\_attrjeod\_\_TimeTDB
  - jeod::TimeTDB, 123
- init\_attrjeod\_\_TimeTT
  - jeod::TimeTT, 125
- init\_attrjeod\_\_TimeUDE
  - jeod::TimeUDE, 134
- init\_attrjeod\_\_TimeUT1
  - jeod::TimeUT1, 140
- init\_attrjeod\_\_TimeUTC
  - jeod::TimeUTC, 142
- init\_converter\_dir\_table
  - jeod::TimeManagerInit, 102
- initial\_value
  - jeod::JeodBaseTime, 28
- initial\_value\_format
  - jeod::TimeUDE, 137
- initialization\_error
  - jeod::TimeMessages, 106
- initialize
  - jeod::TimeConverter, 35
  - jeod::TimeConverter\_Dyn\_TAI, 39
  - jeod::TimeConverter\_Dyn\_TDB, 42
  - jeod::TimeConverter\_Dyn\_UDE, 44
  - jeod::TimeConverter\_STD\_UDE, 48
  - jeod::TimeConverter\_TAI\_GPS, 51
  - jeod::TimeConverter\_TAI\_TDB, 54
  - jeod::TimeConverter\_TAI\_TT, 58
  - jeod::TimeConverter\_TAI\_UT1, 61
  - jeod::TimeConverter\_TAI\_UT1\_tai\_to\_ut1\_default\_data, 65
  - jeod::TimeConverter\_TAI\_UTC, 68
  - jeod::TimeConverter\_TAI\_UTC\_tai\_to\_utc\_default\_data, 71
  - jeod::TimeConverter\_UT1\_GMST, 73
  - jeod::TimeManager, 90
  - jeod::TimeManagerInit, 100
- initialize\_from\_name
  - jeod::JeodBaseTime, 28
- initialize\_from\_parent
  - jeod::JeodBaseTime, 25
  - jeod::TimeStandard, 114
  - jeod::TimeUDE, 129
- initialize\_initializer\_time
  - jeod::JeodBaseTime, 25
  - jeod::TimeDyn, 76
  - jeod::TimeStandard, 115
  - jeod::TimeUDE, 130
- initialize\_leap\_second
  - jeod::TimeConverter\_TAI\_UTC, 68
- initialize\_manager
  - jeod::TimeManagerInit, 100
- initialize\_tai\_to\_ut1
  - jeod::TimeConverter\_TAI\_UT1, 61

- initialize\_time\_types
  - jeod::TimeManagerInit, 100
- initialized
  - jeod::JeodBaseTime, 29
  - jeod::TimeConverter, 37
- initializer
  - jeod::TimeManagerInit, 103
- initializer\_index
  - jeod::TimeManagerInit, 103
- initializing\_data\_present
  - jeod::TimeUDE, 137
- initializing\_value
  - jeod::JeodBaseTime, 29
- InputProcessor
  - jeod::JeodBaseTime, 27
  - jeod::TimeConverter, 36
  - jeod::TimeConverter\_Dyn\_TAI, 40
  - jeod::TimeConverter\_Dyn\_TDB, 42
  - jeod::TimeConverter\_Dyn\_UDE, 45
  - jeod::TimeConverter\_STD\_UDE, 48
  - jeod::TimeConverter\_TAI\_GPS, 51
  - jeod::TimeConverter\_TAI\_TDB, 54
  - jeod::TimeConverter\_TAI\_TT, 58
  - jeod::TimeConverter\_TAI\_UT1, 62
  - jeod::TimeConverter\_TAI\_UTC, 69
  - jeod::TimeConverter\_UT1\_GMST, 74
  - jeod::TimeDyn, 77
  - jeod::TimeGMST, 80
  - jeod::TimeGPS, 84
  - jeod::TimeLinks, 86
  - jeod::TimeManager, 94
  - jeod::TimeManagerInit, 102
  - jeod::TimeMessages, 105
  - jeod::TimeMET, 110
  - jeod::TimeStandard, 117
  - jeod::TimeTAI, 122
  - jeod::TimeTDB, 123
  - jeod::TimeTT, 125
  - jeod::TimeUDE, 134
  - jeod::TimeUT1, 140
  - jeod::TimeUTC, 142
- invalid\_data\_error
  - jeod::TimeMessages, 107
- invalid\_node
  - jeod::TimeMessages, 107
- invalid\_setup\_error
  - jeod::TimeMessages, 107
- is\_initialized
  - jeod::JeodBaseTime, 25
  - jeod::TimeConverter, 35
- JEOD\_FRIEND\_CLASS
  - tai\_to\_ut1.cc, 143
  - tai\_to\_utc.cc, 144
- jeod, 19
- jeod::TimeConverter
  - A\_TO\_B, 33
  - A\_TO\_B\_INIT, 33
  - A\_TO\_B\_UPDATE, 33
  - ANY\_DIRECTION, 33
  - B\_TO\_A, 33
  - B\_TO\_A\_INIT, 33
  - B\_TO\_A\_UPDATE, 33
  - NO\_DIRECTION, 33
- jeod::TimeEnum
  - calendar, 78
  - clock, 78
  - days\_since\_epoch, 78
  - Julian, 78
  - julian, 78
  - modified\_julian, 78
  - seconds\_since\_epoch, 78
  - truncated\_julian, 78
  - undefined, 78
- jeod::JeodBaseTime, 21
  - ~JeodBaseTime, 23
  - add\_parent, 24
  - add\_type\_initialize, 24
  - add\_type\_update, 24
  - clock\_resolution, 28
  - days, 28
  - get\_index, 25
  - index, 28
  - init\_attrjeod\_\_JeodBaseTime, 27
  - initial\_value, 28
  - initialize\_from\_name, 28
  - initialize\_from\_parent, 25
  - initialize\_initializer\_time, 25
  - initialized, 29
  - initializing\_value, 29
  - InputProcessor, 27
  - is\_initialized, 25
  - JeodBaseTime, 23
  - links, 29
  - must\_be\_singleton, 26
  - name, 29
  - operator=, 26
  - override\_initialized, 26
  - seconds, 29
  - set\_index, 26
  - set\_name, 26
  - set\_time\_by\_days, 26
  - set\_time\_by\_seconds, 27
  - time\_manager, 30
  - TimeConverter, 27
  - TimeManagerInit, 28
  - update, 27
  - update\_converter\_direction, 30
  - update\_converter\_ptr, 30
  - update\_from\_name, 30
- jeod::TimeConverter, 31
  - ~TimeConverter, 33
  - a\_name, 36
  - a\_to\_b\_offset, 37
  - b\_name, 37
  - can\_convert, 34
  - convert\_a\_to\_b, 34

- convert\_b\_to\_a, 34
- Direction, 33
- get\_a\_to\_b\_offset, 34
- init\_attrjeod\_\_TimeConverter, 36
- initialize, 35
- initialized, 37
- InputProcessor, 36
- is\_initialized, 35
- JeodBaseTime, 36
- operator=, 35
- override\_initialized, 35
- reset\_a\_to\_b\_offset, 35
- TimeConverter, 33
- valid\_directions, 37
- verify\_setup, 35
- verify\_table\_lookup\_ends, 36
- jeod::TimeConverter\_Dyn\_TAI, 38
  - convert\_a\_to\_b, 39
  - dyn\_ptr, 40
  - initialize, 39
  - InputProcessor, 40
  - operator=, 40
  - tai\_ptr, 40
  - TimeConverter\_Dyn\_TAI, 39
- jeod::TimeConverter\_Dyn\_TDB, 40
  - convert\_a\_to\_b, 42
  - dyn\_ptr, 42
  - initialize, 42
  - InputProcessor, 42
  - operator=, 42
  - tdb\_ptr, 43
  - TimeConverter\_Dyn\_TDB, 41, 42
- jeod::TimeConverter\_Dyn\_UDE, 43
  - convert\_a\_to\_b, 44
  - dyn\_ptr, 45
  - initialize, 44
  - InputProcessor, 45
  - operator=, 45
  - reset\_a\_to\_b\_offset, 45
  - TimeConverter\_Dyn\_UDE, 44
  - ude\_ptr, 45
- jeod::TimeConverter\_STD\_UDE, 46
  - convert\_a\_to\_b, 47
  - convert\_b\_to\_a, 47
  - failed\_null\_test, 48
  - initialize, 48
  - InputProcessor, 48
  - operator=, 48
  - std\_ptr, 49
  - ude\_ptr, 49
- jeod::TimeConverter\_TAI\_GPS, 49
  - convert\_a\_to\_b, 50
  - convert\_b\_to\_a, 50
  - gps\_ptr, 51
  - initialize, 51
  - InputProcessor, 51
  - operator=, 51
  - tai\_ptr, 51
- jeod::TimeConverter\_TAI\_TDB, 52
  - a\_to\_b\_offset, 54
  - convert\_a\_to\_b, 53
  - convert\_b\_to\_a, 53
  - initialize, 54
  - InputProcessor, 54
  - nlter, 55
  - nSteps, 55
  - operator=, 54
  - prev\_tai\_seconds, 55
  - prev\_tdb\_seconds, 55
  - tai\_ptr, 55
  - tdb\_ptr, 56
- jeod::TimeConverter\_TAI\_TT, 56
  - convert\_a\_to\_b, 57
  - convert\_b\_to\_a, 57
  - initialize, 58
  - InputProcessor, 58
  - operator=, 58
  - tai\_ptr, 58
  - tt\_ptr, 58
- jeod::TimeConverter\_TAI\_UT1, 59
  - convert\_a\_to\_b, 61
  - convert\_b\_to\_a, 61
  - gradient, 62
  - index, 62
  - initialize, 61
  - initialize\_tai\_to\_ut1, 61
  - InputProcessor, 62
  - last\_index, 63
  - next\_value, 63
  - next\_when, 63
  - off\_table\_end, 63
  - operator=, 62
  - override\_data\_table, 63
  - prev\_value, 63
  - prev\_when, 64
  - tai\_ptr, 64
  - tai\_to\_ut1\_override\_val, 64
  - ut1\_ptr, 64
  - val\_vec, 64
  - verify\_table\_lookup\_ends, 62
  - when\_vec, 64
- jeod::TimeConverter\_TAI\_UTC, 65
  - convert\_a\_to\_b, 67
  - convert\_b\_to\_a, 68
  - index, 69
  - initialize, 68
  - initialize\_leap\_second, 68
  - InputProcessor, 69
  - last\_index, 69
  - leap\_sec\_override\_val, 69
  - next\_when, 69
  - off\_table\_end, 70
  - operator=, 68
  - override\_data\_table, 70
  - prev\_when, 70
  - tai\_ptr, 70

- utc\_ptr, 70
- val\_vec, 70
- verify\_table\_lookup\_ends, 68
- when\_vec, 71
- jeod::TimeConverter\_UT1\_GMST, 72
  - convert\_a\_to\_b, 73
  - gmst\_ptr, 74
  - initialize, 73
  - InputProcessor, 74
  - operator=, 73
  - ut1\_ptr, 74
- jeod::TimeDyn, 74
  - ~TimeDyn, 75
  - init\_attrjeod\_\_TimeDyn, 77
  - initialize\_initializer\_time, 76
  - InputProcessor, 77
  - offset, 77
  - operator=, 76
  - ref\_scale, 77
  - scale\_factor, 77
  - TimeDyn, 75
  - update, 76
  - update\_offset, 76
- jeod::TimeEnum, 77
  - TimeFormat, 78
- jeod::TimeGMST, 78
  - ~TimeGMST, 79
  - calculate\_calendar\_values, 79
  - init\_attrjeod\_\_TimeGMST, 80
  - InputProcessor, 80
  - operator=, 80
  - set\_epoch, 80
  - set\_time\_by\_trunc\_julian, 80
  - TimeGMST, 79
- jeod::TimeGPS, 80
  - ~TimeGPS, 82
  - calculate\_calendar\_values, 82
  - convert\_from\_calendar, 82
  - day\_of\_week, 84
  - init\_attrjeod\_\_TimeGPS, 84
  - InputProcessor, 84
  - operator=, 83
  - rollover\_count, 84
  - rollover\_count\_13\_bit, 84
  - seconds\_of\_day, 84
  - seconds\_of\_week, 85
  - set\_epoch, 83
  - set\_time\_by\_days, 83
  - set\_time\_by\_seconds, 83
  - set\_time\_by\_trunc\_julian, 83
  - TimeGPS, 82
  - week, 85
  - week\_13\_bit, 85
- jeod::TimeLinks, 85
  - ~TimeLinks, 86
  - default\_path\_size, 86
  - init\_attrjeod\_\_TimeLinks, 86
  - InputProcessor, 86
  - operator=, 86
  - TimeLinks, 86
- jeod::TimeMET, 108
  - ~TimeMET, 109
  - hold, 110
  - init\_attrjeod\_\_TimeMET, 110
  - InputProcessor, 110
  - operator=, 110
  - previous\_hold, 110
  - TimeMET, 109
  - update, 110
- jeod::TimeManager, 87
  - ~TimeManager, 88
  - converter\_vector, 94
  - dyn\_time, 94
  - get\_converter\_ptr, 89
  - get\_jeod\_integration\_time, 89
  - get\_time\_change\_flag, 89
  - get\_time\_ptr, 89, 90
  - get\_time\_scale\_factor, 90
  - get\_timestamp\_time, 90
  - init\_attrjeod\_\_TimeManager, 94
  - initialize, 90
  - InputProcessor, 94
  - num\_types, 94
  - operator=, 91
  - register\_converter, 91
  - register\_time, 91
  - register\_time\_named, 91
  - simtime, 94
  - time\_change\_flag, 95
  - time\_lookup, 92
  - time\_standards\_exist, 92
  - time\_vector, 95
  - TimeManager, 88, 89
  - TimeManagerInit, 94
  - update, 92
  - update\_time, 93
  - verify\_table\_lookup\_ends, 93
- jeod::TimeManagerInit, 95
  - ~TimeManagerInit, 97
  - converter\_ptrs\_index, 102
  - create\_init\_tree, 97
  - create\_update\_tree, 98
  - dyn\_time\_index, 102
  - get\_conv\_dir\_init, 98
  - get\_conv\_dir\_upd, 98
  - get\_conv\_ptr\_index, 99
  - get\_status, 99
  - increment\_status, 99
  - init\_attrjeod\_\_TimeManagerInit, 102
  - init\_converter\_dir\_table, 102
  - initialize, 100
  - initialize\_manager, 100
  - initialize\_time\_types, 100
  - initializer, 103
  - initializer\_index, 103
  - InputProcessor, 102



- num\_added\_pass, 103
- num\_added\_total, 103
- operator=, 101
- organize\_update\_list, 101
- populate\_converter\_registry, 101
- set\_status, 101
- sim\_start\_format, 103
- status, 103
- time\_manager, 104
- TimeManagerInit, 97
- update\_converter\_dir\_table, 104
- verify\_converter\_setup, 101
- verify\_times\_setup, 102
- jeod::TimeMessages, 104
  - duplicate\_methods, 105
  - extension\_error, 106
  - incomplete\_setup\_error, 106
  - init\_attrjeod\_\_TimeMessages, 105
  - initialization\_error, 106
  - InputProcessor, 105
  - invalid\_data\_error, 107
  - invalid\_node, 107
  - invalid\_setup\_error, 107
  - memory\_error, 108
  - operator=, 105
  - redundancy\_error, 108
  - TimeMessages, 105
- jeod::TimeStandard, 110
  - ~TimeStandard, 112
  - add\_type\_initialize, 113
  - calculate\_calendar\_values, 113
  - calendar\_day, 117
  - calendar\_hour, 117
  - calendar\_minute, 118
  - calendar\_month, 118
  - calendar\_second, 118
  - calendar\_update, 113
  - calendar\_year, 118
  - convert\_from\_calendar, 114
  - init\_attrjeod\_\_TimeStandard, 117
  - initialize\_from\_parent, 114
  - initialize\_initializer\_time, 115
  - InputProcessor, 117
  - julian\_date, 118
  - julian\_date\_at\_epoch, 115
  - last\_calendar\_update, 118
  - operator=, 115
  - prev\_julian\_day, 119
  - seconds\_at\_year\_start, 119
  - seconds\_of\_year, 115
  - send\_warning\_pre\_1968, 119
  - set\_epoch, 116
  - set\_time\_by\_days, 116
  - set\_time\_by\_seconds, 116
  - set\_time\_by\_trunc\_julian, 117
  - TimeStandard, 112, 113
  - TimeUDE, 117
  - tjt\_at\_epoch, 119
  - tjt\_jd\_offset, 119
  - tjt\_mjt\_offset, 119
  - trunc\_julian\_time, 120
  - year\_of\_last\_soy, 120
- jeod::TimeTAI, 120
  - ~TimeTAI, 121
  - init\_attrjeod\_\_TimeTAI, 122
  - InputProcessor, 122
  - operator=, 121
  - set\_epoch, 121
  - TimeTAI, 121
- jeod::TimeTDB, 122
  - ~TimeTDB, 123
  - init\_attrjeod\_\_TimeTDB, 123
  - InputProcessor, 123
  - operator=, 123
  - set\_epoch, 123
  - TimeTDB, 123
- jeod::TimeTT, 124
  - ~TimeTT, 124
  - init\_attrjeod\_\_TimeTT, 125
  - InputProcessor, 125
  - operator=, 125
  - set\_epoch, 125
  - TimeTT, 124, 125
- jeod::TimeUDE, 125
  - ~TimeUDE, 128
  - add\_type\_initialize, 128
  - clock\_day, 134
  - clock\_hour, 134
  - clock\_minute, 135
  - clock\_second, 135
  - clock\_update, 129
  - convert\_epoch\_to\_update, 129
  - epoch\_data\_present, 135
  - epoch\_day, 135
  - epoch\_defined\_in\_name, 135
  - epoch\_format, 135
  - epoch\_hour, 136
  - epoch\_index, 136
  - epoch\_initializing\_value, 136
  - epoch\_minute, 136
  - epoch\_month, 136
  - epoch\_second, 136
  - epoch\_value\_is\_set\_calendar, 136
  - epoch\_value\_is\_set\_clock, 137
  - epoch\_value\_is\_set\_number, 137
  - epoch\_year, 137
  - init\_attrjeod\_\_TimeUDE, 134
  - initial\_value\_format, 137
  - initialize\_from\_parent, 129
  - initialize\_initializer\_time, 130
  - initializing\_data\_present, 137
  - InputProcessor, 134
  - last\_clock\_update, 137
  - must\_be\_singleton, 130
  - operator=, 130
  - set\_epoch\_dyn, 131

- set\_epoch\_initializing\_value, 131
- set\_epoch\_std, 131
- set\_epoch\_times, 132
- set\_epoch\_ude, 132
- set\_initial\_times, 132
- set\_time\_by\_clock, 133
- set\_time\_by\_days, 133
- set\_time\_by\_seconds, 133
- TimeUDE, 128
- update\_index, 138
- verify\_epoch, 133
- verify\_init, 134
- verify\_update, 134
- jeod::TimeUT1, 138
  - ~TimeUT1, 139
  - get\_days, 139
  - init\_attrjeod\_\_TimeUT1, 140
  - InputProcessor, 140
  - operator=, 139
  - set\_epoch, 139
  - TimeUT1, 139
  - true\_ut1, 140
- jeod::TimeUTC, 140
  - ~TimeUTC, 141
  - init\_attrjeod\_\_TimeUTC, 142
  - InputProcessor, 142
  - operator=, 141
  - set\_epoch, 141
  - TimeUTC, 141
  - true\_utc, 142
- JeodBaseTime
  - jeod::JeodBaseTime, 23
  - jeod::TimeConverter, 36
- Julian
  - jeod::TimeEnum, 78
- julian
  - jeod::TimeEnum, 78
- julian\_date
  - jeod::TimeStandard, 118
- julian\_date\_at\_epoch
  - jeod::TimeStandard, 115
- last\_calendar\_update
  - jeod::TimeStandard, 118
- last\_clock\_update
  - jeod::TimeUDE, 137
- last\_index
  - jeod::TimeConverter\_TAI\_UT1, 63
  - jeod::TimeConverter\_TAI\_UTC, 69
- leap\_sec\_override\_val
  - jeod::TimeConverter\_TAI\_UTC, 69
- links
  - jeod::JeodBaseTime, 29
- memory\_error
  - jeod::TimeMessages, 108
- Models, 13
- modified\_julian
  - jeod::TimeEnum, 78
- must\_be\_singleton
  - jeod::JeodBaseTime, 26
  - jeod::TimeUDE, 130
- NO\_DIRECTION
  - jeod::TimeConverter, 33
- nIter
  - jeod::TimeConverter\_TAI\_TDB, 55
- nSteps
  - jeod::TimeConverter\_TAI\_TDB, 55
- name
  - jeod::JeodBaseTime, 29
- next\_value
  - jeod::TimeConverter\_TAI\_UT1, 63
- next\_when
  - jeod::TimeConverter\_TAI\_UT1, 63
  - jeod::TimeConverter\_TAI\_UTC, 69
- num\_added\_pass
  - jeod::TimeManagerInit, 103
- num\_added\_total
  - jeod::TimeManagerInit, 103
- num\_types
  - jeod::TimeManager, 94
- off\_table\_end
  - jeod::TimeConverter\_TAI\_UT1, 63
  - jeod::TimeConverter\_TAI\_UTC, 70
- offset
  - jeod::TimeDyn, 77
- operator=
  - jeod::JeodBaseTime, 26
  - jeod::TimeConverter, 35
  - jeod::TimeConverter\_Dyn\_TAI, 40
  - jeod::TimeConverter\_Dyn\_TDB, 42
  - jeod::TimeConverter\_Dyn\_UDE, 45
  - jeod::TimeConverter\_STD\_UDE, 48
  - jeod::TimeConverter\_TAI\_GPS, 51
  - jeod::TimeConverter\_TAI\_TDB, 54
  - jeod::TimeConverter\_TAI\_TT, 58
  - jeod::TimeConverter\_TAI\_UT1, 62
  - jeod::TimeConverter\_TAI\_UTC, 68
  - jeod::TimeConverter\_UT1\_GMST, 73
  - jeod::TimeDyn, 76
  - jeod::TimeGMST, 80
  - jeod::TimeGPS, 83
  - jeod::TimeLinks, 86
  - jeod::TimeManager, 91
  - jeod::TimeManagerInit, 101
  - jeod::TimeMessages, 105
  - jeod::TimeMET, 110
  - jeod::TimeStandard, 115
  - jeod::TimeTAI, 121
  - jeod::TimeTDB, 123
  - jeod::TimeTT, 125
  - jeod::TimeUDE, 130
  - jeod::TimeUT1, 139
  - jeod::TimeUTC, 141
- organize\_update\_list
  - jeod::TimeManagerInit, 101

- override\_data\_table
  - jeod::TimeConverter\_TAI\_UT1, 63
  - jeod::TimeConverter\_TAI\_UTC, 70
- override\_initialized
  - jeod::JeodBaseTime, 26
  - jeod::TimeConverter, 35
- PATH
  - Time, 17
- populate\_converter\_registry
  - jeod::TimeManagerInit, 101
- prev\_julian\_day
  - jeod::TimeStandard, 119
- prev\_tai\_seconds
  - jeod::TimeConverter\_TAI\_TDB, 55
- prev\_tdb\_seconds
  - jeod::TimeConverter\_TAI\_TDB, 55
- prev\_value
  - jeod::TimeConverter\_TAI\_UT1, 63
- prev\_when
  - jeod::TimeConverter\_TAI\_UT1, 64
  - jeod::TimeConverter\_TAI\_UTC, 70
- previous\_hold
  - jeod::TimeMET, 110
- redundancy\_error
  - jeod::TimeMessages, 108
- ref\_scale
  - jeod::TimeDyn, 77
- register\_converter
  - jeod::TimeManager, 91
- register\_time
  - jeod::TimeManager, 91
- register\_time\_named
  - jeod::TimeManager, 91
- reset\_a\_to\_b\_offset
  - jeod::TimeConverter, 35
  - jeod::TimeConverter\_Dyn\_UDE, 45
  - jeod::TimeConverter\_STD\_UDE, 48
- rollover\_count
  - jeod::TimeGPS, 84
- rollover\_count\_13\_bit
  - jeod::TimeGPS, 84
- scale\_factor
  - jeod::TimeDyn, 77
- seconds
  - jeod::JeodBaseTime, 29
- seconds\_since\_epoch
  - jeod::TimeEnum, 78
- seconds\_at\_year\_start
  - jeod::TimeStandard, 119
- seconds\_of\_day
  - jeod::TimeGPS, 84
- seconds\_of\_week
  - jeod::TimeGPS, 85
- seconds\_of\_year
  - jeod::TimeStandard, 115
- send\_warning\_pre\_1968
  - jeod::TimeStandard, 119
- set\_a\_to\_b\_offset
  - jeod::TimeConverter\_TAI\_TDB, 54
- set\_epoch
  - jeod::TimeGMST, 80
  - jeod::TimeGPS, 83
  - jeod::TimeStandard, 116
  - jeod::TimeTAI, 121
  - jeod::TimeTDB, 123
  - jeod::TimeTT, 125
  - jeod::TimeUT1, 139
  - jeod::TimeUTC, 141
- set\_epoch\_dyn
  - jeod::TimeUDE, 131
- set\_epoch\_initializing\_value
  - jeod::TimeUDE, 131
- set\_epoch\_std
  - jeod::TimeUDE, 131
- set\_epoch\_times
  - jeod::TimeUDE, 132
- set\_epoch\_ude
  - jeod::TimeUDE, 132
- set\_index
  - jeod::JeodBaseTime, 26
- set\_initial\_times
  - jeod::TimeUDE, 132
- set\_name
  - jeod::JeodBaseTime, 26
- set\_status
  - jeod::TimeManagerInit, 101
- set\_time\_by\_clock
  - jeod::TimeUDE, 133
- set\_time\_by\_days
  - jeod::JeodBaseTime, 26
  - jeod::TimeGPS, 83
  - jeod::TimeStandard, 116
  - jeod::TimeUDE, 133
- set\_time\_by\_seconds
  - jeod::JeodBaseTime, 27
  - jeod::TimeGPS, 83
  - jeod::TimeStandard, 116
  - jeod::TimeUDE, 133
- set\_time\_by\_trunc\_julian
  - jeod::TimeGMST, 80
  - jeod::TimeGPS, 83
  - jeod::TimeStandard, 117
- sim\_start\_format
  - jeod::TimeManagerInit, 103
- simtime
  - jeod::TimeManager, 94
- status
  - jeod::TimeManagerInit, 103
- std\_ptr
  - jeod::TimeConverter\_STD\_UDE, 49
- TAI\_to\_TT\_offset
  - jeod::TimeConverter\_TAI\_TDB, 55
- tai\_ptr
  - jeod::TimeConverter\_Dyn\_TAI, 40

- jeod::TimeConverter\_TAI\_GPS, 51
- jeod::TimeConverter\_TAI\_TDB, 55
- jeod::TimeConverter\_TAI\_TT, 58
- jeod::TimeConverter\_TAI\_UT1, 64
- jeod::TimeConverter\_TAI\_UTC, 70
- tai\_to\_ut1.cc, 143
- tai\_to\_ut1.hh, 144
- tai\_to\_ut1\_override\_val
  - jeod::TimeConverter\_TAI\_UT1, 64
- tai\_to\_utc.cc, 144
- tai\_to\_utc.hh, 144
- tdb\_ptr
  - jeod::TimeConverter\_Dyn\_TDB, 43
  - jeod::TimeConverter\_TAI\_TDB, 56
- Time, 15
  - PATH, 17
- time.cc, 145
- time.hh, 145
- time\_\_add\_type\_update.cc, 146
- time\_change\_flag
  - jeod::TimeManager, 95
- time\_converter.cc, 146
- time\_converter.hh, 147
- time\_converter\_dyn\_tai.cc, 147
- time\_converter\_dyn\_tai.hh, 148
- time\_converter\_dyn\_tdb.cc, 148
- time\_converter\_dyn\_tdb.hh, 149
- time\_converter\_dyn\_ude.cc, 149
- time\_converter\_dyn\_ude.hh, 150
- time\_converter\_std\_ude.cc, 150
- time\_converter\_std\_ude.hh, 151
- time\_converter\_tai\_gps.cc, 151
- time\_converter\_tai\_gps.hh, 152
- time\_converter\_tai\_tdb.cc, 152
- time\_converter\_tai\_tdb.hh, 153
- time\_converter\_tai\_tt.cc, 153
- time\_converter\_tai\_tt.hh, 154
- time\_converter\_tai\_ut1.cc, 154
- time\_converter\_tai\_ut1.hh, 155
- time\_converter\_tai\_utc.cc, 155
- time\_converter\_tai\_utc.hh, 156
- time\_converter\_ut1\_gmst.cc, 156
- time\_converter\_ut1\_gmst.hh, 157
- time\_dyn.cc, 157
- time\_dyn.hh, 158
- time\_enum.hh, 158
- time\_gmst.cc, 159
- time\_gmst.hh, 159
- time\_gps.cc, 159
- time\_gps.hh, 160
- time\_links.hh, 160
- time\_lookup
  - jeod::TimeManager, 92
- time\_manager
  - jeod::JeodBaseTime, 30
  - jeod::TimeManagerInit, 104
- time\_manager.cc, 161
- time\_manager.hh, 161
- time\_manager\_\_initialize.cc, 162
- time\_manager\_init.cc, 162
- time\_manager\_init.hh, 163
- time\_messages.cc, 164
- time\_messages.hh, 164
- time\_met.cc, 164
- time\_met.hh, 165
- time\_standard.cc, 165
- time\_standard.hh, 166
- time\_standards\_exist
  - jeod::TimeManager, 92
- time\_tai.cc, 166
- time\_tai.hh, 167
- time\_tdb.cc, 167
- time\_tdb.hh, 168
- time\_tt.cc, 168
- time\_tt.hh, 169
- time\_ude.cc, 169
- time\_ude.hh, 170
- time\_ut1.cc, 170
- time\_ut1.hh, 171
- time\_utc.cc, 171
- time\_utc.hh, 172
- time\_vector
  - jeod::TimeManager, 95
- TimeConverter
  - jeod::JeodBaseTime, 27
  - jeod::TimeConverter, 33
- TimeConverter\_Dyn\_TAI
  - jeod::TimeConverter\_Dyn\_TAI, 39
- TimeConverter\_Dyn\_TDB
  - jeod::TimeConverter\_Dyn\_TDB, 41, 42
- TimeConverter\_Dyn\_UDE
  - jeod::TimeConverter\_Dyn\_UDE, 44
- TimeConverter\_STD\_UDE
  - jeod::TimeConverter\_STD\_UDE, 47
- TimeConverter\_TAI\_GPS
  - jeod::TimeConverter\_TAI\_GPS, 50
- TimeConverter\_TAI\_TDB
  - jeod::TimeConverter\_TAI\_TDB, 53
- TimeConverter\_TAI\_TT
  - jeod::TimeConverter\_TAI\_TT, 57
- TimeConverter\_TAI\_UT1
  - jeod::TimeConverter\_TAI\_UT1, 60
- TimeConverter\_TAI\_UTC
  - jeod::TimeConverter\_TAI\_UTC, 67
- TimeConverter\_UT1\_GMST
  - jeod::TimeConverter\_UT1\_GMST, 73
- TimeDyn
  - jeod::TimeDyn, 75
- TimeFormat
  - jeod::TimeEnum, 78
- TimeGMST
  - jeod::TimeGMST, 79
- TimeGPS
  - jeod::TimeGPS, 82
- TimeLinks
  - jeod::TimeLinks, 86

- TimeMET
  - jeod::TimeMET, [109](#)
- TimeManager
  - jeod::TimeManager, [88](#), [89](#)
- TimeManagerInit
  - jeod::JeodBaseTime, [28](#)
  - jeod::TimeManager, [94](#)
  - jeod::TimeManagerInit, [97](#)
- TimeMessages
  - jeod::TimeMessages, [105](#)
- TimeStandard
  - jeod::TimeStandard, [112](#), [113](#)
- TimeTAI
  - jeod::TimeTAI, [121](#)
- TimeTDB
  - jeod::TimeTDB, [123](#)
- TimeTT
  - jeod::TimeTT, [124](#), [125](#)
- TimeUDE
  - jeod::TimeStandard, [117](#)
  - jeod::TimeUDE, [128](#)
- TimeUT1
  - jeod::TimeUT1, [139](#)
- TimeUTC
  - jeod::TimeUTC, [141](#)
- tjt\_at\_epoch
  - jeod::TimeStandard, [119](#)
- tjt\_jd\_offset
  - jeod::TimeStandard, [119](#)
- tjt\_mjt\_offset
  - jeod::TimeStandard, [119](#)
- true\_ut1
  - jeod::TimeUT1, [140](#)
- true\_utc
  - jeod::TimeUTC, [142](#)
- trunc\_julian\_time
  - jeod::TimeStandard, [120](#)
- truncated\_julian
  - jeod::TimeEnum, [78](#)
- tt\_ptr
  - jeod::TimeConverter\_TAI\_TT, [58](#)
- ude\_ptr
  - jeod::TimeConverter\_Dyn\_UDE, [45](#)
  - jeod::TimeConverter\_STD\_UDE, [49](#)
- undefined
  - jeod::TimeEnum, [78](#)
- update
  - jeod::JeodBaseTime, [27](#)
  - jeod::TimeDyn, [76](#)
  - jeod::TimeManager, [92](#)
  - jeod::TimeMET, [110](#)
- update\_converter\_dir\_table
  - jeod::TimeManagerInit, [104](#)
- update\_converter\_direction
  - jeod::JeodBaseTime, [30](#)
- update\_converter\_ptr
  - jeod::JeodBaseTime, [30](#)
- update\_from\_name
  - jeod::JeodBaseTime, [30](#)
- update\_index
  - jeod::TimeUDE, [138](#)
- update\_offset
  - jeod::TimeDyn, [76](#)
- update\_time
  - jeod::TimeManager, [93](#)
- ut1\_ptr
  - jeod::TimeConverter\_TAI\_UT1, [64](#)
  - jeod::TimeConverter\_UT1\_GMST, [74](#)
- utc\_ptr
  - jeod::TimeConverter\_TAI\_UTC, [70](#)
- val\_vec
  - jeod::TimeConverter\_TAI\_UT1, [64](#)
  - jeod::TimeConverter\_TAI\_UTC, [70](#)
- valid\_directions
  - jeod::TimeConverter, [37](#)
- verify\_converter\_setup
  - jeod::TimeManagerInit, [101](#)
- verify\_epoch
  - jeod::TimeUDE, [133](#)
- verify\_init
  - jeod::TimeUDE, [134](#)
- verify\_setup
  - jeod::TimeConverter, [35](#)
- verify\_table\_lookup\_ends
  - jeod::TimeConverter, [36](#)
  - jeod::TimeConverter\_TAI\_UT1, [62](#)
  - jeod::TimeConverter\_TAI\_UTC, [68](#)
  - jeod::TimeManager, [93](#)
- verify\_times\_setup
  - jeod::TimeManagerInit, [102](#)
- verify\_update
  - jeod::TimeUDE, [134](#)
- week
  - jeod::TimeGPS, [85](#)
- week\_13\_bit
  - jeod::TimeGPS, [85](#)
- when\_vec
  - jeod::TimeConverter\_TAI\_UT1, [64](#)
  - jeod::TimeConverter\_TAI\_UTC, [71](#)
- year\_of\_last\_soy
  - jeod::TimeStandard, [120](#)